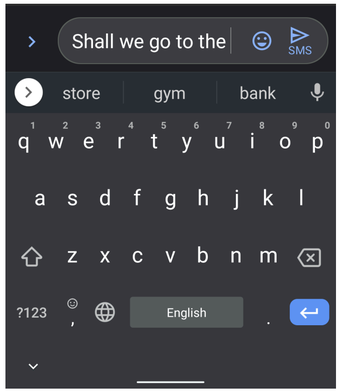
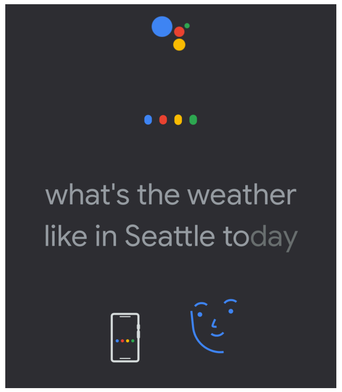
Tackling Distribution Shifts in Federated Learning

Krishna Pillutla

August 6, 2022 @ DRDS Workshop

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING





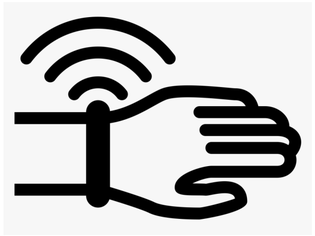


Image Credit: Robotics Business Review

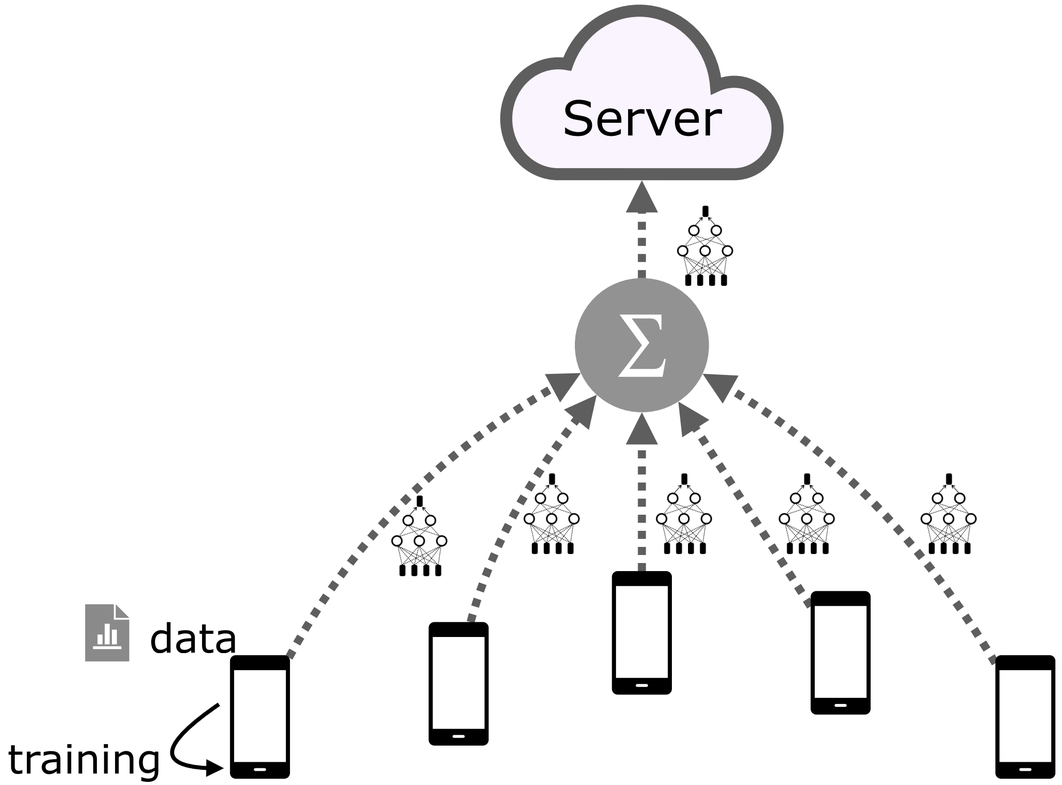
Rieke et al. NPJ Digit. Med. (2020) Image Credit: Wellcome

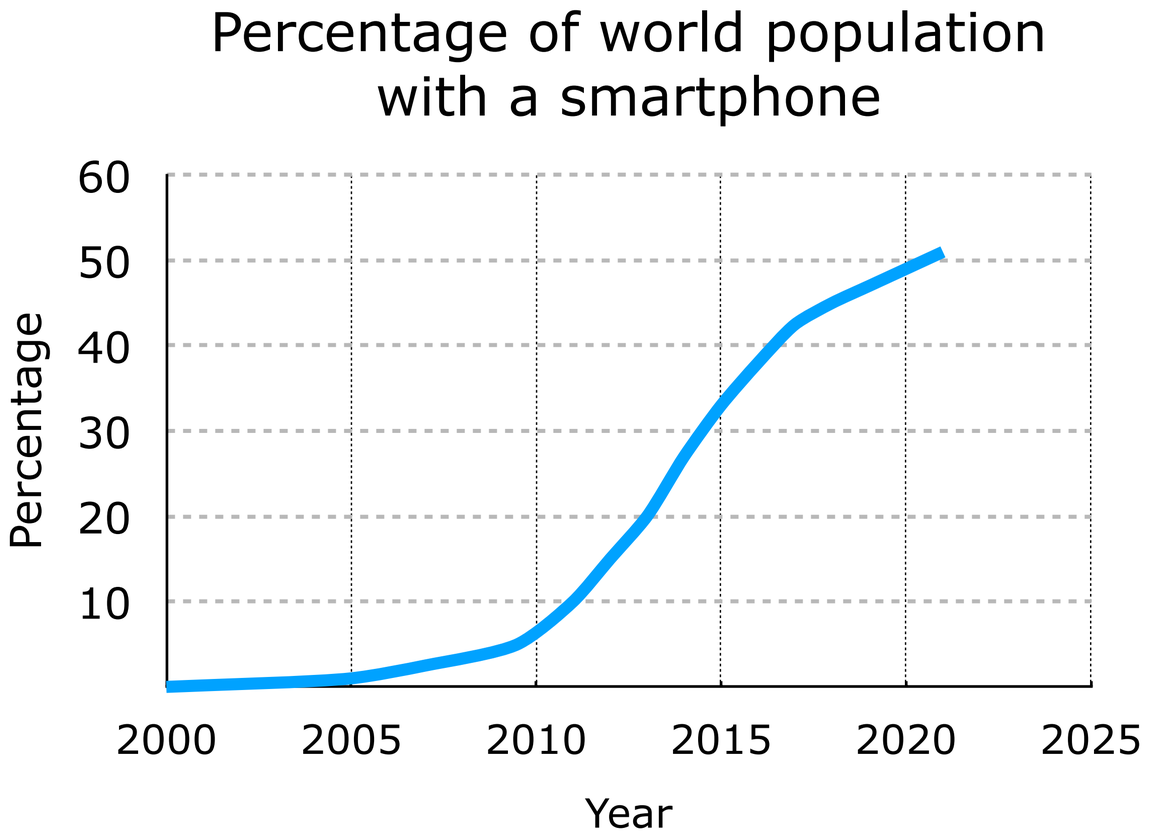
Image Credit: Robotics Business Review

Data is decentralized and private

Rieke et al. NPJ Digit. Med. (2020) Image Credit: Wellcome

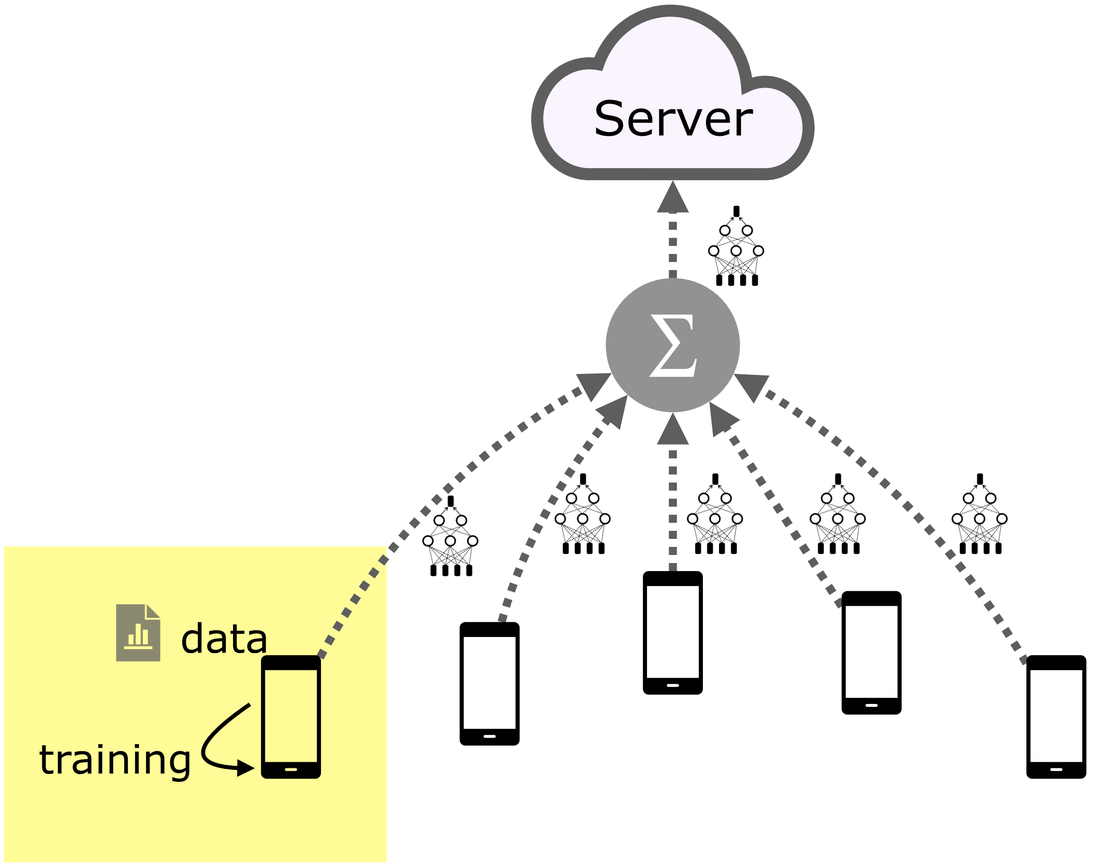
Federated Learning

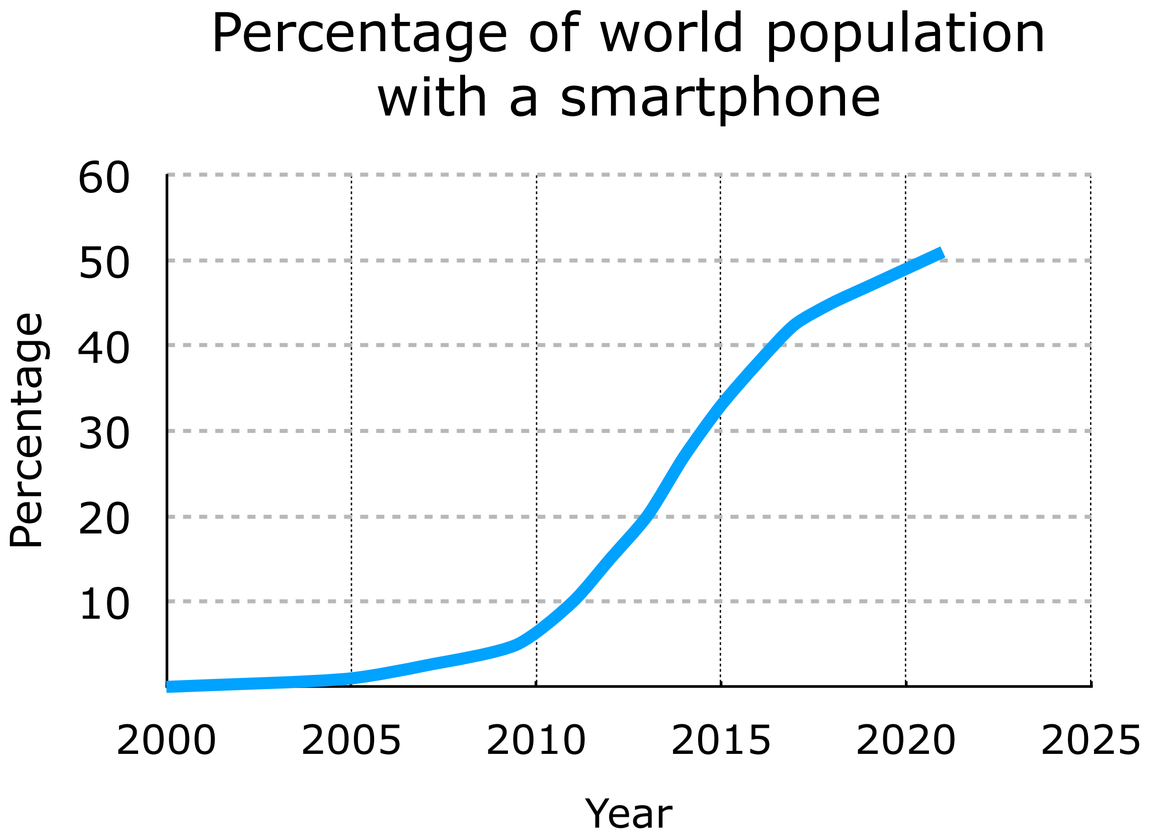




Data Credit: Business Wire

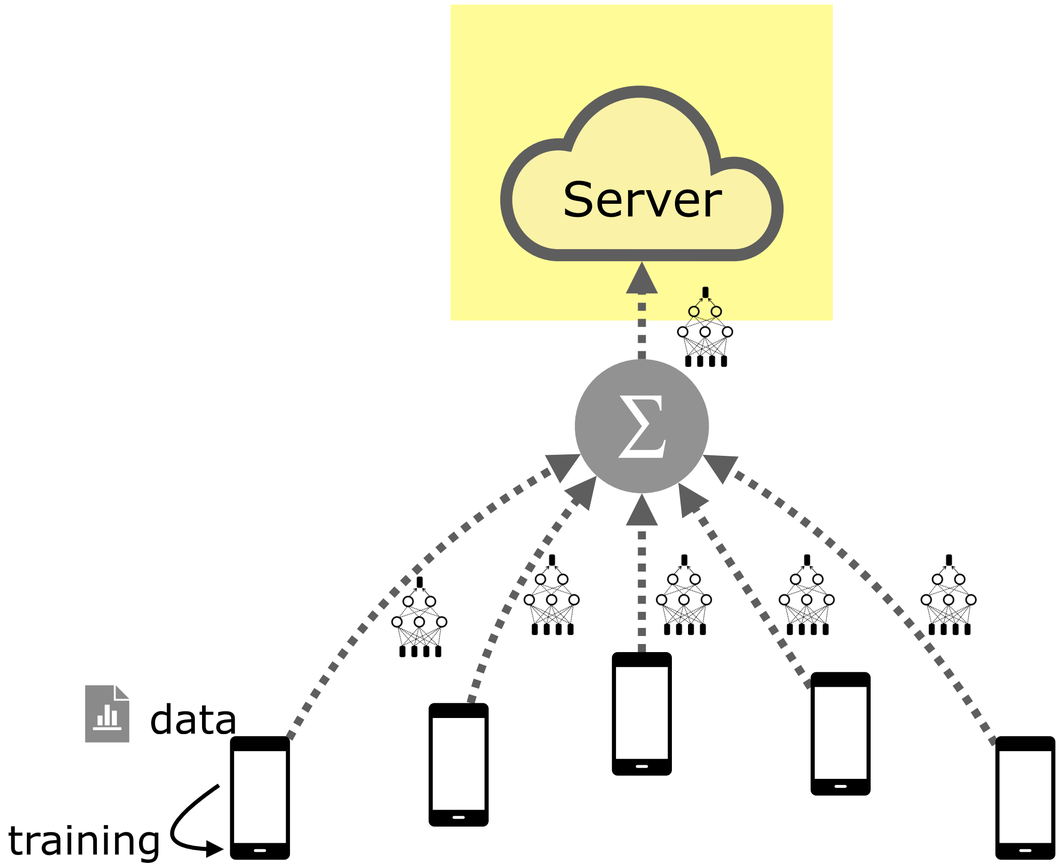
Federated Learning

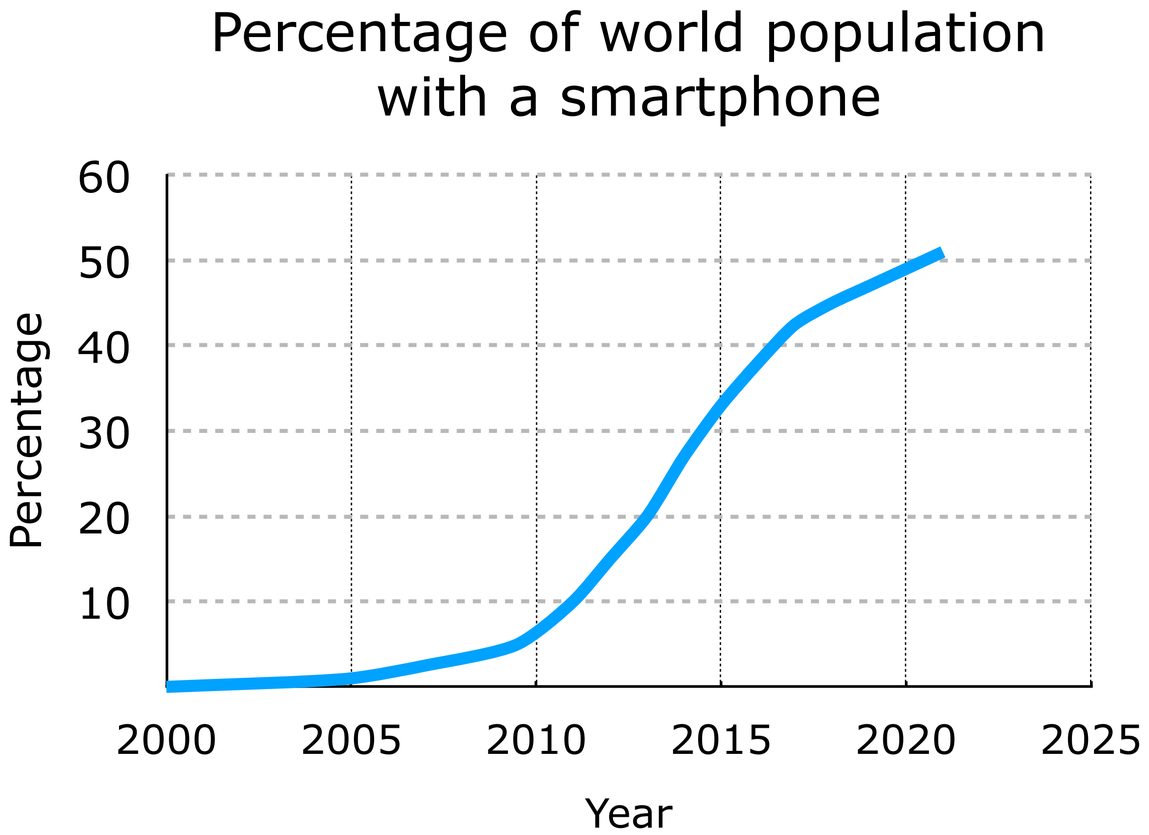




Data Credit: Business Wire

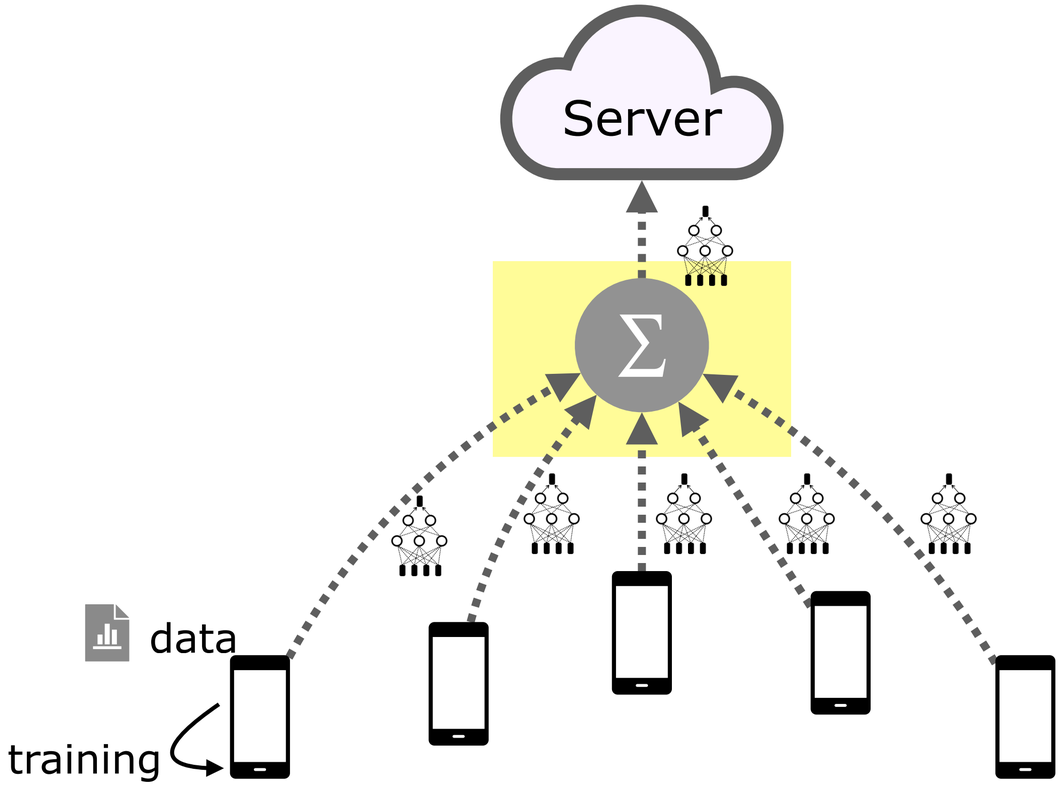
Federated Learning

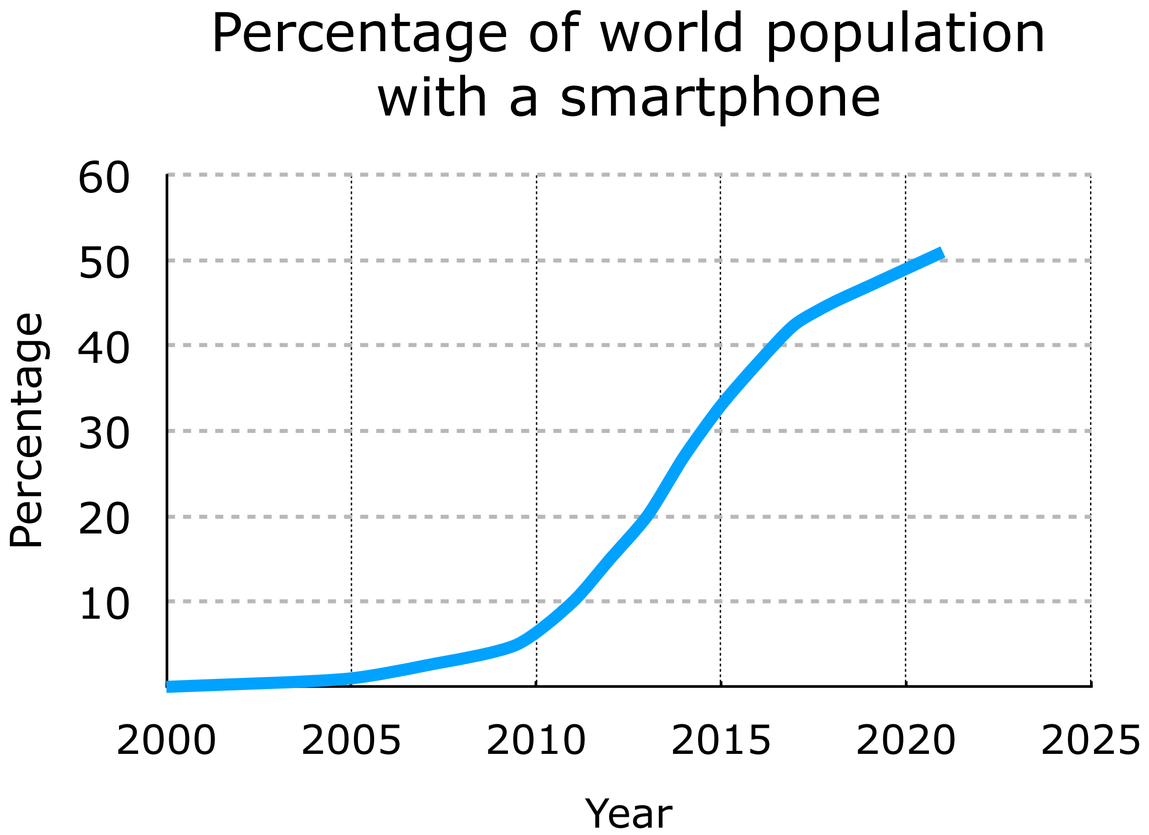




Data Credit: Business Wire

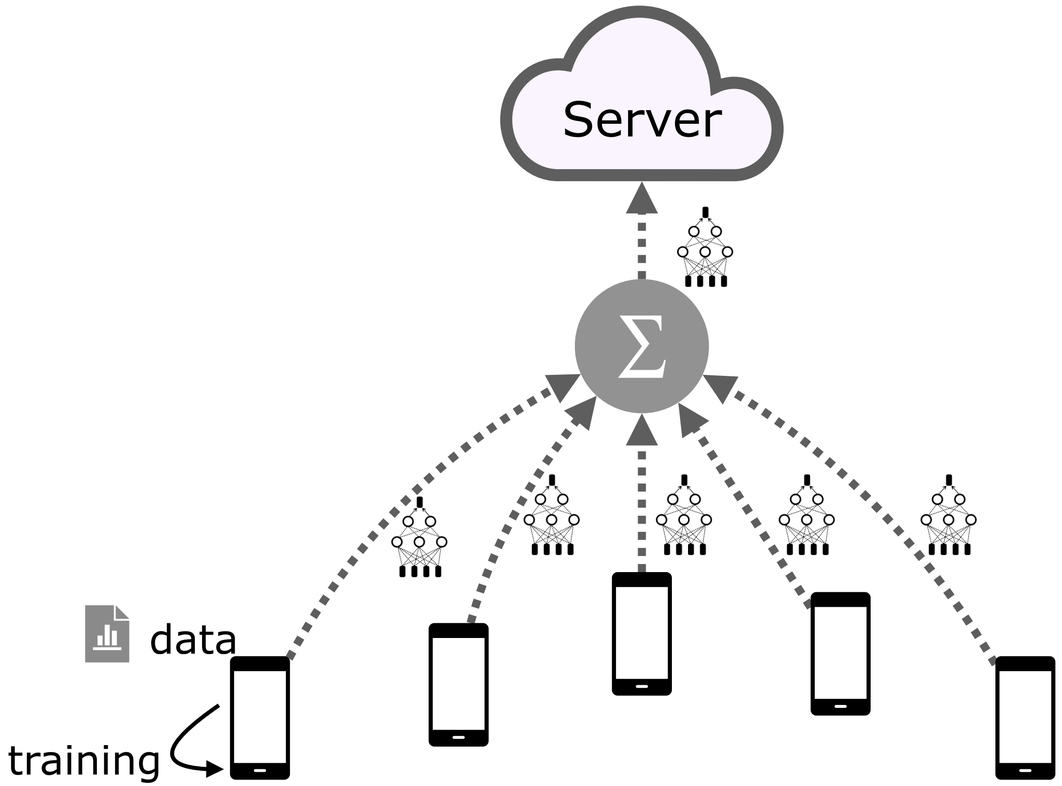
Federated Learning



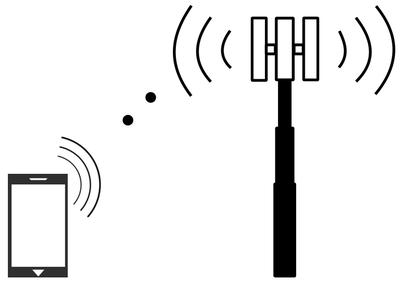


Data Credit: Business Wire

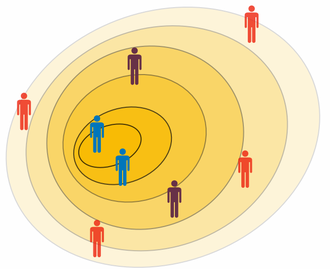
Federated Learning



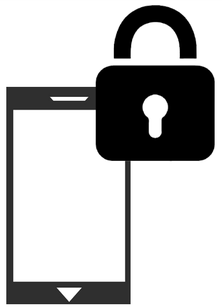
Challenges:



Communication efficiency

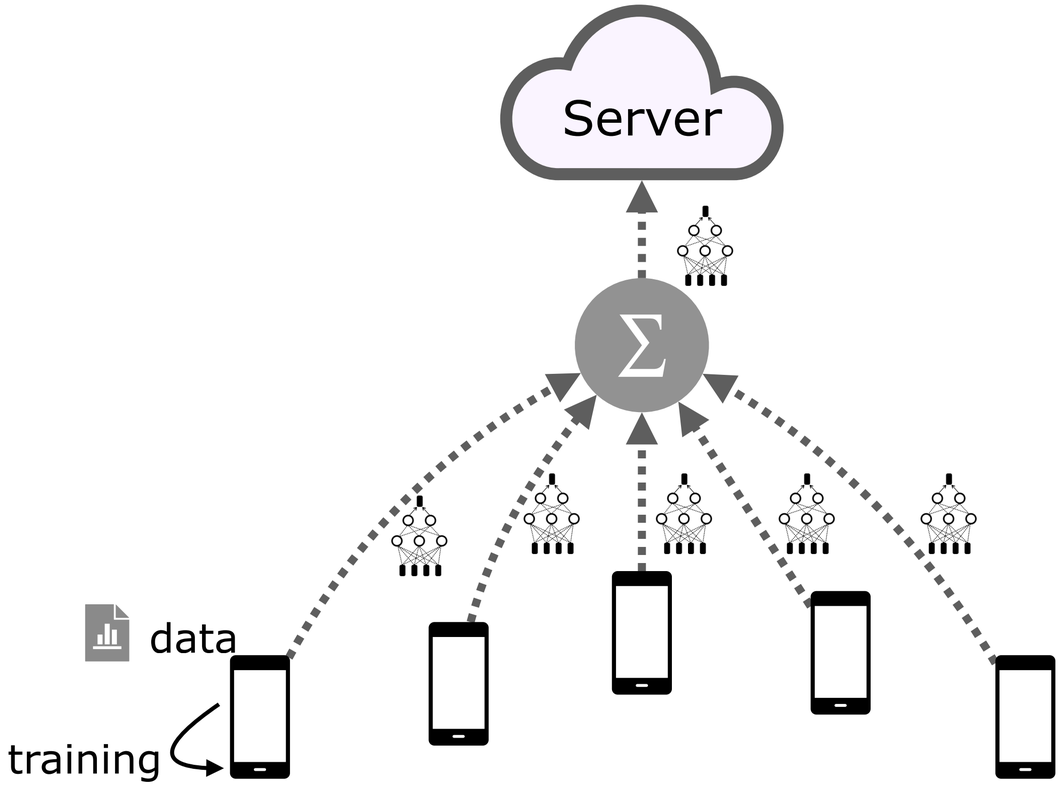


Statistical heterogeneity

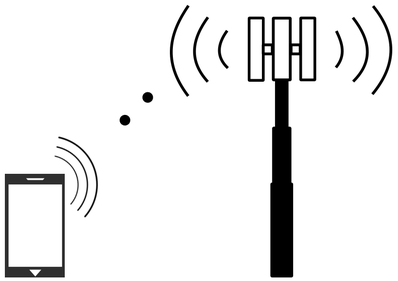


Privacy of user data

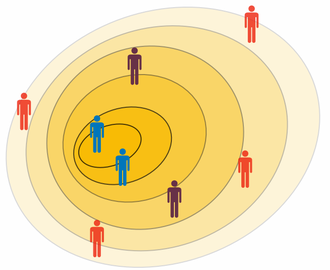
Federated Learning



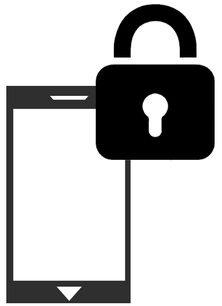
Challenges:



Communication efficiency



Statistical heterogeneity



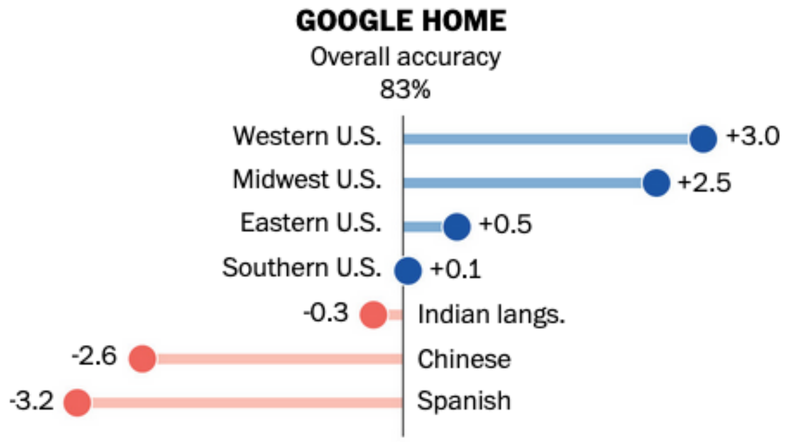
Privacy of user data

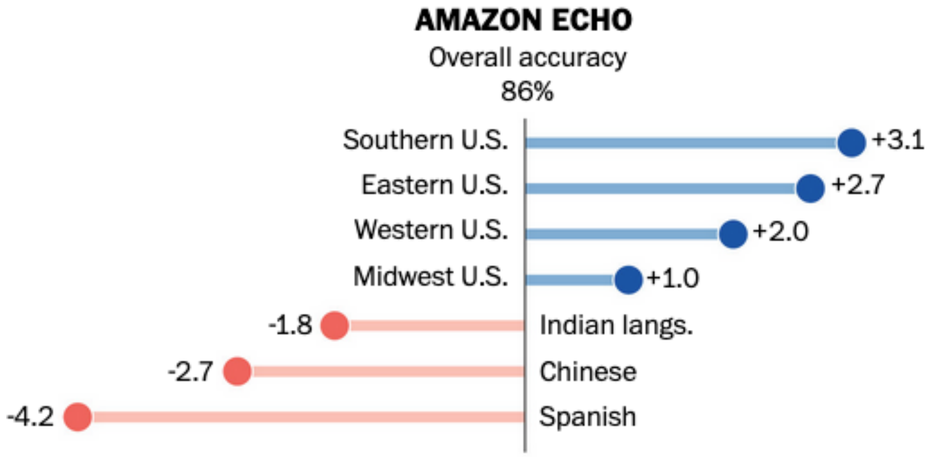
The Washington Post

Democracy Dies in Darkness

THE ACCENT GAP

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





By Drew Harwell July 19, 2018

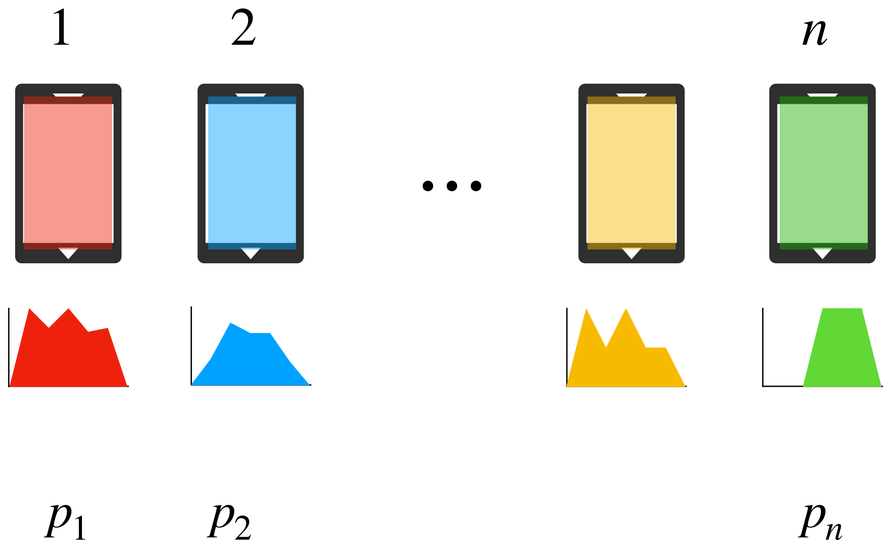
Tackling distribution shifts in federated learning

- **Improving tail performance with a single model**
- Improving overall performance with local adaptation

Problem Setup

Clients

Data Distribution



Data heterogeneity

Usual Learning Objective

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} F_i(w)$$

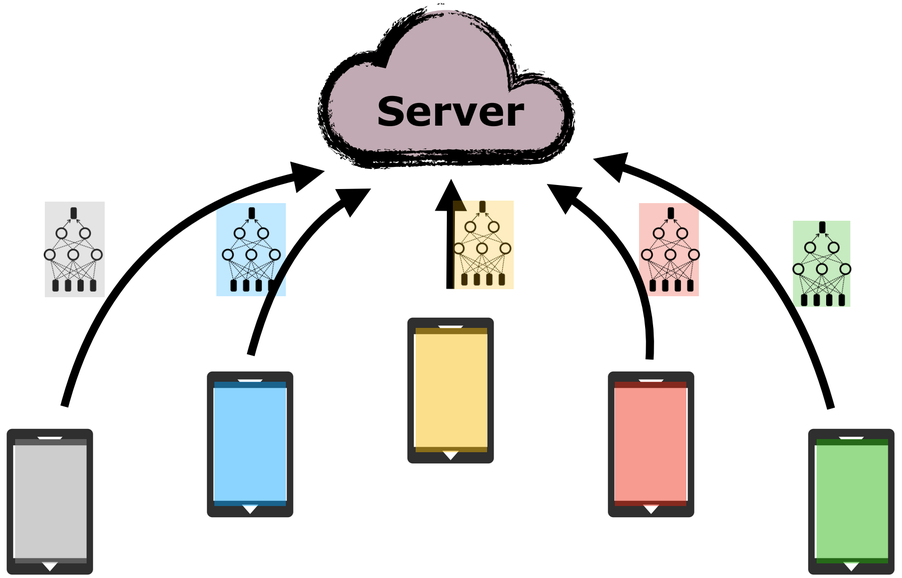
where $F_i(w) = \mathbb{E}_{z \sim p_i} \left[f(w; z) \right]$

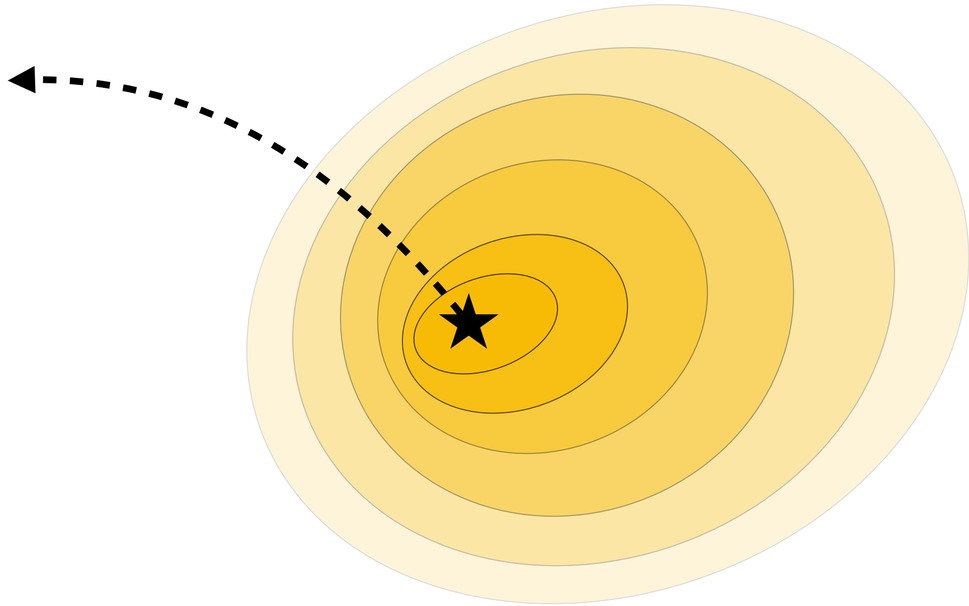
loss on client i

[McMahan et al. AISTATS (2017), Kairouz et al. (2021)]

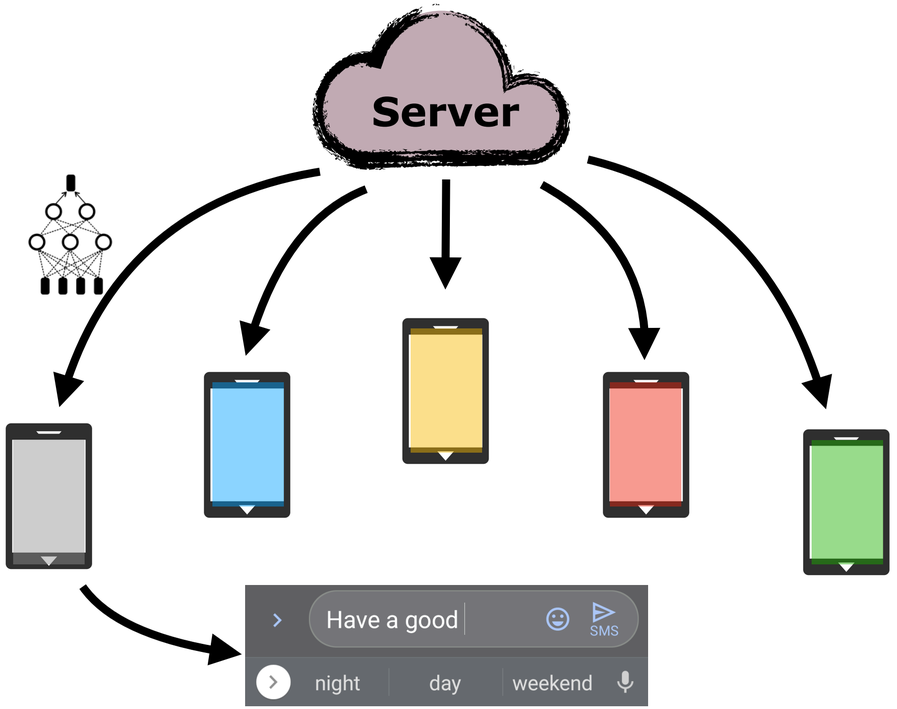
$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} F_i(w)$$

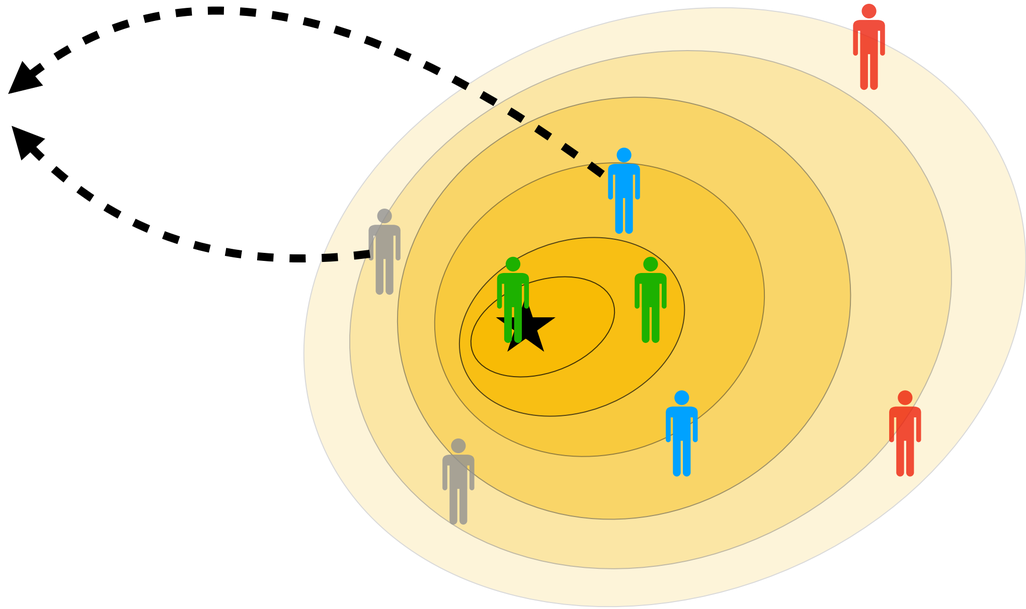
Global model is trained on *average distribution* across clients



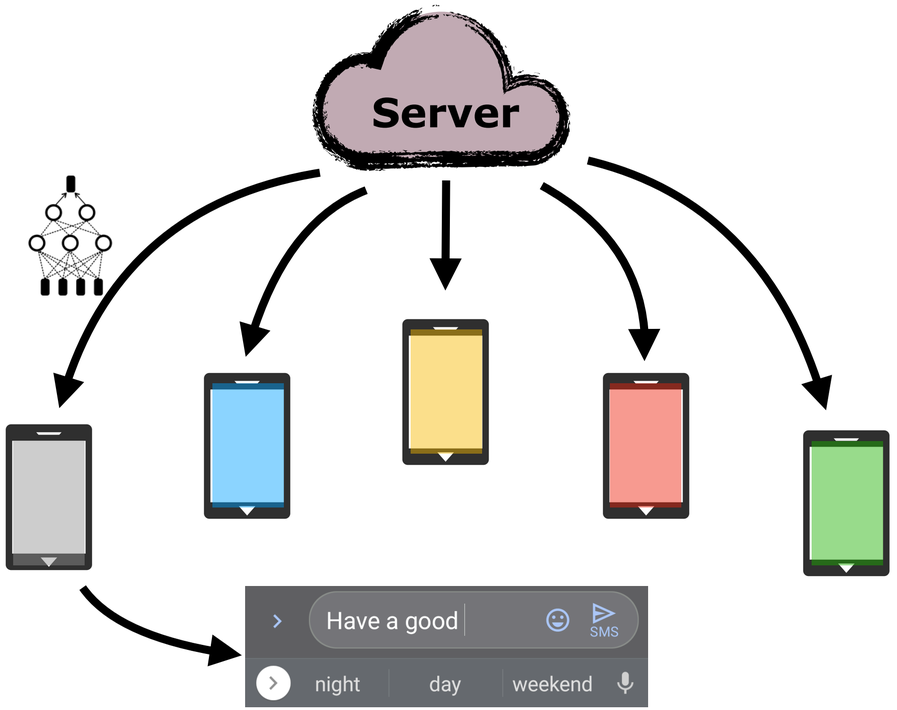


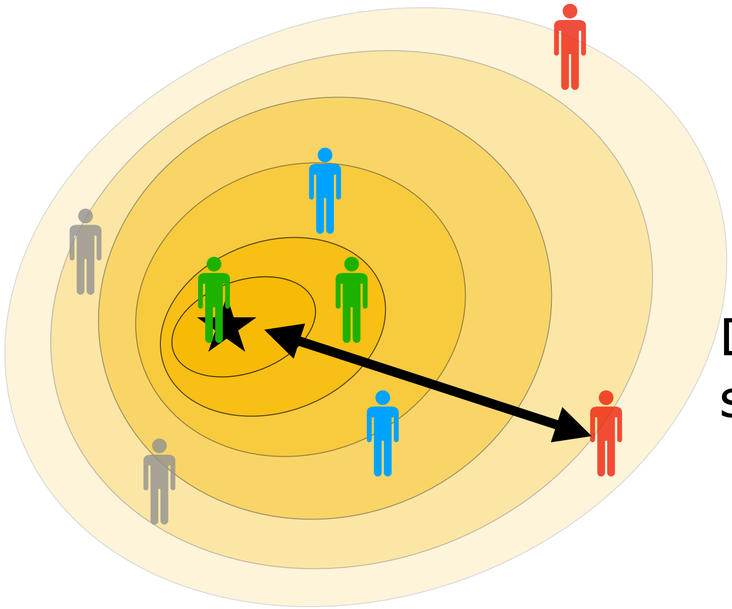
Global model is deployed on *individual* clients





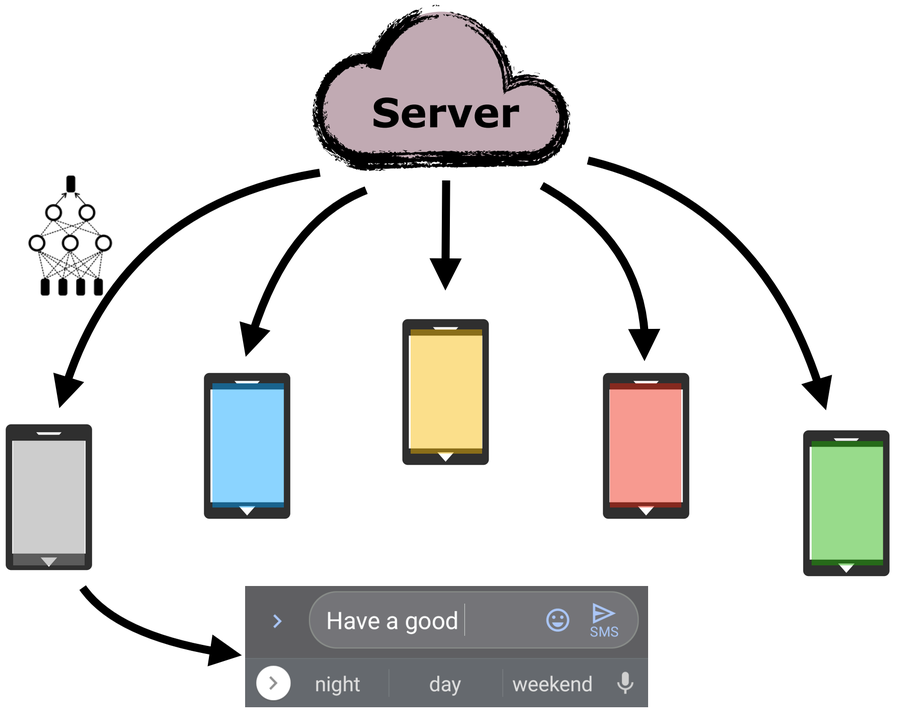
Global model is deployed on *individual* clients

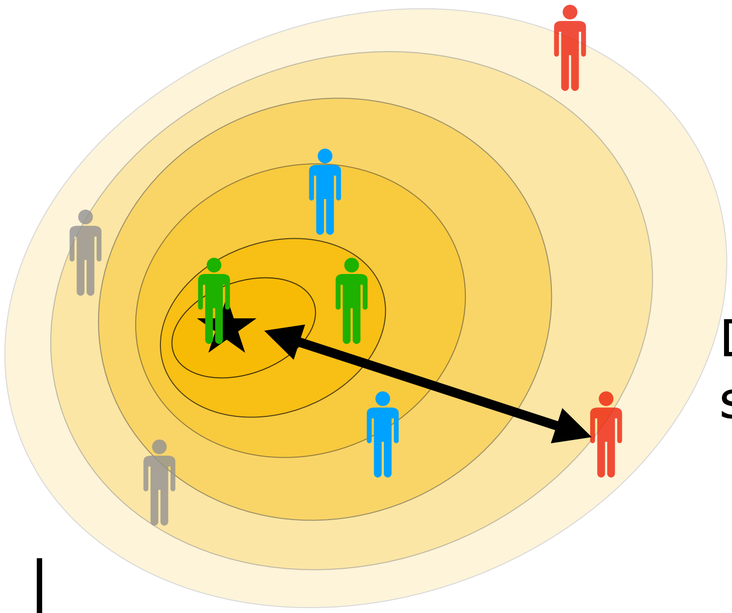




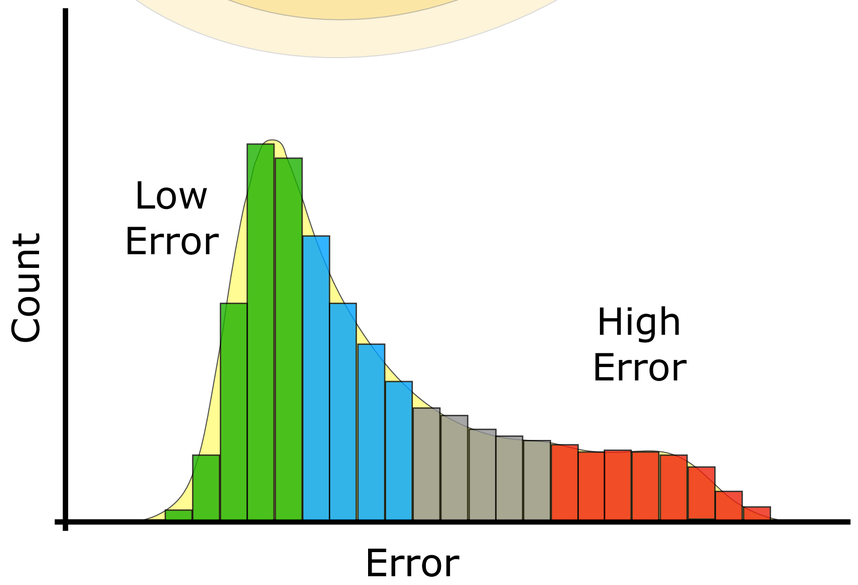
Distribution shift

Global model is deployed on *individual* clients

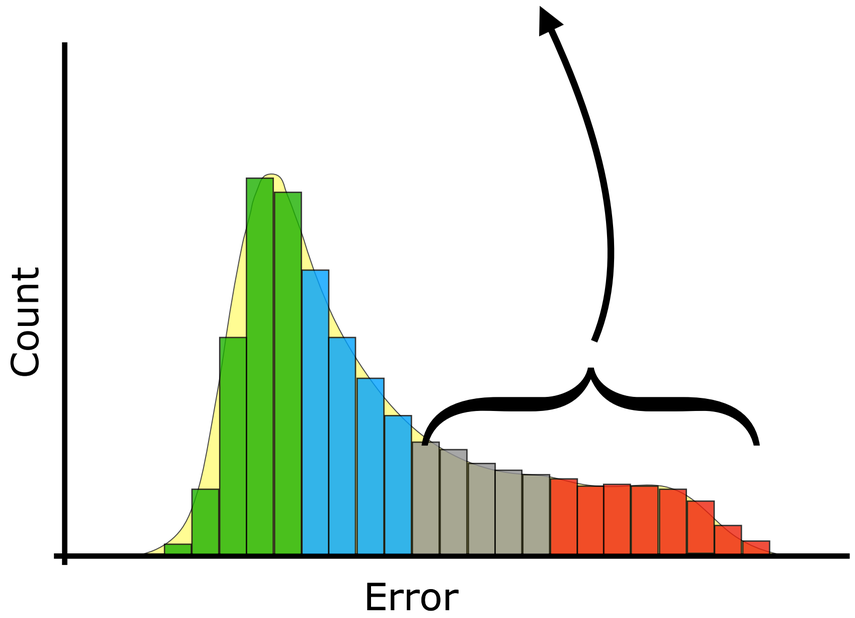




Distribution shift

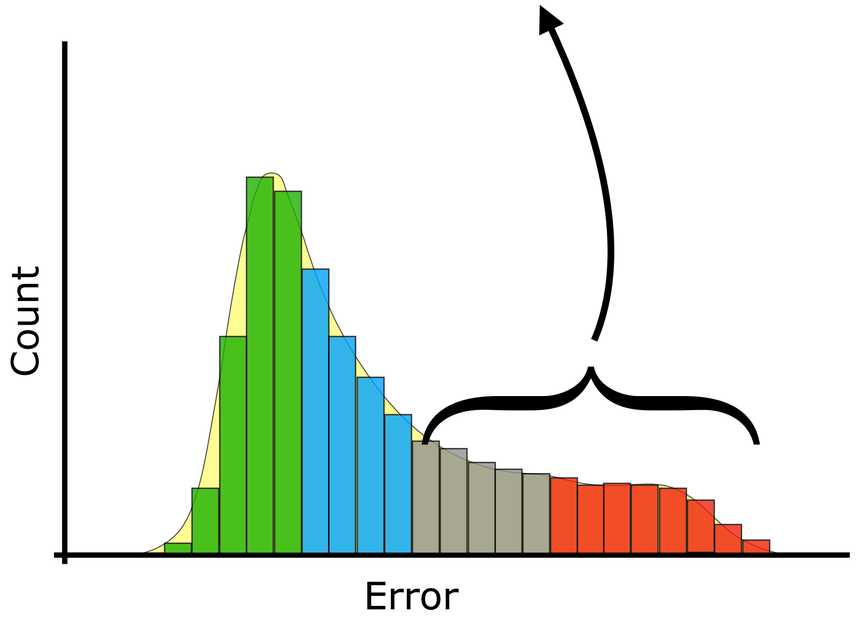


Our goal: improve performance on “tail clients”



Simplicial federated learning

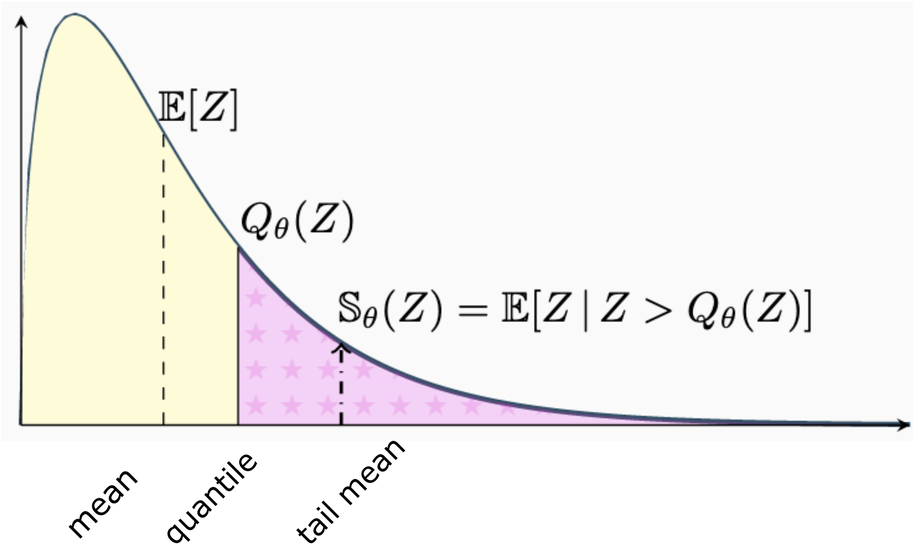
Our Approach: minimize the tail error directly!



Simplicial-FL Objective:

$$\min_{w} \mathbb{S}_{\theta}\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$

Superquantile | Conditional Value at Risk

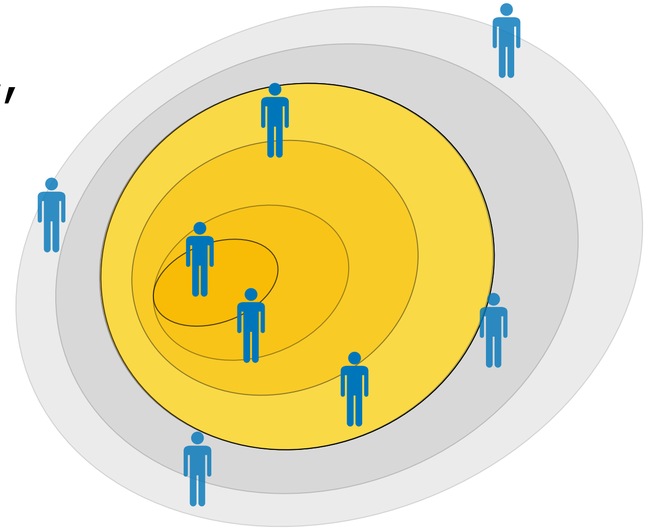


[Rockafellar & Uryasev (2000; 2002)]

Distributional robustness in federated learning:

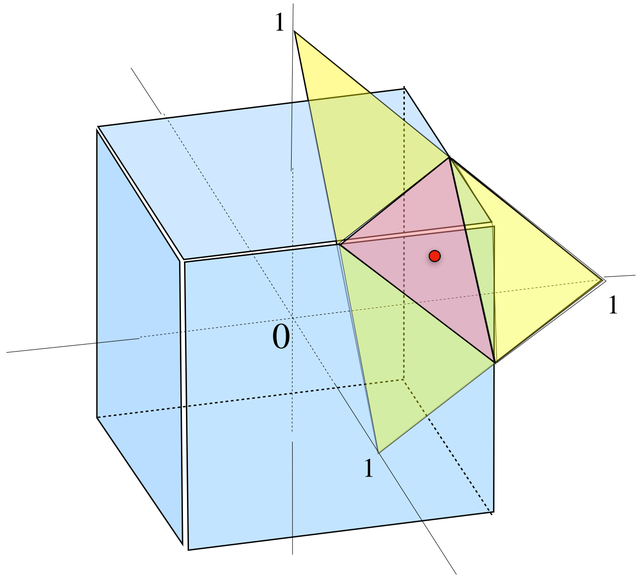
Assuming a new test client with mixture distribution $p_\pi = \sum_i \pi_i p_i$,
Simplicial-FL objective is equivalent to:

$$\min_{w} \max_{\pi\,:\,\pi_i \le (n\theta)^{-1}} \mathbb{E}_{z\sim p_\pi}\left[f(w;z)\right]$$



Dual expression ≡ continuous knapsack problem

$$\mathbb{S}_\theta(x_1,\cdots,x_n) = \max\left\{ \sum_i \pi_i x_i \;:\; \pi_i \ge 0,\ \sum_i \pi_i = 1,\ \pi_i \le (n\theta)^{-1} \right\}$$

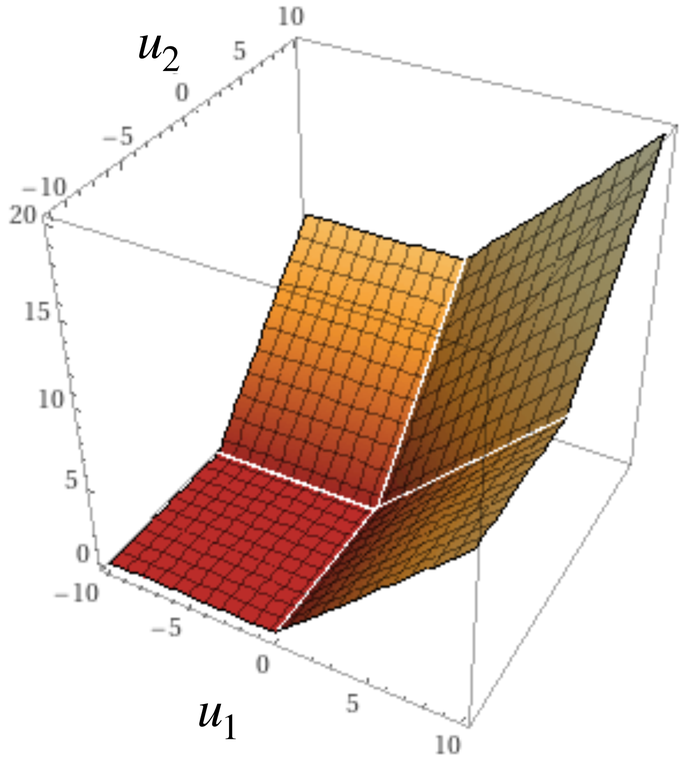


[Dantzig (1957), Ben-Tal & Teboulle (1987), Föllmer & Schied (2002)]

Optimizing Simplicial-FL

Challenge:

The superquantile is non-smooth



plot of $h(u_1, u_2) = \mathbb{S}_{1/2}\big(u_1, u_2, 0, 0\big)$

Nonsmooth: The subdifferential has a tractable form

$$\partial F_\theta(w) \ni \sum_{i=1}^{n} \pi_i^\star \nabla F_i(w) \qquad \text{where} \qquad \pi_i^\star \propto \mathbb{I}\Big(F_i(w) \geq Q_\theta\big(F_1(w), \cdots, F_n(w)\big)\Big)$$

assuming θn is an integer

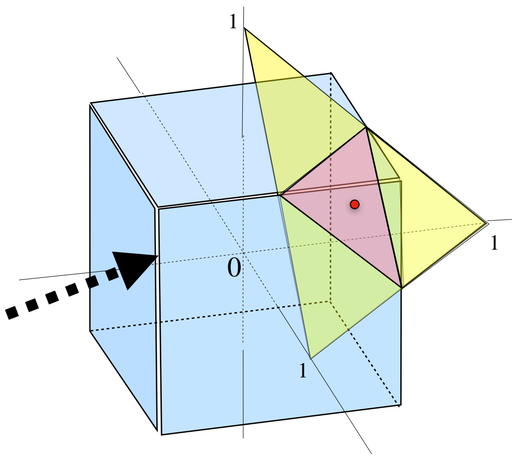
Nonsmooth: The subdifferential has a tractable form

$$\partial F_\theta(w) \ni \sum_{i=1}^{n} \pi_i^\star \nabla F_i(w) \qquad \text{where} \qquad \pi_i^\star \propto \mathbb{I}\Big(F_i(w) \geq Q_\theta\big(F_1(w), \cdots, F_n(w)\big)\Big)$$

assuming θn is an integer

Proof Chain rule $\implies$ subdifferential holds with

$$\pi^\star \in \arg\max_{\pi \in \mathscr{P}_\theta} \sum_i \pi_i F_i(w)$$



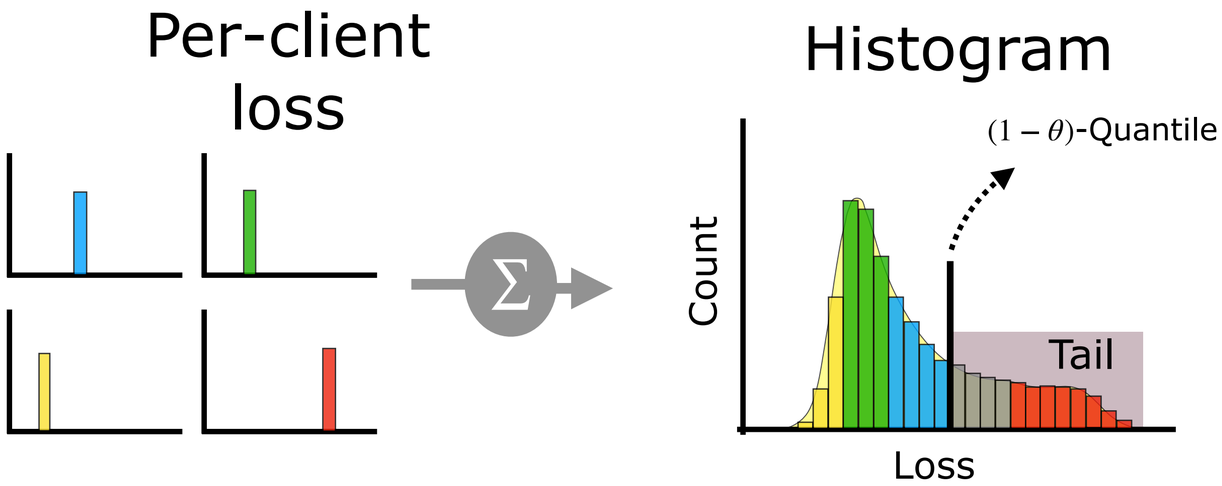
Alternate form of $\pi^\star$ comes from the continuous knapsack problem

[Dantzig. ORIJ (1957)]

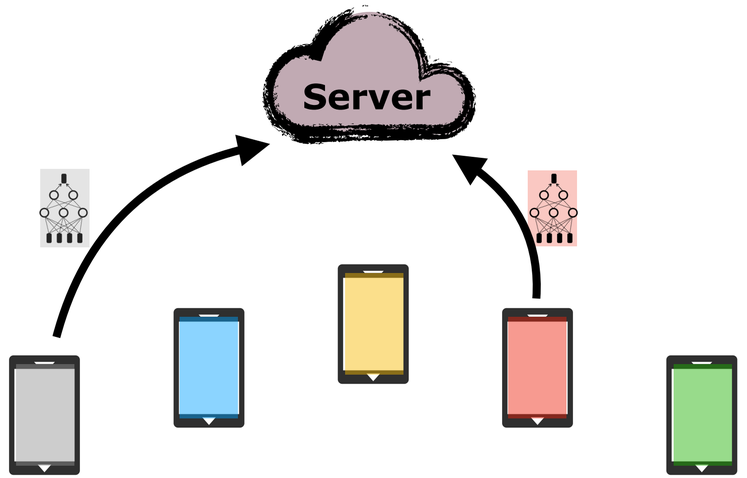
Algorithm

In each communication round:

- Estimate the quantile



- Aggregate over the tail



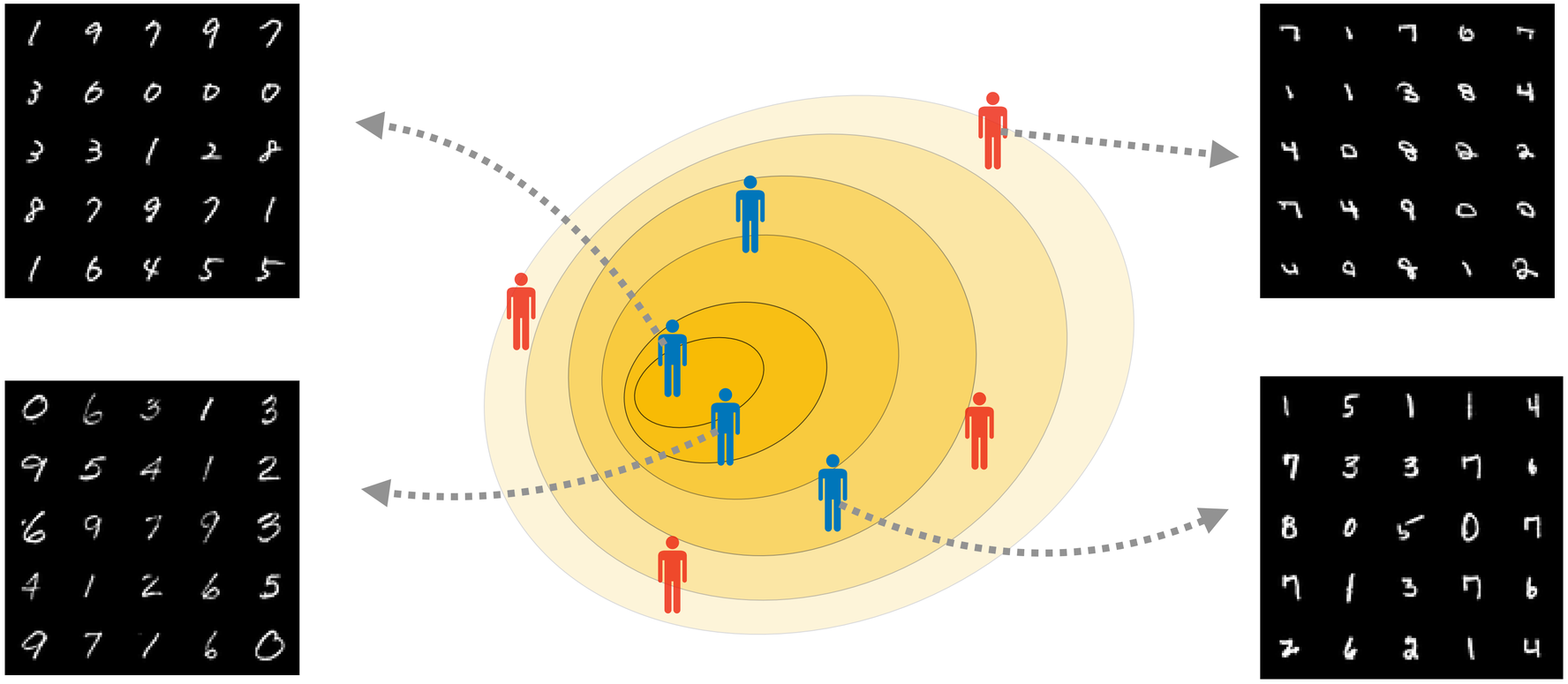
Convergence rates

Non-convex case: $O(1/\sqrt{t})$ + lower order terms

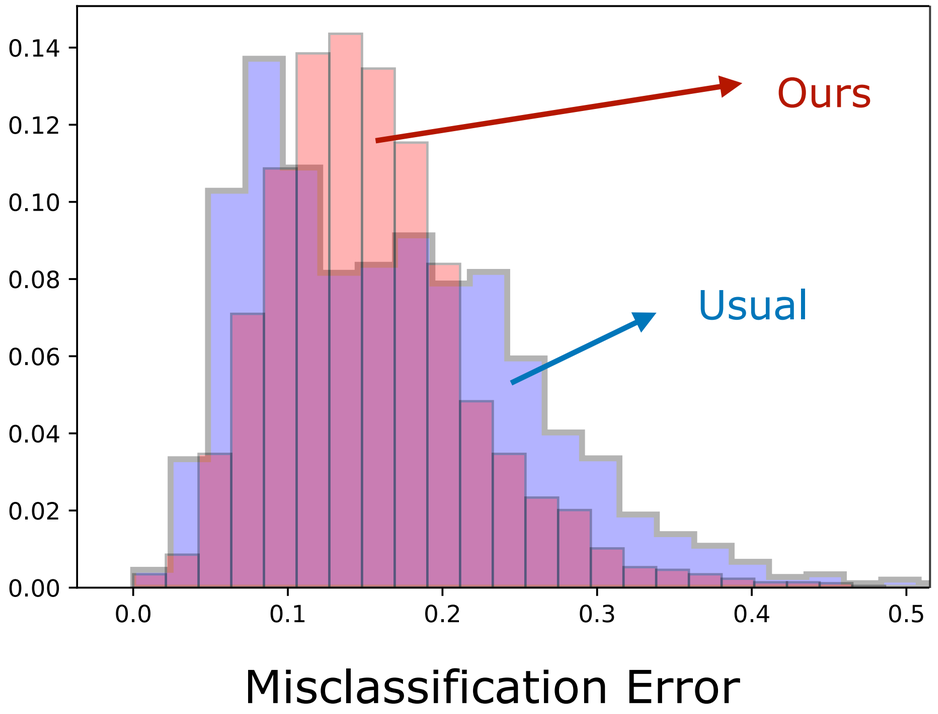
Strongly convex case: $\tilde{O}\left(\kappa^{3/2} + \frac{1}{\lambda\varepsilon}\right)$

κ: condition number
λ: strong convexity

Experiments: EMNIST



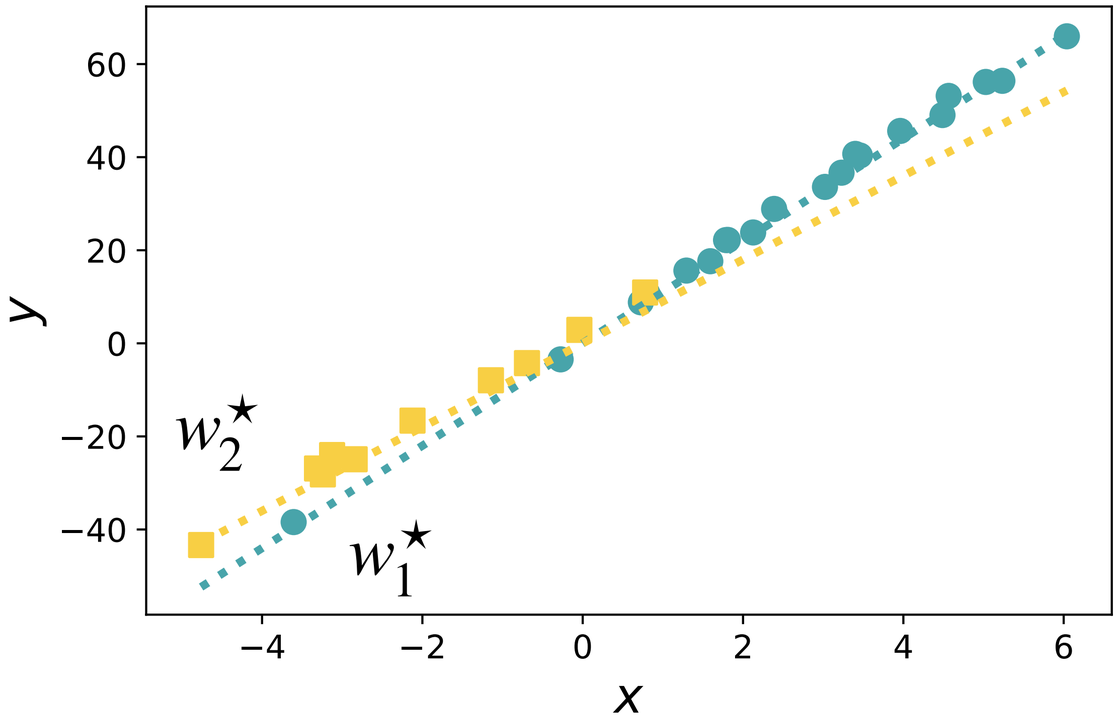
Histogram of per-client errors

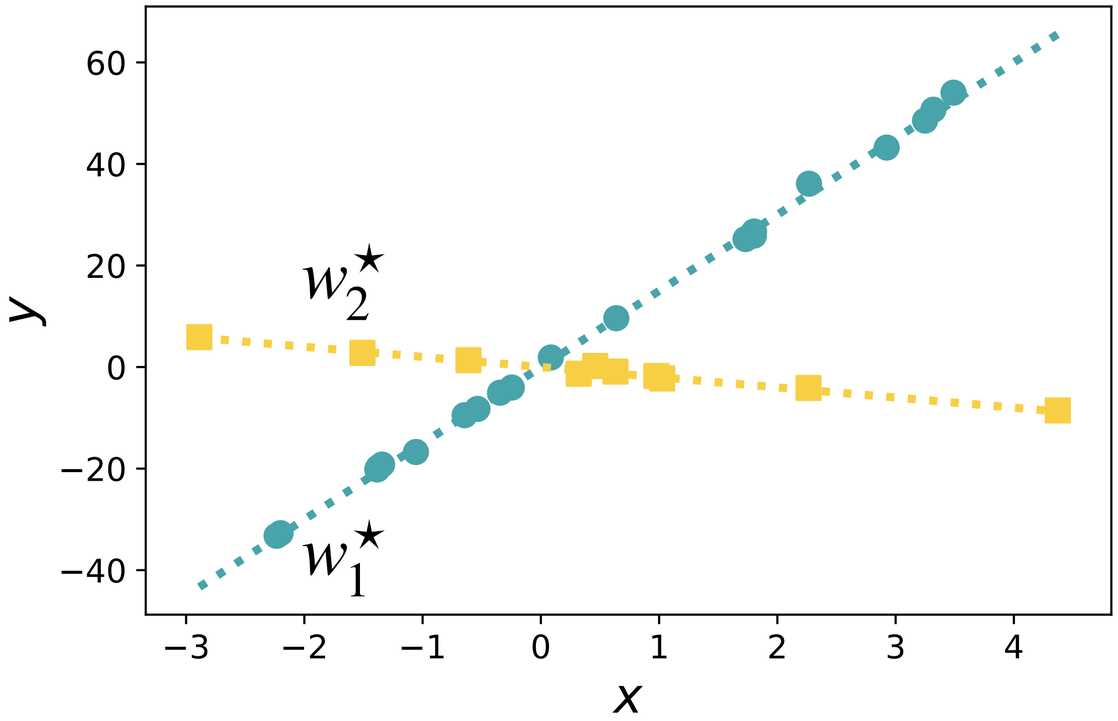


Tackling distribution shifts in federated learning

- Improving tail performance with a single model
- **Improving overall performance with local adaptation**

The need for local adaptation a.k.a. personalization





Objective $\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} F_i(w)$ where $F_i(w) = \mathbb{E}_{z \sim p_i} \left[f(w; z) \right]$

loss on client i

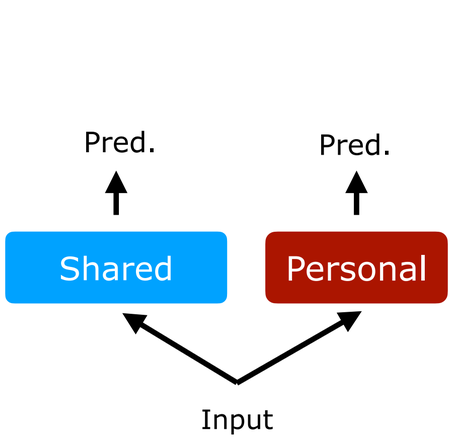
Personalization: Each model has a global component and a per-client component

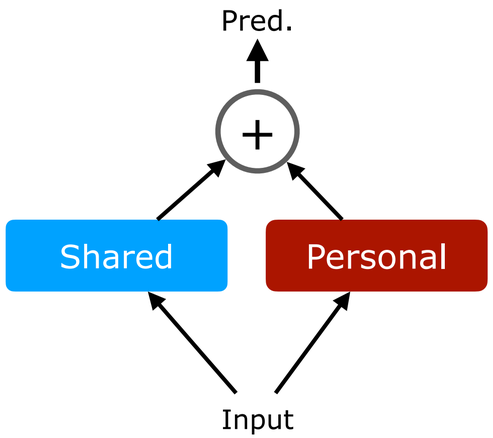
Shared Params u + Personal Params v_i = Full model $w_i = (u, v_i)$

Objective: $$\min_{u, v_1, \cdots, v_n} \frac{1}{n} \sum_{i=1}^{n} F_i(u, v_i)$$

Example: $F_i(u, v_i) = \mathbb{E}_{(X,Y)\sim p_i} \left(\phi_g(X;\, u) + \phi_l(X;\, v_i) - Y \right)^2$

Personalization architectures



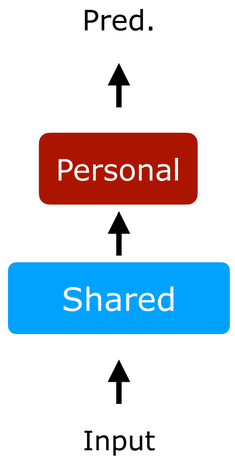


$$F_i(u, v_i) = \mathbb{E}_{(X,Y)\sim p_i}\left(\phi_g(X;\, u) + \phi_l(X;\, v_i) - Y\right)^2$$

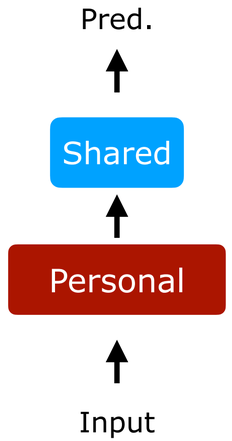
Multi-task learning: Caruana (1997), Baxter (2000), Evgeniou & Pontil (2004), Collobert & Weston (2005), Argyriou et al. (2008), …

Personalization architectures

$$F_i(u, v_i) = \mathbb{E}_{(X,Y)\sim p_i}\left(\phi_g(X; u) + \phi_i(X; v_i) - Y\right)^2$$



Arivazhagan et al. (2019)
Collins et al. (2021)



Liang et al. (2019)

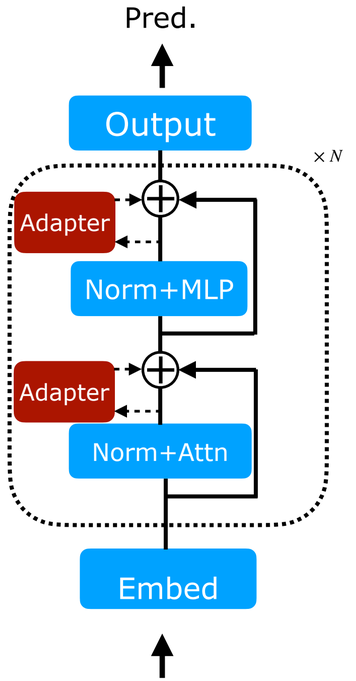
Multi-task learning: Caruana (1997), Baxter (2000), Evgeniou & Pontil (2004), Collobert & Weston (2005), Argyriou et al. (2008), …

Personalization architectures

$$F_i(u, v_i) = \mathbb{E}_{(X,Y)\sim p_i}\left(\phi_g(X; u) + \phi_i(X; v_i) - Y\right)^2$$

Arivazhagan et al. (2019)
Collins et al. (2021)

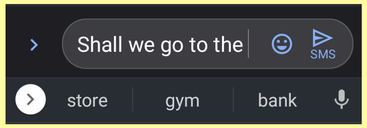
Liang et al. (2019)



Personalized Adapters

Multi-task learning: Caruana (1997), Baxter (2000), Evgeniou & Pontil (2004), Collobert & Weston (2005), Argyriou et al. (2008), …

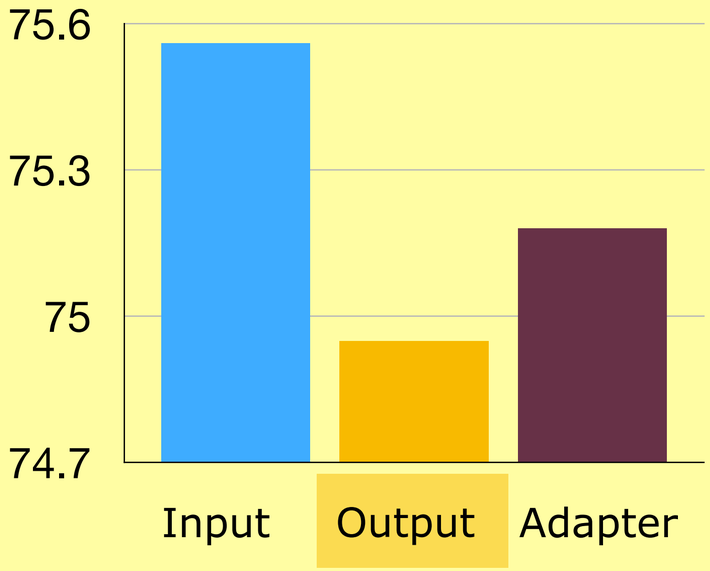
Best personalization architecture depends on task heterogeneity

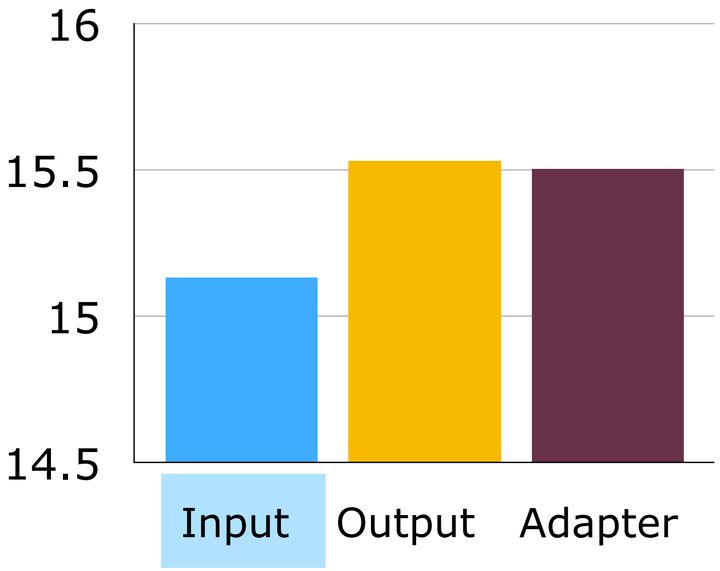


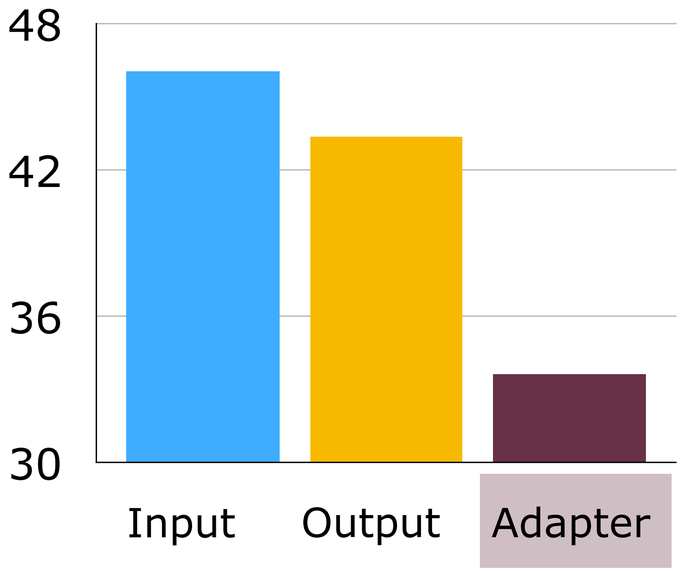
Next word prediction

Speech recognition

Landmark detection

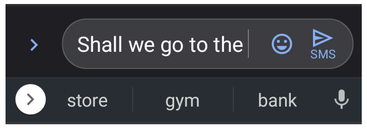






y-axis shows error: lower is better

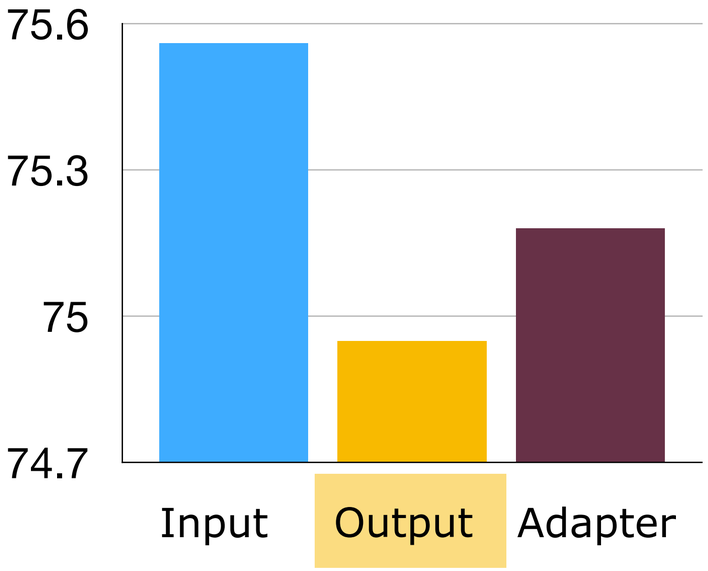
Best personalization architecture depends on task heterogeneity

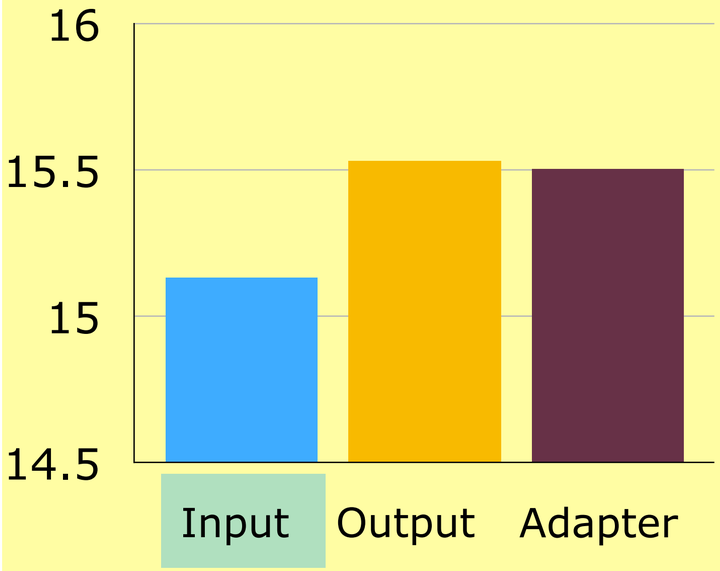


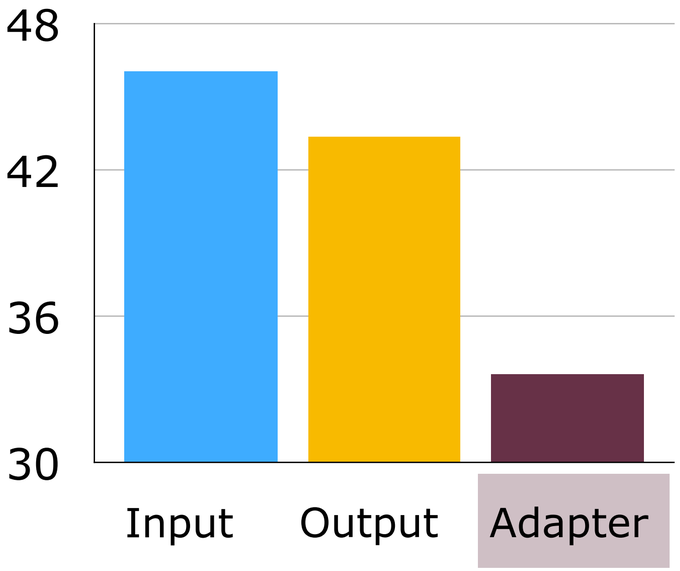
Next word prediction

Speech recognition

Landmark detection

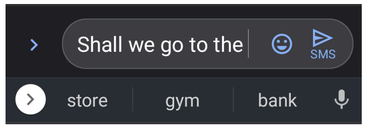






y-axis shows error: lower is better

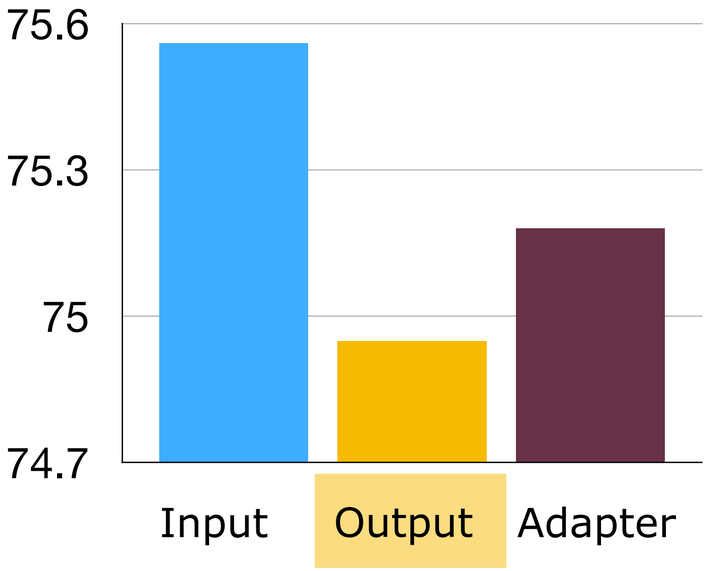
Best personalization architecture depends on task heterogeneity

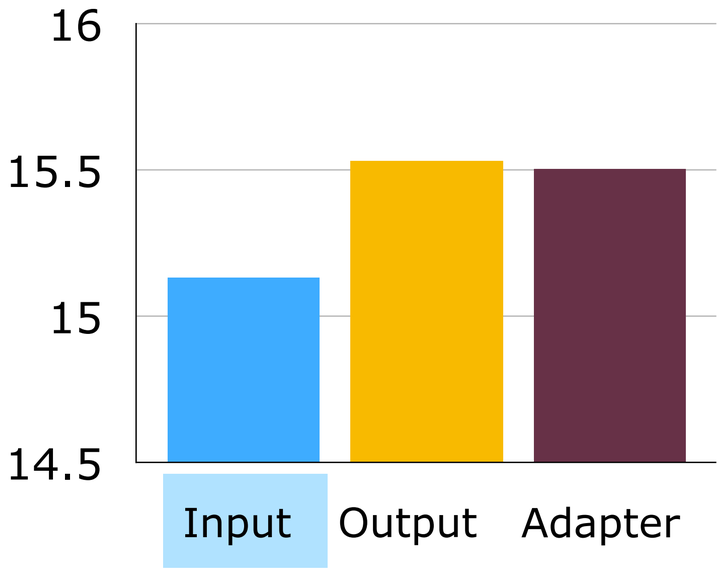


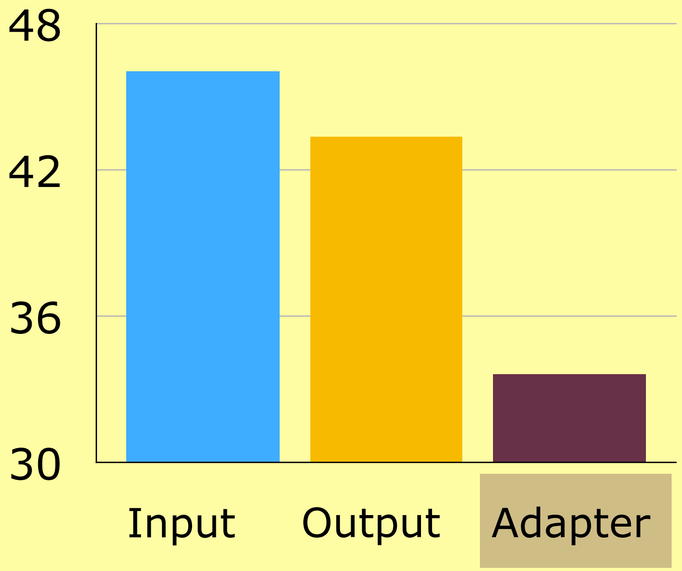
Next word prediction

Speech recognition

Landmark detection







y-axis shows error: lower is better

Open problems: Deeper understanding of shifts

Many negative results: optimization can slow down, makes robustness harder, …

Yet, federated learning is used widely in practice

Open problems: Deeper understanding of shifts

Many negative results: optimization can slow down, makes robustness harder, …

Yet, federated learning is used widely in practice

Quantify heterogeneity:

Measure gaps between distributions: **MAUVE**

[**P**., Swayamdipta, Zellers, Thickstun, Welleck, Choi, Harchaoui. NeurIPS (2021),
Liu, **P**., Welleck, Oh, Choi, Harchaoui. NeurIPS (2021)]

Open problems: Deeper understanding of shifts

Many negative results: optimization can slow down, makes robustness harder, …

Yet, federated learning is used widely in practice

Quantify heterogeneity:

Measure gaps between distributions: **MAUVE**

[**P**., Swayamdipta, Zellers, Thickstun, Welleck, Choi, Harchaoui. NeurIPS (2021),
Liu, **P**., Welleck, Oh, Choi, Harchaoui. NeurIPS (2021)]

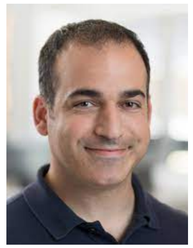
Best algorithms for different types of shifts (subject to federated constraints)

Statistical assumptions under which heterogeneity is benign?

What measures of heterogeneity impact optimization?





Federated Learning with Partial Model Personalization.
Krishna Pillutla, Kshitiz Malick, Abdulrehman Mohamed, Mike Rabbat, Maziar Sanjabi, Lin Xiao
ICML (2022).

Federated Learning with Heterogeneous Devices: A Superquantile Optimization Approach.
Krishna Pillutla*, Yassine Laguel*, Jérôme Malick, Zaid Harchaoui.
Under Review (arXiv 2112.09429)

A Superquantile Approach to Federated Learning with Heterogeneous Devices.
Yassine Laguel*, Krishna Pillutla*, Jérôme Malick, Zaid Harchaoui.
IEEE CISS (2021).

Superquantiles at Work : Machine Learning Applications and Efficient (Sub)gradient Computation.
Yassine Laguel, Krishna Pillutla, Jérôme Malick, Zaid Harchaoui.
Set-Valued and Variational Analysis (2021).

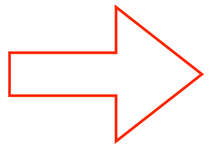
On the Complexity of a Practical Primal-Dual Coordinate Method

Ahmet Alacaoglu
University of Wisconsin-Madison
alacaoglu@wisc.edu

Joint work with Steve Wright and Volkan Cevher

Problem

$$\min_{x\in\mathbb{R}^d} \sum_{i=1}^{n} h_i(\langle a_i, x\rangle) + g(x)$$



$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \underbrace{g(x) + \sum_{i=1}^{n} y^{(i)}\langle a_i, x\rangle - h_i^*(y^{(i)})}_{g(x)+\langle Ax,y\rangle - h^*(y)}$$

convex, nonsmooth

"Generalized" Linear Programs: $\min_{x\in\mathcal{X}} \langle c, x\rangle + r(x) \text{ subject to } Ax = b$

Examples: DRO with Wasserstein (or f-divergence) ambiguity (Steve's talk)

Song, Chaobing, Cheuk Yin Lin, Stephen J. Wright, and Jelena Diakonikolas. "Coordinate linear variance reduction for generalized linear programming." *arXiv:2111.01842* (2021).

Complexity

Cost to obtain $\underbrace{z_{\text{out}}}_{(x,y)}$ such that $\text{OptMeasure}(z_{\text{out}}) \leq \varepsilon$

# of iterations $\times$ cost of each iteration

Recall $A \in \mathbb{R}^{n \times d}$
$x \in \mathbb{R}^d$
$y \in \mathbb{R}^n$

depends on
- ε
- $d, n, \text{nnz}(A)$ (num. of nonzeros)
- $\|A\|, \max_{i \in \{1,\ldots,n\}} \|A_i\|$

Complexity

Cost to obtain $\underbrace{z_{\text{out}}}_{(x,y)}$ such that $\text{OptMeasure}(z_{\text{out}}) \le \varepsilon$

# of iterations $\times$ cost of each iteration

Recall $A \in \mathbb{R}^{n \times d}$
$x \in \mathbb{R}^d$
$y \in \mathbb{R}^n$

depends on ε, $d, n, \text{nnz}(A)$, $\|A\|, \max_{i \in \{1,\dots,n\}} \|A_i\|$

num. of nonzeros

Algorithm: Gradient descent-ascent

Extrapolation

Random coordinate updates

PDHG: "Full gradient"

Randomized PDHG

Example: Standard LP $O\left(\frac{nd\|A\|}{\varepsilon}\right)$ → $O\left(\frac{d\sum_{i=1}^n \|A_i\|}{\varepsilon}\right)$

Context - *Complexity* table

$$\min_{x\in\mathbb{R}^d} \sum_{i=1}^{n} h_i(\langle a_i, x\rangle) + g(x)$$

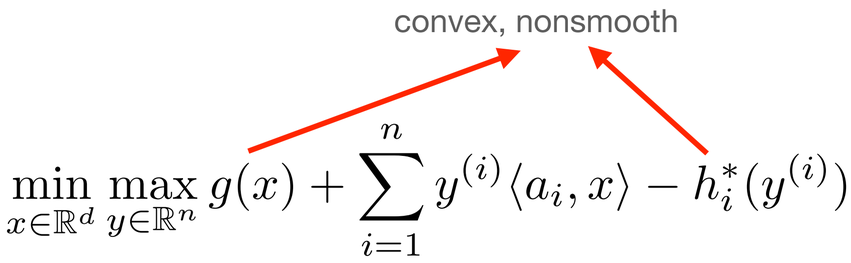
convex, nonsmooth

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} g(x) + \sum_{i=1}^{n} y^{(i)}\langle a_i, x\rangle - h_i^*(y^{(i)})$$

	convex-concave			Strongly convex-concave	Convex-strongly concave	Strongly convex-strongly concave
	h_i Lipschitz ERM with nonsmooth loss	$h_i(z) = \begin{cases} 0, & \text{if } z \in C \\ +\infty, & \text{if } z \notin C \end{cases}$ Linear constraints	h_i general nonsmooth Matrix games	SVM	Lasso	Ridge regression
Dense a_i						
Sparse a_i						

Context - *Complexity* table

Prior work: 8+ different works/algorithms: primal-dual, coordinate descent, variance reduction, acceleration, dual averaging, extragradient…

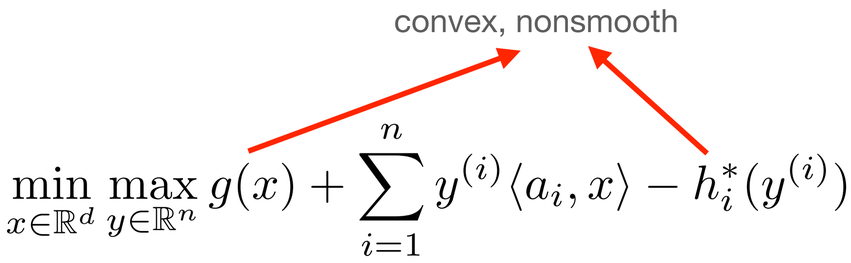


	convex-concave			Strongly convex-concave	Convex-strongly concave	Strongly convex-strongly concave
	h_i Lipschitz ERM with nonsmooth loss	$h_i(z) = \begin{cases} 0, & \text{if } z \in C \\ +\infty, & \text{if } z \notin C \end{cases}$ Linear constraints	h_i general nonsmooth Matrix games	SVM	Lasso	Ridge regression
Dense a_i	Allen-Zhu, JMLR, 2017	Chambolle, Pock, JMIV 2011 A., Malitsky, COLT, 2022	Song et al., ICML, 2021 Chambolle, Pock, JMIV 2011	Chambolle et al., SIOPT: 2018 Song et al., arXiv: 2021	Chambolle et al., SIOPT: 2018	Allen-Zhu, JMLR, 2017 Tan et al., OMS, 2020
Sparse a_i	Chambolle, Pock, JMIV 2011	Chambolle, Pock, JMIV 2011	Chambolle, Pock, JMIV 2011 A., Malitsky, COLT, 2022 Song et al., arXiv: 2021 A. et al., ICML, 2020	Song et al., arXiv: 2021	Chambolle, Pock, JMIV 2011	Tan et al., OMS, 2020

Context - *Complexity* table

Match or improve the complexity with a single (existing) algorithm from A. et al., ICML, 2020

Alacaoglu, Cevher, Wright, arXiv:2201.07684

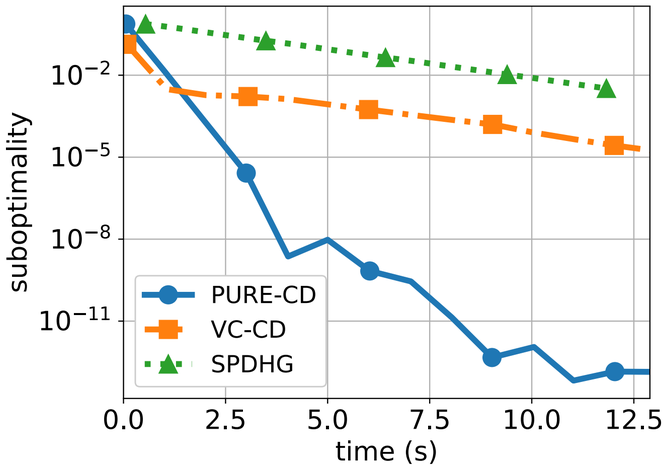


	convex-concave			Strongly convex-concave	Convex-strongly concave	Strongly convex-strongly concave
Color fill: improvement No color fill: match	h_i Lipschitz ERM with nonsmooth loss	$h_i(z) = \begin{cases} 0, & \text{if } z \in C \\ +\infty, & \text{if } z \notin C \end{cases}$ Linear constraints	h_i general nonsmooth Matrix games	SVM	Lasso	Ridge regression
Dense a_i	Allen-Zhu, JMLR, 2017	Chambolle, Pock, JMIV 2011 A., Malitsky, COLT, 2022	Song et al., ICML, 2021 Chambolle, Pock, JMIV 2011	Chambolle et al., SIOPT: 2018 Song et al., arXiv: 2021	Chambolle et al., SIOPT: 2018	Allen-Zhu, JMLR, 2017 Tan et al., OMS, 2020
Sparse a_i	Chambolle, Pock, JMIV 2011	Chambolle, Pock, JMIV 2011	Chambolle, Pock, JMIV 2011 A., Malitsky, COLT, 2022 Song et al., arXiv: 2021 A. et al., ICML, 2020	Song et al., arXiv: 2021	Chambolle, Pock, JMIV 2011	Tan et al., OMS, 2020

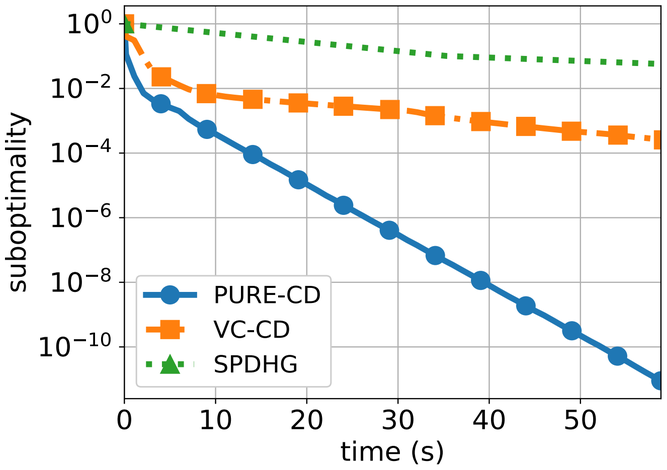
Practicality detour-sparse

Lasso with varying levels of sparsity: $\min_{x \in \mathbb{R}^d} \frac{1}{2}\|Ax - b\|^2 + \lambda \|x\|_1$

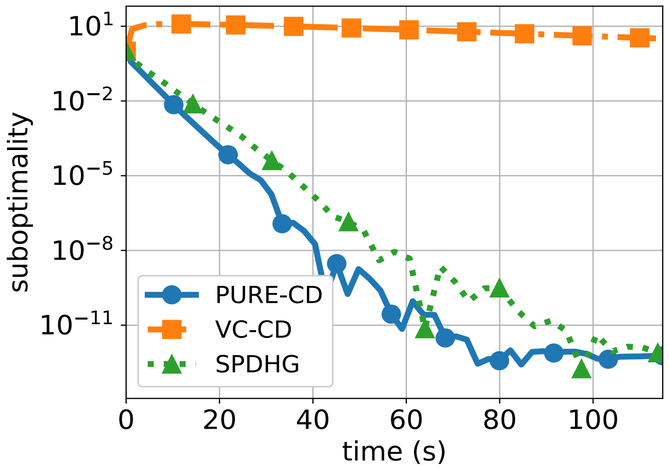
SPDHG: good for dense
VC-CD: good for sparse (Fercoq, Bianchi, 2019)



sparse



moderately sparse



dense

***A.**, Fercoq, Cevher, "Random extrapolation for primal-dual coordinate descent", ICML 2020*

Sparse friendly algorithm

PURE-CD - dense [A., et al, ICML, 2020]

$$\bar{x}_{k+1} = \mathrm{prox}_{\tau g}(x_k - \tau A^\top y_k)$$

$$\text{Random } i_k \in \{1, \ldots, n\}$$

$$y_{k+1}^{(i_k)} = \mathrm{prox}_{\sigma^{(i_k)} h_{i_k}^*}(y_k^{(i_k)} + \sigma^{(i_k)} A_{i_k} \bar{x}_{k+1})$$

$$y_k^{(i)} = y_k^{(i)}, \quad \forall i \neq i_k$$

$$x_{k+1} = \bar{x}_{k+1} - \tau n A^\top (y_{k+1} - y_k)$$

PURE-CD - sparse [A., et al, ICML, 2020]

$$\text{Random } i_k \in \{1, \ldots, n\}$$

$$\bar{x}_{k+1}^{(j)} = \mathrm{prox}_{\tau^{(j)} g_j}(x_k^{(j)} - \tau^{(j)} (A^\top y_k)^{(j)}) \quad \forall j \in J(i_k)$$

$$y_{k+1}^{(i_k)} = \mathrm{prox}_{\sigma^{(i_k)} h_{i_k}^*}(y_k^{(i_k)} - \sigma^{(i_k)} A_{i_k} \bar{x}_{k+1})$$

$$y_{k+1}^{(i)} = y_k^{(i)} \quad \forall i \neq i_k$$

$$x_{k+1}^{(j)} = \bar{x}_{k+1}^{(j)} - \tau^{(j)} \theta^{(j)} A_{i_k, j} (y_{k+1}^{(j)} - y_k^{(j)}) \quad \forall j \in J(i_k)$$

$$x_{k+1}^{(j)} = x_k^{(j)} \quad \forall j \notin J(i_k)$$

Sparse friendly algorithm

$$J(i_k) = \{j \in \{1, \ldots, d\} : A_{i_k,j} \neq 0\}$$

indices of nonzero elements on A_{i_k}

PURE-CD - dense [A., et al, ICML, 2020]

$$\bar{x}_{k+1} = \text{prox}_{\tau g}(x_k - \tau A^\top y_k)$$

$$\text{Random } i_k \in \{1, \ldots, n\}$$

$$y_{k+1}^{(i_k)} = \text{prox}_{\sigma^{(i_k)} h_{i_k}^*}(y_k^{(i_k)} + \sigma^{(i_k)} A_{i_k} \bar{x}_{k+1})$$

$$y_k^{(i)} = y_k^{(i)}, \quad \forall i \neq i_k$$

$$x_{k+1} = \bar{x}_{k+1} - \tau n A^\top (y_{k+1} - y_k)$$

PURE-CD - sparse [A., et al, ICML, 2020]

$$\text{Random } i_k \in \{1, \ldots, n\}$$

$$\bar{x}_{k+1}^{(j)} = \text{prox}_{\tau^{(j)} g_j}(x_k^{(j)} - \tau^{(j)} (A^\top y_k)^{(j)}) \quad \forall j \in J(i_k)$$

$$y_{k+1}^{(i_k)} = \text{prox}_{\sigma^{(i_k)} h_{i_k}^*}(y_k^{(i_k)} - \sigma^{(i_k)} A_{i_k} \bar{x}_{k+1})$$

$$y_{k+1}^{(i)} = y_k^{(i)} \quad \forall i \neq i_k$$

$$x_{k+1}^{(j)} = \bar{x}_{k+1}^{(j)} - \tau^{(j)} \theta^{(j)} A_{i_k,j} (y_{k+1}^{(j)} - y_k^{(j)}) \quad \forall j \in J(i_k)$$

$$x_{k+1}^{(j)} = x_k^{(j)} \quad \forall j \notin J(i_k)$$

Flat minima generalize for low-rank matrix recovery

Lijun Ding

Joint with Dmitriy Drusvyatskiy and Maryam Fazel

IFDS at UW

Overparametrization : Data $\{(x_i, y_i)\}_{i=1}^{n}$

Overparametrization : Data $\{(x_i, y_i)\}_{i=1}^{n}$

$h_\theta(x_i)$

Overparametrization : Data $\{(x_i, y_i)\}_{i=1}^{n}$

$\ell(y_i, h_\theta(x_i))$

Overparametrization: Data $\{(x_i, y_i)\}_{i=1}^n$

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h_\theta(x_i))$$

Overparametrization: Data $\{(x_i, y_i)\}_{i=1}^n$

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h_\theta(x_i))$$

where

$$\underbrace{\#\ \text{parameters}}_{d} \gg \underbrace{\#\ \text{samples}}_{n}$$

Overparametrization: Data $\{(x_i, y_i)\}_{i=1}^n$

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h_\theta(x_i))$$

where $\underbrace{\#\ \text{parameters}}_{d} \gg \underbrace{\#\ \text{samples}}_{n}$

Overparametrization $\Rightarrow$ many zero-loss solutions

Overparametrization: Data $\{(x_i, y_i)\}_{i=1}^n$

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h_\theta(x_i))$$

where $\underbrace{\#\ \text{parameters}}_{d} \gg \underbrace{\#\ \text{samples}}_{n}$

Overparametrization $\Rightarrow$ many zero-loss solutions

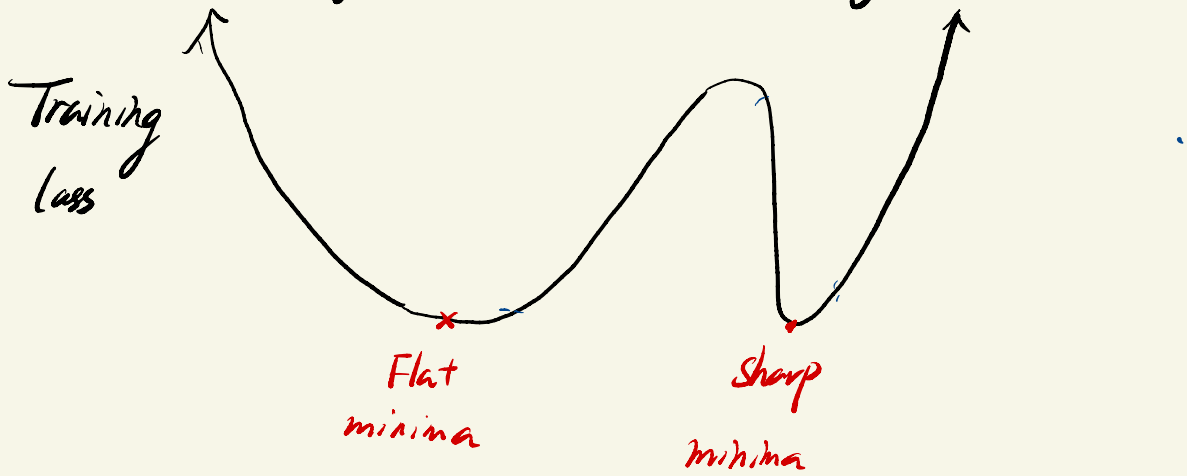
Q: Why do some zero-loss solution generalize & others do not?

Flat Landscape (Hochreiter - Schmidhuber '97)

of loss correlates with generalization

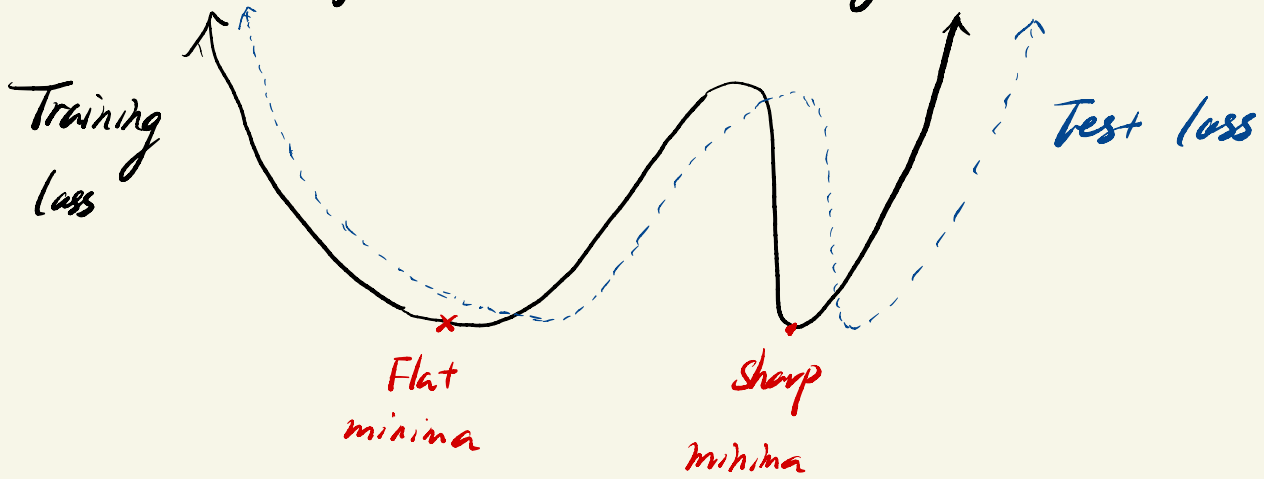
Flat Landscape (Hochreiter - Schmidhuber '97)

of loss correlates with generalization



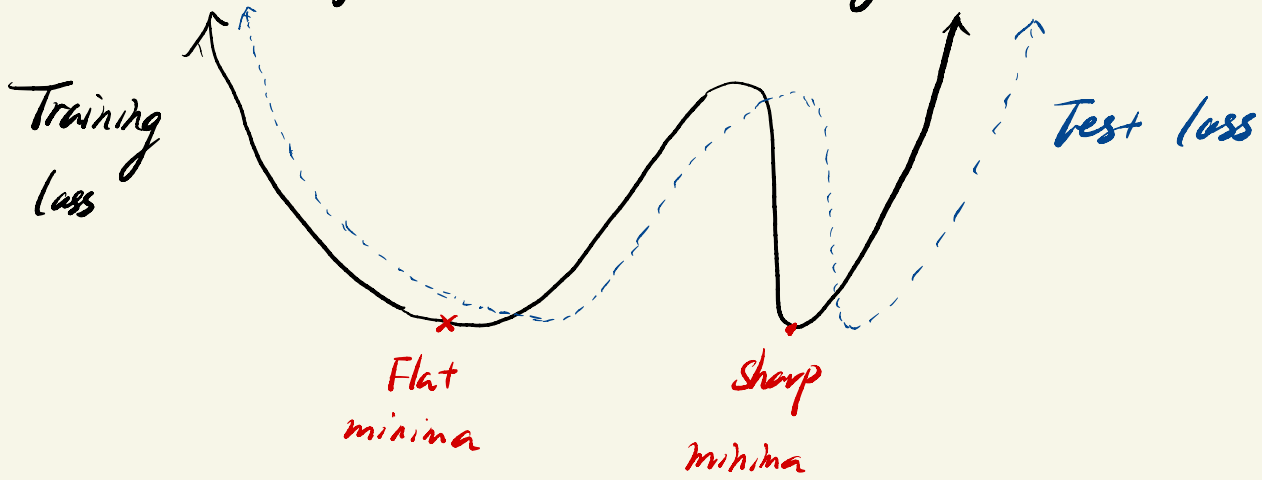
Flat Landscape (Hochreiter - Schmidhuber '97)

of loss correlates with generalization



Flat Landscape (Hochreiter-Schmidhuber '97)

of loss correlates with generalization



Lots of Recent Interests, e.g. Foret-Kleiner-Mobahi-Neyshabur '21.

Q: Do flat minimizers generalize

for a broad family of overparametrized problems?

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

For many overparametrized low-rank recovery problems.

- matrix sensing & bilinear sensing
- one hidden-layer NN with quadratic activation
- matrix completion
- robust PCA

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

For many overparametrized low-rank recovery problems.

- matrix sensing & bilinear sensing
- one hidden-layer NN with quadratic activation
- matrix completion
- robust PCA

Flat minima exactly Recover the ground truth!

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

overparametrized low-rank recovery problems.

Flat solution $(\underline{L}, \underline{R})$ of $\min_{L,R} \|\mathcal{A}(LR^\top - M_\#)\|_2^2$

($\underline{L}, \underline{R} \in \mathbb{R}^{d\times k}$; $\mathcal{A}$: a linear map; $M_\# \in \mathbb{R}^{d\times d}$)

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

overparametrized low-rank recovery problems.

Flat solution (L, R) of $\min_{L,R} \|\mathcal{A}(LR^\top - M_\#)\|_2^2$

($\mathcal{A}$: a linear map; $L, R \in \mathbb{R}^{d\times k}$, $M_\# \in \mathbb{R}^{d\times d}$)

$k \geq \operatorname{rank}(M_\#)$

Q: Do flat minimizers generalize for a broad family of overparametrized problems?

Our contribution: Yes!

overparametrized low-rank recovery problems.

Flat solution (L, R) of $\min_{L,R} \|\mathcal{A}(LR^\top - M_\#)\|_2^2$

($\mathcal{A}$: a linear map; $L, R \in \mathbb{R}^{d\times k}$; $M_\# \in \mathbb{R}^{d\times d}$)

$k \geq \operatorname{rank}(M_\#)$

satisfies $LR^\top = M_\#$.

Stochastic Optimization under Distributional Drift

Zaid Harchaoui
Department of Statistics, University of Washington
Joint work with **J. Cutler** and D. Drusvyatskiy

IFDS 2022

Problem

Stochastic optimization with an evolution over time:

$$\min_x \ \varphi_t(x) := f_t(x) + r_t(x)$$

indexed by time $t \in \mathbb{N}$, where

1. loss $f_t : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly convex,

$$\frac{\mu}{2}\|y-x\|^2 \le f_t(y) - f_t(x) - \langle \nabla f_t(x), y-x\rangle \le \frac{L}{2}\|y-x\|^2;$$

2. regularizer $r_t : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is closed and convex;
3. objective φ_t may evolve stochastically in time.

Goal: Track the optimum "as closely as possible" in "shortest amount of time".

Online proximal stochastic gradient method:

$$\text{Set } x_{t+1} = \operatorname{prox}_{\eta_t r_t}(x_t - \eta_t g_t),$$

where g_t is an unbiased estimator of $\nabla f_t(x_t)$ and

$$\operatorname{prox}_{\eta r}(y) = \arg\min_u \left\{ r(u) + \tfrac{1}{2\eta}\|u-y\|^2 \right\}.$$

Motivation

Supervised learning as stochastic optimization:

$$\min_{x \in \mathcal{X}} \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(x, z)]$$

assuming we can draw $z_1, \ldots, z_n \sim \mathcal{D}$.

When... test and training data are often **not drawn from same distribution**.

Two common reasons:

1. (temporal effects) distribution evolves in time
   - parameter tracking, concept drift (Priouret and Juditsky '94, Bartlett et al. '00)
2. (state effects) deployed model influences population data
   - strategic classification, performative prediction (Hardt et al. '16, Perdomo et al. '20)

Optimization problems indexed by time:

$$\min_{x} \underbrace{\mathop{\mathbb{E}}_{z \sim \mathcal{D}_t} [\ell(x, z)]}_{f_t(x)} + \underbrace{\delta_{\mathcal{X}_t}(x)}_{r_t(x)}$$

Stochastic framework

Gradient noise: Let

$$\xi_t = \nabla f_t(x_t) - g_t$$

denote the gradient noise at time t.

Filtration: Let $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space with filtration $\mathbb{F} = (\mathcal{F}_t)_{t \geq 0}$ such that $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and the following hold for all t:

1. $x_t, x_t^\star : \Omega \to \mathbb{R}^d$ are $\mathcal{F}_t$-measurable;
2. $\xi_t : \Omega \to \mathbb{R}^d$ is $\mathcal{F}_{t+1}$-measurable with $\mathbb{E}[\xi_t \,|\, \mathcal{F}_t] = 0$.

Progress bound

Suppose henceforth that $\eta_t \leq 1/2L$.

Lemma: For all $x \in \mathbb{R}^d$,

$$2\eta_t(\varphi_t(x_{t+1}) - \varphi_t(x)) \leq (1-\mu\eta_t)\|x_t - x\|^2 - \|x_{t+1} - x\|^2 + 2\eta_t\langle \xi_t, x_t - x\rangle + 2\eta_t^2\|\xi_t\|^2.$$

Distance recursion: Let $\Delta_t := \|x_t^\star - x_{t+1}^\star\|$ denote the *minimizer drift*. Then

$$\|x_{t+1} - x_{t+1}^\star\|^2 \leq (1-\mu\eta_t)\|x_t - x_t^\star\|^2 + 2\eta_t\langle \xi_t, x_t - x_t^\star\rangle + 2\eta_t^2\|\xi_t\|^2 + \tfrac{2}{\mu\eta_t}\Delta_t^2.$$

- Immediate from lemma by taking $x = x_t^\star$ and applying μ-strong convexity:

$$\frac{\mu}{2}\|x_{t+1} - x_t^\star\|^2 \leq \varphi_t(x_{t+1}) - \varphi_t^\star.$$

Error decomposition

Last-iterate progress: Using constant step size η,

$$\|x_t - x_t^\star\|^2 \le (1-\mu\eta)^t \|x_0 - x_0^\star\|^2 + 2\eta \sum_{i=0}^{t-1} \langle \xi_i, x_i - x_i^\star \rangle (1-\mu\eta)^{t-1-i}$$
$$+ 2\eta^2 \sum_{i=0}^{t-1} \|\xi_i\|^2 (1-\mu\eta)^{t-1-i} + \tfrac{2}{\mu\eta} \sum_{i=0}^{t-1} \Delta_i^2 (1-\mu\eta)^{t-1-i}.$$

Drift and noise: Suppose there are $\Delta, \sigma > 0$ such that for all t,

$$\mathbb{E}\Delta_t^2 \le \Delta^2 \quad \text{and} \quad \mathbb{E}\|\xi_t\|^2 \le \sigma^2.$$

Error decomposition: Using constant step size η,

$$\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \underbrace{(1-\mu\eta)^t \cdot \|x_0 - x_0^\star\|^2}_{\text{optimization}} + \underbrace{\frac{\eta\sigma^2}{\mu}}_{\text{noise}} + \underbrace{\left(\frac{\Delta}{\mu\eta}\right)^2}_{\text{drift}}.$$

Asymptotic error and optimal step size:

$$\mathcal{E} := \min_{\eta \in (0, 1/2L]} \left\{ \frac{\eta\sigma^2}{\mu} + \left(\frac{\Delta}{\mu\eta}\right)^2 \right\} \quad \text{and} \quad \eta_\star := \min\left\{ \frac{1}{2L}, \left(\frac{2\Delta^2}{\mu\sigma^2}\right)^{1/3} \right\}.$$

Numerical illustration

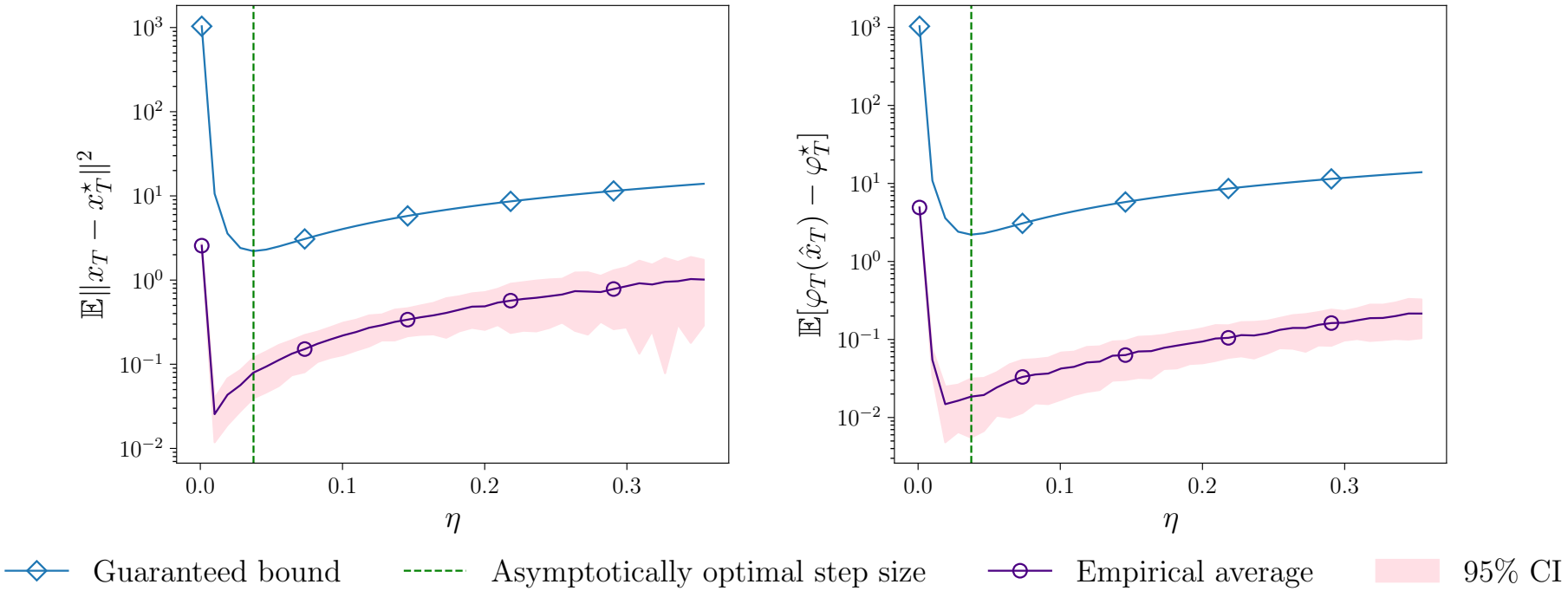


Figure: Semilog plots of guaranteed bounds and empirical tracking errors at horizon T with respect to step size η for logistic regression with stochastically evolving labels.

$$f_t(x) = \frac{1}{n}\left(\sum_{i=1}^{n} \log(1 + \exp\langle a_i, x\rangle) - \langle Ax, b_t\rangle\right) + \frac{1}{2}\|x\|^2$$

Two regimes of variation

Asymptotically optimal step size:

$$\eta_\star = \begin{cases} \frac{1}{2L} & \text{if } \frac{\Delta}{\sigma} \geq \sqrt{\frac{\mu}{16L^3}} \\ \left(\frac{2\Delta^2}{\mu\sigma^2}\right)^{1/3} & \text{otherwise.} \end{cases}$$

Thm (CDH21): In the low drift-to-noise regime, a step-decay schedule $\{\eta_t\}$ ensures:

$$\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \mathcal{E} \quad \text{after time} \quad t \lesssim \frac{L}{\mu}\log\left(\frac{\|x_0 - x_0^\star\|^2}{\mathcal{E}}\right) + \frac{\sigma^2}{\mu^2\mathcal{E}}.$$

- Matches the static setting with $\mathcal{E}$ in place of target accuracy ε.
- Starting at $\eta_0 = 1/2L$, the k^{th} epoch uses step size $\eta_\star + 2^{-k}(\eta_0 - \eta_\star)$.

High-probability guarantees

Sub-Gaussian drift and noise: Suppose there are $\Delta, \sigma > 0$ such that for all t,

1. The drift Δ_t^2 is sub-exponential conditioned on $\mathcal{F}_t$ with parameter Δ^2:

$$\mathbb{E}\Big[\exp(\lambda \Delta_t^2) \,|\, \mathcal{F}_t\Big] \leq \exp(\lambda \Delta^2) \quad \text{for all} \quad 0 \leq \lambda \leq \Delta^{-2}.$$

2. The noise ξ_t is norm sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter $\sigma/2$:

$$\mathbb{P}\Big\{\|\xi_t\| \geq \tau \,|\, \mathcal{F}_t\Big\} \leq 2\exp(-2\tau^2/\sigma^2) \quad \text{for all} \quad \tau > 0.$$

Thm (CDH21): Given $t \in \mathbb{N}$ and $\delta \in (0,1)$, the bound

$$\|x_t - x_t^\star\|^2 \lesssim \left(1 - \frac{\mu\eta}{2}\right)^t \|x_0 - x_0^\star\|^2 + \left(\frac{\eta\sigma^2}{\mu} + \left(\frac{\Delta}{\mu\eta}\right)^2\right) \log\left(\frac{e}{\delta}\right)$$

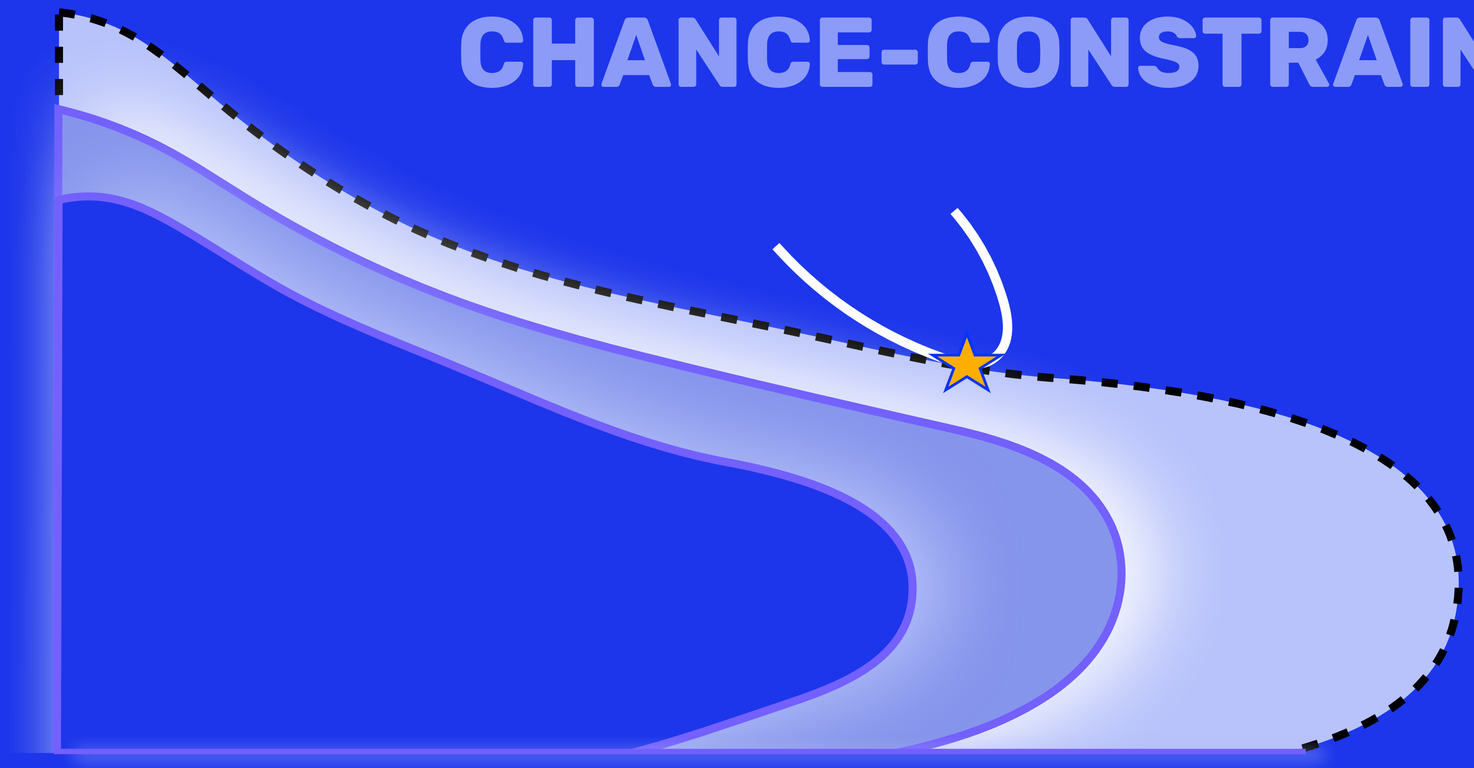
holds with probability at least $1 - \delta$.

- Proof uses the distance recursion to control $\mathbb{E}[\exp(\lambda \|x_t - x_t^\star\|^2)]$.
- Step-decay schedule yields high-probability efficiency estimate as before.

For further details:

- J. Cutler, D. Drusvyatskiy, and Z. Harchaoui. **Stochastic optimization under time drift: iterate averaging, step decay, and high probability guarantees**. In *Advances in Neural Information Processing Systems*, 2021.
- **Stochastic optimization under distributional drift**, `https://arxiv.org/abs/2108.07356`, in revision, 2022.

TACO : A TOOLBOX FOR CHANCE-CONSTRAINED PROGRAMMING



Yassine LAGUEL
https://yassine-laguel.github.io
Rutgers University
Joint Work with J. Malick and W. Van Ackooij

Talk at the IFDS Workshop
on Distributionally Robust Leaning
Seattle, Washington - 3rd, 2022

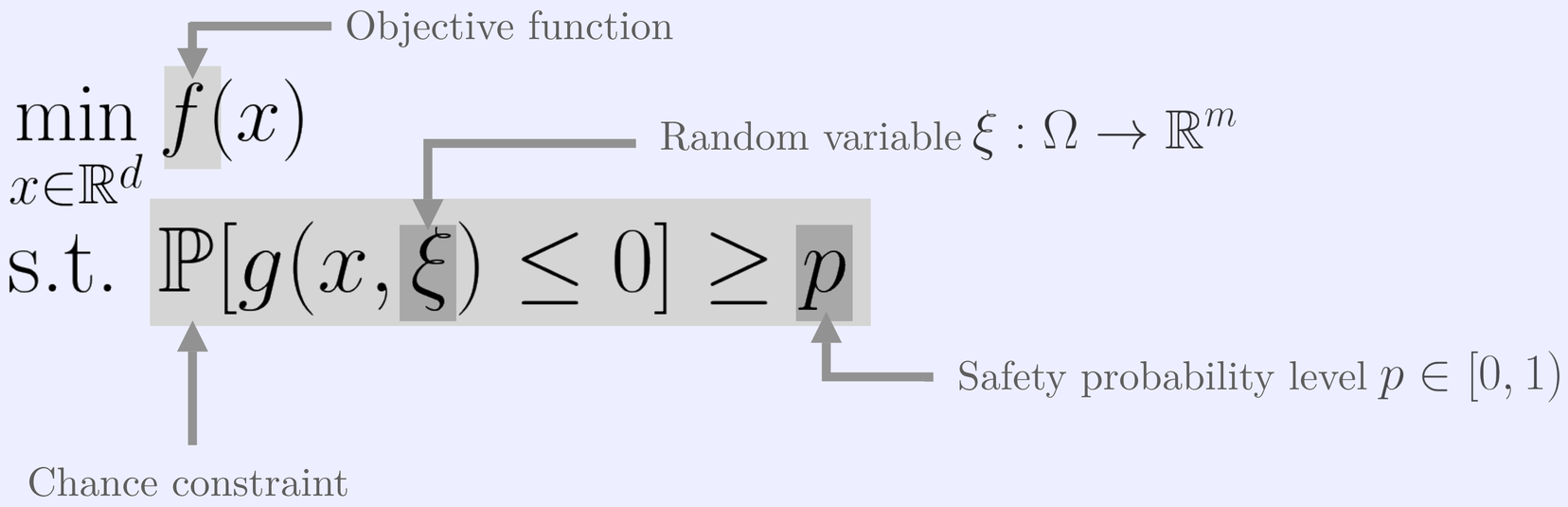
Definition of Chance Constraints

- A chance constraint problem is of the form:

$$\begin{aligned} &\min_{x \in \mathbb{R}^d} f(x) \\ &\text{s.t. } \mathbb{P}[g(x, \xi) \leq 0] \geq p \end{aligned}$$

Definition of Chance Constraints

- A chance constraint problem is of the form:



Definition of Chance Constraints

- A chance constraint problem is of the form:

$$\begin{aligned} &\min_{x \in \mathbb{R}^d} f(x) \\ &\text{s.t. } \mathbb{P}[g(x, \xi) \leq 0] \geq p \end{aligned}$$

- Chance constrained optimization problems are difficult:
  - non-convex
  - non-smooth

From Chance Constraints to Bilevel Programs

- **Our approach**: rewrite chance constraints as

$$\mathbb{P}[g(x,\xi) \leq 0] \geq p \iff Q_p(g(x,\xi)) \leq 0$$

From Chance Constraints to Bilevel Programs

- **Our approach**: rewrite chance constraints as

$$\mathbb{P}[g(x,\xi) \leq 0] \geq p \iff Q_p(g(x,\xi)) \leq 0$$

$$\iff \quad \eta \leq 0$$

$$\eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

- We obtain the following bilevel program:

Upper Level

$$\min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x)$$
$$\text{s.t.} \quad \eta \leq 0$$

Lower Level

$$\eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

We propose a Double Penalization Procedure

- First penalization

$$(\mathcal{P}) \quad \min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x)$$
$$\text{s.t.} \quad \eta \leq 0$$
$$\eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

We propose a Double Penalization Procedure

- First penalization

$$(\mathcal{P}) \quad \min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x) \quad \text{s.t.} \quad \eta \leq 0, \quad \eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

$$\Rightarrow \quad (\mathcal{P}_\mu) \quad \min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x) + \mu \max(\eta, 0) \quad \text{s.t.} \quad \eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

We propose a Double Penalization Procedure

- First penalization

$$(\mathcal{P}) \quad \min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x) \quad \text{s.t.} \quad \eta \leq 0, \quad \eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

$$(\mathcal{P}_\mu) \quad \min_{x \in \mathbb{R}^d, \eta \in \mathbb{R}} f(x) + \mu \max(\eta, 0) \quad \text{s.t.} \quad \eta \in \operatorname*{argmin}_{s \in \mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi) - s, 0)\right]$$

- In practice, the constant μ is a hyperparameter to tune.

We propose a Double Penalization Procedure

- First penalization

$$(\mathcal{P}) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) \quad \text{s.t.} \quad \eta \leq 0, \quad \eta \in \operatorname*{argmin}_{s\in\mathbb{R}} s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi)-s,0)\right]$$

$$(\mathcal{P}_\mu) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu\max(\eta,0) \quad \text{s.t.} \quad \eta \in \operatorname*{argmin}_{s\in\mathbb{R}} \underbrace{s + \frac{1}{1-p}\mathbb{E}\left[\max(g(x,\xi)-s,0)\right]}_{=G(x,s)\geq \bar{Q}_p(g(x,\xi))}$$

- In practice, the constant μ is a hyperparameter to tune.

- Using Rockafellar property

$$\min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu\max(\eta,0) \quad \text{s.t.} \quad G(x,\eta) - \bar{Q}_p(g(x,\xi)) \leq 0$$

We propose a Double Penalization Procedure

- Second penalization

$$(\mathcal{P}_\mu) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu \max(\eta, 0) \quad \text{s.t.} \quad G(x,\eta) - \bar{Q}_p(g(x,\xi)) \leq 0$$

$$(\mathcal{P}_{\lambda,\mu}) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu \max(\eta, 0) + \lambda \left(G(x,\eta) - \bar{Q}_p(g(x,\xi))\right)$$

We propose a Double Penalization Procedure

- Second penalization

$$(\mathcal{P}_\mu) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu \max(\eta, 0) \quad \text{s.t.} \quad G(x,\eta) - \bar{Q}_p(g(x,\xi)) \leq 0$$

$$\Rightarrow \quad (\mathcal{P}_{\lambda,\mu}) \quad \min_{x\in\mathbb{R}^d,\eta\in\mathbb{R}} f(x) + \mu \max(\eta, 0) + \lambda \left(G(x,\eta) - \bar{Q}_p(g(x,\xi))\right)$$

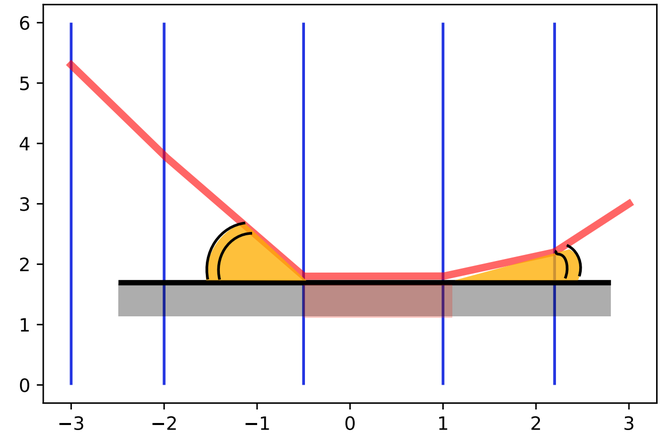
- This penalization is exact.

Theorem Let $\mu > 0$ be given and fixed and assume that the solution set of problem $(\mathcal{P}_\mu)$ is not empty. Then for any $\lambda > \lambda_\mu = \frac{\mu}{\delta}$ where:

$$\delta = \begin{cases} \frac{1}{n(1-p)} & \text{if } p \in \mathcal{I} \\ \frac{d_{\mathcal{I}}(p)}{1-p} & \text{otherwise.} \end{cases}$$

the solution set of $(\mathcal{P}_\mu)$ coincides with the solution set of $(\mathcal{P}_{\lambda,\mu})$

$$\eta \mapsto G(x,\eta) = \eta + \frac{1}{1-p}\mathbb{E}[\max(g(x,\xi) - \eta, 0)]$$



TACO : a Toolbox for chAnce Constrained Optimization

- Goal : solve a problem of the form $\min_{x \in \mathbb{R}^d} f(x)$ s.t. $\mathbb{P}[g(x, \xi) \leq 0] \geq p$
- Input : the class `Problem`
  - First-order oracles for f and g.
  - A sampled dataset for the values of ξ.
  - A python dictionary of parameters.

Example : Kataoka's Example

In[1]:

```
import numpy as np

class Kataoka:

    def __init__(self, nb_samples=10000, nb_features=2, seed=42):

        np.random.seed(seed)
        mean = np.array([1.0, 1.0])
        cov =  np.eye(2)
        self.data = np.random.multivariate_normal(mean, cov,
size=self.nb_samples)

    def objective_func(self, x):
         return 0.5*np.dot(x,x)

    def objective_grad(self,x):
        return x

    def constraint_func(self, x, z):
        return np.dot(x,z)

    def constraint_grad(self, x, z):
        return z
```

TACO : a Toolbox for chAnce Constrained Optimization

- Goal : solve a problem of the form $\min_{x \in \mathbb{R}^d} f(x)$ s.t. $\mathbb{P}[g(x, \xi) \leq 0] \geq p$
- Input : the class `Problem`
  - First-order oracles for f and g.
  - A sampled dataset for the values of ξ.
  - A python dictionary of parameters.
- The class `Optimizer`
  - Instantiate with the inputs.
  - Optimization launched with the method `run`.

Example : Kataoka's Example

```
In[2]: optimizer = Optimizer(problem, params=params)
       optimiser.run()
```

TACO : a Toolbox for chAnce Constrained Optimization

- Goal : solve a problem of the form $\min_{x \in \mathbb{R}^d} f(x)$ s.t. $\mathbb{P}[g(x, \xi) \leq 0] \geq p$
- Input : the class `Problem`
  - First-order oracles for f and g.
  - A sampled dataset for the values of ξ.
  - A python dictionary of parameters.
- The class `Optimizer`
  - Instantiate with the inputs.
  - Optimization launched with the method `run`.
- Output
  - Retrieved from the `Optimizer` class.

Example : Kataoka's Example

```
In[2]: optimizer = Optimizer(problem, params=params)
       optimiser.run()

In[3]: sol = optimizer.solution
```

TACO : a Toolbox for chAnce Constrained Optimization

- Goal : Solve a problem of the form $\min_{x \in \mathbb{R}^d} f(x)$ s.t. $\mathbb{P}[g(x, \xi) \leq 0] \geq p$
- Input : the class `Problem`
  - First-order oracles for f and g.
  - A sampled dataset for the values of ξ.
  - A python dictionary of parameters.
- The class `Optimizer`
  - Instantiate with the inputs.
  - Optimization launched with the method `run`.
- Output
  - Retrieved from the `Optimizer` class.

Example : Kataoka's Example

```
In[2]: optimizer = Optimizer(problem, params=params)
       optimiser.run()

In[3]: sol = optimizer.solution
```

- Hyperparameters
  - Probability threshold p
  - Penalization parameters μ, λ
  - Number of iterations, starting point, target precision, etc.

Proof of concept on a quadratic Chance constraint Problem

- 2D quadratic problem

$$\min_{x \in \mathbb{R}^d} f(x) \qquad f(x) = (x - c)^\top A (x - c)$$
$$\text{s.t. } \mathbb{P}[g(x, \xi) \leq 0] \geq p \qquad g(x, z) = z^\top W(x)^\top z + p^\top z + b$$
$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

Proof of concept on a quadratic Chance constraint Problem

- 2D quadratic problem

$$\min_{x \in \mathbb{R}^d} f(x) \quad \text{s.t. } \mathbb{P}[g(x, \xi) \leq 0] \geq p$$

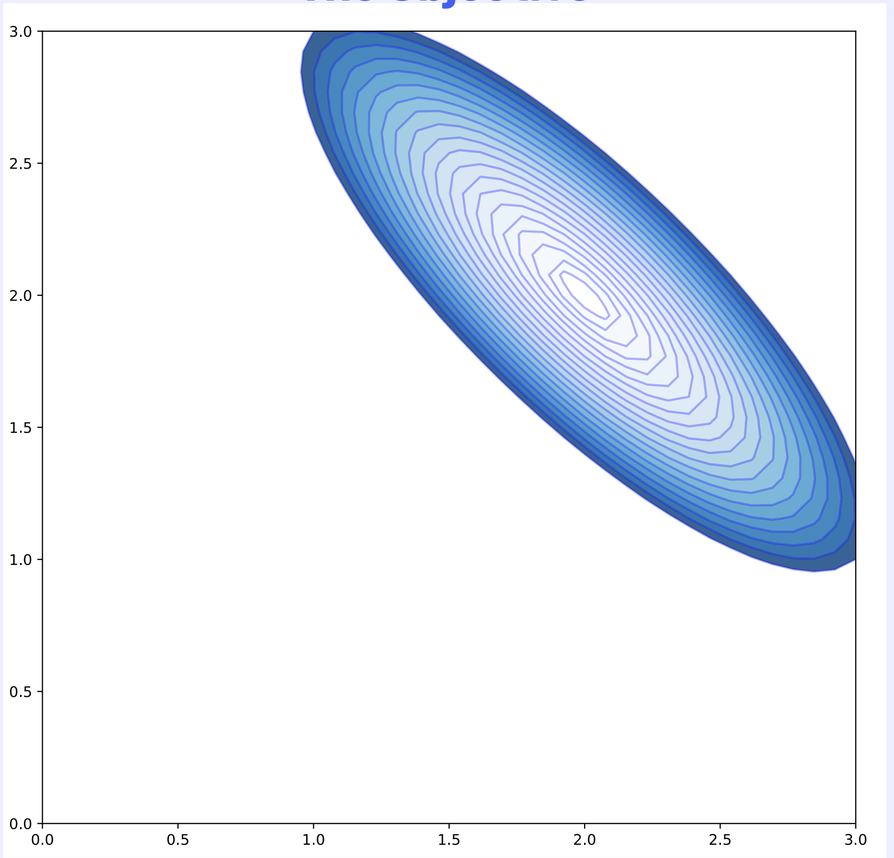
$$f(x) = (x - c)^\top A (x - c)$$

$$g(x, z) = z^\top W(x)^\top z + p^\top z + b$$

$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

$$c = \begin{pmatrix} 2. \\ 2. \end{pmatrix} \quad A = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix}$$

The Objective



Proof of concept on a quadratic Chance constraint Problem

- 2D quadratic problem

$$\min_{x \in \mathbb{R}^d} f(x) \qquad f(x) = (x-c)^\top A (x-c)$$

$$\text{s.t. } \mathbb{P}[g(x,\xi) \leq 0] \geq p \qquad g(x,z) = z^\top W(x)^\top z + q^\top z + r$$

$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

$$c = \begin{pmatrix} 2. \\ 2. \end{pmatrix} \quad A = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix}$$

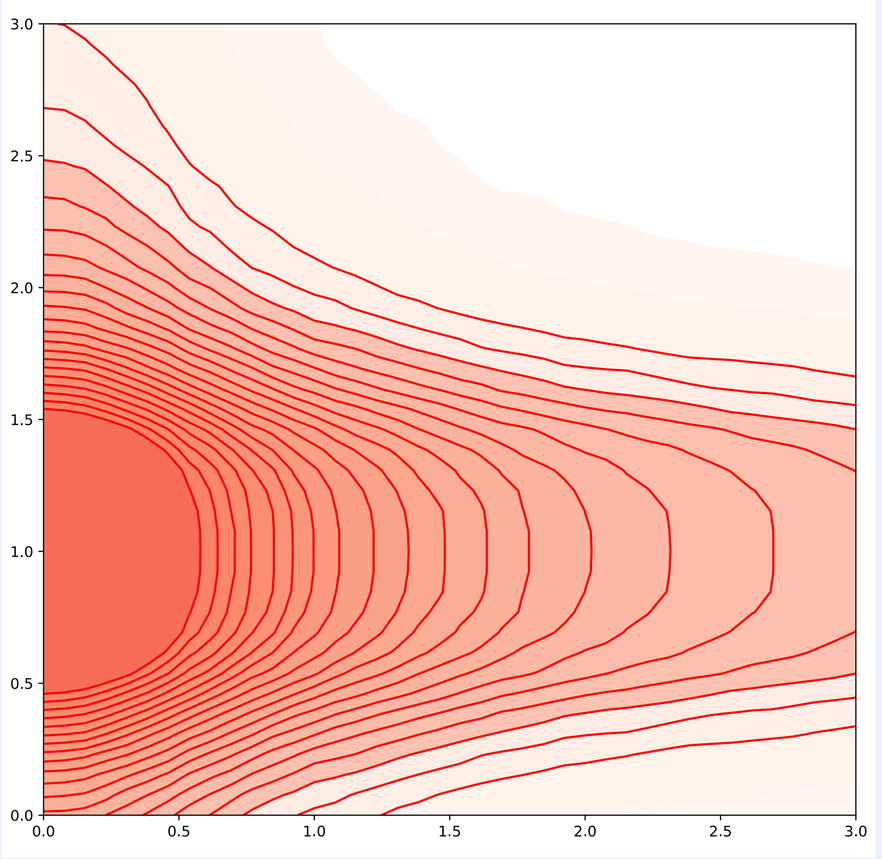
$$W : x = (x_1, x_2)^\top \mapsto \begin{pmatrix} x_1^2 + 0.5 & 0.0 \\ 0.0 & |x_2 - 1|^3 + 1. \end{pmatrix}$$

$$q = \begin{pmatrix} 1. \\ 1. \end{pmatrix}, \; r = -1$$

ξ is sampled 10000 times with parameters $\mu = \begin{pmatrix} 1. \\ 1. \end{pmatrix}$

$$\Sigma = \begin{pmatrix} 20. & 0. \\ 0. & 20. \end{pmatrix}$$

The Chance Constraint



Proof of concept on a quadratic Chance constraint Problem

- 2D quadratic problem

$$\min_{x \in \mathbb{R}^d} f(x) \quad \text{s.t.} \quad \mathbb{P}[g(x,\xi) \leq 0] \geq p$$

$$f(x) = (x-c)^\top A (x-c)$$

$$g(x,z) = z^\top W(x)^\top z + q^\top z + r$$

$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

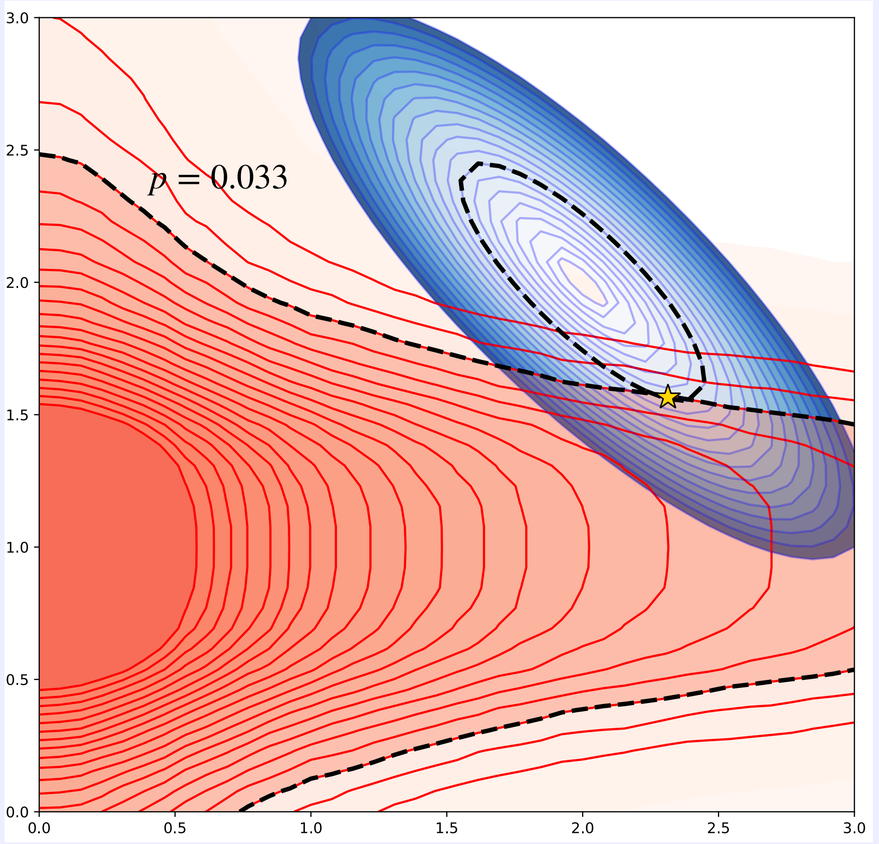
$$c = \begin{pmatrix} 2. \\ 2. \end{pmatrix} \quad A = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix}$$

$$W : x = (x_1, x_2)^\top \mapsto \begin{pmatrix} x_1^2 + 0.5 & 0.0 \\ 0.0 & |x_2 - 1|^3 + 1. \end{pmatrix}$$

$$q = \begin{pmatrix} 1. \\ 1. \end{pmatrix}, \; r = -1$$

ξ is sampled 10000 times with parameters $\mu = \begin{pmatrix} 1. \\ 1. \end{pmatrix}$

$$\Sigma = \begin{pmatrix} 20. & 0. \\ 0. & 20. \end{pmatrix}$$



Proof of concept on a quadratic Chance constraint Problem

- 2D quadratic problem

$$\min_{x \in \mathbb{R}^d} f(x) \quad \text{s.t.} \quad \mathbb{P}[g(x,\xi) \leq 0] \geq p$$

$$f(x) = (x-c)^\top A (x-c)$$

$$g(x,z) = z^\top W(x)^\top z + q^\top z + r$$

$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

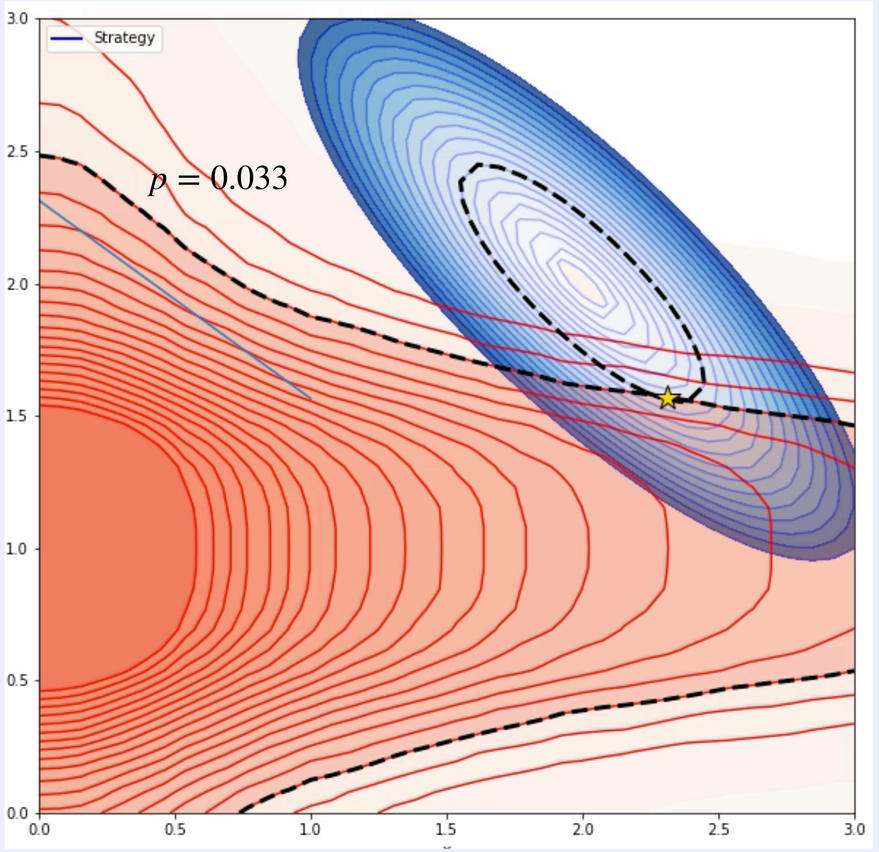
$$c = \begin{pmatrix} 2. \\ 2. \end{pmatrix} \quad A = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix}$$

$$W : x = (x_1, x_2)^\top \mapsto \begin{pmatrix} x_1^2 + 0.5 & 0.0 \\ 0.0 & |x_2 - 1|^3 + 1. \end{pmatrix}$$

$$q = \begin{pmatrix} 1. \\ 1. \end{pmatrix}, \; r = -1$$

ξ is sampled 10000 times with parameters $\mu = \begin{pmatrix} 1. \\ 1. \end{pmatrix}$

$$\Sigma = \begin{pmatrix} 20. & 0. \\ 0. & 20. \end{pmatrix}$$



Conclusion

- We propose a new approach to chance constraints via Bilevel Programming.
- We derive a double penalization method for this approach, with an exact penalty for the hard constraint.
- We propose a python toolbox to test out your problems.
- Derive more methods from the bilevel approach

yassine.laguel@rutgers.edu

Orthogonal Statistical Learning with Self-Concordant Loss

Lang Liu, Carlos Cinelli, Zaid Harchaoui

University of Washington

Orthogonal Statistical Learning

Orthogonal statistical learning (OSL)

- **Data**: $\mathcal{D} := \{Z_1, \ldots, Z_{2n}\}$ i.i.d. sample from $\mathbb{P}$.
- **Target parameter**: $\theta \in \Theta \subset \mathbb{R}^d$.
- **Nuisance**: $g \in (\mathcal{G}, \|\cdot\|_{\mathcal{G}})$
- **Loss**: $\ell_z : \Theta \times \mathcal{G} \to \mathbb{R}_+$.
- **Risk**: $L(\theta, g) := \mathbb{E}_{Z \sim \mathbb{P}}[\ell_Z(\theta, g)]$.
- **Goal**: assuming a true nuisance g_0, want to estimate

$$\theta_\star := \arg\min_{\theta \in \Theta} L(\theta, g_0).$$

Orthogonal Statistical Learning

OSL meta-algorithm

- **Sample splitting**: $\mathcal{D}_1 := \{Z_1, \ldots, Z_n\}$ and $\mathcal{D}_2 := \{Z_{n+1}, \ldots, Z_{2n}\}$.
- **Nuisance parameter**: outputs $\hat{g}$ based on $\mathcal{D}_2$.
- **Target parameter**: outputs $\hat{\theta}$ by minimizing

$$\min_{\theta \in \Theta} L_n(\theta, \hat{g}) := \frac{1}{n} \sum_{i=1}^{n} \ell_{Z_i}(\theta, \hat{g}).$$

- **Excess risk**: $\mathcal{E}(\hat{\theta}, g_0) := L(\hat{\theta}, g_0) - L(\theta_\star, g_0)$.

Assumption: Neyman Orthogonality

For $F : \mathcal{F} \to \mathbb{R}^m$, we define $\mathrm{D}F(f)[h] := \frac{\mathrm{d}}{\mathrm{d}t} F(f + th)|_{t=0}$ for $f, h \in \mathcal{F}$.

Assumption: Neyman Orthogonality

For $F : \mathcal{F} \to \mathbb{R}^m$, we define $\mathrm{D}F(f)[h] := \frac{\mathrm{d}}{\mathrm{d}t} F(f + th)|_{t=0}$ for $f, h \in \mathcal{F}$.

Definition (Neyman orthogonality)

We say L is *Neyman orthogonal* at $(\theta_\star, g_0)$ if

$$\mathrm{D}_g \nabla_\theta L(\theta_\star, g_0)[g - g_0] = 0, \quad \text{for all } g \in \mathcal{G}.$$

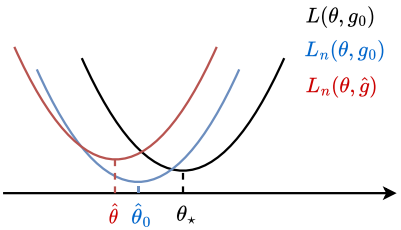
Assumption: Neyman Orthogonality

For $F : \mathcal{F} \to \mathbb{R}^m$, we define $\mathrm{D}F(f)[h] := \frac{\mathrm{d}}{\mathrm{d}t} F(f + th)|_{t=0}$ for $f, h \in \mathcal{F}$.

Definition (Neyman orthogonality)

We say L is *Neyman orthogonal* at $(\theta_\star, g_0)$ if

$$\mathrm{D}_g \nabla_\theta L(\theta_\star, g_0)[g - g_0] = 0, \quad \text{for all } g \in \mathcal{G}.$$



Effective Dimension

Effective dimension

- **Gradient**: $S_z(\theta, g) := \nabla_\theta \ell_z(\theta, g)$.
- **Covariance**: $\Sigma(\theta, g) := \mathsf{Cov}(S_Z(\theta, g))$.
- **Hessian**: $H_\star := \nabla^2_\theta L(\theta_\star, g_0)$.

Effective Dimension

Effective dimension

- **Gradient**: $S_z(\theta, g) := \nabla_\theta \ell_z(\theta, g)$.
- **Covariance**: $\Sigma(\theta, g) := \mathrm{Cov}(S_Z(\theta, g))$.
- **Hessian**: $H_\star := \nabla^2_\theta L(\theta_\star, g_0)$.
- **Effective dimension**: $d_\star := \sup_{g \in \mathcal{G}_{g_0}} \mathrm{Tr}(H_\star^{-1/2} \Sigma(\theta_\star, g) H_\star^{-1/2})$.
  - ▷ Well-specified model—$d_\star = d$.
  - ▷ Mis-specified model—problem-specific characterization of the complexity of Θ.

Main Result

Theorem (Simplified)

Under suitable assumptions, the OSL estimator $\hat{\theta}$ has excess risk, with probability at least $1-\delta$,

$$\mathcal{E}(\hat{\theta}, g_0) \lesssim O\left(\frac{d_\star}{\lambda_\star}\frac{1}{n} + \frac{1}{\lambda_\star}\|\hat{g} - g_0\|_{\mathcal{G}}^4\right)$$

whenever n sufficiently large, where $\lambda_\star := \lambda_{\min}(H_\star)$.

Main Result

Theorem (Simplified)

Under suitable assumptions, the OSL estimator $\hat{\theta}$ has excess risk, with probability at least $1-\delta$,

$$\mathcal{E}(\hat{\theta}, g_0) \lesssim O\left(\frac{d_\star}{\lambda_\star}\frac{1}{n} + \frac{1}{\lambda_\star}\|\hat{g} - g_0\|_{\mathcal{G}}^4\right)$$

whenever n sufficiently large, where $\lambda_\star := \lambda_{\min}(H_\star)$.

Remark

Foster and Syrgkanis (2020) obtained the rate

$$O\left(\frac{d^2}{\lambda_\star^2}\frac{1}{n} + \frac{d}{\lambda_\star^2}\|\hat{g} - g_0\|_{\mathcal{G}}^4\right).$$

Summary

- Novel non-asymptotic bound for the OSL estimator.
- The bound depends on the effective dimension instead of d.
- It improves previous work at least by a factor of d.

Paper

Assumption: Pseudo Self-Concordance

For $f : \mathbb{R}^d \to \mathbb{R}$, we define the derivative operator D as

$$\mathrm{D}^k f(x)[u] := \frac{\mathrm{d}^k}{\mathrm{d}t^k} f(x + tu)|_{t=0}, \quad \text{for } x, u \in \mathbb{R}^d, k \in \mathbb{Z}_+$$

Assumption (Pseudo self-concordance)

For any z and g, the loss $\ell_z(\cdot, g)$ is pseudo self-concordant with parameter R, i.e.,

$$|\mathrm{D}^3_\theta \ell_z(\theta, g)[\eta, \eta, \eta]| \leq R\|\eta\|_2 \mathrm{D}^2_\theta \ell_z(\theta, g)[\eta, \eta], \quad \textit{for all } \theta, \eta.$$

Assumption: Pseudo Self-Concordance

For $f : \mathbb{R}^d \to \mathbb{R}$, we define the derivative operator D as

$$\mathrm{D}^k f(x)[u] := \frac{\mathrm{d}^k}{\mathrm{d}t^k} f(x + tu)|_{t=0}, \quad \text{for } x, u \in \mathbb{R}^d, k \in \mathbb{Z}_+$$

Assumption (Pseudo self-concordance)

For any z and g, the loss $\ell_z(\cdot, g)$ is pseudo self-concordant with parameter R, i.e.,

$$|\mathrm{D}^3_\theta \ell_z(\theta, g)[\eta, \eta, \eta]| \le R\|\eta\|_2 \mathrm{D}^2_\theta \ell_z(\theta, g)[\eta, \eta], \quad \textit{for all } \theta, \eta.$$

Proposition (Bach '10)

Assume that $\ell_z(\cdot, g)$ is pseudo self-concordant with parameter R. For any θ, θ', we have

$$e^{-R\|\theta' - \theta\|_2} \nabla^2_\theta \ell_z(\theta, g) \preceq \nabla^2_\theta \ell_z(\theta', g) \preceq e^{R\|\theta' - \theta\|_2} \nabla^2_\theta \ell_z(\theta, g).$$

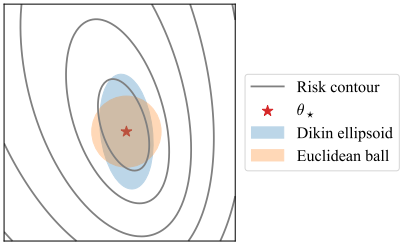
Localization and Dikin Ellipsoid

Assumption (Localization)

There exists $N > 0$ such that for all $n > N$, we have $\hat{\theta} \in \Theta_{\theta_\star}$ and $\hat{g} \in \mathcal{G}_{g_0}$.

Dikin ellipsoid

- **Hessian**: $H(\theta, g) := \nabla^2_\theta L(\theta, g)$ and $H_\star := H(\theta_\star, g_0)$.
- **Dikin ellipsoid**: $\Theta_{\theta_\star, r} := \{\theta \in \Theta : \|\theta - \theta_\star\|_{H_\star} := \|H_\star^{1/2}(\theta - \theta_\star)\|_2 < r\}$.



Main Result

Table: In their simplified version, our bound scales as $O(d_\star/n)$ and Foster and Syrgkanis's bound scales as $O(d'/n)$ where $d' := d^2/\lambda_\star$. We compare them in different regimes of eigendecays.

	Eigendecay		**Ratio**
	$\Sigma_\star$	$H_\star$	$d'/d_\star$
Poly-Poly	$i^{-\alpha}$	$i^{-\beta}$	$d^{(\alpha+1)\wedge(\beta+2)}$
Poly-Exp	$i^{-\alpha}$	$e^{-\nu i}$	$d^{1\wedge(3-\alpha)}$
Exp-Poly	$e^{-\mu i}$	$i^{-\beta}$	$d^{\beta+2}$
Exp-Exp	$e^{-\mu i}$	$e^{-\nu i}$	$de^{\nu d}$ if $\mu = \nu$ $d^2e^{\nu d}$ if $\mu > \nu$ $d^2e^{\mu d}$ if $\mu < \nu$

Proof Sketch

By Taylor's theorem,

$$\mathcal{E}(\hat{\theta}, g_0) = L(\hat{\theta}, g_0) - L(\theta_\star, g_0) = S(\theta_\star, g_0)^\top (\hat{\theta} - \theta_\star) + \|\hat{\theta} - \theta_\star\|^2_{H(\bar{\theta}, g_0)}/2 \lesssim \|\hat{\theta} - \theta_\star\|^2_{H_\star}.$$

Proof Sketch

By Taylor's theorem,

$$\mathcal{E}(\hat{\theta}, g_0) = L(\hat{\theta}, g_0) - L(\theta_\star, g_0) = S(\theta_\star, g_0)^\top (\hat{\theta} - \theta_\star) + \|\hat{\theta} - \theta_\star\|^2_{H(\bar{\theta}, g_0)}/2 \lesssim \|\hat{\theta} - \theta_\star\|^2_{H_\star}.$$

By Taylor's theorem again,

$$\begin{aligned} L_n(\hat{\theta}, \hat{g}) - L_n(\theta_\star, \hat{g}) &= S_n(\theta_\star, \hat{g})^\top (\hat{\theta} - \theta_\star) + \|\hat{\theta} - \theta_\star\|^2_{H_n(\bar{\theta}', \hat{g})}/2 \\ &\gtrsim -\left[\sqrt{d_\star/n} + \|\hat{g} - g_0\|^2_{\mathcal{G}}\right] \|\hat{\theta} - \theta_\star\|_{H_\star} + \|\hat{\theta} - \theta_\star\|^2_{H_\star}. \end{aligned}$$

It follows that

$$\mathcal{E}(\hat{\theta}, g_0) \lesssim \|\hat{\theta} - \theta_\star\|^2_{H_\star} \lesssim \frac{d_\star}{n} + \|\hat{g} - g_0\|^4_{\mathcal{G}}.$$

Proof Sketch

By Taylor's theorem,

$$\mathcal{E}(\hat{\theta}, g_0) := L(\hat{\theta}, g_0) - L(\theta_\star, g_0) = S(\theta_\star, g_0)^\top (\hat{\theta} - \theta_\star) + \|\hat{\theta} - \theta_\star\|^2_{H(\bar{\theta}, g_0)}/2 \lesssim \|\hat{\theta} - \theta_\star\|^2_{H_\star}.$$

By Taylor's theorem again,

$$\begin{aligned} L_n(\hat{\theta}, \hat{g}) - L_n(\theta_\star, \hat{g}) &= S_n(\theta_\star, \hat{g})^\top (\hat{\theta} - \theta_\star) + \|\hat{\theta} - \theta_\star\|^2_{H_n(\bar{\theta}', \hat{g})}/2 \\ &\gtrsim -\left[\sqrt{d_\star/n} + \|\hat{g} - g_0\|^2_{\mathcal{G}}\right] \|\hat{\theta} - \theta_\star\|_{H_\star} + \|\hat{\theta} - \theta_\star\|^2_{H_\star}. \end{aligned}$$

Missing steps

- Control $S_n(\theta_\star, g)$ for every $g \in \mathcal{G}_{g_0}$.
- Relate $H_n(\theta, g)$ to $H(\theta, g)$ and then to $H(\theta_\star, g_0)$ for every $(\theta, g) \in \Theta_{\theta_\star} \times \mathcal{G}_{g_0}$.

Assumptions

Step 1: Relate $S_n(\theta_\star, g)$ to $S(\theta_\star, g)$ and then to $S(\theta_\star, g_0) = 0$.

- Sub-Gaussian score.
- Neyman orthogonal score.

Assumptions

Step 1: Relate $S_n(\theta_\star, g)$ to $S(\theta_\star, g)$ and then to $S(\theta_\star, g_0) = 0$.

- Sub-Gaussian score.
- Neyman orthogonal score.

Step 2: Relate $H_n(\theta, g)$ to $H(\theta, g)$ and then to $H(\theta_\star, g_0)$.

- Matrix Bernstein.
- Pseudo self-concordance.

Assumptions

Step 1: Relate $S_n(\theta_\star, g)$ to $S(\theta_\star, g)$ and then to $S(\theta_\star, g_0) = 0$.

- Sub-Gaussian score.
- Neyman orthogonal score.

Step 2: Relate $H_n(\theta, g)$ to $H(\theta, g)$ and then to $H(\theta_\star, g_0)$.

- Matrix Bernstein.
- Pseudo self-concordance.

Theorem (Informal)

Under assumptions above, with probability at least $1 - \delta$,

$$\mathcal{E}(\hat{\theta}, g_0) \lesssim \frac{e^R}{\kappa^2} \left[K_1^2 \log(1/\delta) \frac{d_\star}{n} + \beta_2^2 \|\hat{g} - g_0\|_{\mathcal{G}}^4 \right]$$

whenever $n \gtrsim \max\{N, (K_2^2 + \sigma_H^2) d^2\}$.