Luiz F. O. Chamon

supervised and reinforcement learning under requirements

SIMPAS group meeting
Apr. 7th, 2025

Who am I?

Luiz

- 2025–: École Polytechnique de Paris (Professor)
- 2022–2024: ELLIS-SimTech (Research group leader) IMPRS-IS faculty
- 2021–2022: Simons Institute, UC Berkeley (Postdoc)
- 2020: University of Pennsylvania (PhD)
- < 2015: University of São Paulo (BSc. & MSc.)
- I speak 4.5 languages

Agenda

I. Constrained supervised learning
   - Constrained learning theory
   - Constrained learning algorithms
   - Resilient constrained learning

Break (10 min)

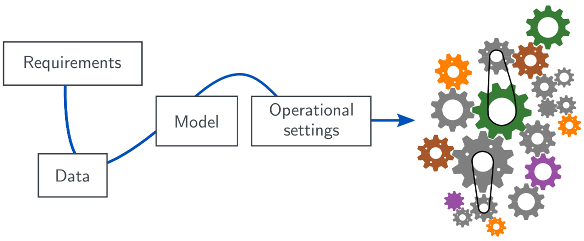
II. Constrained reinforcement learning
   - Constrained RL duality
   - Constrained RL algorithms

Q&A and discussions

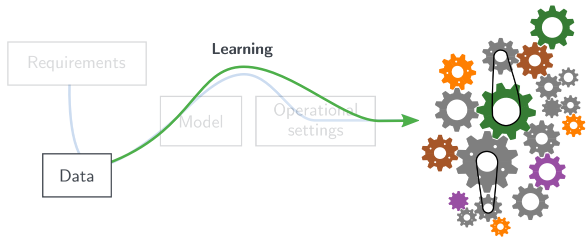
https://luizchamon.com/sgm

Why learning under requirements?

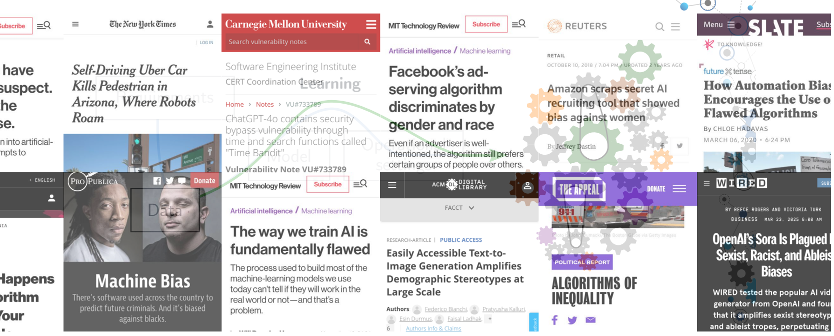
Why learning under requirements?



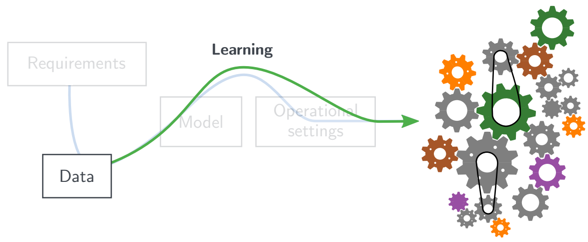
Why learning under requirements?



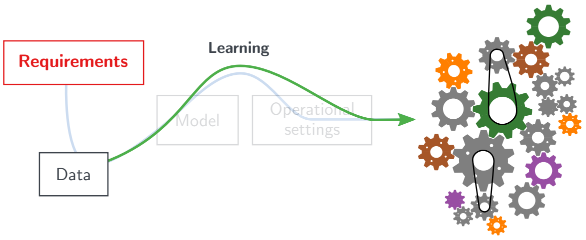
Why learning under requirements?



Why learning under requirements?

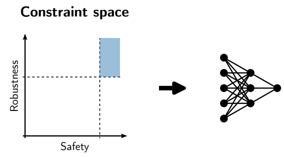


Why learning under requirements?



What is a requirements?

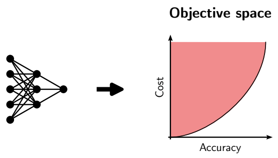
- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide



[NASA. "Systems engineering handbook," 2019]

What is a requirements?

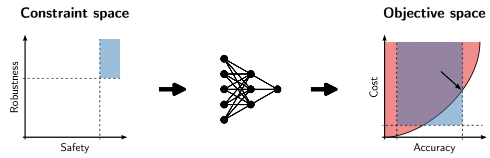
- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide
- **Goals are "should" statements:** express recommendations (once "shall" statements are satisfied)
  - *Objective space*: things the system achieves



[NASA. "Systems engineering handbook," 2019]

What is a requirements?

- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide
- **Goals are "should" statements:** express recommendations (once "shall" statements are satisfied)
  - *Objective space*: things the system achieves



[NASA. "Systems engineering handbook," 2019]

What is (un)constrained learning?

$$P_U^\star = \min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $\mathfrak{D}, \mathfrak{A}, \mathfrak{P}$ unknown

[Chamon et al., IEEE ICASSP'20 (best student paper); Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

What is (un)constrained learning?

$$\begin{aligned} P^\star = \min_{\theta} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $\mathfrak{D}, \mathfrak{A}, \mathfrak{P}$ unknown

[Chamon et al., IEEE ICASSP'20 (best student paper); Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

What about penalties?

$$\begin{aligned} P^\star = \min_{\theta} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

$$\updownarrow$$

$$\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \lambda\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{P}}\Big[\mu(\boldsymbol{x},y)h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$

What about penalties?

NON-CONVEX

$$\begin{aligned} P^\star = \min_{\theta} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \\ \text{subject to} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big) \leq u, \quad \mathfrak{P}\text{-a.e.} \end{aligned}$$

$$\not\updownarrow$$

$$\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \lambda\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{P}}\Big[\mu(\boldsymbol{x},y)h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big]$$

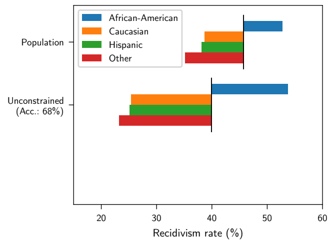
- ✗ There may not exist (λ, μ) such that the penalized solution is optimal *and* feasible
- ✗ Even if such (λ, μ) exist, they are not easy to find (hyperparameter search, cross-validation...)
- ✓ Constrained learning yields stronger guarantees, better performance, better trade-offs...

Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- …

Fairness

Problem
Predict whether an individual will recidivate



*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

Fairness: "Equality" of odds

Problem
Predict whether an individual will recidivate at the same rate across races

$$\begin{array}{ll} \min\limits_{\theta} & \text{Prediction error} \\ \text{subject to} & \text{Prediction rate disparity (Race)} \leq c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{array}$$

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]

Fairness: "Equality" of odds

Problem
Predict whether an individual will recidivate at the same rate across races

$$\begin{array}{ll} \min\limits_{\theta} & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \\ \text{subject to} & \text{Prediction rate disparity (Race)} \leq c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{array}$$

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]

Fairness: "Equality" of odds

Problem
Predict whether an individual will recidivate at the same rate across races

$$\begin{array}{ll} \min\limits_{\theta} & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \\ \text{subject to} & \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[f_{\theta}(x_n) = 1 \mid \text{Race}] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[f_{\theta}(x_n) = 1] + c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{array}$$

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23]

Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- …

Federated learning

Problem
Learn a common model using data from K clients

$$\min_{\theta} \ \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_{\theta})$$

- k-th client loss: $\text{Loss}_k(f_{\theta}) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_{\theta}(x_{n_k}), y_{n_k}\big)$

[Shen et al., ICLR'22]

Heterogeneous federated learning

Problem
Learn a common model using data from K clients

$$\min_{\theta} \ \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_{\theta})$$

- k-th client loss: $\text{Loss}_k(f_{\theta}) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_{\theta}(x_{n_k}), y_{n_k}\big)$

[Shen et al., ICLR'22]

Heterogeneous federated learning

Problem
Learn a common model using data from K clients that is good for all clients

$$\min_{\theta} \quad \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta)$$
$$\text{subject to} \quad \text{Loss}_k(f_\theta) \leq \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta) + c,$$
$$k = 1, \dots, K$$

- k-th client loss: $\text{Loss}_k(f_\theta) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(x_{n_k}), y_{n_k}\big)$

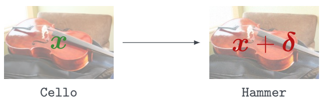
[Shen et al., ICRL'22]

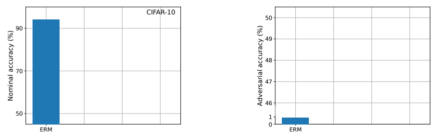
Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- ...

Robustness

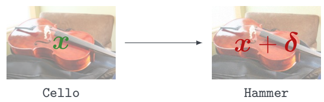
Problem
Learn an accurate classifier that is robust to input perturbations





Robustness

Problem
Learn an accurate classifier that is robust to input perturbations

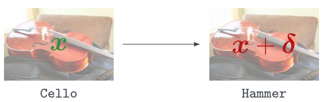


$$\min_{\theta} \quad \text{Nominal loss}$$
$$\text{subject to} \quad \text{Robustness loss} \leq c$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robustness

Problem
Learn an accurate classifier that is robust to input perturbations

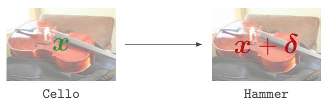


$$\min_{\theta} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$
$$\text{subject to} \quad \text{Robustness loss} \leq c$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robustness

Problem
Learn an accurate classifier that is robust to input perturbations

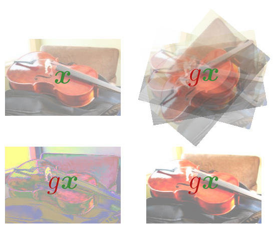


$$\min_{\theta} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$
$$\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\delta\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(x_n + \delta), y_n\big)\right] \leq c$$

[Chamon and Ribeiro, NeurIPS'20; Robey*, Chamon*, Pappas, Hassani, and Ribeiro, NeurIPS'21; Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Invariance

Problem
Learn an accurate classifier that is invariant to transformation $g \in \mathcal{G}$, e.g., $\mathcal{G} = \left\{\begin{array}{l}\text{Rotate, TranslateX(Y),} \\ \text{ShearX(Y), Crop, Invert,} \\ \text{Solarize, Contrast,} \\ \text{Brightness, Sharpness}\dots\end{array}\right\}$



$$\min_{\theta} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n), y_n\big)$$
$$\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N} \left[\max_{g \in \mathcal{G}} \text{Loss}\big(f_\theta(gx_n), y_n\big)\right] \leq c$$

[Hounie, Chamon, Ribeiro, NeurIPS'23]

Applications

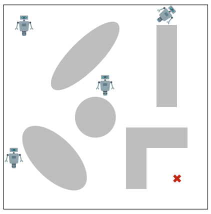
- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- ...

Safety

Problem
Find a control policy that navigates the environment effectively and safely

Safety

Problem
Find a control policy that navigates the environment effectively and safely

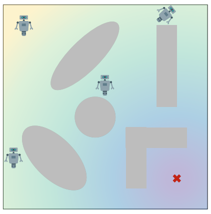


$$\begin{aligned} &\underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} && \text{Task reward} \\ &\text{subject to} && \mathbb{P}\left[\text{Colliding with } \mathcal{O}_i\right] \leq \delta, \\ & && \text{for } i = 1, 2, \ldots \end{aligned}$$

[Paternain, Calvo-Fullana, Chamon, Ribeiro, IEEE TAC'23]

Safety

Problem
Find a control policy that navigates the environment effectively and safely

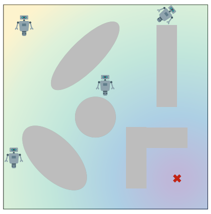


$$\begin{aligned} &\underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} && \mathbb{E}_{s,a\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} r_0(s_t, a_t)\right] \\ &\text{subject to} && \mathbb{P}\left[\text{Colliding with } \mathcal{O}_i\right] \leq \delta, \\ & && \text{for } i = 1, 2, \ldots \end{aligned}$$

[Paternain, Calvo-Fullana, Chamon, Ribeiro, IEEE TAC'23]

Safety

Problem
Find a control policy that navigates the environment effectively and safely



$$\begin{aligned} &\underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} && \mathbb{E}_{s,a\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} r_0(s_t, a_t)\right] \\ &\text{subject to} && \mathbb{P}\left(\bigcap_{t=0}^{T-1}\{s_t \notin \mathcal{O}_i\} \,\Big|\, \pi\right) \geq 1 - \delta_i, \\ & && \text{for } i = 1, 2, \ldots \end{aligned}$$

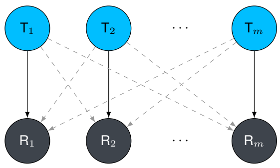
[Paternain, Calvo-Fullana, Chamon, Ribeiro, IEEE TAC'23]

Applications

- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23])
- Safe learning
  (e.g., [Paternain et al., IEEE TAC'23])
- Wireless resource allocation
  (e.g., [Eisen et al., IEEE TSP'19; NaderiAlizadeh et al., IEEE TSP'22; Chowdhury et al., Asilomar'23])
- . . .

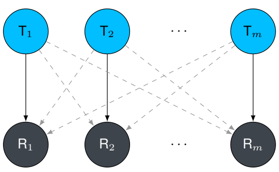
Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate



Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate



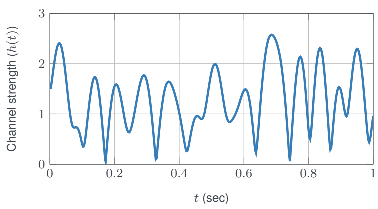
Input (p)

Channel (h)

Output = Input × Channel + Noise
$\Rightarrow$ Rate(p, h)

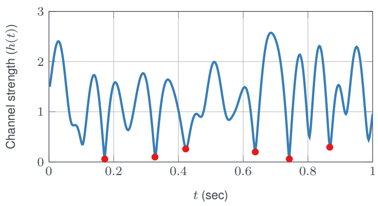
Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate



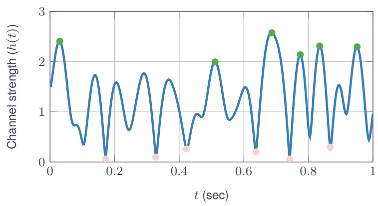
Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate



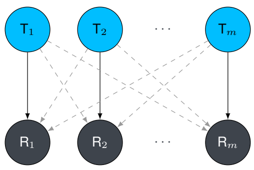
Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate



Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate

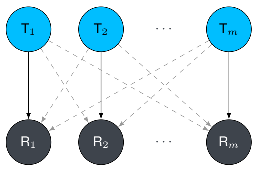


$$\begin{aligned} \min_{\pi \in \mathcal{P}(\mathcal{S})} \quad & \text{Total transmit power} \\ \text{s. to} \quad & \text{Rate } T_i \to R_i \geq c_i \end{aligned}$$

[Eisen, Zhang, Chamon, Lee, and Ribeiro, IEEE TSP'19]

Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate

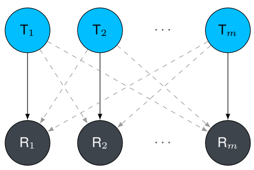


$$\begin{aligned} \max_{\pi \in \mathcal{P}(\mathcal{S})} \quad & -\sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h})}\left[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{m} p_{i,t}\right] \\ \text{s. to} \quad & \text{Rate } T_i \to R_i \geq c_i \end{aligned}$$

[Eisen, Zhang, Chamon, Lee, and Ribeiro, IEEE TSP'19]

Wireless resource allocation

Problem
Allocate the least transmit power to m devices to achieve a communication rate

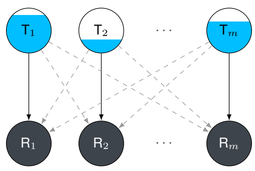


$$\begin{aligned} \max_{\pi \in \mathcal{P}(\mathcal{S})} \quad & -\sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h})}\left[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{m} p_{i,t}\right] \\ \text{s. to} \quad & \mathbb{E}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h})}\left[\frac{1}{T}\sum_{t=0}^{T-1}\text{Rate}_i(\boldsymbol{p}_t,\boldsymbol{h}_t)\right] \geq c_i \end{aligned}$$

[Eisen, Zhang, Chamon, Lee, and Ribeiro, IEEE TSP'19]

Wireless resource allocation

Problem
Allocate power without depleting the battery of m devices to achieve a communication rate

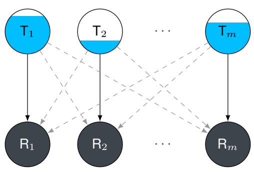


$$\begin{aligned} \min_{\pi \in \mathcal{P}(\mathcal{S})} \quad & \text{Total probability of depleting battery} \\ \text{s. to} \quad & \mathbb{E}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h},\boldsymbol{b})}\left[\frac{1}{T}\sum_{t=0}^{T-1}\text{Rate}_i(\boldsymbol{p}_t,\boldsymbol{h}_t)\right] \geq c_i \end{aligned}$$

[Chowdhury, Paternain, Verma, Swami, Segarra, Asilomar'23]

Wireless resource allocation

Problem
Allocate power without depleting the battery of m devices to achieve a communication rate



$$\begin{aligned} \max_{\pi \in \mathcal{P}(\mathcal{S})} \quad & -\sum_{i=1}^{m} \mathbb{P}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h},\boldsymbol{b})}\left[\bigcap_{t=0}^{T-1}\{b_{i,t}=0\}\right] \\ \text{s. to} \quad & \mathbb{E}_{\boldsymbol{h},\boldsymbol{p}\sim\pi(\boldsymbol{h},\boldsymbol{b})}\left[\frac{1}{T}\sum_{t=0}^{T-1}\text{Rate}_i(\boldsymbol{p}_t,\boldsymbol{h}_t)\right] \geq c_i \end{aligned}$$

[Chowdhury, Paternain, Verma, Swami, Segarra, Asilomar'23]

And many more...

- Precision, recall, churn (e.g., [Cotter et al., JMLR'19])
- Scientific priors (e.g., [Lu et al., SIAM J. Sci. Comp.'21; Moro and Chamon, ICLR'25])
- Continual learning (e.g., [Peng et al., ICML'23])
- Active learning (e.g., [Elenter et al., NeurIPS'22])
- Semi-supervised learning (e.g., [Cerviño et al., ICML'23])
- Minimum norm interpolation, SVM...

Constrained supervised learning

What is (un)constrained learning?

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u, \quad r = 1, \dots, N\end{aligned}$$

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- $f_{\boldsymbol{\theta}}$ is a (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]
- $(\boldsymbol{x}_n, y_n) \sim \mathfrak{D}$, $(\boldsymbol{x}_m, y_m) \sim \mathfrak{A}$, $(\boldsymbol{x}_r, y_r) \sim \mathfrak{P}$ (i.i.d.)

[Chamon et al., IEEE ICASSP'20 (best student paper); Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

Constrained learning challenges

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u\end{aligned} \quad \xrightarrow{?} \quad \begin{aligned}P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.}\end{aligned}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?

Constrained learning challenges

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u\end{aligned} \quad \xrightarrow{?} \quad \begin{aligned}P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.}\end{aligned}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?

Constrained learning challenges

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u\end{aligned} \quad \xrightarrow{?} \quad \begin{aligned}P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.}\end{aligned}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?

Constrained learning challenges

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u\end{aligned} \quad \xrightarrow{?} \quad \begin{aligned}P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.}\end{aligned}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?

Agenda

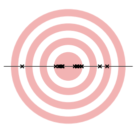
Constrained learning theory

Constrained learning algorithms

What classical learning theory says?

$$\min_{\boldsymbol{\theta}} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \xrightarrow{\text{"LLN"}} \min_{\boldsymbol{\theta}} \ \mathbb{E}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big]$$

- ✓ $f_{\boldsymbol{\theta}}$ is *probably approximately correct (PAC)* learnable
  e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...
  $(N \approx 1/\epsilon^2)$



[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning..., 2014]

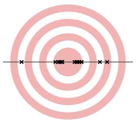
What's in a solution?

Definition (PAC learnability)

$f_{\boldsymbol{\theta}}$ is a *probably approximately correct* (PAC) learnable if for every ϵ, δ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\boldsymbol{\theta}^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1 - \delta$,

- near-optimal

$$P^\star - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\Big] \leq \epsilon$$



What's in a solution?

Definition (PACC learnability)

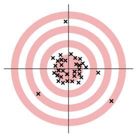
$f_{\boldsymbol{\theta}}$ is a *probably approximately correct constrained* (PACC) learnable if for every ϵ, δ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\boldsymbol{\theta}^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1 - \delta$,

- near-optimal

$$\left|P^\star - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\Big]\right| \leq \epsilon$$

- approximately feasible

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\Big] \leq c + \epsilon$$



When is constrained learning possible?

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \end{aligned} \quad \xrightarrow{?} \quad \begin{aligned} P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \end{aligned}$$

Proposition

$f_{\boldsymbol{\theta}}$ is PAC learnable $\not\Rightarrow$ $f_{\boldsymbol{\theta}}$ is PACC learnable

ECRM is not a PACC learner

Counter-example

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

- $\tau \sim \text{Uniform}\big(-1/2, 1/2\big)$

ECRM is not a PACC learner

Counter-example

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

- $\tau \sim \text{Uniform}\big(-1/2, 1/2\big)$

ECRM is not a PACC learner

Counter-example

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2 \bar{\tau}_N \leq \theta_1 - 1 \\ & -\theta_1 \bar{\tau}_N \leq 1 - \theta_2 \end{aligned} \qquad \mathbb{P}\big[|\hat{P}^\star - P^\star| \leq 1/32\big] = \mathbb{P}\big[\bar{\tau}_N = 0\big] = 0$$

- $\tau \sim \text{Uniform}\big(-1/2, 1/2\big) \to \bar{\tau}_N = \frac{1}{N}\sum_{n=1}^{N} \tau_n$

ECRM is not a PACC learner

Counter-example

$$\begin{aligned} P^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) = \frac{1}{8} \\ \text{subject to} \quad & \theta_2\, \mathbb{E}_\tau[\tau] \leq \theta_1 - 1 \Rightarrow \theta_1 \geq 1 \\ & -\theta_1\, \mathbb{E}_\tau[\tau] \leq \theta_2 - 1 \Rightarrow \theta_2 \leq 1 \end{aligned} \qquad J(\boldsymbol{\theta}) = \begin{cases} 1/16, & \boldsymbol{\theta} = [1/2, 1/2] \\ 1/8, & \boldsymbol{\theta} = [1, 1] \\ 1/4, & \boldsymbol{\theta} = [1, 0] \end{cases}$$

$$\begin{aligned} \hat{P}_r^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & \theta_2 \bar{\tau}_N \leq \theta_1 - 1 + r_1 \\ & -\theta_1 \bar{\tau}_N \leq 1 - \theta_2 + r_2 \end{aligned} \qquad \mathbb{P}\big[|\hat{P}_r^\star - P^\star| \leq 1/32\big] \leq 4e^{-0.001N},$$

unless $\bar{\tau}_N \leq r_1 < \dfrac{\bar{\tau}_N + 1}{2}$ and $r_2 \geq \bar{\tau}_N$

- $\tau \sim \text{Uniform}\big(-1/2, 1/2\big) \to \bar{\tau}_N = \frac{1}{N}\sum_{n=1}^{N} \tau_n$

Constrained learning challenges

$$\begin{aligned} \hat{P}^\star = \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u \end{aligned} \quad \xrightarrow{\text{PACC}} \quad \begin{aligned} P^\star = \min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \\ \text{subject to} \quad & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c \\ & h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.} \end{aligned}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?
2) *Computational*: can we solve the constrained empirical problem?

Constrained learning challenges

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \qquad P^\star = \min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big]$$

$$\text{subject to} \quad \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \quad \xrightarrow{\text{PACC}} \quad \text{subject to} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c$$

$$h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r, y_r)\big) \leq u \qquad h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big) \leq u \text{ a.e.}$$

Challenges

1) *Statistical*: does the solution of the constrained empirical problem generalize?

2) *Computational*: can we solve the constrained empirical problem?

Duality

PRIMAL

$\updownarrow$

DUAL

Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

DUAL

Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\updownarrow$

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

- In general, $\hat{D}^\star \leq \hat{P}^\star$
- But in some cases, $\hat{D}^\star = \hat{P}^\star$ (strong duality) [e.g., convex optimization]

Duality

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \text{ subject to } \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$\not\updownarrow$

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

- In general, $\hat{D}^\star \leq \hat{P}^\star$
- ~~But in some cases, $\hat{D}^\star = \hat{P}^\star$ (strong duality) [e.g., convex optimization]~~

An alternative path

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

PRIMAL $\not\longleftrightarrow$ DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\big[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\big] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\big[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\big] \leq c$$

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

An alternative path

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

PRIMAL $\not\longleftrightarrow$ DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\big[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\big] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\big[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\big] \leq c$$

$\big\downarrow \mathcal{H}_\theta \subset \mathcal{H}$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\big[\ell(\phi, \boldsymbol{z})\big] \quad \text{s. to} \quad \mathbb{E}_{\boldsymbol{z}}\big[g(\phi, \boldsymbol{z})\big] \leq c \quad \overset{?}{\longleftrightarrow} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\big[\ell(\phi, \boldsymbol{z})\big] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\big[g(\phi, \boldsymbol{z})\big] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Non-convex variational duality

Convex optimization: Primal $\longleftrightarrow$ Dual

Non-convex, finite dimensional optimization: Primal $\not\longleftrightarrow$ Dual

Non-convex variational duality

Convex optimization: Primal $\longleftrightarrow$ Dual

Non-convex, finite dimensional optimization: Primal $\not\longleftrightarrow$ Dual

Non-convex, **infinite** dimensional optimization: Primal $\longleftrightarrow$ Dual

[Chamon, Eldar, Ribeiro, IEEE TSP'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

Discrete, non-convex

[Chen et al., JMLR'19]: NP-hard

Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

Discrete, non-convex

[Chen et al., JMLR'19]: NP-hard

$$\min_{\theta\in L_2} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \int \theta(t) x_n(t) dt\right)\right]$$

$$\text{s. to } \|\theta\|_{L_0} = \int \mathbb{I}\left[\theta(t) \neq 0\right] dt \leq \frac{k}{p}$$

Continuous, non-convex

[Chamon et al., IEEE TSP'20]: tractable

Sparse logistic regression

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$

$$\text{s. to } \|\boldsymbol{\theta}\|_0 = \sum_{t=1}^{p} \mathbb{I}\left[\boldsymbol{\theta}_t \neq 0\right] \leq k$$

Discrete, non-convex

[Chen et al., JMLR'19]: NP-hard

$$\min_{\theta\in L_2} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \int \theta(t) x_n(t) dt\right)\right]$$

$$\text{s. to } \|\theta\|_{L_0} = \int \mathbb{I}\left[\theta(t) \neq 0\right] dt \leq \frac{k}{p}$$

Continuous, non-convex

[Chamon et al., IEEE TSP'20]: tractable

An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

$\not\longleftrightarrow$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c$$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \overset{=}{\longleftrightarrow} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

$\not\longleftrightarrow$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c \quad \overset{\epsilon_0}{\longleftrightarrow} \quad D^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] - c\right)$$

$\updownarrow \epsilon_0$ $\qquad$ $\updownarrow \epsilon_0$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \overset{=}{\longleftrightarrow} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

An alternative path

PRIMAL

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \quad \text{s. to } \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c$$

$\not\longleftrightarrow$

DUAL

$$\hat{D}^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

$\not\updownarrow$ PAC $\qquad$ $\updownarrow O(\epsilon)$

$$P^\star = \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] \leq c \quad \overset{\epsilon_0}{\longleftrightarrow} \quad D^\star = \max_{\lambda\geq 0} \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{z}}\left[\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(f_{\boldsymbol{\theta}}, \boldsymbol{z})\right] - c\right)$$

$\updownarrow \epsilon_0$ $\qquad$ $\updownarrow \epsilon_0$

$$\tilde{P}^\star = \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] \quad \text{s. to } \mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] \leq c \quad \overset{=}{\longleftrightarrow} \quad \tilde{D}^\star = \max_{\lambda\geq 0} \min_{\phi\in\mathcal{H}} \mathbb{E}_{\boldsymbol{z}}\left[\ell(\phi, \boldsymbol{z})\right] + \lambda\left(\mathbb{E}_{\boldsymbol{z}}\left[g(\phi, \boldsymbol{z})\right] - c\right)$$

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Dual (near-)PACC learning

Theorem

Let f be ν-universal, i.e., for each $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\gamma \in [0,1]$ there exists $\boldsymbol{\theta}$ such that

$$\mathbb{E}\left[\left|\gamma f_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + (1-\gamma) f_{\boldsymbol{\theta}_2}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right|\right] \le \nu$$

[$\{f_{\boldsymbol{\theta}}\}$ is a good covering of $\overline{\operatorname{conv}}(\{f_{\boldsymbol{\theta}}\})$]

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

Dual (near-)PACC learning

Theorem

Let f be ν-universal, i.e., for each $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\gamma \in [0,1]$ there exists $\boldsymbol{\theta}$ such that

$$\mathbb{E}\left[\left|\gamma f_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + (1-\gamma) f_{\boldsymbol{\theta}_2}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right|\right] \le \nu$$

Then $\hat{D}^\star$ is a (near-)PACC learner, i.e., with probability $1-\delta$,

Near-optimal: $\left|P^\star - \hat{D}^\star\right| \le \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

(mild additional conditions apply)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

Dual (near-)PACC learning

Theorem

Let f be ν-universal, i.e., for each $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\gamma \in [0,1]$ there exists $\boldsymbol{\theta}$ such that

$$\mathbb{E}\left[\left|\gamma f_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + (1-\gamma) f_{\boldsymbol{\theta}_2}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right|\right] \le \nu$$

Then $\hat{D}^\star$ is a (near-)PACC learner, i.e., for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, with probability $1-\delta$,

Near-optimal: $\left|P^\star - \hat{D}^\star\right| \le \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

Approximately feasible: $\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

(ℓ_0 strongly convex and g, h convex) $h\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right) \le r$, with $\mathfrak{P}$-prob. $1 - \tilde{O}\left(\nu + \frac{1}{\sqrt{N}}\right)$

(mild additional conditions apply)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$, ℓ_0 strongly convex, and g convex. Then, $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL) for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, i.e., with probability at least $1-\delta$,

$$\left|P^\star - \hat{D}^\star\right| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + (1+\Delta)^{3/2}\left(M\sqrt{\epsilon_0} + \epsilon\right)$$

$\epsilon_0 = M\nu$ $\qquad$ $\epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\qquad$ $\Delta = \max\left(\|\boldsymbol{\lambda}^\star\|_1, \|\hat{\boldsymbol{\lambda}}^\star\|_1, \|\tilde{\boldsymbol{\lambda}}^\star\|_1\right)$

Sources of error

parametrization richness (ν) $\qquad$ sample size (N) $\qquad$ requirements difficulty ($\boldsymbol{\lambda}^\star$)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$, ℓ_0 strongly convex, and g convex. Then, $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL) for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, i.e., with probability at least $1-\delta$,

$$\left|P^\star - \hat{D}^\star\right| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + (1+\Delta)^{3/2}\left(M\sqrt{\epsilon_0} + \epsilon\right)$$

$\epsilon_0 = M\nu$ $\qquad$ $\epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\qquad$ $\Delta = \max\left(\|\boldsymbol{\lambda}^\star\|_1, \|\hat{\boldsymbol{\lambda}}^\star\|_1, \|\tilde{\boldsymbol{\lambda}}^\star\|_1\right)$

Sources of error

parametrization richness (ν) $\qquad$ sample size (N) $\qquad$ requirements difficulty ($\boldsymbol{\lambda}^\star$)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

38

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$, ℓ_0 strongly convex, and g convex. Then, $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL) for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, i.e., with probability at least $1-\delta$,

$$\left|P^\star - \hat{D}^\star\right| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + (1+\Delta)^{3/2}\left(M\sqrt{\epsilon_0} + \epsilon\right)$$

$\epsilon_0 = M\nu$ $\qquad$ $\epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\qquad$ $\Delta = \max\left(\|\boldsymbol{\lambda}^\star\|_1, \|\hat{\boldsymbol{\lambda}}^\star\|_1, \|\tilde{\boldsymbol{\lambda}}^\star\|_1\right)$

Sources of error

parametrization richness (ν) $\qquad$ sample size (N) $\qquad$ requirements difficulty ($\boldsymbol{\lambda}^\star$)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

38

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$, ℓ_0 strongly convex, and g convex. Then, $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL) for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, i.e., with probability at least $1-\delta$,

$$\left|P^\star - \hat{D}^\star\right| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + (1+\Delta)^{3/2}\left(M\sqrt{\epsilon_0} + \epsilon\right)$$

$\epsilon_0 = M\nu$ $\qquad$ $\epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\qquad$ $\Delta = \max\left(\|\boldsymbol{\lambda}^\star\|_1, \|\hat{\boldsymbol{\lambda}}^\star\|_1, \|\tilde{\boldsymbol{\lambda}}^\star\|_1\right)$

Sources of error

parametrization richness (ν) $\qquad$ sample size (N) $\qquad$ requirements difficulty ($\boldsymbol{\lambda}^\star$)

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

38

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$, ℓ_0 strongly convex, and g convex. Then, $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL) for all $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ that achieve $\hat{D}^\star$, i.e., with probability at least $1-\delta$,

$$\left|P^\star - \hat{D}^\star\right| \le (1+\Delta)(\epsilon_0 + \epsilon)$$

$$\mathbb{E}\left[g\left(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\right)\right] \le c + (1+\Delta)^{3/2}\left(M\sqrt{\epsilon_0} + \epsilon\right)$$

$\epsilon_0 = M\nu$ $\qquad$ $\epsilon = B\sqrt{\frac{1}{N}\left[1 + \log\left(\frac{4m(2N)^{d_{\text{VC}}}}{\delta}\right)\right]}$ $\qquad$ $\Delta = \max\left(\|\boldsymbol{\lambda}^\star\|_1, \|\hat{\boldsymbol{\lambda}}^\star\|_1, \|\tilde{\boldsymbol{\lambda}}^\star\|_1\right)$

Sources of error

parametrization richness (ν) $\qquad$ sample size (N) $\qquad$ requirements difficulty ($\boldsymbol{\lambda}^\star$)

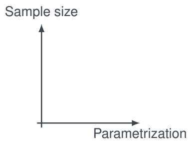
[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23; Elenter, Chamon, Ribeiro, ICLR'24]

38

Dual learning trade-offs

- Unconstrained learning

  parametrization $\times$ sample size



[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

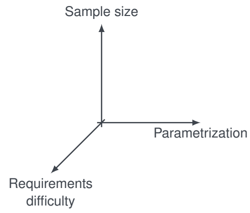
Dual learning trade-offs

- Unconstrained learning

  parametrization $\times$ sample size

- Constrained learning

  parametrization $\times$ sample size $\times$ requirements



[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

When is constrained learning possible?

Corollary

f_θ is PAC learnable $\approx^*$ f_θ is PACC learnable

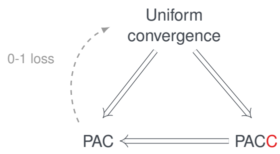
Constrained learning is **essentially as hard as** unconstrained learning

[mild conditions apply]

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

When is constrained learning possible?

Corollary



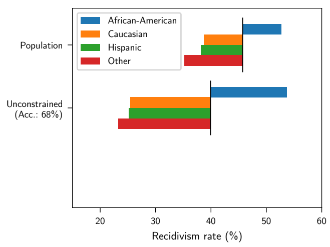
[mild conditions apply]

[Chamon and Ribeiro, NeurIPS'20; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness

Problem

Predict whether an individual will recidivate



*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \ \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1\right] + c,$$

for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

[Cotter et al., JMLR'19; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \ \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_\theta(\boldsymbol{x}_n) = 1\right] + c,$$

for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

[Cotter et al., JMLR'19; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: "Equality" of odds

Problem

Predict whether an individual will recidivate at the same rate across races

$$\min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \ \ \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_\theta(\boldsymbol{x}_n) - 0.5\big)\, \mathbb{I}\left[\boldsymbol{x}_n \in \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_\theta(\boldsymbol{x}_n) - 0.5\big) + c,$$

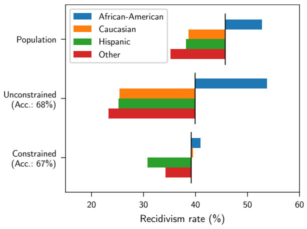
for Race $\in$ {African-American, Caucasian, Hispanic, Other}

*We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

[Cotter et al., JMLR'19; Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: “Equality” of odds

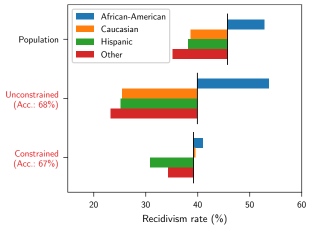
Problem
Predict whether an individual will recidivate at the same rate across races



*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: “Equality” of odds

Problem
Predict whether an individual will recidivate at the same rate across races



*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: “Equality” of odds

Prediction

	True label	Unconstrained 0	Unconstrained 1	Constrained 0	Constrained 1
African-American	0	31%	16%	36%	11%
	1	16%	37%	23%	30%
Caucasian	0	52%	9%	44%	17%
	1	23%	16%	16%	23%

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: “Equality” of odds

Prediction

	True label	Unconstrained 0	Unconstrained 1	Constrained 0	Constrained 1
African-American	0	31%	16%	36%	11%
	1	16%	37%	23%	30%
Caucasian	0	52%	9%	44%	17%
	1	23%	16%	16%	23%

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Fairness: “Equality” of odds

Prediction

	True label	Unconstrained 0	Unconstrained 1	Constrained 0	Constrained 1
African-American	0	31%	16%	36%	11%
	1	16%	37%	23%	30%
Caucasian	0	52%	9%	44%	17%
	1	23%	16%	16%	23%

*We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Agenda

Constrained learning theory

Constrained learning algorithms

Constrained optimization methods

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\theta}(x_n), y_n\big)$$

$$\text{subject to} \quad \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\theta}(x_m), y_m\big) \le c$$

$$h\big(f_{\theta}(x_r), y_r\big) \le u$$

Constrained optimization methods

$$\hat{P}^\star = \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\theta}(x_n), y_n\big)$$

$$\text{subject to} \quad \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\theta}(x_m), y_m\big) \le c$$

$$h\big(f_{\theta}(x_r), y_r\big) \le u$$

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
  - Tractability [non-convex constraints]
  - Feasible candidate solution
- Interior point methods
  e.g., barriers, projection, polyhedral approx.
  - Tractability [non-convex constraints]
  - Feasible candidate solution

Constrained optimization methods

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \quad \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c$$

$$h\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_r), y_r\big) \leq u$$

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution
- Interior point methods
  e.g., barriers, projection, polyhedral approx.
  - ✗ Tractability [non-convex constraints]
  - ✓ Feasible candidate solution
- Duality
  e.g., (augmented) Lagrangian
  - ✓ Tractability
  - ✓ (near-)feasible solution [small duality gap]

Dual learning algorithm

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^\dagger \in \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\operatorname{argmin}} \; \frac{1}{N}\sum_{n=1}^{N} \left[\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right)\right]$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

[Haeffele et al., CVPR'17; Ge et al., ICLR'18; Mei et al., PNAS'18; Kawaguchi et al., AISTATS'20...]

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

Dual learning algorithm

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

A (near-)PACC learner

Theorem
Suppose $\boldsymbol{\theta}^\dagger$ is a ρ-approximate solution of the regularized ERM:

$$\boldsymbol{\theta}^\dagger \approx \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\operatorname{argmin}} \; \frac{1}{N}\sum_{n=1}^{N} \Big(\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right)\Big).$$

Then, after $T = \left\lceil \frac{\|\boldsymbol{\lambda}^\star\|^2}{2\eta M \nu} \right\rceil + 1$ dual iterations with step size $\eta \leq \frac{2\epsilon}{mB^2}$,

the iterates $\big(\boldsymbol{\theta}^{(T)}, \boldsymbol{\lambda}^{(T)}\big)$ are such that

$$\Big|P^\star - L\Big(\boldsymbol{\theta}^{(T)}, \boldsymbol{\lambda}^{(T)}\Big)\Big| \leq (2 + \Delta)(\epsilon_0 + \epsilon) + \rho$$

with probability $1 - \delta$ over sample sets.

[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

In practice...

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ \approx \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

In practice...

- Minimize the primal ($\equiv$ **ERM**)

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \Big[\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big], \quad n = 1, 2, \dots, N$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_m), y_m\big) - c\right)\right]_+$$

$$\hat{D}^\star = \max_{\lambda \geq 0} \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \quad \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda \left[\frac{1}{N}\sum_{m=1}^{N} g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\right) - c\right]$$

In practice...

Initialize: $\boldsymbol{\theta}_0, \lambda_0$
for $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  for $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) \right]$
  end
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_n\big) - c \right) \right]_+$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$

SGD

Dual update

PyTorch

https://github.com/lfochamon/csl

In practice...

Initialize: $\boldsymbol{\theta}_0, \lambda_0$
for $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  for $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) \right]$
  end
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_n\big) - c \right) \right]_+$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$

Use adaptive method (e.g., ADAM)

PyTorch

https://github.com/lfochamon/csl

In practice...

Initialize: $\boldsymbol{\theta}_0, \lambda_0$
for $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  for $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) \right]$
  end
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_n\big) - c \right) \right]_+$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$

Use adaptive method (e.g., ADAM)
Use different time-scales ($\eta_\lambda = 0.1\eta_\theta$)

PyTorch

https://github.com/lfochamon/csl

In practice...

Initialize: $\boldsymbol{\theta}_0, \lambda_0$
for $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  for $n = 1, \dots, N$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta_\theta \nabla_{\boldsymbol{\beta}} \left[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) \right]$
  end
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_n\big) - c \right) \right]_+$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$

Check slack:
- feasibility: $s_t \leq 0$
- "duality gap": $\lambda_t s_t$

$$s_t = \frac{1}{N} \sum_{n=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n), y_n\big) - c$$

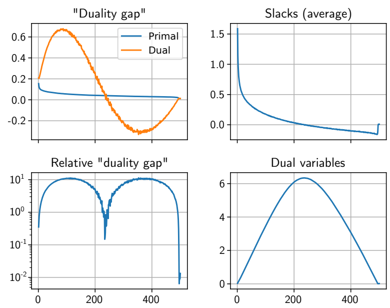
Use adaptive method (e.g., ADAM)
Use different time-scales ($\eta_\lambda = 0.1\eta_\theta$)

PyTorch

https://github.com/lfochamon/csl

In practice...

Initialize: $\boldsymbol{\theta}_0, \lambda_0$
for $t = 1, \dots, T$
  $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
  for $n = 1, \dots, l$
    $\boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n$
  end
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \big[\lambda_{t-1} +$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$



ethod (e.g., ADAM)
ne-scales ($\eta_\lambda = 0.1\eta_\theta$)

rch

https://github.com/lfochamon/csl

Penalty-based vs. dual learning

Penalty-based learning

$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta}} \ \text{Loss}(\boldsymbol{\theta}) + \lambda \cdot \text{Penalty}(\boldsymbol{\theta})$$

- Parameter: λ (data-dependent)
- Generalizes with respect to Loss + λPenalty

Dual learning

$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta}} \ \text{Loss}(\boldsymbol{\theta}) + \lambda \cdot \text{Penalty}(\boldsymbol{\theta})$$

$$\lambda^+ = \Big[\lambda + \eta \Big(\text{Penalty}(\boldsymbol{\theta}^\dagger) - c \Big) \Big]_+$$

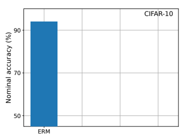
- Parameter: c (requirement-dependent)
- Generalizes with respect to Loss and Penalty $\leq c$

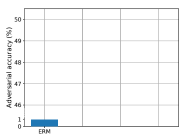
Robust learning

Problem
Learn an accurate classifier that is robust to input perturbations

Cello

Hammer





Adversarial training

Problem
Learn an accurate classifier that is robust to input perturbations

- Adversarial training (e.g., [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18])

$$\min_{\boldsymbol{\theta}} \ \frac{1}{N} \sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \longrightarrow \min_{\boldsymbol{\theta}} \ \frac{1}{N} \sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \ \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right]$$

Adversarial training

Problem
Learn an accurate classifier that is robust to input perturbations

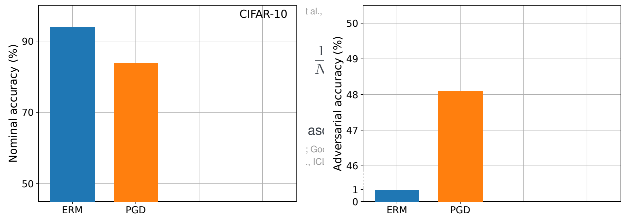
- Adversarial training (e.g., [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18])

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \longrightarrow \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\Big]$$

$\approx$ gradient ascent
[Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]

Adversarial training

Problem
Learn an accurate classifier that is robust to input perturbations



Adversarial training

Problem
Learn an accurate classifier that is robust to input perturbations

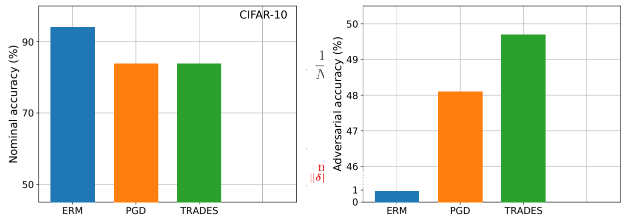
- Adversarial training (e.g., [Zhang et al., ICML'19])

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \longrightarrow \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\Big]$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda\Big[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\Big]$$

Adversarial training

Problem
Learn an accurate classifier that is robust to input perturbations



Constrained learning for robustness

Problem
Learn an accurate classifier that is robust to input perturbations

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\Big] \leq c \end{aligned}$$

[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]

Constrained learning for robustness

Problem
Learn an accurate classifier that is robust to input perturbations

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\Big] \leq c \end{aligned}$$

[**C.** and Ribeiro, NeurIPS'20; Robey*, **C.***, Pappas, Hassani, and Ribeiro, NeurIPS'21; **C.**, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness

Problem
Learn an accurate classifier that is robust to input perturbations

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} u_n \leq c \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) \leq u_n, \quad \text{for all } \|\boldsymbol{\delta}_n\|_\infty \leq \epsilon \end{aligned}$$

[**C.** and Ribeiro, NeurIPS'20; Robey*, **C.***, Pappas, Hassani, and Ribeiro, NeurIPS'21; **C.**, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness

Problem
Learn an accurate classifier that is robust to input perturbations

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} u_n \leq c \end{aligned}$$

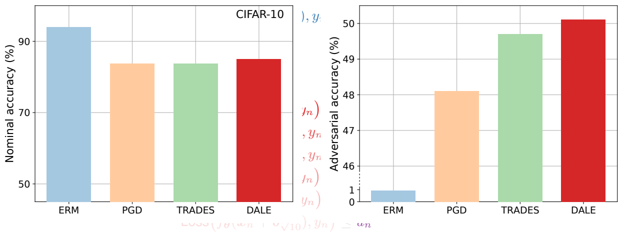
Sampling (e.g., LMC)

$$\begin{aligned} & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_0), y_n\big) \leq u_n \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\sqrt{2}}), y_n\big) \leq u_n \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\sqrt{3}}), y_n\big) \leq u_n \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{e}), y_n\big) \leq u_n \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\pi}), y_n\big) \leq u_n \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\sqrt{10}}), y_n\big) \leq u_n \end{aligned}$$

[**C.** and Ribeiro, NeurIPS'20; Robey*, **C.***, Pappas, Hassani, and Ribeiro, NeurIPS'21; **C.**, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness

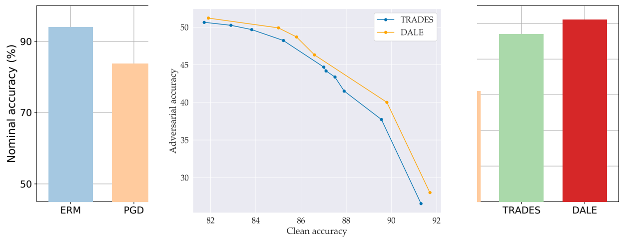
Problem
Learn an accurate classifier that is robust to input perturbations



[C. and Ribeiro, NeurIPS'20; Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21; C., Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

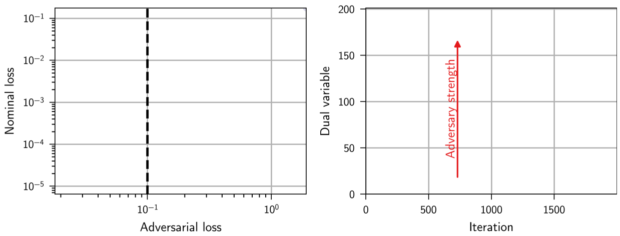
Constrained learning for robustness

Problem
Learn an accurate classifier that is robust to input perturbations



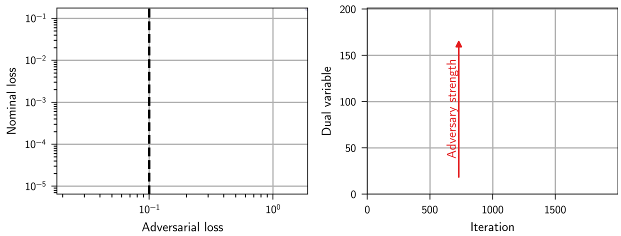
[C. and Ribeiro, NeurIPS'20; Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21; C., Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness



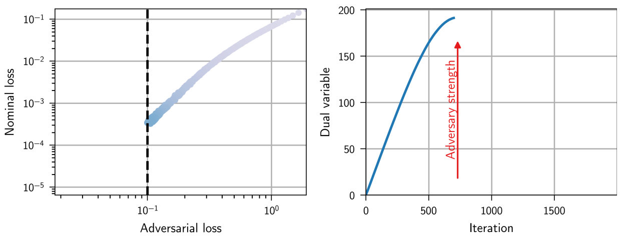
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness



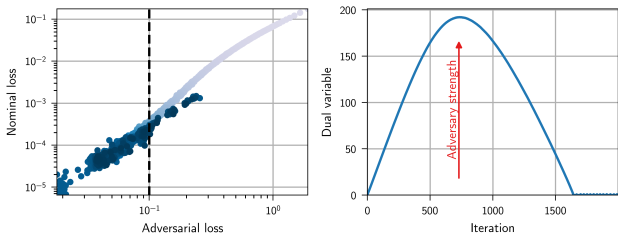
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness



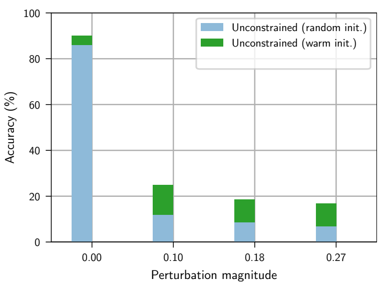
[Chamon, Paternain, Calvo-Fullana, Ribeiro, IEEE TIT'23]

Constrained learning for robustness



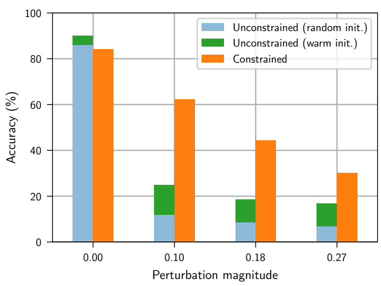
Empirical observations: [Zhang et al., ICML'20; Sitawarin, arXiv'20]

Constrained learning for robustness



[Chamon et al., IEEE TIT'23]

Constrained learning for robustness



[Chamon et al., IEEE TIT'23]

Penalty-based vs. dual learning

Penalty-based learning

$$\theta^\dagger \in \underset{\theta}{\operatorname{argmin}}\ \text{Loss}(\theta) + \lambda \cdot \text{Penalty}(\theta)$$

- Parameter: λ (data-dependent)
- Generalizes with respect to Loss $+\ \lambda$Penalty

Dual learning

$$\theta^\dagger \in \underset{\theta}{\operatorname{argmin}}\ \text{Loss}(\theta) + \lambda \cdot \text{Penalty}(\theta)$$

$$\lambda^+ = \left[\lambda + \eta\left(\text{Penalty}(\theta^\dagger) - c\right)\right]_+$$

- Parameter: c (requirement-dependent)
- Generalizes with respect to Loss and Penalty $\leq c$

Summary

- **Constrained learning is ~~the~~ a tool to learn under requirements**
- **Constrained learning is hard...**
- **...but possible. How?**

Summary

- **Constrained learning is ~~the~~ a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
- **...but possible. How?**

Summary

- **Constrained learning is ~~the~~ a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
  Constrained, non-convex, statistical optimization problem
- **...but possible. How?**

Summary

- **Constrained learning is ~~the~~ a tool to learn under requirements**
  Constrained learning imposes generalizable requirements organically during training, e.g., fairness [Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23], heterogeneity [Shen et al., ICRL'22]...
- **Constrained learning is hard...**
  Constrained, non-convex, statistical optimization problem
- **...but possible. How?**
  We can learn under requirements (essentially) whenever we can learn at all by solving *(penalized) ERM problems.*

Robust/resilient constraints

Agenda

Resilient constrained learning

Semi-infinite learning

Probabilistic robustness

Agenda

Resilient constrained learning

Semi-infinite learning

Probabilistic robustness

Heterogeneous federated learning

Problem
Learn a common model using data from K clients that is good for all clients

$$\min_{\theta} \ \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta)$$
$$\text{subject to} \quad \text{Loss}_k(f_\theta) \le \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta) + c$$
$$k = 1, \dots, K$$

- k-th client loss: $\text{Loss}_k(\phi) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(x_{n_k}), y_{n_k}\big)$

Heterogeneous federated learning

Problem
Learn a common model using data from K clients that is good for all clients

$$\min_{\theta} \ \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta)$$
$$\text{subject to} \quad \text{Loss}_k(f_\theta) \le \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_\theta) + c_k$$
$$k = 1, \dots, K$$

- k-th client loss: $\text{Loss}_k(\phi) = \frac{1}{N_k}\sum_{n_k=1}^{N_k} \text{Loss}\big(f_\theta(x_{n_k}), y_{n_k}\big)$

Resilient constrained learning

Definition (Resilience)
(ecology) ability of an ecosystem to adapt its function to accommodate operating conditions

Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$P^\star = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$
$$\text{subject to} \quad \mathbb{E}_{(x,y)\sim\mathfrak{A}_i}\Big[g_i\big(f_\theta(x_m), y_m\big)\Big] \le c_i$$

Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$P^\star(r) = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$
$$\text{subject to} \quad \mathbb{E}_{(x,y)\sim\mathfrak{A}_i}\Big[g_i\big(f_\theta(x_m), y_m\big)\Big] \le c_i + r_i$$

Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$P^\star(r) = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$
$$\text{subject to} \quad \mathbb{E}_{(x,y)\sim\mathfrak{A}_i}\Big[g_i\big(f_\theta(x_m), y_m\big)\Big] \le c_i + r_i$$

- Larger relaxations r decrease the objective $P^\star(r)$ (benefit), but increase specification violation $c_i + r_i$ (cost)

Resilient constrained learning

Definition (Resilience)
~~(ecology)~~ ability of an ~~ecosystem~~ to adapt its ~~function~~ to accommodate ~~operating conditions~~
(learning) learning system specification data properties

$$P^\star(r) = \min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x), y\big)\Big]$$
$$\text{subject to} \quad \mathbb{E}_{(x,y)\sim\mathfrak{A}_i}\Big[g_i\big(f_\theta(x_m), y_m\big)\Big] \le c_i + r_i$$

- Larger relaxations r decrease the objective $P^\star(r)$ (benefit), but increase specification violation $c_i + r_i$ (cost)
- Resilience is a compromise!

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

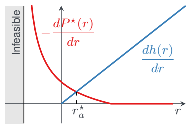
[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing



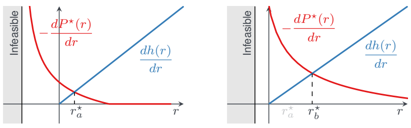
[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing



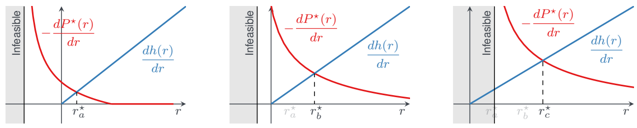
[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) \quad \leftarrow (\partial\text{: subdifferential})$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing



[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) = \boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)

[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

$$\nabla h(\boldsymbol{r}^\star) \in -\partial P^\star(\boldsymbol{r}^\star) = \boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)
- The resilient equilibrium *exists and is unique* (because h is strictly convex)

[Hounie, Chamon, Ribeiro, NeurIPS'23]

Resilient constrained learning

Definition (Resilient equilibrium)
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

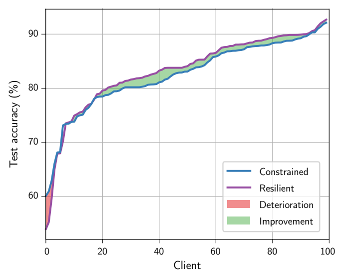
$$P^\star(\boldsymbol{r}^\star) = \min_{\boldsymbol{\theta},\boldsymbol{r}} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\big)\Big] + h(\boldsymbol{r})$$
$$\text{subject to} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\Big[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m),y_m\big)\Big] \leq c_i + r_i$$

In words: at the resilient equilibrium the marginal cost of relaxing equals the marginal gain of relaxing

- After relaxing, $\boldsymbol{\lambda}^\star(\boldsymbol{r}^\star)$ is *smaller* than $\boldsymbol{\lambda}^\star(\mathbf{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)
- The resilient equilibrium *exists and is unique* (because h is strictly convex)

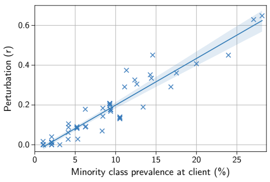
[Hounie, Chamon, Ribeiro, NeurIPS'23]

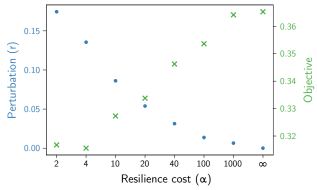
Heterogeneous federated learning



[Hounie, Chamon, Ribeiro, NeurIPS'23]

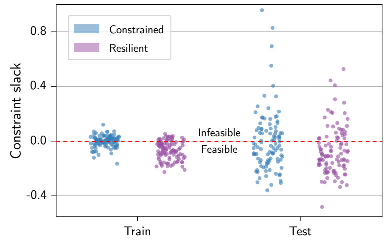
Heterogeneous federated learning





[Hounie, Chamon, Ribeiro, NeurIPS'23]

Heterogeneous federated learning



[Hounie, Chamon, Ribeiro, NeurIPS'23]

Agenda

Resilient constrained learning

Semi-infinite learning

Probabilistic robustness

Semi-infinite constrained learning

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta} \in \Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right]$$

Semi-infinite constrained learning

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N} \sum_{n=1}^{N} \left[t(\boldsymbol{x}_n, y_n) \right] \\ \text{subject to} \quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \leq t(\boldsymbol{x}_n, y_n), \\ & \text{for all } (\boldsymbol{x}_n, y_n) \text{ and } \boldsymbol{\delta} \in \Delta \end{aligned}$$

- Epigraph formulation:

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t \iff \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t, \text{ for all } \|\boldsymbol{\delta}\|_\infty \leq \epsilon$$

Semi-infinite constrained learning

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N} \sum_{n=1}^{N} \left[t(\boldsymbol{x}_n, y_n) \right] \\ \text{subject to} \quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_0), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\sqrt{2}}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_e), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_\pi), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_4), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{e^2}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{\pi^e}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{e^\pi}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\ & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_{25}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \end{aligned}$$

- Epigraph formulation:

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t \iff \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \leq t, \text{ for all } \|\boldsymbol{\delta}\|_\infty \leq \epsilon$$

- Semi-infinite program

Duality

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta} \in \Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right]$$

$$\Big\updownarrow =$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[t(\boldsymbol{x}_n, y_n) \right] \text{ s. to } \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \leq t(\boldsymbol{x}_n, y_n), \forall (\boldsymbol{x}_n, y_n, \boldsymbol{\delta})$$

$$\Big\updownarrow =$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu \in \mathcal{P}} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \int_\Delta \mu_n(\boldsymbol{\delta}) \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) d\boldsymbol{\delta}}_{L(\theta, \mu_n)}$$

Duality

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[\max_{\boldsymbol{\delta} \in \Delta} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right]$$

$$\Big\updownarrow =$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[t(\boldsymbol{x}_n, y_n) \right] \text{ s. to } \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \leq t(\boldsymbol{x}_n, y_n), \forall (\boldsymbol{x}_n, y_n, \boldsymbol{\delta})$$

$$\Big\updownarrow =$$

$$\min_{\boldsymbol{\theta}} \sup_{\mu \in \mathcal{P}} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{\delta} \sim \mu(\cdot | \boldsymbol{x}_n, y_n)} \left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right]}_{L(\theta, \mu_n)}$$

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For all $\epsilon > 0$, there exists $\gamma(x, y) < \max_{\delta\in\Delta}$ Loss$\big(f_{\theta}(x+\delta), y\big)$ s.t. $L(\theta,\mu_{\gamma}) \geq \sup_{\mu\in\mathcal{P}^2} L(\theta,\mu) - \xi$ for

$$\mu_{\gamma}(\delta|x,y) \propto \Big[\ell\big(f_{\theta}(x+\delta), y\big) - \gamma(x,y)\Big]_{+}$$

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

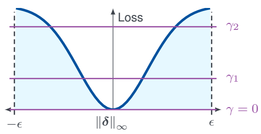
$$\updownarrow \approx$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For any approximation error, $\exists\, \gamma(x, y)$ such that

$$\mu_{\gamma}(\delta|x,y) \propto \Big[\text{Loss}\big(f_{\theta}(x+\delta), y\big) - \gamma(x,y)\Big]_{+}$$



[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

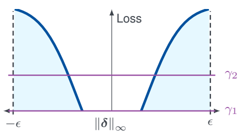
$$\updownarrow \approx$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For any approximation error, $\exists\, \gamma(x, y)$ such that

$$\mu_{\gamma}(\delta|x,y) \propto \Big[\text{Loss}\big(f_{\theta}(x+\delta), y\big) - \gamma(x,y)\Big]_{+}$$



[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

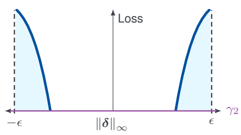
$$\updownarrow \approx$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For any approximation error, $\exists\, \gamma(x, y)$ such that

$$\mu_{\gamma}(\delta|x,y) \propto \Big[\text{Loss}\big(f_{\theta}(x+\delta), y\big) - \gamma(x,y)\Big]_{+}$$



[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

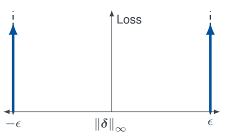
$$\updownarrow =$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For any approximation error, $\exists\, \gamma(x, y)$ such that

$$\mu_{\gamma}(\delta|x,y) \propto \Big[\text{Loss}\big(f_{\theta}(x+\delta), y\big) - \gamma(x,y)\Big]_{+}$$



[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

From optimization to sampling

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

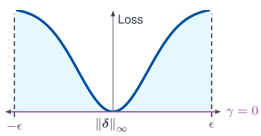
$$\updownarrow \approx$$

$$\min_{\theta} \sup_{\mu\in\mathcal{P}^2} \underbrace{\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\delta\sim\mu_{\gamma}(\cdot|x_n,y_n)}\left[\text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]}_{L(\theta,\mu)}$$

Proposition

For any approximation error, $\exists\, \gamma(x, y)$ such that

$$\mu_{0}(\delta|x,y) \propto \text{Loss}\big(f_{\theta}(x+\delta), y\big)$$



[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

Constrained learning for robustness

Problem

Learn an image classifier that is robust to input perturbations

$$\max_{\lambda\geq 0} \min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) + \lambda\left[\max_{\delta\in\Delta} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning

✗ Computing the worst-case perturbations

- gradient ascent → non-convex, underparametrized

Constrained learning for robustness

Problem

Learn an image classifier that is robust to input perturbations

$$\max_{\lambda\geq 0} \min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) + \lambda\left[\cancel{\max_{\delta\in\Delta}}^{\,\mathbb{E}_{\delta\sim\mu_0(\cdot|x_n,y_n)}} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right]$$

✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning

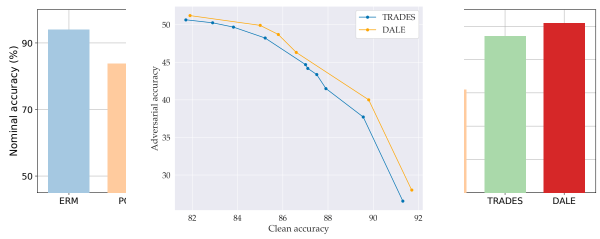
✓ Computing the worst-case perturbations

- ~~gradient ascent → non-convex, underparametrized~~ ⇒ sampling

Dual Adversarial LEarning

Problem

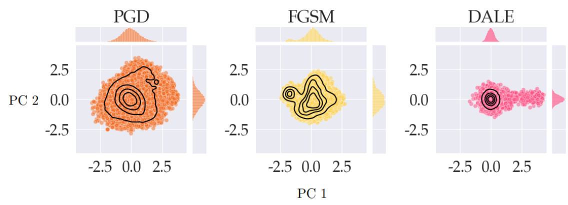
Learn an image classifier th



Dual Adversarial LEarning

Problem

Learn an image classifier that is robust to input perturbations



Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

HMC sampling: $\delta \sim \mu_0(\cdot|x_n, y_n)$

SGD

GA

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

HMC sampling: $\delta \sim \mu_0(\cdot|x_n, y_n)$

SGD

GA

[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21] 79

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

HMC sampling: $\delta \sim \mu_0(\cdot|x_n, y_n)$

SGD

GA

[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21] 79

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

HMC sampling: $\delta \sim \mu_0(\cdot|x_n, y_n)$

SGD

GA

[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21] 79

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

HMC sampling: $\delta \sim \mu(\cdot|x_n, y_n)$

SGD

GA

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\delta_n \sim$ Random(Δ)

  for $k = 1, \dots, K$:

    $\zeta \sim$ Laplace$(0, I)$

    $\delta_n \leftarrow \text{proj}_{\Delta}\left[\delta_n + \eta\,\text{sign}\left[\nabla_\delta \log\left(\text{Loss}\big(f_{\theta_t}(x_n + \delta_n), y_n\big)\right)\right] + \sqrt{2\eta T}\zeta\right]$

  end

  $\theta \leftarrow \theta - \eta \nabla_\theta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big)\right]$

end

$\lambda \leftarrow \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(x_n + \delta_n), y_n\big) - c\right)\right]_+$

Gaussian
[Lopes et al., arXiv'19]
[Rusak et al., ECCV'20]

Patches
[Zhong et al., AAAI'20]
[Yun et al., ICCV'19]

...

SGD

GA

Dual Adversarial LEarning

for $n = 1, \dots, N$:

  $\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$

  for $k = 1, \dots, K$:

    $\zeta \sim \text{Laplace}(0, I)$

    $\boldsymbol{\delta}_n \leftarrow \underset{\Delta}{\operatorname{proj}} \left[\boldsymbol{\delta}_n + \eta \operatorname{sign} \left[\nabla_{\boldsymbol{\delta}} \log \left(\text{Loss}\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) \right) \right] + \sqrt{2\eta T}\zeta \right]$

  end

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) + \lambda \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) \right]$

end

$\lambda \leftarrow \left[\lambda + \eta \left(\frac{1}{N} \sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}_n), y_n\big) - c \right) \right]_+$

$T \to 0$: "PGD"

[Szegedy et al., ICLR'14]

[Goodfellow et al., ICLR'15]

[Madry et al., ICLR'18]

SGD

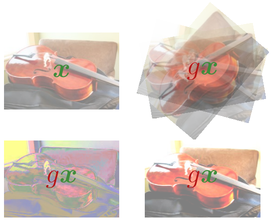
GA

[Robey*, Chamon*, Pappas, Hassani, Ribeiro, NeurIPS'21]

Invariance

Problem

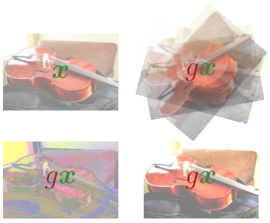
Learn a classifier that is invariant to transformation $g \in \mathcal{G}$



Invariance

Problem

Learn a classifier that is invariant to transformation $g \in \mathcal{G}$



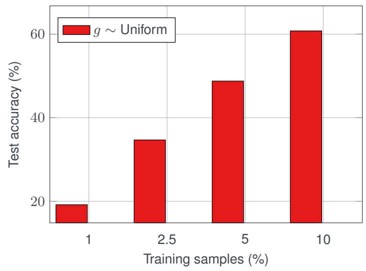
$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{g \sim \mathfrak{m}} \left[\text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \right]$$

$\mathcal{G} =$

- Identity
- ShearX(Y), Flip, Rotate, TranslateX(Y), Cutout, Crop
- AutoContrast, Invert, Equalize, Color, Solarize, Posterize, Contrast, Brightness, Sharpness

[Hounie, Chamon, Ribeiro, ICML'23]

Training on a subset of ImageNet-100



[Hounie, Chamon, Ribeiro, ICML'23]

Invariance

Problem

Learn a classifier that is invariant to transformation $g \in \mathcal{G}$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{g \sim \mathfrak{m}} \left[\text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \right]$$

$\mathcal{G} =$

- Identity
- ShearX(Y), Flip, Rotate, TranslateX(Y), Cutout, Crop
- AutoContrast, Invert, Equalize, Color, Solarize, Posterize, Contrast, Brightness, Sharpness

[Hounie, Chamon, Ribeiro, ICML'23]

Invariance

Problem

Learn a classifier that is invariant to transformation $g \in \mathcal{G}$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[\max_{g \in \mathcal{G}} \text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \right]$$

$\mathcal{G} =$

- Identity
- ShearX(Y), Flip, Rotate, TranslateX(Y), Cutout, Crop
- AutoContrast, Invert, Equalize, Color, Solarize, Posterize, Contrast, Brightness, Sharpness

[Hounie, Chamon, Ribeiro, ICML'23]

Invariance

Problem

Learn a classifier that is invariant to transformation $g \in \mathcal{G}$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left[\cancel{\max_{g \in \mathcal{G}}}^{\mathbb{E}_{g \sim \mu_0(\cdot \mid \boldsymbol{x}_n, y_n)}} \text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \right]$$

$\mathcal{G} =$

- Identity
- ShearX(Y), Flip, Rotate, TranslateX(Y), Cutout, Crop
- AutoContrast, Invert, Equalize, Color, Solarize, Posterize, Contrast, Brightness, Sharpness

[Hounie, Chamon, Ribeiro, ICML'23]

Invariance

Problem

Learn a classifier that is invariant to transformation $g \in \mathcal{G}$

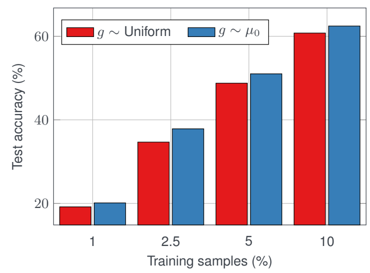
$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N} \sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N} \sum_{n=1}^{N} \left[\cancel{\max_{g \in \mathcal{G}}}^{\mathbb{E}_{g \sim \mu_0(\cdot \mid \boldsymbol{x}_n, y_n)}} \text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \right] \leq c \end{aligned}$$

$\mathcal{G} =$

- ~~Identity~~
- ShearX(Y), Flip, Rotate, TranslateX(Y), Cutout, Crop
- AutoContrast, Invert, Equalize, Color, Solarize, Posterize, Contrast, Brightness, Sharpness

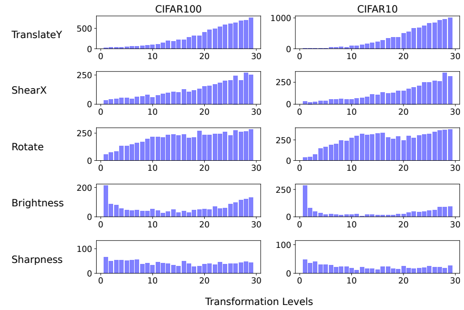
[Hounie, Chamon, Ribeiro, ICML'23]

Training on a subset of ImageNet-100

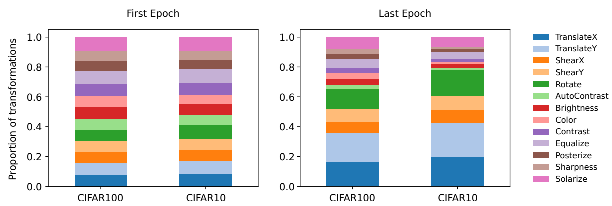


[Hounie, Chamon, Ribeiro, ICML'23]

Not all transformations are created equal



Not all transformations are created equal



"Identifying" invariances

		Synthetic Invariance		
Dataset	Dual variable (λ)	Rotation	Translation	Scale
MNIST	Rotation	**0.000**	2.724	0.012
	Translation	1.218	**0.439**	0.006
	Scale	2.026	4.029	**0.003**
F-MNIST	Rotation	**0.000**	3.301	1.352
	Translation	3.572	**0.515**	**0.441**
	Scale	4.144	2.725	0.904

[Hounie, Chamon, Ribeiro, ICML'23]

Agenda

Resilient constrained learning

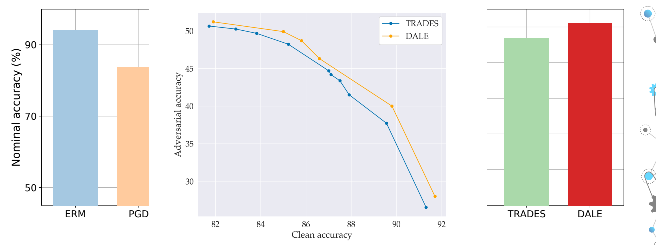
Semi-infinite learning

Probabilistic robustness

Constrained learning for robustness

Problem

Learn an accurate classifier



[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]

Constrained learning for robustness

Problem

Learn an accurate classifier that is (mostly) robust to input perturbations

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big) \right] \leq c \end{aligned}$$

[Chamon and Ribeiro, NeurIPS'20; Robey et al., NeurIPS'21; Chamon et al., IEEE TIT'23]

"Softer" robustness

- Softmax or *log-sum-exp* [Li et al., ICLR'21]

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{(x,y)} \left[\frac{1}{\tau} \log \left(\mathbb{E}_{\boldsymbol{\delta} \sim \mathfrak{m}} \left[e^{\tau \cdot \text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\right)} \right] \right) \right]$$

  - $\tau \to 0$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

- L_p norms [Rice et al., NeurIPS'21]

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{(x,y)} \left[\mathbb{E}_{\boldsymbol{\delta} \sim \mathfrak{m}} \left[\left| \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big) \right|^{\tau} \right]^{1/\tau} \right]$$

  - $\tau = 1$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

“Softer” robustness

- Softmax or *log-sum-exp* [Li et al., ICLR'21]

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\frac{1}{\tau}\log\left(\mathbb{E}_{\delta\sim\mathfrak{m}}\left[e^{\tau\cdot\text{Loss}\left(f_{\theta}(x+\delta),y\right)}\right]\right)\right]$$

- L_p norms [Rice et al., NeurIPS'21]

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\mathbb{E}_{\delta\sim\mathfrak{m}}\left[\left|\text{Loss}\left(f_{\theta}(x+\delta),y\right)\right|^{\tau}\right]^{1/\tau}\right]$$

✖ Computationally challenging (especially as $\tau \to \infty$, i.e., stronger robustness)

✖ No guaranteed advantages (lower sample complexity? improved trade-offs?)

Towards probabilistic robustness

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N}\left[t(x_n,y_n)\right] \\ \text{subject to} \quad & \text{Loss}\left(f_{\theta}(x_n+\delta_0),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_1),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{\sqrt{2}}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_e),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{\pi}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_4),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{e^2}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{\pi^e}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{e^\pi}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{25}),y_n\right) \leq t(x_n,y_n) \end{aligned}$$

Towards probabilistic robustness

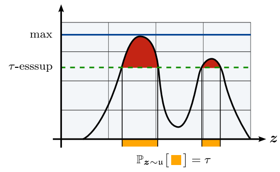
$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N}\left[t(x_n,y_n)\right] \\ \text{subject to} \quad & \text{Loss}\left(f_{\theta}(x_n+\delta_0),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_1),y_n\right) \leq t(x_n,y_n) \\ & \cancel{\text{Loss}\left(f_{\theta}(x_n+\delta_{\sqrt{2}}),y_n\right) \leq t(x_n,y_n)} \\ & \cancel{\text{Loss}\left(f_{\theta}(x_n+\delta_e),y_n\right) \leq t(x_n,y_n)} \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{\pi}),y_n\right) \leq t(x_n,y_n) \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_4),y_n\right) \leq t(x_n,y_n) \\ & \cancel{\text{Loss}\left(f_{\theta}(x_n+\delta_{e^2}),y_n\right) \leq t(x_n,y_n)} \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{\pi^e}),y_n\right) \leq t(x_n,y_n) \\ & \cancel{\text{Loss}\left(f_{\theta}(x_n+\delta_{e^\pi}),y_n\right) \leq t(x_n,y_n)} \\ & \text{Loss}\left(f_{\theta}(x_n+\delta_{25}),y_n\right) \leq t(x_n,y_n) \end{aligned}$$

Probabilistic robustness

- Probabilistic robustness

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\tau\text{-esssup}_{\delta\in\Delta} \text{Loss}\left(f_{\theta}(x+\delta),y\right)\right]$$

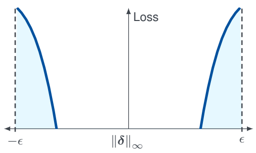
  - $\tau = 1/2$: classical learning (for symmetric $\mathfrak{m}$)
  - $\tau = 0$: adversarial robustness (ess sup)



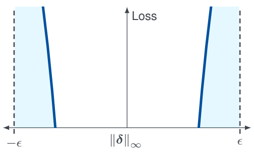
[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

Probabilistic robustness

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\xcancel{\max_{\delta\in\Delta}} \text{Loss}\left(f_{\theta}(x+\delta),y\right)\right] \quad \mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}$$



$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\xcancel{\tau\text{-esssup}_{\delta\in\Delta}} \text{Loss}\left(f_{\theta}(x+\delta),y\right)\right] \quad \mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}$$



[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

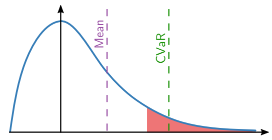
Probabilistic robustness and Risk

- Conditional value at risk:

$$\text{CVaR}_{\rho}(f) = \mathbb{E}_z\left[f(z) \mid f(z) \geq F_z^{-1}(\rho)\right]$$

$$= \inf_{\alpha\in\mathbb{R}} \alpha + \frac{\mathbb{E}_z\left[[f(z)-\alpha]_+\right]}{1-\rho}$$

  - $\text{CVaR}_0(f) = \mathbb{E}_z[f(z)]$
  - $\text{CVaR}_1(f) = \text{ess}\sup_z f(z)$



Proposition

CVaR is the tightest *convex* upper bound of τ-esssup, i.e.,
$\tau\text{-esssup}_z f(z) \leq \text{CVaR}_{1-\tau}(f)$ with equality when $\rho = 0$ or $\rho = 1$.

[Shapiro et al. Lectures on Stochastic Programming, 2014; Kalogerias et al., IEEE ICASSP'20]

Probabilistically robust learning

for $n = 1, \dots, N$:

$\alpha_0 = 0$
for $t = 1, \dots, T$:
$\delta_t \sim \text{Random}(\Delta)$
$\alpha \leftarrow \alpha - \frac{\eta}{\tau}\left(\tau - \mathbb{I}\left[\text{Loss}\left(f_{\theta}(x_n+\delta_t),y_n\right) \geq \alpha\right]\right)$
end — SGD (CVaR)

$$\theta \leftarrow \theta - \eta\nabla_{\theta}\underbrace{\left[\text{Loss}\left(f_{\theta}(x_n+\delta_T),y_n\right)-\alpha\right]_+}_{\approx\text{CVaR}_{1-\tau}\left[\text{Loss}\left(f_{\theta}(x_n+\delta),y_n\right)\right]}$$ — SGD (θ)

end

[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

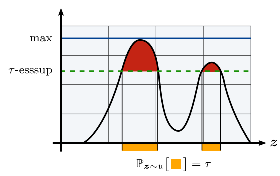
Probabilistic robustness

- Probabilistic robustness

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\tau\text{-esssup}_{\delta\in\Delta} \text{Loss}\left(f_{\theta}(x+\delta),y\right)\right]$$

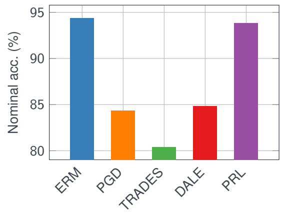
  - $\tau = 1/2$: classical learning (for symmetric $\mathfrak{m}$)
  - $\tau = 0$: adversarial robustness (ess sup)

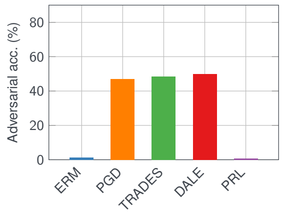
    - Potentially better sample complexity
      [Robey et al., ICML'22 (spotlight)] ✔
      [Raman et al., NeurIPS ML Safety Workshop'22] ✔ ✖
    - Better performance trade-off
      [Robey et al., ICML'22 (spotlight)] ✔



[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

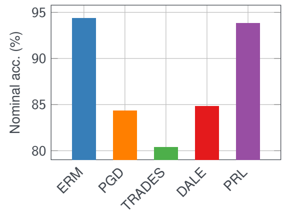
Probabilistically robust learning

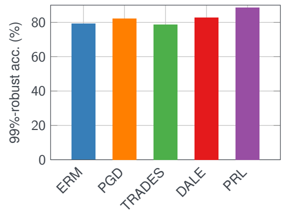




[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

Probabilistically robust learning





[Robey, Chamon, Pappas, Hassani, ICML'22 (spotlight)]

Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
- **Semi-infinite constrained learning...**
- **...but possible. How?**

Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]...
- **Semi-infinite constrained learning...**
- **...but possible. How?**

Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]...
- **Semi-infinite constrained learning...**
  Learning problem with an infinite number of constraints
- **...but possible. How?**

Summary

- **Semi-infinite constrained learning is the a tool to enforce worst-case requirements**
  e.g., robustness [Robey et al., NeurIPS'21], invariance [Hounie et al., ICML'23], smoothness [Cerviño et al., ICML'23]...
- **Semi-infinite constrained learning...**
  Learning problem with an infinite number of constraints
- **...but possible. How?**
  Using a hybrid sampling–optimization algorithm or, in the case of probabilistic robustness, a *tight* convex relaxation (CVaR) [Robey et al., ICML'22]

Agenda

I. Constrained supervised learning
   - Constrained learning theory
   - Constrained learning algorithms
   - Resilient constrained learning

   Break (10 min)

II. Constrained reinforcement learning
   - Constrained RL duality
   - Constrained RL algorithms

   Q&A and discussions

https://luizchamon.com/sgm