Miguel Calvo-Fullana
Universitat Pompeu Fabra, Spain

Luiz F. O. Chamon
Universität Stuttgart, Germany

Santiago Paternain
Rensselaer Polytechnic Institute, USA

Alejandro Ribeiro
University of Pennsylvania, USA

AAAI tutorial
Feb. 20, 2023

supervised and reinforcement learning under requirements

Agenda

I. Constrained supervised learning

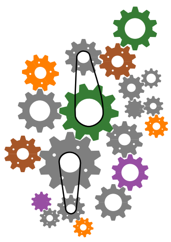
II. Robustness-constrained learning

Break (30 min)

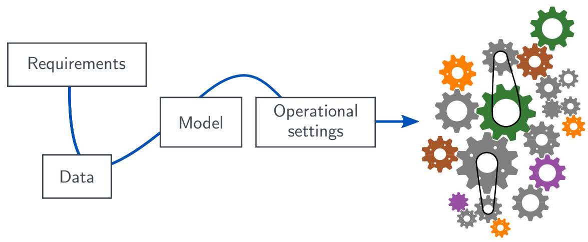
III. Constrained reinforcement learning

https://luizchamon.com/aaai

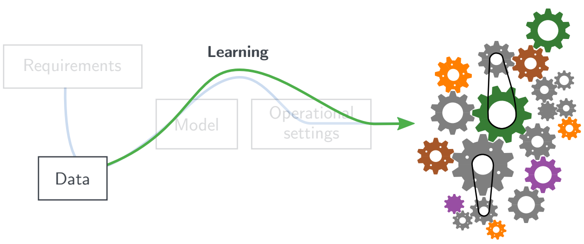
Why requirements?



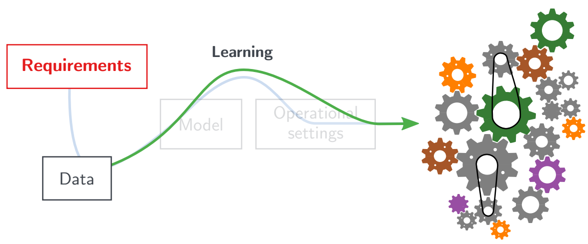
Why requirements?



Why requirements?

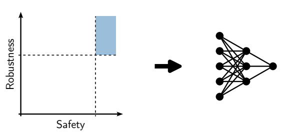


Why requirements?



What is a requirements?

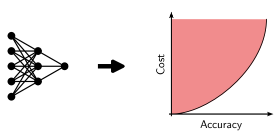
- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide



[NASA. "Systems engineering handbook," 2019]

What is a requirements?

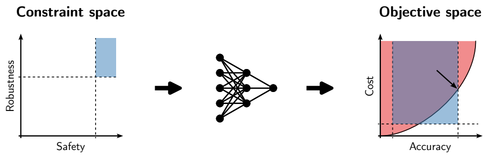
- **Requirements are "shall" statements:** describe *necessary* features subject to verification
  - *Constraint space*: things we decide
- **Goals are "should" statements:** express recommendations (once "shall" statements are satisfied)
  - *Objective space*: things the system achieves



[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?

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

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- f_θ 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?

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

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- f_θ 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?

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

$$\Updownarrow$$

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

What about penalties?

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

NON-CONVEX

$$\not\Updownarrow$$

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

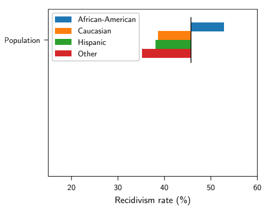
- ✗ There may not exist (λ, μ) such that the penalized solution is optimal *and* feasible
- ✗ Even if such (λ, μ) exist, they are not easy to find (hyperparameter search, cross-validation...)
- ✓ Constrained learning yields better guarantees, better performance, better trade-offs...

Applications

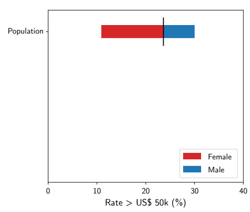
- Fairness
  (e.g., [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; Chamon et al., IEEE TIT'23])
- Federated learning
  (e.g., [Shen et al., ICLR'22; Hounie et al., NeurIPS'23])
- Adversarially robust learning
  (e.g., [Chamon et al., NeurIPS'20; Robey et al., NeurIPS'21; Chamon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

Fairness

Problem
Predict whether an individual will recidivate



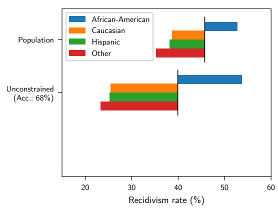
Problem
Predict whether an individual makes > $50k



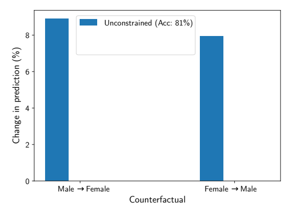
* We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.

Fairness

Problem
Predict whether an individual will recidivate



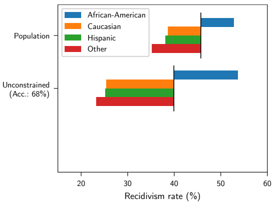
Problem
Predict whether an individual makes > $50k



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

Fairness

Problem
Predict whether an individual will recidivate



Problem
Predict whether an individual makes > \$50k

* 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

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

* 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

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

* 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

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1\right] + c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

* 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

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1\right] + c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

* 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

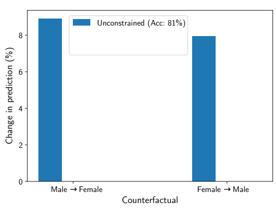
$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1 \mid \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[f_{\theta}(\boldsymbol{x}_n) = 1\right] + c, \\ & \quad \text{for Race} \in \{\text{African-American, Caucasian, Hispanic, Other}\} \end{aligned}$$

* 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

Problem
Predict whether an individual will recidivate

Problem
Predict whether an individual makes > \$50k



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

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

$$\begin{aligned} \min_{\theta} \quad & \text{Prediction error} \\ \text{subject to} \quad & \text{Change in prediction } (\rho x) \leq c \quad \text{a.e.} \\ & \quad (\rho : \text{Male} \leftrightarrow \text{Female}) \end{aligned}$$

* We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

$$\begin{aligned}\min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \text{Change in prediction } (\rho x) \leq c \quad \text{a.e.} \\ & (\rho : \text{Male} \leftrightarrow \text{Female})\end{aligned}$$

*"We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

$$\begin{aligned}\min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \mathrm{D}_{\mathrm{KL}}\big(f_{\theta}(\boldsymbol{x}_n) \| f_{\theta}(\rho \boldsymbol{x}_n)\big) \leq c, \quad \text{for all } n \\ & (\rho : \text{Male} \leftrightarrow \text{Female})\end{aligned}$$

*"We say "Race" to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon and Ribeiro, NeurIPS'20]

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])
- ...

Federated learning

Problem
Learn a common model using data using data distributed among K clients

$$\min_{\theta} \quad \text{Average loss across clients}$$

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

[Shen et al., ICRL'22]

Federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

[Shen et al., ICRL'22]

Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

[Shen et al., ICRL'22]

Federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

[Shen et al., ICRL'22]

Federated learning

Problem
Learn a common model using data using data distributed among K clients

$$\begin{aligned}\min_{\theta} \quad & \frac{1}{K}\sum_{k=1}^{K} \text{Loss}_k(f_{\theta}) \\ \text{subject to} \quad & \text{Loss disparity } (k\text{-th client}) \leq c, \\ & k = 1, \dots, K\end{aligned}$$

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

[Shen et al., ICRL'22]

Federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

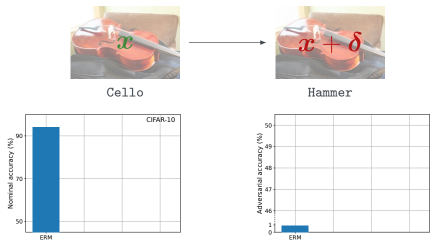
[Shen et al., ICRL'22]

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])
- …

Robustness

Problem
Learn a classifier that is robust to input perturbations



Robustness

Problem
Learn a classifier that is robust to input perturbations

$$\begin{aligned} \min_{\theta} \quad & \text{Nominal loss} \\ \text{subject to} \quad & \text{Adversarial loss} \leq c \end{aligned}$$

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

Robustness

Problem
Learn a classifier that is robust to input perturbations

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

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

Robustness

Problem
Learn a classifier that is robust to input perturbations

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

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

Invariance

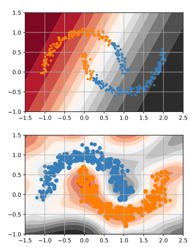
Problem
Learn a classifier that is invariant to transformation $g \in \mathcal{G}$

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

[Hounie et al., ICML'23]

Smoothness

Problem
Learn a classifier that is smooth, i.e., Lipschitz continuous (on a manifold)



$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) \\ \text{subject to} \quad & \max_{\boldsymbol{x} \in \mathcal{M}} \|\nabla_{\mathcal{M}} f_{\theta}(\boldsymbol{x})\|^2 \leq L \end{aligned}$$

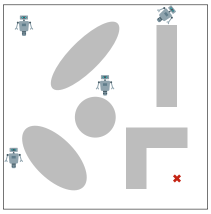
[Cerviño et al., ICML'23]

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])
- ...

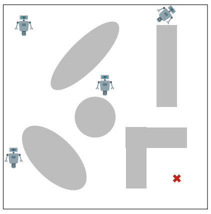
Safety

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

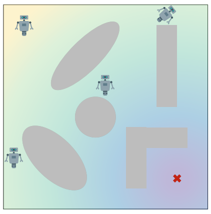


$$\begin{aligned} \underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} \quad & \text{Task reward} \\ \text{subject to} \quad & \Pr\left[\text{Colliding with } \mathcal{O}_i\right] \leq \delta, \\ & \text{for } i = 1, 2, \dots \end{aligned}$$

[Paternain et al., IEEE TAC'23]

Safety

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

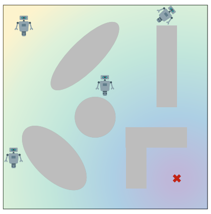


$$\begin{aligned} \underset{\pi \in \mathcal{P}(\mathcal{S})}{\text{maximize}} \quad & \mathbb{E}_{s,a\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} r_0(s_t, a_t)\right] \\ \text{subject to} \quad & \Pr\left[\text{Colliding with } \mathcal{O}_i\right] \leq \delta, \\ & \text{for } i = 1, 2, \dots \end{aligned}$$

[Paternain et al., IEEE TAC'23]

Safety

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



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

[Paternain et al., IEEE TAC'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])
- Wireless resource allocation (e.g., [Eisen et al., IEEE TSP'19])
- 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]

What is (un)constrained learning?

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

- ℓ, g are bounded, Lipschitz continuous (possibly non-convex) functions
- f_{θ} 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]

Constrained learning challenges

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

Challenges

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

Constrained learning challenges

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

Challenges

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

Constrained learning challenges

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

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

Resilient constrained learning

Constrained learning challenges

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

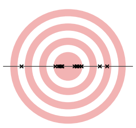
Challenges

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

What classical learning theory says?

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

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



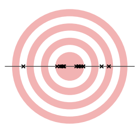
[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning…, 2014]

What classical learning theory says?

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

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

✗ Requirements?



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

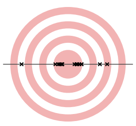
What's in a solution?

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

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



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

What's in a solution?

Definition (PACC learnability)

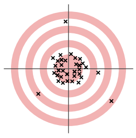
$f_{\boldsymbol{\theta}}$ is a *probably approximately correct constrained (PACC)* learnable if for every ϵ, δ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\boldsymbol{\theta}^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1-\delta$,

- 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

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



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

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

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

ECRM is not a PACC learner

Counter-example

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

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

ECRM is not a PACC learner

Counter-example

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

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

ECRM is not a PACC learner

Counter-example

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

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

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

ECRM is not a PACC learner

Counter-example

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

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

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

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

Constrained learning challenges

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

Challenges

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

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

Constrained learning challenges

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

$$\text{subject to} \ \frac{1}{N}\sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m), y_m\big) \leq c \ \overset{\text{PAC}}{\not\longrightarrow} \ \text{subject to} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\Big[g\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\Big] \leq c$$

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

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

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

PRIMAL

$\updownarrow$

DUAL

Duality

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

$$\updownarrow$$

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

PRIMAL DUAL

Duality

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

$$\updownarrow$$

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

PRIMAL DUAL

- 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

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

$$\not\updownarrow$$

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

PRIMAL DUAL

- 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

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \ & \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \\ \text{s. to} \ & \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c\end{aligned} \ \not\longleftrightarrow \ \hat{D}^\star = \max_{\lambda\geq 0} \ \min_{\boldsymbol{\theta}\in\Theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

PRIMAL DUAL

$$\not\updownarrow \ \text{PAC}$$

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

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

An alternative path

$$\begin{aligned}\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \ & \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \\ \text{s. to} \ & \frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) \leq c\end{aligned} \ \not\longleftrightarrow \ \hat{D}^\star = \max_{\lambda\geq 0} \ \min_{\boldsymbol{\theta}\in\Theta} \ \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) + \lambda\left(\frac{1}{N}\sum_{n=1}^{N} g(f_{\boldsymbol{\theta}}, \boldsymbol{z}_n) - c\right)$$

PRIMAL DUAL

$$\not\updownarrow \ \text{PAC}$$

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

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

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

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., 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 et al., IEEE TSP'20]

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 et al., 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 et al., 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

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

$\updownarrow O(\epsilon)$

$\not\updownarrow$ 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 et al., 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 $\overline{\text{conv}}(\{f_{\boldsymbol{\theta}}\})$]

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., 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

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

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

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

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

(mild conditions apply)

[Chamon and Ribeiro, NeurIPS'20; Chamon et al., 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

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

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

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

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

(if losses are convex) $h\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big) \le r$, with $\mathfrak{P}$-prob. $1 - \widetilde{O}\left(\frac{1}{\sqrt{N}}\right)$

(mild conditions apply)

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

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

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

$$\mathbb{E}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \le c + \epsilon$$

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

Sources of error

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

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

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

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

$$\mathbb{E}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \le c + \epsilon$$

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

Sources of error

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

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

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

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

$$\mathbb{E}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \le c + \epsilon$$

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

Sources of error

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

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

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

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

$$\mathbb{E}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \le c + \epsilon$$

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

Sources of error

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

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

Dual (near-)PACC learning

Theorem

Let f be ν-universal with VC dimension $d_{\text{VC}} < \infty$. There exists $(\boldsymbol{\theta}^\dagger, \boldsymbol{\lambda}^\dagger)$ achieving $\hat{D}^\star$ such that $f_{\boldsymbol{\theta}^\dagger}$ is a (near-)PACC solution of (P-CSL), i.e., with probability at least $1-\delta$,

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

$$\mathbb{E}\left[g\big(f_{\boldsymbol{\theta}^\dagger}(\boldsymbol{x}), y\big)\right] \le c + \epsilon$$

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

Sources of error

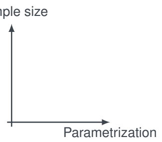
parametrization richness (ν) sample size (N) requirements difficulty ($\boldsymbol{\lambda}^\star$)

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

Dual learning trade-offs

- Unconstrained learning

  parametrization × sample size



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

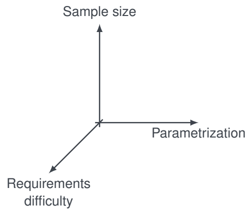
Dual learning trade-offs

- Unconstrained learning

  parametrization × sample size

- Constrained learning

  parametrization × sample size × requirements



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

When is constrained learning possible?

Corollary

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

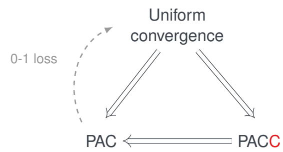
Constrained learning is **essentially as hard as** unconstrained learning

[mild conditions apply]

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

When is constrained learning possible?

Corollary

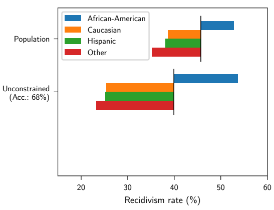


[mild conditions apply]

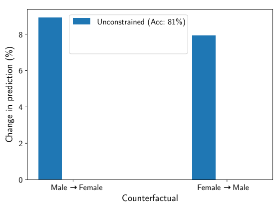
[Chamon and Ribeiro, NeurIPS'20; Chamon et al., IEEE TIT'23]

Fairness

Problem
Predict whether an individual will recidivate



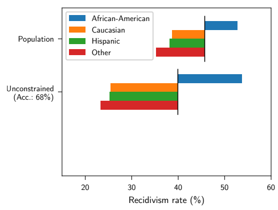
Problem
Predict whether an individual makes > \$50k



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

Fairness

Problem
Predict whether an individual will recidivate



Problem
Predict whether an individual makes > \$50k

* 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 et al., 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 et al., IEEE TIT'23]

Fairness: “Equality” of odds

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

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

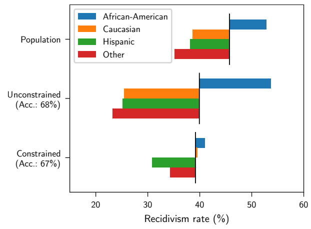
$$\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - 0.5\big)\,\mathbb{I}\left[\boldsymbol{x}_n \in \text{Race}\right] \leq \frac{1}{N}\sum_{n=1}^{N} \sigma\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - 0.5\big) + c,$$

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 et al., IEEE TIT'23]

Fairness: “Equality” of odds

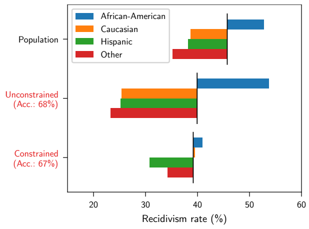
Problem
Predict whether an individual will recidivate at the same rate across races



* We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., 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 et al., IEEE TIT'23]

Fairness: “Equality” of odds

	True label	Unconstrained: Prediction 0	Unconstrained: Prediction 1	Constrained: Prediction 0	Constrained: Prediction 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 et al., IEEE TIT'23]

Fairness: “Equality” of odds

	True label	Unconstrained: Prediction 0	Unconstrained: Prediction 1	Constrained: Prediction 0	Constrained: Prediction 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 et al., IEEE TIT'23]

Fairness: “Equality” of odds

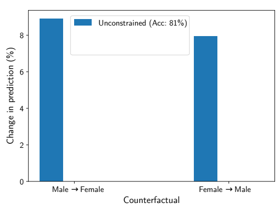
	True label	Unconstrained: Prediction 0	Unconstrained: Prediction 1	Constrained: Prediction 0	Constrained: Prediction 1
African-American	0	31%	16%	36%	11%
	1	16%	37%	23%	30%
Caucasian	0	52%	9%	44%	17%
	1	23%	16%	16%	23%

* We say “Race” to follow the terminology used during the data collection of the COMPAS dataset.
[Chamon et al., IEEE TIT'23]

Fairness

Problem
Predict whether an individual will recidivate

Problem
Predict whether an individual makes > \$50k



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

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

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

$$\text{subject to} \quad \mathrm{D}_{\mathrm{KL}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \| f_{\boldsymbol{\theta}}(\rho \boldsymbol{x}_n)\right) \leq c, \quad \text{for all } n$$

(ρ : Male $\leftrightarrow$ Female)

[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

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

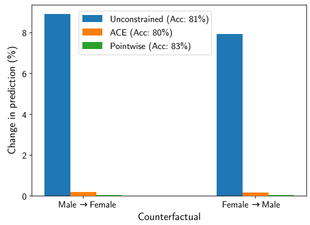
$$\text{subject to} \ \ \frac{1}{N}\sum_{n=1}^{N} \mathrm{D}_{\mathrm{KL}}\left(f_{\theta}(\boldsymbol{x}_n) \| f_{\theta}(\rho\boldsymbol{x}_n)\right) \leq c \quad \text{for all } n$$

(ρ : Male $\leftrightarrow$ Female)

[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender



[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender

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

$$\text{subject to} \ \ \mathrm{D}_{\mathrm{KL}}\left(f_{\theta}(\boldsymbol{x}_n) \| f_{\theta}(\rho\boldsymbol{x}_n)\right) \leq c, \quad \text{for all } n$$

(ρ : Male $\leftrightarrow$ Female)

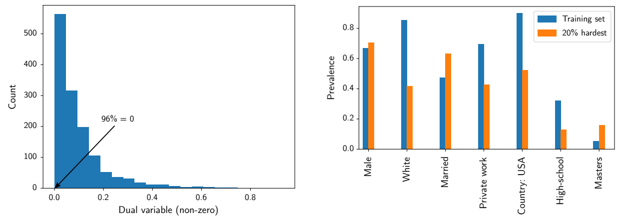
$\updownarrow$

$$\max_{\lambda_n \geq 0} \min_{\theta} \ \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(\boldsymbol{x}_n), y_n\big) + \sum_{n=1}^{N} \lambda_n \left[\mathrm{D}_{\mathrm{KL}}\left(f_{\theta}(\boldsymbol{x}_n) \| f_{\theta}(\rho\boldsymbol{x}_n)\right) - c\right]$$

[Chamon and Ribeiro, NeurIPS'20]

Counterfactual fairness

Problem
Predict whether an individual makes > \$50k while being invariant to gender



[Chamon and Ribeiro, NeurIPS'20]

Agenda

Constrained learning theory

Constrained learning algorithms

Resilient constrained learning

Constrained optimization methods

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

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

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

Constrained optimization methods

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

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

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

- Feasible update methods
  e.g., conditional gradients (Frank-Wolfe)
- Interior point methods
  e.g., barriers, projection, polyhedral approx.

Constrained optimization methods

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

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

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

- 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

Constrained optimization methods

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

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

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

- 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

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

Dual learning algorithm

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

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

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

Dual learning algorithm

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

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

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

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

Dual learning algorithm

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

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

- Update the dual

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

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

A (near-)PACC learner

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

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

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 $\left(\boldsymbol{\theta}^{(T)}, \boldsymbol{\lambda}^{(T)}\right)$ are such that

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

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

[Chamon et al., IEEE TIT'23]

In practice. . .

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

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

- Update the dual

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

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

In practice. . .

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

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

- Update the dual

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

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

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
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\big) - c \right) \right]_+$
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
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\big) - c \right) \right]_+$
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
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\big) - c \right) \right]_+$
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
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
  $\lambda_t = \left[\lambda_{t-1} + \eta_\lambda \left(\frac{1}{N} \sum_{m=1}^{N} g\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\big) - c \right) \right]_+$
end
Output: $\boldsymbol{\theta}_T, \lambda_T$

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

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

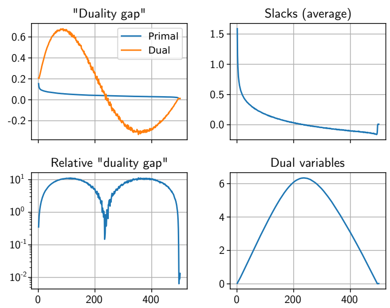
Use adaptive method (e.g., ADAM)
Use different time-scales ($\eta_\lambda = 0.1\eta_\theta$)

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

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

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

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

Agenda

Constrained learning theory

Constrained learning algorithms

Resilient constrained learning

Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

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

Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

Heterogeneous federated learning

Problem
Learn a common model using data using data distributed among K clients

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

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

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

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

Resilient constrained learning

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

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

Resilient constrained learning

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

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

- Larger relaxations $\boldsymbol{r}$ decrease the objective $P^\star(\boldsymbol{r})$ (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

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

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

Resilient constrained learning

Definition (Resilience)

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

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

- Larger relaxations r decrease the objective $P^\star(\boldsymbol{r})$ (benefit), but increase specification violation $c_i + r_i$ (cost) $\Rightarrow h(\boldsymbol{r})$
- 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 et al., NeurIPS'23]

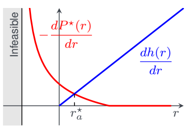
Resilient constrained learning

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 et al., NeurIPS'23]

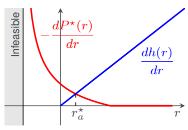
Resilient constrained learning

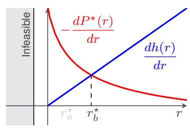
Definition (Resilient equilibrium)

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 et al., NeurIPS'23]

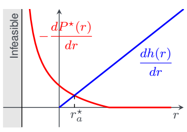
Resilient constrained learning

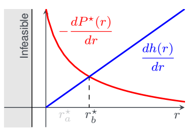
Definition (Resilient equilibrium)

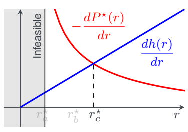
For a strictly convex function $h(\boldsymbol{r})$, we say the relaxation $\boldsymbol{r}^\star$ achieves the resilient equilibrium if

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

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







[Hounie et al., 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 $\boldsymbol{\lambda}^\star(\boldsymbol{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)

[Hounie et al., 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 $\boldsymbol{\lambda}^\star(\boldsymbol{0})$
  $\Rightarrow$ Resilient constrained learning "generalizes better" (lower sample complexity)
- ✅ The resilient equilibrium *exists and is unique* (because h is strictly convex)

[Hounie et al., 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

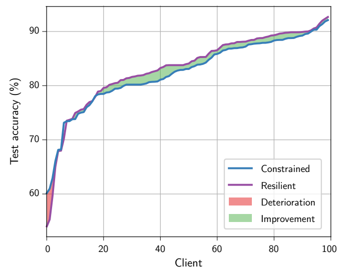
$$P^\star(\boldsymbol{r}^\star) = \min_{\boldsymbol{\theta},\boldsymbol{r}} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] + h(\boldsymbol{r})$$
$$\text{subject to } \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}_i}\left[g_i\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_m),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

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

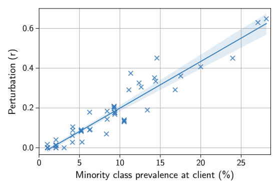
[Hounie et al., NeurIPS'23]

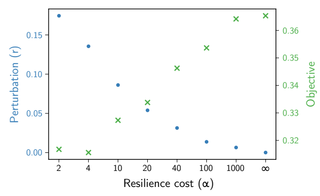
Heterogeneous federated learning



[Hounie et al., NeurIPS'23]

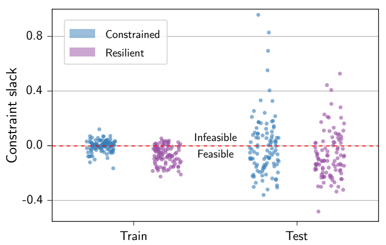
Heterogeneous federated learning





[Hounie et al., NeurIPS'23]

Heterogeneous federated learning



[Hounie et al., NeurIPS'23]

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.* Resilient learning can then be used to adapt the requirements to the task difficulty [Hounie et al., NeurIPS'23]

Robustness constraints

Agenda

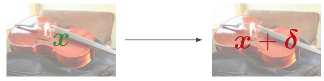
Adversarially robust learning

Semi-infinite learning

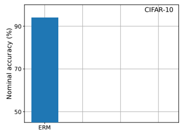
Probabilistic robustness

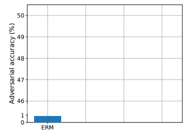
Robust learning

Problem
Learn an image classifier that is robust to input perturbations



Cello Hammer





Adversarial training

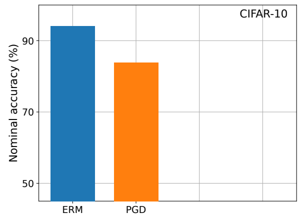
Problem
Learn an image classifier that is robust to input perturbations

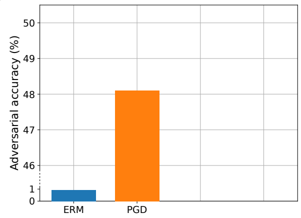
- Adversarial training [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; . . .]

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

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations





[Robey et al., NeurIPS'21]

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations

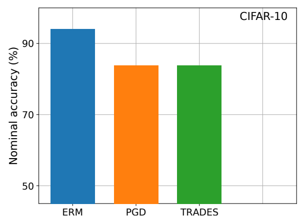
- Adversarial training [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; . . .]

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

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

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations



[Zhang et al., ICML'19]

Constrained learning for robustness

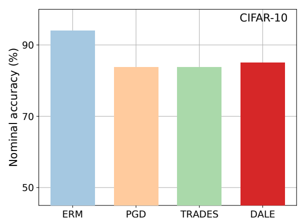
Problem
Learn an image classifier that is robust to input perturbations

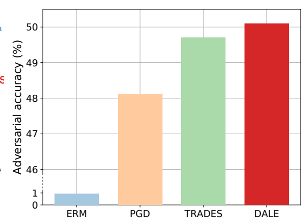
$$\begin{aligned} &\min_{\theta} && \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\theta}(x_n), y_n\big) \\ &\text{subject to} && \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\delta\|_\infty \le \epsilon} \text{Loss}\big(f_{\theta}(x_n+\delta), y_n\big)\right] \le c \end{aligned}$$

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

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations





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

Constrained learning for robustness

Problem

Learn an image classifier th

Penalty-based vs. dual learning

Penalty-based learning

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

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

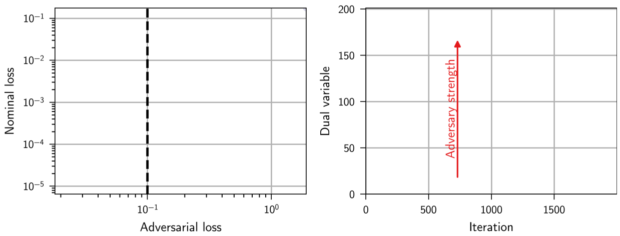
Dual learning

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

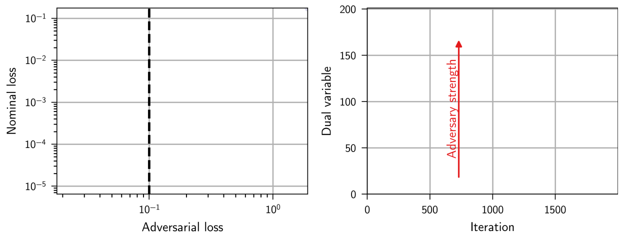
$$\lambda^+ = \left[\lambda + \eta\left(\text{Penalty}(\theta^\dagger) - c\right)\right]_+$$

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

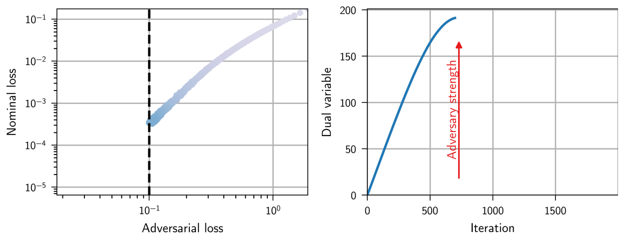
Constrained learning for robustness



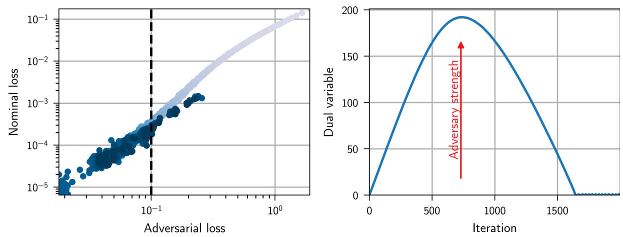
Constrained learning for robustness



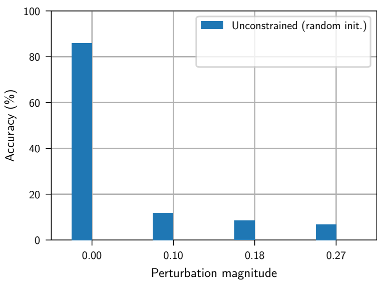
Constrained learning for robustness



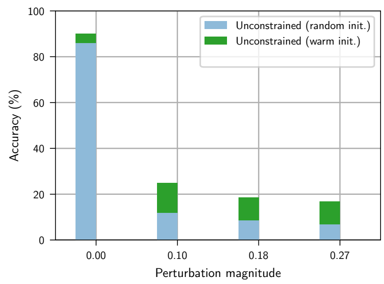
Constrained learning for robustness



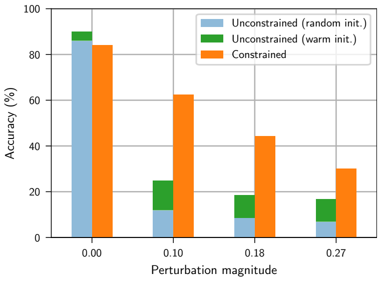
Constrained learning for robustness



Constrained learning for robustness



Constrained learning for robustness



[Chamon et al., IEEE TIT'23]

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

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

Adversarial training

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

- "PGD" [Mądry et al., ICLR'18]

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\delta^{k+1} \leftarrow \text{proj}_{\Delta} \left[\delta^k + \eta \,\text{sign}\left(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x + \delta^k), y\big)\right)\right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x + \delta_t), y\Big)$

Adversarial training

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

- "PGD" [Mądry et al., ICLR'18]

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\delta^{k+1} \leftarrow \text{proj}_{\Delta} \left[\delta^k + \eta \,\text{sign}\left(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x + \delta^k), y\big)\right)\right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x + \delta_t), y\Big)$

- Random initialization
- Restarts
- Pruning
- Adaptive step size

[Dhillon et al., ICLR'18; Carmon et al., NeurIPS'19; Wu et al., NeurIPS'20; Cheng et al., IJCAI'22]

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~~

Agenda

Adversarially robust learning

Semi-infinite learning

Probabilistic robustness

Semi-infinite constrained learning

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

Semi-infinite constrained learning

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

- Epigraph formulation:

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

Semi-infinite constrained learning

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

- Epigraph formulation:

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

- Semi-infinite program

Duality

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

$$\updownarrow =$$

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

$$\updownarrow =$$

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

Duality

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

$$\updownarrow =$$

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

$$\updownarrow =$$

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

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 $\gamma(x, y) < \max_{\delta\in\Delta} \text{Loss}\big(f_\theta(x+\delta), y\big)$ s.t. $L(\theta, \mu_\gamma) \ge \sup_{\mu\in\mathcal{P}^2} L(\theta, \mu) - \xi$ for

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

From optimization to sampling

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

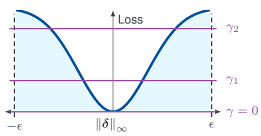
$$\updownarrow \approx$$

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

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 et al., NeurIPS'21]

From optimization to sampling

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

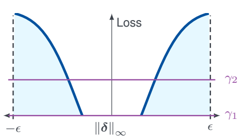
$$\updownarrow \approx$$

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

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 et al., NeurIPS'21]

From optimization to sampling

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

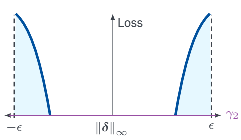
$$\updownarrow \approx$$

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

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 et al., NeurIPS'21]

From optimization to sampling

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

$$\updownarrow =$$

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

Proposition

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

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

[Robey et al., NeurIPS'21]

From optimization to sampling

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

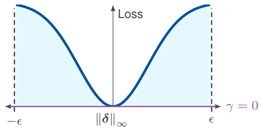
$$\updownarrow \approx$$

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

Proposition

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

$$\mu_0(\boldsymbol{\delta}|\boldsymbol{x}, y) \propto \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)$$



[Robey et al., NeurIPS'21]

Constrained learning for robustness

Problem

Learn an image classifier that is robust to input perturbations

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

- ✓ 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

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

- ✓ Balancing nominal accuracy and robustness ⇒ Dual constrained learning
- ✓ Computing the worst-case perturbations
  - ~~gradient ascent → non-convex, underparametrized~~ ⇒ sampling

Dual Adversarial LEarning

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

$\quad\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$ | HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

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

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

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

$\quad$**end**

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

end

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

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

$\quad\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$ | HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

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

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

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

$\quad$**end**

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

end

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

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

$\quad\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$ | HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

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

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

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

$\quad$**end**

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

end

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

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

$\quad\boldsymbol{\delta}_n \sim \text{Random}(\Delta)$ | HMC sampling: $\boldsymbol{\delta} \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)$

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

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

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

$\quad$**end**

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

end

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

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

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

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

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

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

$\quad$ **end**

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

end

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

HMC sampling:
$\boldsymbol{\delta} \sim \mu(\cdot|\boldsymbol{x}_n, y_n)$

SGD

GA

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

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

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

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

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

$\quad$ **end**

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

end

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

Gaussian
[Lopes et al., arXiv'19]
[Rusak et al., ECCV'20]
Patches
[Zhong et al., AAAI'20]
[Yun et al., ICCV'19]
...

SGD

GA

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

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

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

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

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

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

$\quad$ **end**

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

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

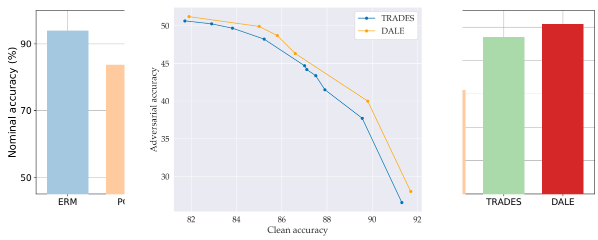
GA

[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

Problem

Learn an image classifier th

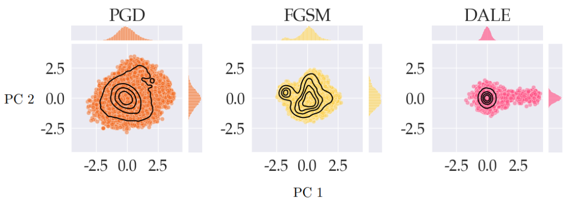


[Robey et al., NeurIPS'21]

Dual Adversarial LEarning

Problem

Learn an image classifier that is robust to input perturbations



[Robey et al., NeurIPS'21]

Invariance

Problem

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

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

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

[Hounie et al., ICML'23]

Invariance

Problem

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

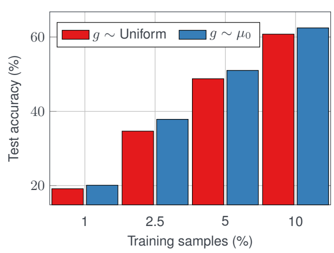
$$\min_{\boldsymbol{\theta}} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N}\left[\mathbb{E}_{g \sim \mu_0(\cdot|\boldsymbol{x}_n, y_n)} \text{Loss}\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\right] \leq c$$

- No differentiability required (e.g., Metropolis-Hastings)

[Hounie et al., ICML'23]

Training on a subset of ImageNet-100



- Transformations ($\mathcal{G}$)
  - ShearX, ShearY, Flips, Rotate, TranslateX, TranslateY
  - Cutout, Crop
  - AutoContrast, Invert, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness

[Hounie et al., ICML'23]

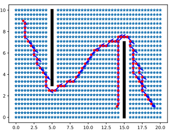
“Identifying” invariances

Dataset	Dual variable (λ)	Synthetic Invariance		
		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

(Manifold) smoothness

Problem
Leveraging unlabeled data

- Labeled data ({Position, Action})

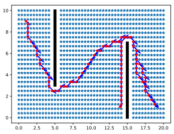


Dataset

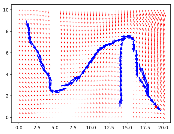
(Manifold) smoothness

Problem
Leveraging unlabeled data

- Labeled data ({Position, Action})



Dataset

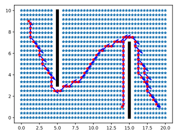


ERM
(85% success)

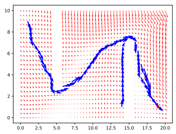
(Manifold) smoothness

Problem
Leveraging unlabeled data

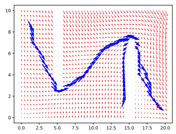
- Labeled data ({Position, Action})



Dataset



ERM
(85% success)



Lipschitz regularization
(66% success)

(Manifold) smoothness

Problem
Leveraging unlabeled data

- Labeled data ({Position, Action}) and unlabeled data ({Position})
- Use {Position} to estimate a data manifold $\mathcal{M}$

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \|f_{\theta}(\text{Position}_n) - \text{Action}_n\|^2$$
$$\text{subject to} \quad \max_{x} \|\nabla_{\mathcal{M}} f_{\theta}(x)\|^2 \le c$$

(Manifold) smoothness

Problem
Leveraging unlabeled data

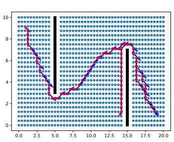
- Labeled data ({Position, Action}) and unlabeled data ({Position})
- Use {Position} to estimate a data manifold $\mathcal{M}$

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \|f_{\theta}(\text{Position}_n) - \text{Action}_n\|^2$$
$$\text{subject to} \quad \cancel{\max_{x}} \|\nabla_{\mathcal{M}} f_{\theta}(x)\|^2 \le c$$
$$\mathbb{E}_{x\sim\mu_0}$$

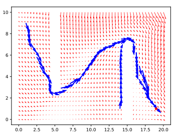
(Manifold) smoothness

Problem
Leveraging unlabeled data

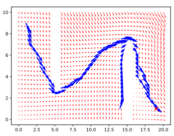
- Labeled data ({Position, Action}) and unlabeled data ({Position})



Dataset



ERM
(85% success)



Manifold smoothness
(94% success)

Agenda

Adversarially robust learning

Semi-infinite learning

Probabilistic robustness

Constrained learning challenges

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

$$\xrightarrow{\text{PACC}}$$

$$\text{s. to } \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\boldsymbol{\delta}\in\Delta} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\Big] \leq c \xrightarrow{\text{PAC}} \text{s. to } \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\max_{\boldsymbol{\delta}\in\Delta} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\Big] \leq c$$

Challenges

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

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

Constrained learning challenges

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

$$\xrightarrow{\text{PACC}}$$

$$\text{s. to } \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\boldsymbol{\delta}\in\Delta} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\Big] \leq c \xrightarrow[\text{PAC?}]{} \text{s. to } \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\max_{\boldsymbol{\delta}\in\Delta} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\big)\Big] \leq c$$

Challenges

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

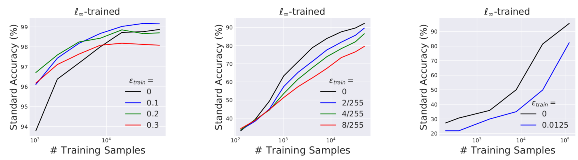
2) *Computational*: can we solve the constrained empirical problem?

Statistical complexity

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \Big[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\Big)\Big] \xrightarrow{?} \min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\Big[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\Big)\Big]$$

- Is robust learning harder than non-robust learning? Do we need more samples?
  A: YES *and* NO
  [Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]
  [Yin, Ramchandran, Bartlett. Rademacher Complexity for Adversarially Robust Generalization, ICML'19]
  [Montasser, Hanneke, Srebro. VC Classes are Adversarially Robustly Learnable, but Only Improperly, COLT'19]
  [Awasthi, Frank, Mohri. Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks, ICML'20]
  [Montasser, Hanneke, Srebro. Adversarially robust learning: A generic minimax optimal learner & characterization, NeurIPS'22]

Nominal performance of robust models



(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

[Tsipras et al., ICLR'19]

"Softer" robustness

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

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

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

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

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

  - $\tau = 1$: 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]

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

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

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

- Computationally challenging (especially as $\tau \to \infty$, i.e., stronger robustness)
- No guaranteed advantages (lower sample complexity? improved trade-offs?)

Towards probabilistic robustness

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N}\big[t(\boldsymbol{x}_n, y_n)\big] \\
\text{subject to} \quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}), y_n\big) \leq t(\boldsymbol{x}_n, y_n)
\end{aligned}$$

- 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

Towards probabilistic robustness

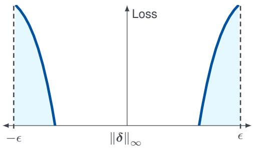
$$\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & \frac{1}{N}\sum_{n=1}^{N}\big[t(\boldsymbol{x}_n, y_n)\big] \\
\text{subject to} \quad & \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_0), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_1), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \cancel{\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\sqrt{2}}), y_n\big) \leq t(\boldsymbol{x}_n, y_n)} \\
& \cancel{\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_e), y_n\big) \leq t(\boldsymbol{x}_n, y_n)} \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_\pi), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_4), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \cancel{\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^2}), y_n\big) \leq t(\boldsymbol{x}_n, y_n)} \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{\pi^e}), y_n\big) \leq t(\boldsymbol{x}_n, y_n) \\
& \cancel{\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{e^\pi}), y_n\big) \leq t(\boldsymbol{x}_n, y_n)} \\
& \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}_{25}), y_n\big) \leq t(\boldsymbol{x}_n, y_n)
\end{aligned}$$

Towards probabilistic robustness

$$\min_{\theta} \; \frac{1}{N}\sum_{n=1}^{N} \left[t(x_n, y_n)\right]$$

$$\begin{aligned}
\text{subject to}\quad & \text{Loss}\big(f_\theta(x_n+\delta_0), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_1), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_{\sqrt{2}}), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_e), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_\pi), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_4), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_{e^2}), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_{\pi^e}), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_{e^\pi}), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_{25}), y_n\big) \le t(x_n,y_n)
\end{aligned}$$

$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\underbrace{\max_{\delta\in\Delta}}_{\approx \mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}} \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$

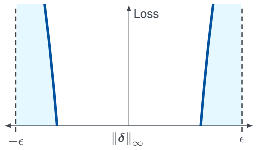


Towards probabilistic robustness

$$\min_{\theta} \; \frac{1}{N}\sum_{n=1}^{N} \left[t(x_n, y_n)\right]$$

$$\begin{aligned}
\text{subject to}\quad & \text{Loss}\big(f_\theta(x_n+\delta_0), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_1), y_n\big) \le t(x_n,y_n)\\
& \cancel{\text{Loss}\big(f_\theta(x_n+\delta_{\sqrt{2}}), y_n\big) \le t(x_n,y_n)}\\
& \cancel{\text{Loss}\big(f_\theta(x_n+\delta_e), y_n\big) \le t(x_n,y_n)}\\
& \text{Loss}\big(f_\theta(x_n+\delta_\pi), y_n\big) \le t(x_n,y_n)\\
& \text{Loss}\big(f_\theta(x_n+\delta_4), y_n\big) \le t(x_n,y_n)\\
& \cancel{\text{Loss}\big(f_\theta(x_n+\delta_{e^2}), y_n\big) \le t(x_n,y_n)}\\
& \text{Loss}\big(f_\theta(x_n+\delta_{\pi^e}), y_n\big) \le t(x_n,y_n)\\
& \cancel{\text{Loss}\big(f_\theta(x_n+\delta_{e^\pi}), y_n\big) \le t(x_n,y_n)}\\
& \text{Loss}\big(f_\theta(x_n+\delta_{25}), y_n\big) \le t(x_n,y_n)
\end{aligned}$$

$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\underbrace{\max_{\delta\in\Delta}}_{\approx \mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}} \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$

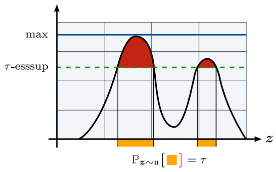


Probabilistic robustness

- Probabilistic robustness

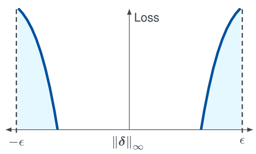
$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\tau\text{-}\underset{\delta\in\Delta}{\text{esssup}}\; \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$

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

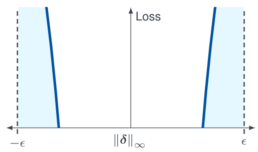


Probabilistic robustness

$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\underbrace{\max_{\delta\in\Delta}}_{\mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}} \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$



$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\underbrace{\tau\text{-}\underset{\delta\in\Delta}{\text{esssup}}}_{\mathbb{E}_{\delta\sim\mu_0(\cdot|x,y)}} \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$



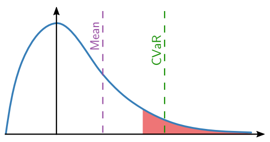
Probabilistic robustness and Risk

- Conditional value at risk:

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

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

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



Proposition

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

Probabilistically robust learning

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

2: $\alpha_0 = 0$

3: **for** $t = 1, \dots, T$:

4: $\delta_t \sim$ Random(Δ)

5: $\alpha \leftarrow \alpha - \frac{\eta}{\tau}\Big(\tau - \mathbb{I}\left[\text{Loss}(f_\theta(x_n+\delta_t), y_n) \ge \alpha\right]\Big)$

6: **end**

SGD (CVaR)

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

SGD (θ)

8: **end**

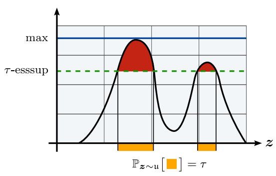
Probabilistic robustness

- Probabilistic robustness

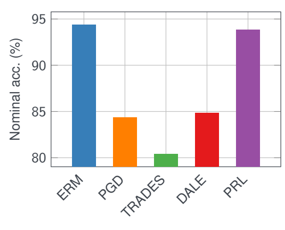
$$\min_{\theta} \; \mathbb{E}_{(x,y)}\Big[\tau\text{-}\underset{\delta\in\Delta}{\text{esssup}}\; \text{Loss}\big(f_\theta(x+\delta), y\big)\Big]$$

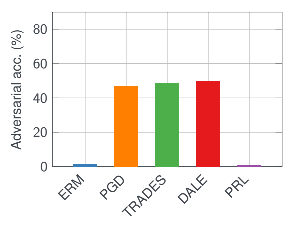
  - $\tau = 1/2$: classical learning (for symmetric m)
  - $\tau = 0$: adversarial robustness (ess sup)

  - 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)]

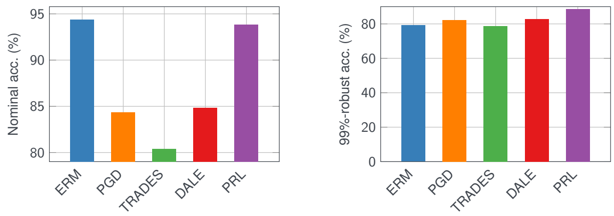


Probabilistically robust learning





Probabilistically robust learning



[Robey et al., 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]

Break