Universität Stuttgart

Luiz F. O. Chamon

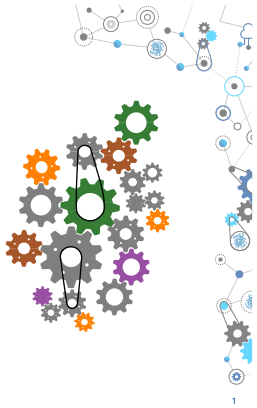
learning under requirements

SimTech ML session
June 21, 2023

ellis

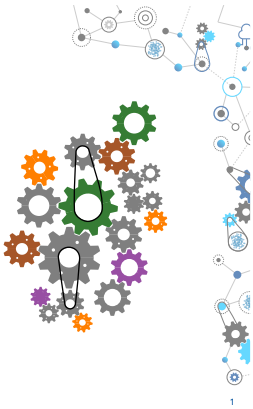
SimTech Cluster of Excellence

System engineering cycle



System engineering cycle

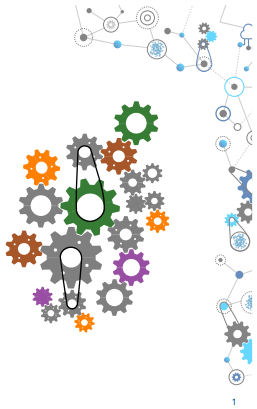
Requirements



System engineering cycle

Requirements

Data

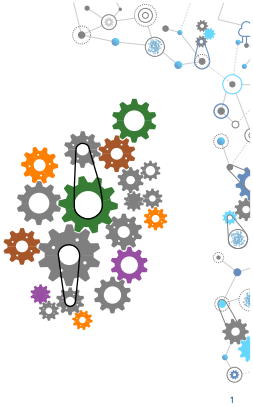


System engineering cycle

Requirements

Model

Data



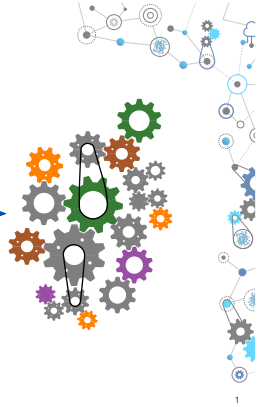
System engineering cycle

Requirements

Model

Operational settings

Data



The promise of learning

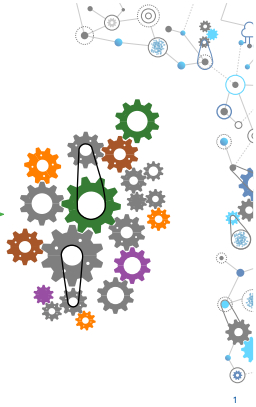
Requirements

Learning

Model

Operational settings

Data



The ~~promise~~ emerging reality of learning

Microsoft Azure will become the preferred cloud platform for Johns Hopkins inHealth precision medicine initiative

[Stokes et al., Cell'20]

amazon

Google

[Ma et al., Nature Photonics'20]

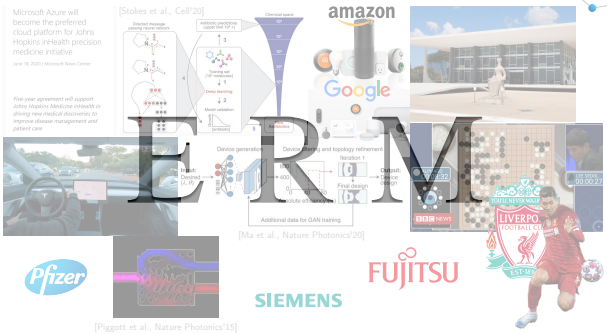
Pfizer

SIEMENS

FUJITSU

[Piggott et al., Nature Photonics'15]

The promise emerging reality of learning



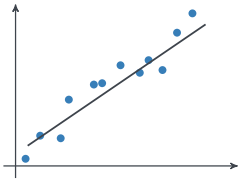
[Stokes et al., Cell'20]
[Ma et al., Nature Photonics'20]
[Piggott et al., Nature Photonics'15]

Empirical Risk Minimization

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\Big) \qquad \text{(ERM)}$$

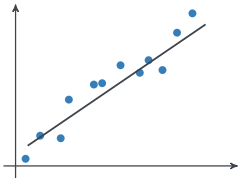
Empirical Risk Minimization

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\Big) \qquad \text{(ERM)}$$



Empirical Risk Minimization

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\Big) \qquad \text{(ERM)}$$



Learning breakthroughs

Learning breakthroughs

1970s
(theoretical)
Learning theory

Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

- e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

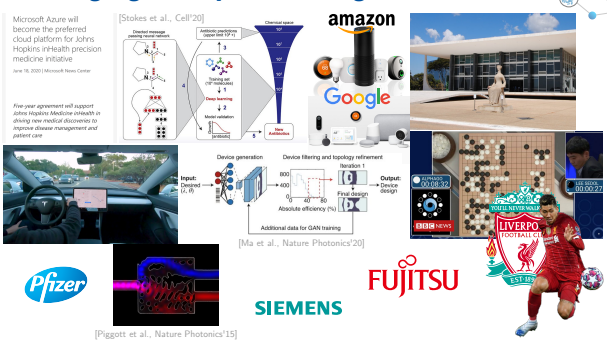
Learning breakthroughs

1970s
(theoretical)
Learning theory

2000s
(technological)
Internet

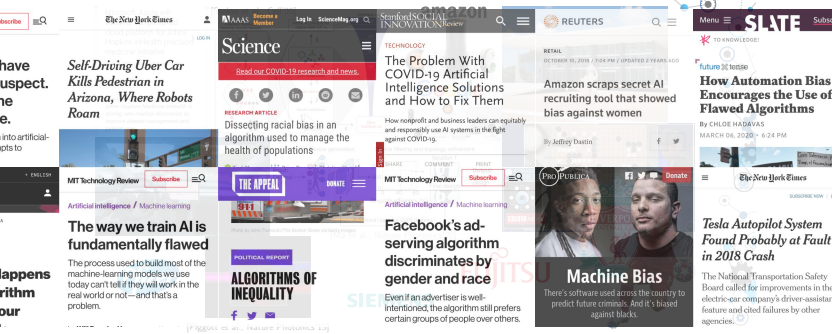
2010s
(computational)
Deep learning

The emerging reality of learning



[Stokes et al., Cell'20]
[Ma et al., Nature Photonics'20]
[Piggott et al., Nature Photonics'15]

The ~~emerging reality~~ limitations of learning



Improving ERM

Improving ERM

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

[Xie & Yuille, ICLR'20; Guo et al., CVPR'20; Finzi et al., ICML'20; Li et al., ICRL'21; Lu et al., Nature Mach. Intel.'21; Raissi et al., J. Comp. Phys.'19; ...]

Improving ERM

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

[Xie & Yuille, ICLR'20; Guo et al., CVPR'20; Finzi et al., ICML'20; Li et al., ICRL'21; Lu et al., Nature Mach. Intel.'21; Raissi et al., J. Comp. Phys.'19; ...]

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

[Kamiran & Calders, KIS'12; Feldman et al., SIGKDD'15; Calmon et al., NeurIPS'17; Chen et al., ICML'20; Müller & Hutter, ICCV'21; Zheng et al., ICRL'22; ...]

Improving ERM

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

[Xie & Yuille, ICLR'20; Guo et al., CVPR'20; Finzi et al., ICML'20; Li et al., ICRL'21; Lu et al., Nature Mach. Intel.'21; Raissi et al., J. Comp. Phys.'19; ...]

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

[Kamiran & Calders, KIS'12; Feldman et al., SIGKDD'15; Calmon et al., NeurIPS'17; Chen et al., ICML'20; Müller & Hutter, ICCV'21; Zheng et al., ICRL'22; ...]

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

[Goodfellow et al., ICLR'15; Arjovsky et al., ICML'17; Madry et al., ICLR'18; Zhang et al., ICML'19; Raissi et al., J. Comp. Phys.'19; Krishnan et al., NeurIPS'20; ...]

Improving ERM

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

[Xie & Yuille, ICLR'20; Guo et al., CVPR'20; Finzi et al., ICML'20; Li et al., ICRL'21; Lu et al., Nature Mach. Intel.'21; Raissi et al., J. Comp. Phys.'19; ...]

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

[Kamiran & Calders, KIS'12; Feldman et al., SIGKDD'15; Calmon et al., NeurIPS'17; Chen et al., ICML'20; Müller & Hutter, ICCV'21; Zheng et al., ICRL'22; ...]

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

[Goodfellow et al., ICLR'15; Arjovsky et al., ICML'17; Madry et al., ICLR'18; Zhang et al., ICML'19; Raissi et al., J. Comp. Phys.'19; Krishnan et al., NeurIPS'20; ...]

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

[Helmbold & Long, JMLR'15; Mianjy et al., ICML'18; Tashiro et al., NeurIPS'20; Li et al., AISTATS'20; Lin et al., ICML'20; Foret et al., ICRL'21; ...]

A different paradigm...

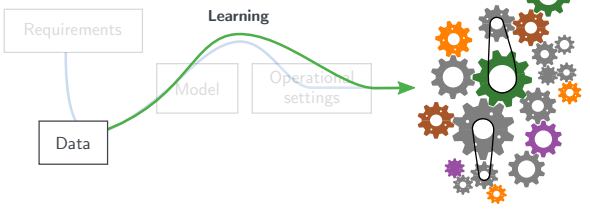
Learning is doing exactly what we asked for.

A different paradigm...

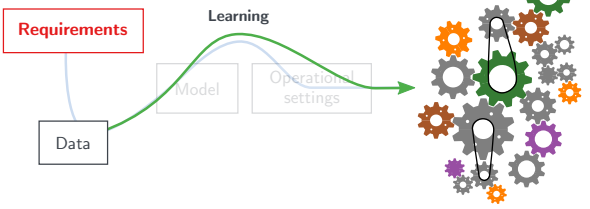
Learning is doing exactly what we asked for.

How can we learn models that do what we want?

The promise of learning



The promise of learning



A different paradigm…

Learning is doing exactly what we asked for.

How can we learn models that do what we want?

Constrained learning

Claims

Constrained learning is the right tool to learn under requirements

Constrained learning is hard…

…but possible

Claims

Constrained learning is the right tool to learn under requirements

- Wireless resource allocation
- Robust image recognition

Constrained learning is hard…

…but possible

Claims

Constrained learning is the right tool to learn under requirements

- Wireless resource allocation
- Robust image recognition

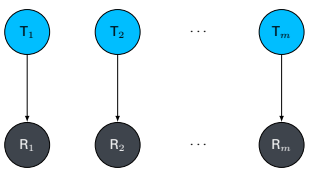
Constrained learning is hard…

…but possible

Wireless resource allocation

Problem

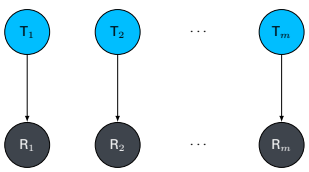
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

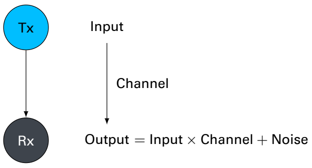
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

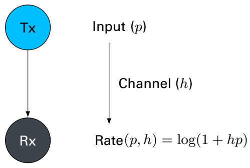
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

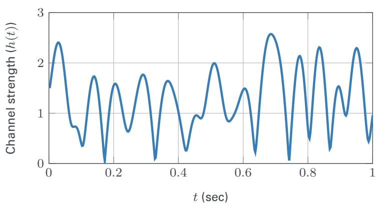
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

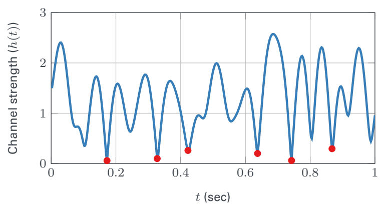
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

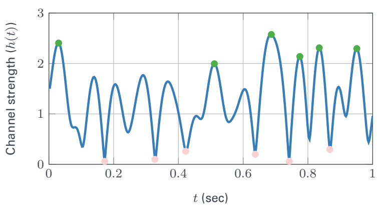
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

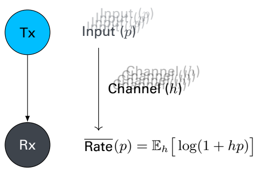
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

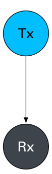
Allocate the least transmit power to m device pairs to achieve a communication rate



Wireless resource allocation

Problem

Allocate the least transmit power to m device pairs to achieve a communication rate

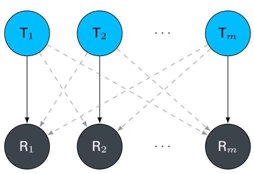


$$\min_{p} \quad \mathbb{E}_h\big[p(h)\big]$$
$$\text{s. to} \quad \mathbb{E}_h\big[\log(1+hp)\big] \geq c$$

Wireless resource allocation

Problem

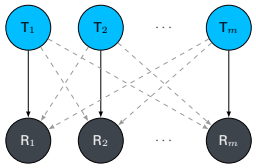
Allocate the least transmit power to m device pairs to achieve a communication rate



$$\min_{p} \quad \sum_{i=1}^{m} \mathbb{E}_{\mathbf{h}}\big[p_i(\mathbf{h})\big]$$
$$\text{s. to} \quad \mathbb{E}_{\mathbf{h}}\left[\log\left(1+\frac{h^{ii}p_i(\mathbf{h})}{1+\sum_{j\in\mathcal{N}_i} h^{ij}p_j(\mathbf{h})}\right)\right] \geq c_i$$

Wireless resource allocation

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

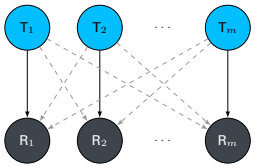


$$\min_{\boldsymbol{p}} \quad \sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{h}}\left[p_i(\boldsymbol{h})\right]$$
$$\text{s. to} \quad \mathbb{E}_{\boldsymbol{h}}\left[\log\left(1+\frac{h^{ii}p_i(\boldsymbol{h})}{1+\sum_{j\in\mathcal{N}_i}h^{ij}p_j(\boldsymbol{h})}\right)\right] \geq c_i$$

[Eisen, Zhang, **C.**, Lee, and Ribeiro, IEEETSP'19]

Wireless resource allocation

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

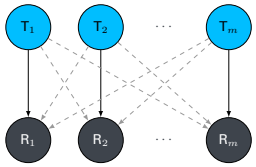


$$\min_{\boldsymbol{p}} \quad \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{m} p_i(\boldsymbol{h}_n)$$
$$\text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N}\log\left(1+\frac{h_n^{ii}p_i(\boldsymbol{h}_n)}{1+\sum_{j\in\mathcal{N}_i}h_n^{ij}p_j(\boldsymbol{h}_n)}\right) \geq c_i$$

[Eisen, Zhang, **C.**, Lee, and Ribeiro, IEEETSP'19]

Wireless resource allocation

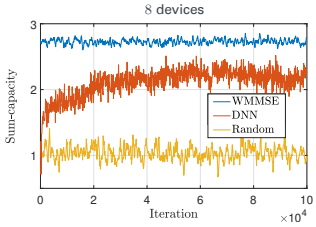
Problem
Allocate the least transmit power to m device pairs to achieve a communication rate



$$\min_{\boldsymbol{\theta}} \quad \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{m} p_i(\boldsymbol{h}_n;\boldsymbol{\theta})$$
$$\text{s. to} \quad \frac{1}{N}\sum_{n=1}^{N}\log\left(1+\frac{h_n^{ii}p_i(\boldsymbol{h}_n;\boldsymbol{\theta})}{1+\sum_{j\in\mathcal{N}_i}h_n^{ij}p_j(\boldsymbol{h}_n;\boldsymbol{\theta})}\right) \geq c_i$$

[Eisen, Zhang, **C.**, Lee, and Ribeiro, IEEETSP'19]

Wireless resource allocation



[Eisen, Zhang, **C.**, Lee, and Ribeiro, IEEETSP'19; Eisen and Ribeiro, IEEETSP'20]

Claims

Constrained learning is the right tool to learn under requirements
- Wireless resource allocation
- Robust image recognition

Constrained learning is hard...

...but possible

Robust image recognition

Problem
Learn an image classifier

Robust image recognition

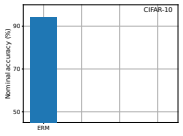
Problem
Learn an image classifier

Cello

Robust image recognition

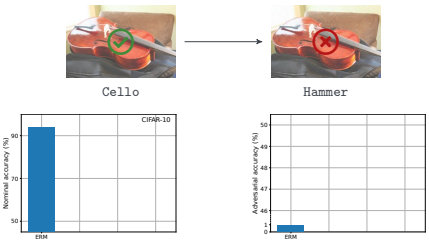
Problem
Learn an image classifier

Cello



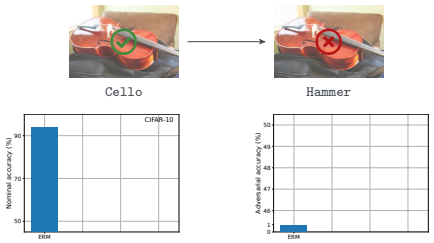
Robust image recognition

Problem
Learn an image classifier



Robust image recognition

Problem
Learn an image classifier that is robust to input perturbations



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; ...]

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; ...]

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

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; ...]

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

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations

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

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations

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

- Computing the worst-case perturbations

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations

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

- Computing the worst-case perturbations
  - $\approx$ gradient ascent

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

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

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

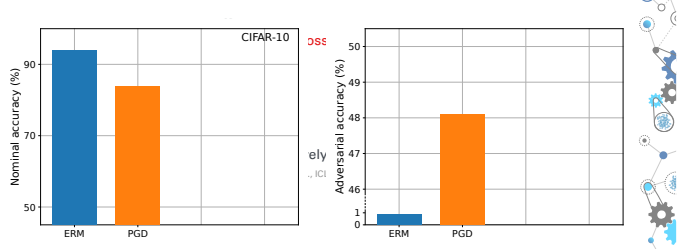
- Computing the worst-case perturbations
  - ≈ gradient ascent: non-convex and (severely) underparametrized

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

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

- 

[R^{1}C^{1}APH, NeurIPS'21]

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

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

- Computing the worst-case perturbations
  - ≈ gradient ascent: non-convex and (severely) underparametrized

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

- Balancing nominal accuracy and robustness
  - Penalty-based method (e.g., [Zhang et al., ICML'19])

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

- Adversarial training

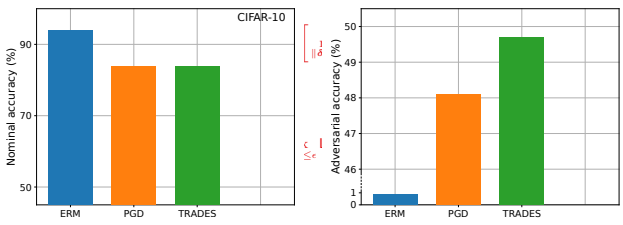
$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_\theta(x_n), y_n\big) \rightarrow \min_{\theta} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\|\delta\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(x_n+\delta), y_n\big)\right]$$

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

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

- 

[Zhang et al., ICML'19]

Adversarial training

Problem

Learn an image classifier that is robust to input perturbations

- Adversarial training

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

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

Penalty-based methods

Problem

Learn an image classifier that is robust to input perturbations

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

- No straightforward relation between λ and adversarial loss

Penalty-based methods

Problem

Learn an image classifier that is robust to input perturbations

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

- No straightforward relation between λ and adversarial loss
- λ depends on the values of the losses (dataset, model, performance measure)

Penalty-based methods

Problem
Learn an image classifier that is robust to input perturbations

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

- No straightforward relation between λ and adversarial loss
- λ depends on the values of the losses (dataset, model, performance measure)
- Generalization

Adversarial training

Problem
Learn an image classifier that is robust to input perturbations

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

- Computing the worst-case perturbations
  - ✗ $\approx$ gradient ascent: non-convex and (severely) underparametrized
    [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]
- Balancing nominal accuracy and robustness
  - ✗ Penalty-based method: value of λ and generalization

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations

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

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

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations

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

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

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations

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

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

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations

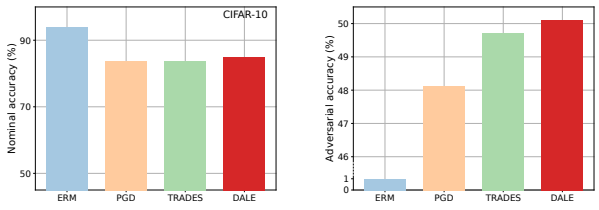
$$\begin{aligned} \min_{\theta} \quad & \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right] \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n),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

- More natural: nominal performance is a constraint, not a cost
- Decouple performance and requirements

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

Constrained learning for robustness

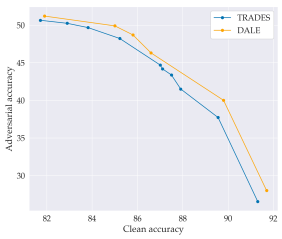
Problem
Learn an image classifier that is robust to input perturbations



[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Constrained learning for robustness

Problem
Learn an image classifier that is robust to input perturbations



[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Claims

Constrained learning is the right tool to learn under requirements

Constrained learning is hard...

...but possible

What is (un)constrained learning?

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

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

What is (un)constrained learning?

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

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

What is (un)constrained learning?

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

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

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.)

Constrained learning challenges

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

Challenges

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

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.)

Constrained learning challenges

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

Challenges

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

Claims

Constrained learning is the right tool to learn under requirements

Constrained learning is hard...

...but possible

Constrained learning challenges

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

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

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

Challenges

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

Constrained learning challenges

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

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

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

Challenges

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

Constrained learning challenges

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

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

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

Challenges

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

A PAC learning primer

- Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

  - e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

A PAC learning primer

- Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

  - e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

- $f_{\boldsymbol{\theta}}$ is *probably approximately correct (PAC)* learnable

A PAC learning primer

- Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

  - e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

- $f_{\boldsymbol{\theta}}$ is *probably approximately correct (PAC)* learnable

A PAC learning primer

- Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

  - e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

- $f_{\boldsymbol{\theta}}$ is *probably approximately correct (PAC)* learnable

- Tools: Rademacher complexity, VC dimensionality...

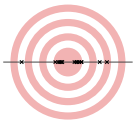
A PAC learning primer

- Classical learning theory [Vapnik & Chervonenkis, TP'71; Valiant, CACM'84]:

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

  - e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...

- f_θ is *probably approximately correct (PAC)* learnable
- Tools: Rademacher complexity, VC dimensionality...
- Requirements?



What's in a solution?

Definition (PACC learnability)

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

What's in a solution?

Definition (PACC learnability)

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

1) near-optimal

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

What's in a solution?

Definition (PACC learnability)

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

1) near-optimal (≈ **PAC learnable**)

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

What's in a solution?

Definition (PACC learnability)

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

1) near-optimal

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

2) approximately feasible

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

What's in a solution?

Definition (PACC learnability)

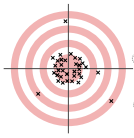
f_θ is a *probably approximately correct constrained (PACC)* learnable if for every ϵ, δ and every distributions $\mathfrak{D}, \mathfrak{A}$, we can obtain $f_{\theta^\dagger}$ from $N_f(\epsilon, \delta)$ samples such that, with prob. $1-\delta$,

1) near-optimal

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

2) approximately feasible

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



When is it possible to learn under constraints?

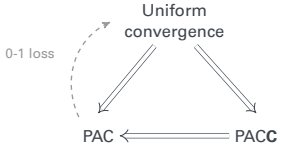
Theorem

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

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

When is it possible to learn under constraints?

Theorem



Constrained learning challenges

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

PAC(C)

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

Challenges

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

Constrained learning challenges

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

PAC(C)

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

Challenges

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

Constrained optimization methods

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

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)
- Interior point methods
  e.g., barriers, projection, polyhedral approx.

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

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
  - Feasible candidate solution ~~[strong duality]~~

Duality

PRIMAL

↕

DUAL

Duality

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

↕

DUAL

Duality

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

$$\updownarrow$$

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

Duality

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

$$\updownarrow$$

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

- Unconstrained optimization is "easier" than constrained optimization

Duality

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

$$\updownarrow$$

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

- Unconstrained optimization is "easier" than constrained optimization
- In general, $\hat{D}^\star \leq \hat{P}^\star$

Duality

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

$$\updownarrow$$

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

- Unconstrained optimization is "easier" than constrained optimization
- 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]

Break
Robust learning revisited

Adversarial training: Challenges

Problem
Learn an image classifier that is robust to input perturbations

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)\right]$$
$$\text{subject to} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \leq c$$

- Computing the worst-case perturbations
  - ✗ ≈ gradient ascent: non-convex and (severely) underparametrized
    [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]
- Balancing nominal accuracy and robustness
  - ✓ Constrained learning [CR, NeurIPS'20; R*C*APH, NeurIPS'21; CPCR, IEEE TIT'23]

Adversarial training: Challenges

Problem
Learn an image classifier that is robust to input perturbations

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)\right]$$
$$\text{subject to} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\big)\right] \leq c$$

- Computing the worst-case perturbations
  - ✗ ≈ gradient ascent: non-convex and (severely) underparametrized
    [Szegedy et al., ICLR'14; Goodfellow et al., ICLR'15; Madry et al., ICLR'18; ...]
- Balancing nominal accuracy and robustness
  - ✓ Constrained learning [CR, NeurIPS'20; R*C*APH, NeurIPS'21; CPCR, IEEE TIT'23]

Semi-infinite constrained learning

Problem
Learn an image classifier that is robust to input perturbations

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)\right]$$

Semi-infinite constrained learning

Problem
Learn an image classifier that is robust to input perturbations

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[t(x,y)\right] \\ \text{subject to} \quad & \text{Loss}\big(f_\theta(x+\delta),y\big) \le t(x,y), \\ & \text{for a.e. } (x,y) \text{ and } \|\delta\|_\infty \le \epsilon \end{aligned}$$

- 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

Problem
Learn an image classifier that is robust to input perturbations

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[t(x,y)\right] \\ \text{subject to} \quad & \text{Loss}\big(f_\theta(x+\delta_0),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{\sqrt{2}}),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_\pi),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{\pi^2}),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_4),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{e^2}),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{e^\pi}),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{\pi^\pi}),y\big) \le t(x,y) \\ & \text{Loss}\big(f_\theta(x+\delta_{25}),y\big) \le t(x,y) \end{aligned}$$

- 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

Semi-infinite constrained learning

Proposition
Let $(x,y) \mapsto \ell(f_\theta(x,y)) \in L^2$ and $\mathcal{P}^2$ be the space of square integrable probability distributions. Then,

$$P^\star = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[t(x,y)\right] \text{ subject to } \ell\big(f_\theta(x+\delta),y\big) \le t(x,y), \forall (x,y,\delta)$$
$$\Big\updownarrow =$$
$$D^\star = \min_{\theta} \max_{\mu\in\mathcal{P}^2} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\int \mu(x,y,\delta)\ell\big(f_\theta(x+\delta),y\big)d\delta\right]}_{L(\theta,\mu)}$$

[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Semi-infinite constrained learning

Proposition
Let $(x,y) \mapsto \ell(f_\theta(x,y)) \in L^2$ and $\mathcal{P}^2$ be the space of square integrable probability distributions. Then,

$$P^\star = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)\right]$$
$$\Big\updownarrow =$$
$$D^\star = \min_{\theta} \max_{\mu\in\mathcal{P}^2} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\int \mu(x,y,\delta)\ell\big(f_\theta(x+\delta),y\big)d\delta\right]}_{L(\theta,\mu)}$$

[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Semi-infinite constrained learning

Proposition
Let $(x,y) \mapsto \ell(f_\theta(x,y)) \in L^2$ and $\mathcal{P}^2$ be the space of square integrable probability distributions. Then,

$$P^\star = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)\right]$$
$$\Big\updownarrow =$$
$$D^\star = \min_{\theta} \max_{\mu\in\mathcal{P}^2} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)}\left[\ell\big(f_\theta(x+\delta),y\big)\right]\right]}_{L(\theta,\mu)}$$

- $\mu(x,y)$ is a conditional probability (Radon-Nikodym derivative) of perturbations

[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Semi-infinite constrained learning

Proposition
Let $(x,y) \mapsto \ell(f_\theta(x,y)) \in L^2$ and $\mathcal{P}^2$ be the space of square integrable probability distributions. Then,

$$P^\star = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)\right]$$
$$\Big\updownarrow =$$
$$D^\star = \min_{\theta} \max_{\mu\in\mathcal{P}^2} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)}\left[\ell\big(f_\theta(x+\delta),y\big)\right]\right]}_{L(\theta,\mu)}$$

Proposition
For all $\xi > 0$, there exists $\gamma(x,y) < \max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)$ s.t. $L(\theta,\mu_\gamma) \ge \max_{\mu\in\mathcal{P}^2} L(\theta,\mu) - \xi$ for

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

[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Semi-infinite constrained learning

Proposition
Let $(x,y) \mapsto \ell(f_\theta(x,y)) \in L^2$ and $\mathcal{P}^2$ be the space of square integrable probability distributions. Then,

$$P^\star = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)\right]$$
$$\Big\updownarrow \approx$$
$$D^\star = \min_{\theta} \max_{\mu\in\mathcal{P}^2} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{E}_{\delta\sim\mu_\gamma(\delta|x,y)}\left[\ell\big(f_\theta(x+\delta),y\big)\right]\right]}_{L(\theta,\mu)}$$

Proposition
For all $\xi > 0$, there exists $\gamma(x,y) < \max_{\|\delta\|_\infty\le\epsilon} \ell\big(f_\theta(x+\delta),y\big)$ s.t. $L(\theta,\mu_\gamma) \ge \max_{\mu\in\mathcal{P}^2} L(\theta,\mu) - \xi$ for

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

[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Dual Adversarial LEarning

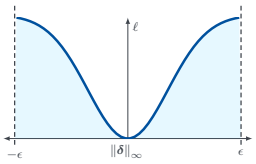
$$\begin{cases} \delta' \sim \mu_\gamma(\cdot|x,y) \\ \theta^+ = \theta - \eta\nabla_\theta \ell\big(f_\theta(x+\delta'),y\big) \end{cases}$$

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

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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

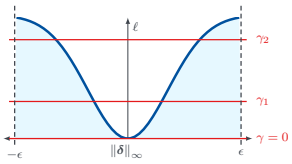


- γ control the smoothness of the approximation

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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



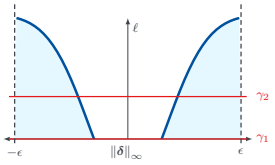
- γ control the smoothness of the approximation

$\gamma = 0$
(oversmoothed)

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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



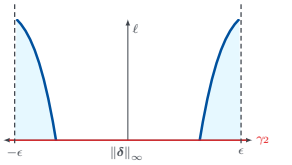
- γ control the smoothness of the approximation

$\gamma = 0$
(oversmoothed)

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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



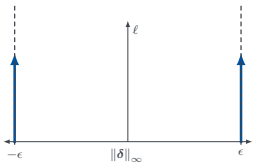
- γ control the smoothness of the approximation

$\gamma = 0$
(oversmoothed)

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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



- γ control the smoothness of the approximation

$\gamma = 0$
(oversmoothed)

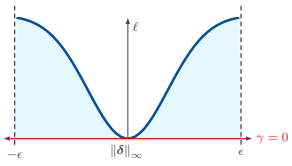
$\gamma \to \max_{\boldsymbol{\delta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big)$
(undersmoothed)

hard to sample

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \mu_\gamma(\cdot|\boldsymbol{x}, y) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

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



- γ control the smoothness of the approximation

$\gamma = 0$
(oversmoothed)

easy to sample

$\gamma \to \max_{\boldsymbol{\delta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big)$
(undersmoothed)

hard to sample

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

- $\mu_0 \in L^2$ and μ_0 as "differentiable" as $\ell \circ f_{\boldsymbol{\theta}}$ $\Rightarrow$ Langevin/Hamiltonian Monte Carlo

$$\boldsymbol{\delta}^+ = \underset{\|\boldsymbol{\delta}\|_\infty \le \epsilon}{\text{proj}} \Big[\boldsymbol{\delta} + \eta \,\text{sign}\Big[\nabla_{\boldsymbol{\delta}} \log\big(\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big)\big)\Big] + \sqrt{2\eta T}\boldsymbol{\zeta}\Big], \quad \boldsymbol{\zeta} \sim \text{Laplace}(0, I)$$

Dual Adversarial LEarning

$$\begin{cases} \boldsymbol{\delta}' \sim \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big) \\ \boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}'), y\big) \end{cases}$$

- $\mu_0 \in L^2$ and μ_0 as "differentiable" as $\ell \circ f_{\boldsymbol{\theta}}$ $\Rightarrow$ Langevin/Hamiltonian Monte Carlo

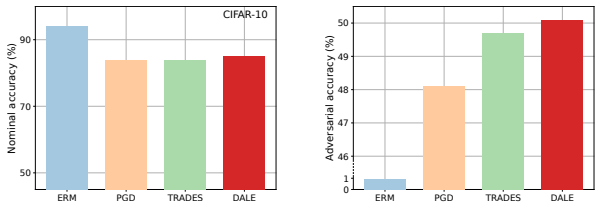
$$\boldsymbol{\delta}^+ = \underset{\|\boldsymbol{\delta}\|_\infty \le \epsilon}{\text{proj}} \Big[\boldsymbol{\delta} + \eta \,\text{sign}\Big[\nabla_{\boldsymbol{\delta}} \log\big(\ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y\big)\big)\Big] + \sqrt{2\eta T}\boldsymbol{\zeta}\Big], \quad \boldsymbol{\zeta} \sim \text{Laplace}(0, I)$$

- If $\ell \circ f_{\boldsymbol{\theta}}$ is not differentiable $\Rightarrow$ Metropolis-Hastings

Constrained learning for robustness

Problem

Learn an image classifier that is robust to input perturbations

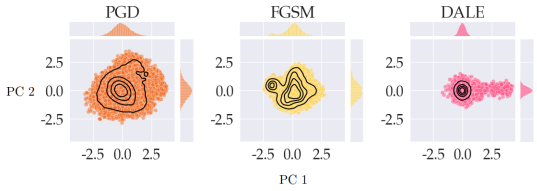


[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Constrained learning for robustness

Problem

Learn an image classifier that is robust to input perturbations



[Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]

Constrained learning challenges

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

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

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

PAC(C)

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

$$\text{subject to} \ \ \mathbb{E}_{(x,y)\sim\mathcal{R}}\Big[g\big(f_{\theta}(x), y\big)\Big] \leq c$$

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

Challenges

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

Duality

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

$$\updownarrow$$

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

- Unconstrained optimization is "easier" than constrained optimization
- 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

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

$$\updownarrow$$

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

- Unconstrained optimization is "easier" than constrained optimization
- 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]~~

Non-convex variational duality

Convex optimization: Primal ⟷ Dual

Non-convex, finite dimensional optimization: Primal ⟷ Dual

Non-convex variational duality

Convex optimization: Primal ⟷ Dual

Non-convex, finite dimensional optimization: Primal ⟷ Dual

Non-convex, infinite dimensional optimization: Primal ⟷ Dual

[C., Eldar, Ribeiro, IEEE TSP'20]

Sparse logistic regression

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

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

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

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$
$$\text{s. to } \|\boldsymbol{\theta}\|_1 = \sum_{t=1}^{p} |\boldsymbol{\theta}_t| \leq c$$

Discrete, convex

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

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$
$$\text{s. to } \|\boldsymbol{\theta}\|_1 = \sum_{t=1}^{p} |\boldsymbol{\theta}_t| \leq c$$

Discrete, convex

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

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \boldsymbol{\theta}^T \boldsymbol{x}_n\right)\right]$$
$$\text{s. to } \|\boldsymbol{\theta}\|_1 = \sum_{t=1}^{p} |\boldsymbol{\theta}_t| \leq c$$

Discrete, convex

e.g., [Foucart & Rauhut, 2013]

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

[C., Eldar, and Ribeiro, 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

$$\min_{\theta\in L_2} -\sum_{n=1}^{N} \log\left[1+\exp\left(y_n \cdot \int \theta(t) x_n(t) dt\right)\right]$$
$$\text{s. to } \|\theta\|_{L_0} = \int \mathbb{I}\left[\theta(t) \neq 0\right] dt \leq \frac{k}{p}$$

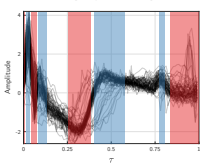
Continuous, non-convex

[C., Eldar, and Ribeiro, IEEE TSP'20]: tractable

Non-convex variational duality

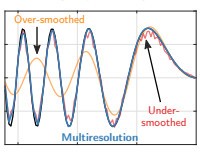
EEG classification

[CER, IEEE TSP'20]



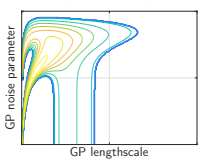
Multiresolution RKHS

[PCPR, IEEE TSP'20]



Bayesian posterior learning

[CPR, Asilomar'19]



[C., Eldar, and Ribeiro, IEEE TSP'20; Peifer, C., Paternain, and Ribeiro, IEEE TSP'20; C., Paternain, and Ribeiro, Asilomar'19]

How can we learn under constraints?

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

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

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

How can we learn under constraints?

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

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

$$P^\star = \min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right)\right] \text{ subject to } \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{A}}\left[g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right)\right] \leq c$$

How can we learn under constraints?

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

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

$$P^\star = \min_{\boldsymbol{\theta}} \int \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) d\mathfrak{D}(\boldsymbol{x}, y) \text{ subject to } \int g\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) d\mathfrak{A}(\boldsymbol{x}, y) \leq c$$

[C. and Ribeiro, NeurIPS'20; C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23] 56

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

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

[$\mathcal{H}$ is a good covering of $\overline{\text{conv}}(\mathcal{H})$]

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

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

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

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

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

(mild conditions apply)

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

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

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

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

(mild conditions apply)

[C., Paternain, Calvo-Fullana, and Ribeiro, ICASSP'20 (best student paper); C. and Ribeiro, NeurIPS'20; C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23] 57

Dual 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 $1-\delta$,

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

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

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

Source of error
parametrization richness (ν) sample size (N) requirements difficulty ($\lambda^\star$)

Dual 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 $1-\delta$,

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

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

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

Source of error
parametrization richness (ν) sample size (N) requirements difficulty ($\lambda^\star$)

[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23] 58

Dual 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 $1-\delta$,

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

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

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

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

[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Dual 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 $1-\delta$,

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

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

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

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

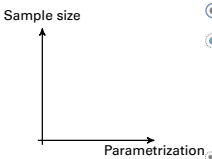
[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

58

Dual learning trade-offs

- Unconstrained learning

  parametrization $\times$ sample size



[C., Paternain, Calvo-Fullana, and Ribeiro, ICASSP'20 (best student paper); C. and Ribeiro, NeurIPS'20; C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

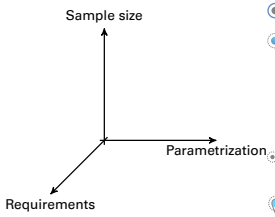
Dual learning trade-offs

- Unconstrained learning

  parametrization $\times$ sample size

- Constrained learning

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



[C., Paternain, Calvo-Fullana, and Ribeiro, ICASSP'20 (best student paper); C. and Ribeiro, NeurIPS'20; C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23] 59

Claims

Constrained learning is the right tool to learn under requirements

Constrained learning is hard...

...but possible. How?

Break
Learning
with invariance

Learning with invariance

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

Learning with invariance

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

62

Learning with invariance

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


Cello


Cello


Cello


Cello

- Why?
  - Improve accuracy
  - Reduce sample complexity
  - Robustness

Learning with invariance

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

[Musil et al., Chem. Rev.'21; Lu et al., SIAM JSC'21; Jumper et al., Nature'21...]

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

[Cohen et al., ICML'16, ; Anderson et al., NeurIPS'19; Finzi et al., ICML'20; Li et al., ICRL'21...]

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

[Lagaris et al., IEEE TNN'98; Raisi et al., JCP'19...]

Learning with invariance

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

[Musil et al., Chem. Rev.'21; Lu et al., SIAM JSC'21; Jumper et al., Nature'21...]

- ✗ Manual design ⇒ Limited $\mathcal{G}$

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

[Lagaris et al., IEEE TNN'98; Raisi et al., JCP'19...]

- ✗ Time-consuming hyperparameter tuning
- ✗ Guarantee?

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

[Cohen et al., ICML'16, ; Anderson et al., NeurIPS'19; Finzi et al., ICML'20; Li et al., ICRL'21...]

- ✗ Manual design ⇒ Limited $\mathcal{G}$
- ✗ Computational complexity

Learning with invariance

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

[Musil et al., Chem. Rev.'21; Lu et al., SIAM JSC'21; Jumper et al., Nature'21...]

- ✗ Manual design ⇒ Limited $\mathcal{G}$

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

[Lagaris et al., IEEE TNN'98; Raisi et al., JCP'19...]

- ✗ Time-consuming hyperparameter tuning
- ✗ Guarantee?

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

[Cohen et al., ICML'16, ; Anderson et al., NeurIPS'19; Finzi et al., ICML'20; Li et al., ICRL'21...]

- ✗ Manual design ⇒ Limited $\mathcal{G}$
- ✗ Computational complexity

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

[...]

Data augmentation

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

Data: $\{(\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

Data augmentation

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

Data: $\{(\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

$$\downarrow$$

$$\min_{\boldsymbol{\theta}} \hat{R}_{\text{aug}}(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \left[\frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} \ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\right]$$

Data: $\bigcup_{g\in\mathcal{G}} \{(g\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

- $\mathcal{G}$: set of transformations (id $\in \mathcal{G}$)

Data augmentation

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

Data: $\{(\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

$$\downarrow$$

$$\min_{\boldsymbol{\theta}} \hat{R}_{\text{aug}}(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{G}}\Big[\ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\Big]$$

Data: $\bigcup_{g\in\mathcal{G}} \{(g\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

- $\mathcal{G}$: set of transformations (id $\in \mathcal{G}$)
- $\mathfrak{G}$: distribution over $\mathcal{G}$ (typically, uniform)

Data augmentation

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

Data: $\{(\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

$$\downarrow$$

$$\min_{\boldsymbol{\theta}} \hat{R}_{\text{aug}}(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{G}}\Big[\ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\Big]$$

Data: $\bigcup_{g\in\mathcal{G}} \{(g\boldsymbol{x}_n, y_n)\}_{n=1,\dots,N}$

$$\downarrow$$

$$\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big)\Big]$$

- $\mathcal{G}$: set of transformations (id $\in \mathcal{G}$)
- $\mathfrak{G}$: distribution over $\mathcal{G}$ (typically, uniform)

Data augmentation

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

$$\min_{\theta} \hat{R}_{\text{aug}}(\theta) \triangleq \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{(x,y)\sim\mathfrak{G}}\Big[\ell\big(f_\theta(gx_n), y_n\big)\Big]$$

$$\min_{\theta} R(\theta) \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\big(f_\theta(x_n), y_n\big)\Big]$$

- $\mathcal{G}$: set of transformations (id $\in \mathcal{G}$)
- $\mathfrak{G}$: distribution over $\mathcal{G}$ (typically, uniform)

Data: $\{(x_n, y_n)\}_{n=1,\dots,N}$

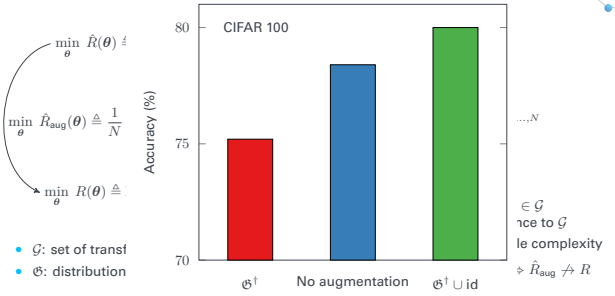
Data: $\bigcup_{g\in\mathcal{G}} \{(gx_n, y_n)\}_{n=1,\dots,N}$

- If...
  - $\mathcal{D}$ is invariant to $g \in \mathcal{G}$
  - $\mathfrak{G}$ induces invariance to $\mathcal{G}$

  $\Rightarrow$ reduce sample complexity
- If $\mathfrak{G}$ is chosen poorly $\Rightarrow \hat{R}_{\text{aug}} \not\to R$

[Chen et al., NeurIPS'20; Bietti et al., NeurIPS'21; Shao et al., NeurIPS'22; Hounie, **Chamon**, Ribeiro, ICML'23]

Data augmentation



From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

[Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - $f_\theta(gx) \overset{d}{=} f_\theta(x)$, for $x \sim \mathcal{D}$
  - $f_\theta(gx) = f_\theta(x)$, for all $g \in \mathcal{G}$
  - $\ell\big(f_\theta(gx), y\big) = \ell\big(f_\theta(x), y\big)$, for all $g \in \mathcal{G}$

[Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - $f_\theta(gx) \overset{d}{=} f_\theta(x)$, for $x \sim \mathcal{D}$
  - $f_\theta(gx) = f_\theta(x)$, for all $g \in \mathcal{G}$
  - $\ell\big(f_\theta(gx), y\big) = \ell\big(f_\theta(x), y\big)$, for all $g \in \mathcal{G}$

  $\Rightarrow \hat{R}_{\text{aug}} \approx R$

[Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✗ $f_\theta(gx) \overset{d}{=} f_\theta(x)$, for $x \sim \mathcal{D}$
  - ✗ $f_\theta(gx) = f_\theta(x)$, for all $g \in \mathcal{G}$
  - ✓ $\ell\big(f_\theta(gx), y\big) = \ell\big(f_\theta(x), y\big)$, for all $g \in \mathcal{G}$

  $\Rightarrow \hat{R}_{\text{aug}} \approx R$

[Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal

Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✓ $\ell\big(f_\theta(gx), y\big) = \ell\big(f_\theta(x), y\big)$, for all $g \in \mathcal{G}$ $(\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?

[Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✅ $\ell\big(f_\theta(g\boldsymbol{x}), y\big) = \ell\big(f_\theta(\boldsymbol{x}), y\big), \quad \text{for all } g \in \mathcal{G} \quad (\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?
  ⇒ Fit the data while maintaining a certain level of invariance

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✅ $\ell\big(f_\theta(g\boldsymbol{x}), y\big) = \ell\big(f_\theta(\boldsymbol{x}), y\big), \quad \text{for all } g \in \mathcal{G} \quad (\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?
  ⇒ Fit the data while maintaining a certain level of invariance

$$\ell_{\text{inv}}\big(f_\theta(\boldsymbol{x}), y\big) = \max_{g\in\mathcal{G}} \ell\big(f_\theta(g\boldsymbol{x}), y\big) - \ell\big(f_\theta(\boldsymbol{x}), y\big) \geq 0 \quad (\text{id} \in \mathcal{G})$$

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✅ $\ell\big(f_\theta(g\boldsymbol{x}), y\big) = \ell\big(f_\theta(\boldsymbol{x}), y\big), \quad \text{for all } g \in \mathcal{G} \quad (\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?
  ⇒ Fit the data while maintaining a certain level of invariance

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell_{\text{inv}}\big(f_\theta(\boldsymbol{x}), y\big)\Big] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\max_{g\in\mathcal{G}} \ell\big(f_\theta(g\boldsymbol{x}), y\big)\Big] - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(\boldsymbol{x}), y\big)\Big] \leq c$$

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✅ $\ell\big(f_\theta(g\boldsymbol{x}), y\big) = \ell\big(f_\theta(\boldsymbol{x}), y\big), \quad \text{for all } g \in \mathcal{G} \quad (\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?
  ⇒ Fit the data while maintaining a certain level of invariance

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell_{\text{inv}}\big(f_\theta(\boldsymbol{x}), y\big)\Big] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\max_{g\in\mathcal{G}} \ell\big(f_\theta(g\boldsymbol{x}), y\big)\Big] - \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(\boldsymbol{x}), y\big)\Big]}_{R(\theta)} \leq c$$

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

- *Which* invariance?
  - ✅ $\ell\big(f_\theta(g\boldsymbol{x}), y\big) = \ell\big(f_\theta(\boldsymbol{x}), y\big), \quad \text{for all } g \in \mathcal{G} \quad (\Rightarrow \hat{R}_{\text{aug}} \approx R)$
- What if f_θ cannot express invariance to $\mathcal{G}$?
  ⇒ Fit the data while maintaining a certain level of invariance

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell_{\text{inv}}\big(f_\theta(\boldsymbol{x}), y\big)\Big] = \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\max_{g\in\mathcal{G}} \ell\big(f_\theta(g\boldsymbol{x}), y\big)\Big]}_{\text{"Adversarial robustness"}} - \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\Big[\ell\big(f_\theta(\boldsymbol{x}), y\big)\Big]}_{R(\theta)} \leq c$$

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

$$\begin{aligned} \min_{\theta} \quad & \text{Accuracy} \\ \text{subject to} \quad & \text{Invariance} \leq c \end{aligned}$$

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

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

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance → Robustness → Data Augmentation

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

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance ⇢ Robustness ⇢ Data Augmentation

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

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

$$\equiv \ \text{(DALE)}$$

$$\min_{\boldsymbol{\theta}} \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{g\sim\mu_n}\Big[\ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\Big], \quad \mu_n = \begin{cases} \text{id} & \text{w.p.} \ \propto \frac{1}{\lambda} + \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ g & \text{w.p.} \ \propto \ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \end{cases}$$

[Robey*, **Chamon***, Pappas, Hassani, and Ribeiro, NeurIPS'21; Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

Invariance ⇢ Robustness ⇢ Data Augmentation

- μ_γ is a data augmentation distribution ($\mathfrak{G}$) that yields the best fit for a given invariance level (c)

$$\min_{\boldsymbol{\theta}} \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{g\sim\mu_n}\Big[\ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\Big], \quad \mu_n = \begin{cases} \text{id} & \text{w.p.} \ \propto \frac{1}{\lambda} + \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ g & \text{w.p.} \ \propto \ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \end{cases}$$

[Robey*, **Chamon***, Pappas, Hassani, and Ribeiro, NeurIPS'21; Hounie, **Chamon**, Ribeiro, ICML'23]

From invariance to data augmentation

Goal
Which (if any) data augmentation ($\mathfrak{G}$) induces invariance?

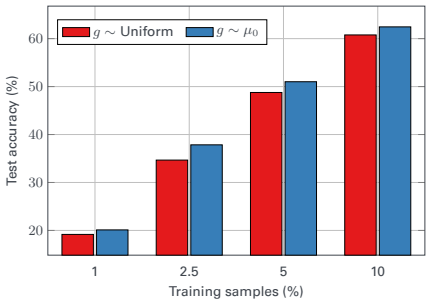
Invariance ⇢ Robustness ⇢ Data Augmentation

- μ_γ is a data augmentation distribution ($\mathfrak{G}$) that yields the best fit for a given invariance level (c)
- No differentiability required (MH sampling)

$$\min_{\boldsymbol{\theta}} \ \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{g\sim\mu_n}\Big[\ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big)\Big], \quad \mu_n = \begin{cases} \text{id} & \text{w.p.} \ \propto \frac{1}{\lambda} + \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\big) \\ g & \text{w.p.} \ \propto \ell\big(f_{\boldsymbol{\theta}}(g\boldsymbol{x}_n), y_n\big) \end{cases}$$

[Robey*, **Chamon***, Pappas, Hassani, and Ribeiro, NeurIPS'21; Hounie, **Chamon**, Ribeiro, ICML'23]

Training on a subset of ImageNet-100



[Hounie, **Chamon**, Ribeiro, ICML'23]

"Identifying" invariances

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

[Hounie, **Chamon**, Ribeiro, ICML'23]

Claims

Constrained learning is the right tool to learn under requirements

Constrained learning is hard...

...but possible. How?

Primal-dual algorithms

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

Primal-dual algorithms

- Minimize the primal (≡ **ERM**)

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

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

Primal-dual algorithms

- Minimize the primal (≡ **ERM**)

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

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

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

Primal-dual algorithms

- Minimize the primal (≡ **ERM**)

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

- Update the dual

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

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

Primal-dual algorithms

- Minimize the primal (≡ **ERM**)

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

- Update the dual

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

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

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 $T = \left\lceil \frac{\|\boldsymbol{\lambda}^\star\|^2}{2\eta \Delta l \nu} \right\rceil + 1$ 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.

[Paternain, **C.**, Calvo-Fullana, and Ribeiro, NeurIPS'19; **C.** and Ribeiro, NeurIPS'20; **C.**, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

In practice...

- Minimize the primal (≡ **ERM**)

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

- Update the dual

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

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

In practice...

- Minimize the primal (≡ **ERM**)

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

- Update the dual

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

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

In practice...

1: Initialize: $\boldsymbol{\theta}_0$, λ_0
2: **for** $t = 1, \dots, T$
3: $\quad \boldsymbol{\beta}_1 \leftarrow \boldsymbol{\theta}_{t-1}$
4: $\quad$ **for** $n = 1, \dots, N$
5: $\quad\quad \boldsymbol{\beta}_{n+1} \leftarrow \boldsymbol{\beta}_n - \eta \nabla_{\boldsymbol{\beta}} \big[\ell\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) + \lambda_{t-1} g\big(f_{\boldsymbol{\beta}_n}(\boldsymbol{x}_n), y_n\big) \big]$
6: $\quad$ **end**
7: $\quad \boldsymbol{\theta}_t \leftarrow \boldsymbol{\beta}_{N+1}$
8: $\quad \lambda_t = \left[\lambda_{t-1} + \eta \left(\frac{1}{N} \sum_{m=1}^{N} \ell\big(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_m), y_m\big) - c \right) \right]_+$
9: **end**
10: Output: $\boldsymbol{\theta}_T, \lambda_T$

PyTorch
https://github.com/lfochamon/csl

Penalty-based vs. dual learning

Penalty-based learning

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

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

Dual learning

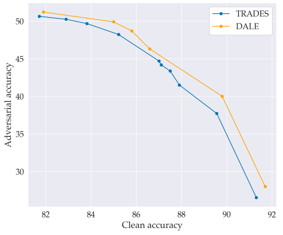
$$\boldsymbol{\theta}^\dagger \in \operatorname*{argmin}_{\boldsymbol{\theta}} \; \text{Loss}(\boldsymbol{\theta}) + \lambda \cdot \text{Penalty}(\boldsymbol{\theta})$$
$$\lambda^+ = \Big[\lambda + \eta \Big(\text{Penalty}(\boldsymbol{\theta}^\dagger) - c \Big) \Big]_+$$

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

Penalty-based vs. dual learning

Penalty-b

$\theta^\dagger \in \operatorname{argmin}_{\theta}$ Loss



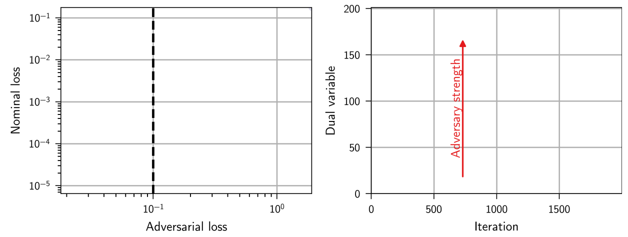
arning

$(\theta) + \lambda \cdot \text{Penalty}(\theta)$

$\text{lty}(\theta^\dagger) - c\Big)\Big]_+$

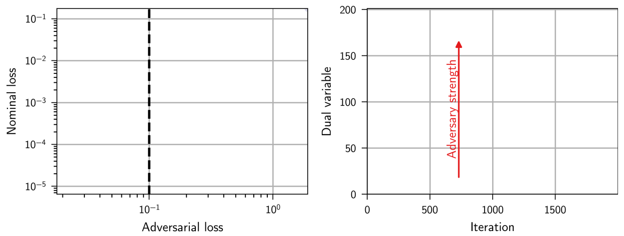
- Parameter: λ (data- ... nent-dependent)
- Generalizes with re ... ect to Loss

Robust image recognition



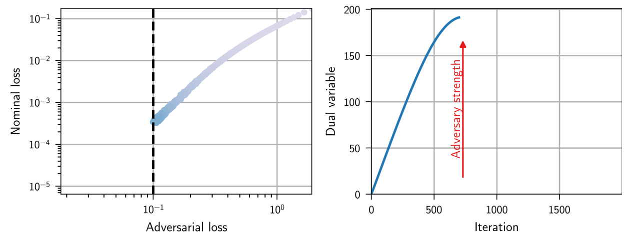
[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robust image recognition



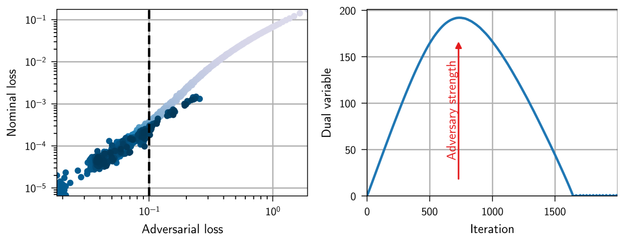
[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robust image recognition



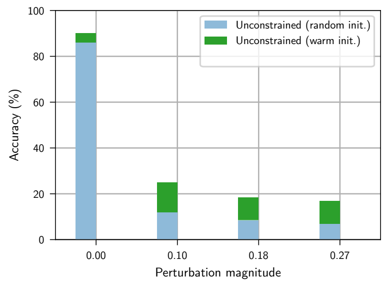
[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robust image recognition



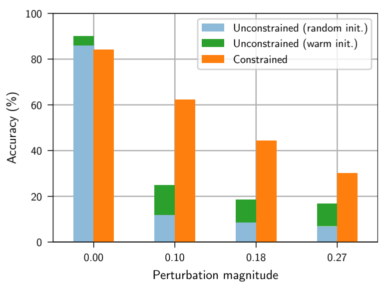
Empirical observations: [Zhang et al., ICML'20; Sitawarin, ArXiv'20]

Robust image recognition



[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Robust image recognition



[C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Claims

(1) Constrained learning is the right tool to learn under requirements

(2) Constrained learning is hard...

(3) ...but possible

Claims

(1) **Constrained learning is ~~the right~~ A tool to learn under requirements**
Constrained learning imposes generalizable requirements organically during training (resource budget, robustness…)

(2) **Constrained learning is hard…**

(3) **…but possible**

Claims

(1) **Constrained learning is ~~the right~~ A tool to learn under requirements**
Constrained learning imposes generalizable requirements organically during training (resource budget, robustness…)

(2) **Constrained learning is hard…**
Constrained, non-convex, statistical optimization problem

(3) **…but possible**

Claims

(1) **Constrained learning is ~~the right~~ A tool to learn under requirements**
Constrained learning imposes generalizable requirements organically during training (resource budget, robustness…)

(2) **Constrained learning is hard…**
Constrained, non-convex, statistical optimization problem

(3) **…but possible**
We can learn under requirements (essentially) whenever we can learn at all

Claims

(1) **Constrained learning is ~~the right~~ A tool to learn under requirements**
Constrained learning imposes generalizable requirements organically during training (resource budget, robustness…)

(2) **Constrained learning is hard…**
Constrained, non-convex, statistical optimization problem

(3) **…but possible. How?**
We can learn under requirements (essentially) whenever we can learn at all:
by solving *(penalized) ERM problems*

The current landscape

- Constrained learning beyond resource allocation and adversarial robustness
  - Fairness [CPCR, ICASSP'20 (best student paper); CR, NeurIPS'20; CPCR, IEEE TIT'23], Invariance [HCR, ICML'23], (manifold) smoothness [CCHVR, ICML'23]…
  - Different forms of robustness [CR, NeurIPS'20; CPCR, IEEE TIT'23; RCPH, ICML'22]
- Constrained (reinforcement) learning [PCCR, NeurIPS'19; PCCR, IEEE TAC'23; CPCR, arXiv'21 (submitted to IEEE TAC)]
- Gradient descent-ascent dynamics and primal recovery
  [PCCR, NeurIPS'19; CR, NeurIPS'20; CPCR, IEEE TSP'23; CPCR, arXiv'21]
- Constrained inference and control
  - Functional sparsity [CPR, Asilomar'19; PCPR, IEEE TSP'20; CER, IEEE TSP'20]
  - (non-submodular) sensor/actuator selection/scheduling [CR, NeurIPS'17; CPR, IEEE TAC'21; CAR, IEEE TAC'22]
  - Risk-aware estimation and control [KCPR, ICASSP'20 (best paper award); TKCRP, CDC'20]
  - Graph(on) signal processing/neural networks [RCR, NeurIPS'20; RCR, IEEE TSP'21; RCR, arXiv'21]

Universität Stuttgart

Collaborators:
Miguel Calvo-Fullana, Juan Cerviño, Mark Eisen, Yonina C. Eldar, Benjamin D. Haeffele, Hamed Hassani, Ignacio Hounie, Dyonisios Kalogerias, Dan D. Lee, George J. Pappas, Santiago Paternain, Alejandro Ribeiro, Alexander Robey, Luana Ruiz, Anastasios Tsiamis, Rene Vidal, Clark Zhang

Luiz F. O. Chamon

www.luizchamon.com
luiz.chamon@simtech.uni-stuttgart.de

learning under requirements

Index

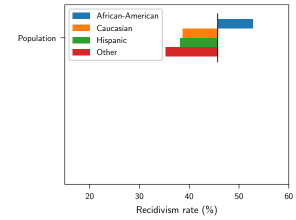
Fair learning

Problem
Predict whether an individual will recidivate

Fair learning

Problem

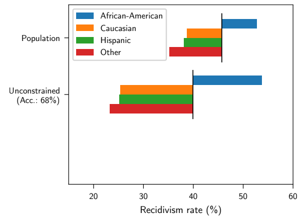
Predict whether an individual will recidivate



Fair learning

Problem

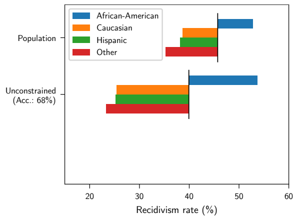
Predict whether an individual will recidivate



Fair learning

Problem

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



Fair learning

Problem

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

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

Fair learning

Problem

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

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

Fair learning

Problem

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

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

Fair learning

Problem

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

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

Fair learning

Problem

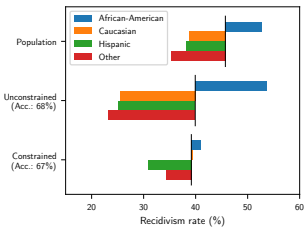
Predict whether an individual will recidivate at the same rate across races

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

- Generalization guarantees (randomized solutions)
  [Goh et al., NeurIPS'16; Kearns et al., ICML'18; Cotter et al., JMLR'19; ...]

Fair learning

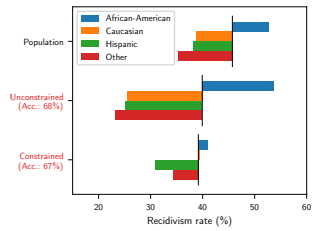
Problem
Predict whether an individual will recidivate at the same rate across races



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

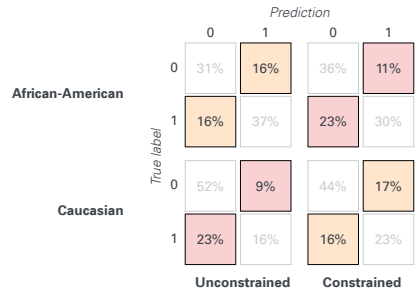
Fair learning

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



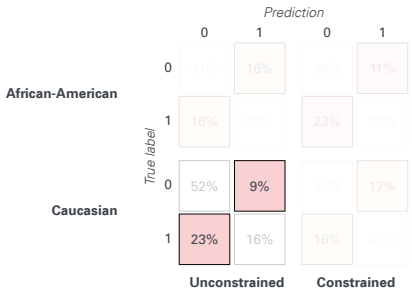
[Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Fair prediction



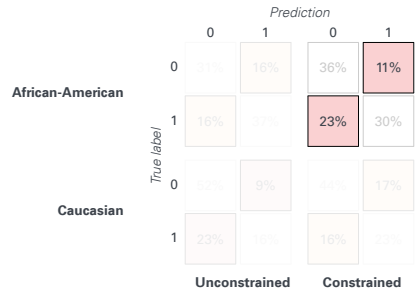
[Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Fair prediction



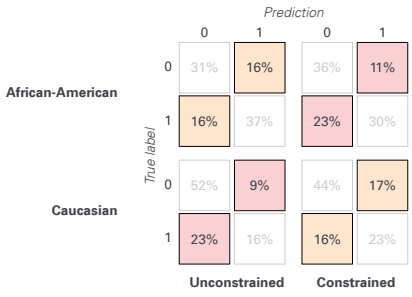
[Chamon, Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Fair prediction



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

Fair prediction



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

Index

Approximate primal-dual for rate constraints

- Minimize the primal

$$\theta^+ = \theta - \eta \left[\text{Loss}\big(f_\theta(x_n), y_n\big) + \lambda \nabla_\theta \left[\mathbb{I}\left(f_\theta(x_n) > 0.5\right) \mathbb{I}(\text{Race}) - \mathbb{I}\left(f_\theta(x_n) > 0.5\right) \right] \right]$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta \left(\frac{1}{N} \sum_{n=1}^{N} \mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right) \mathbb{I}(\text{Race}) - \mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right) - c \right) \right]_+$$

Approximate primal-dual for rate constraints

- Minimize the primal

$$\theta^+ = \theta - \eta\left[\text{Loss}\left(f_\theta(x_n), y_n\right) + \lambda\nabla_\theta\left[\mathbb{I}\left(f_\theta(x_n) > 0.5\right)\mathbb{I}(\text{Race}) - \mathbb{I}\left(f_\theta(x_n) > 0.5\right)\right]\right]$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right)\mathbb{I}(\text{Race}) - \mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right) - c\right)\right]_+$$

Approximate primal-dual for rate constraints

- Minimize (approximately) the primal

$$\theta^+ = \theta - \eta\left[\text{Loss}\left(f_\theta(x_n), y_n\right) + \lambda\nabla_\theta\left[\sigma\left(f_\theta(x_n) - 0.5\right)\mathbb{I}(\text{Race}) - \sigma\left(f_\theta(x_n) - 0.5\right)\right]\right]$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right)\mathbb{I}(\text{Race}) - \mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right) - c\right)\right]_+$$

Approximate primal-dual for rate constraints

- Minimize (approximately) the primal

$$\theta^+ = \theta - \eta\left[\text{Loss}\left(f_\theta(x_n), y_n\right) + \lambda\nabla_\theta\left[\sigma\left(f_\theta(x_n) - 0.5\right)\mathbb{I}(\text{Race}) - \sigma\left(f_\theta(x_n) - 0.5\right)\right]\right]$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right)\mathbb{I}(\text{Race}) - \mathbb{I}\left(f_{\theta^+}(x_n) > 0.5\right) - c\right)\right]_+$$

86

Approximate primal-dual for rate constraints

- Minimize (approximately) the primal

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta\left[\text{Loss}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), y_n\right) + \lambda\nabla_{\boldsymbol{\theta}}\left[\sigma\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - 0.5\right)\mathbb{I}(\text{Race}) - \sigma\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - 0.5\right)\right]\right]$$

- Update the dual

$$\lambda^+ = \left[\lambda + \eta\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_n) > 0.5\right)\mathbb{I}(\text{Race}) - \mathbb{I}\left(f_{\boldsymbol{\theta}^+}(\boldsymbol{x}_n) > 0.5\right) - c\right)\right]_+$$

86