Habilitation à diriger des recherches

Selected Results on First-order Optimization and Collaborative Learning

A walk in the past and a glimpse into the future.

Aymeric DIEULEVEUT
Assistant Professor, École Polytechnique,
Institut Polytechnique de Paris.

Preambule: This slide is empty[1]

[1] Well, not completely.

Contributions to WC analysis and alg. design for deterministic first-order optimization. Part 1
with **B. Goujaud,** A. Taylor,
and C. Moucer, F. Pedregosa, D. Scieur, J. Hendrickx, F. Glineur.

Stochastic Approximation... Part 2
with F. Bach, N. Flammarion, S. Pesme, K. K. Patel, A. Durmus, E. Moulines, G. Fort.

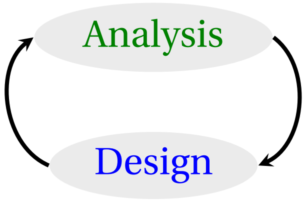
and towards distributed and federated settings
with **C. Philippenko**
and G. Fort, E. Moulines, G. Robin, M. Jaggi, E. Oyallon, L. Leconte, G. Pagès, V. Plassier, M. Vono, M. Noble, A. Bellet.

Learning with Missing Data, Uncertainty quantification and applications Part 3
with **M. Zaffran, A. Ayme,** J. Josse, C. Boyer, E. Scornet, Y. Goude, O. Féron
and A. Sportisse.

My mission: Design and Analysis of Optimization Algorithms

Continuous Optimization

$$\min_{w \in \mathcal{W} \subset \mathbb{R}^d} f(w).$$



- **Design:** Build algorithms to minimize a function f
- **Assumption:** f belongs to a class $\mathcal{F}$.
- **Analysis:** guarantee convergence and rates.

$\rightarrow$ Applications: statistics, machine learning, control.

! Ubiquitously methods not fully understood.

Approach: Iterative algorithms $\rightarrow$ generate $w_0, w_1, \dots, w_t$ from *oracle information*.

	Deterministic	Stochastic	Multi-agent & Federated
	f	$f(w) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(w,z)]$	$f(w) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{z \sim \mathcal{D}_i}[\ell(w,z)]$
Accessed information (oracle)	First-order (FO): ∇f	Stochastic FO	Partial Stochastic FO

Worst-case analysis: ensure convergence **uniformly** over a class $\mathcal{F}$.

- **Trust** in a black-box optimization method,
- **Design** algorithms based on the worst-case guarantees (WCG).

Part 1:
Contributions to WC analysis and alg. design for deterministic first-order optimization.

Designing an optimal algorithms for quadratic functions

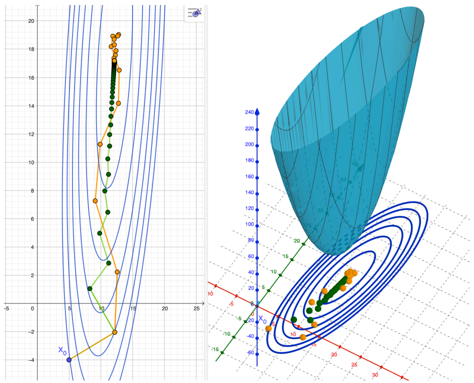
- Quadratic function f_H, $H \in \mathcal{S}_d^+$:

$$f_H(w) - f_H(w_*) = \frac{1}{2}(w - w_*)^\top H(w - w_*).$$

- Class of L-smooth and μ-strongly convex quadratics:

$$\mathcal{Q}_{\mu,L} = \{f_H, \ \ \mathrm{Sp}(H) \subset [\mu, L]\}.$$

$\rightarrow$ *Parametric description of the class!*



First-order methods: $w_T = w_{T-1} - \sum_{i=0}^{T-1} h_{T,i} \nabla f_H(w_i) \Leftrightarrow$ **Link with polynomials:** $w_T - w_\star = P_T(H)(w_0 - w_\star)$.

	Criterion ?	Algorithm?
Optimal method?	$\sup_{f_H \in \mathcal{Q}_{\mu,L}} \frac{\lVert w_T - w_* \rVert^2}{\lVert w_0 - w_* \rVert^2}$	$\rightarrow$ Polyak momentum[1]: $w_{t+1} = w_t - \underbrace{\gamma_t}_{\text{step}} \nabla f_H(w_t) + \underbrace{\beta_t}_{\text{momentum}} (w_t - w_{t-1})$

$\rightarrow$ Limit parameters as $t \rightarrow \infty$

$$\beta^* = \left(\frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}}\right)^2, \quad \gamma^* = \frac{2}{\mu + L}(1 + \beta^*).$$

Also optimal rate for HB (PM with constant β, γ) on $\mathcal{Q}_{\mu,L}$!

WCG and design on $\mathcal{Q}_{\mu,L}$:

- ♡ Success story of worst-case design
- $\rightarrow$ Optimal algorithm – **Heavy Ball**
- $\rightarrow$ Rate $O((1 - 4\sqrt{\kappa})^T)$, $\kappa := \frac{\mu}{L}$
- ? Extending such optimality results?

[1] Polyak, "Some methods of speeding up the convergence of iteration methods"

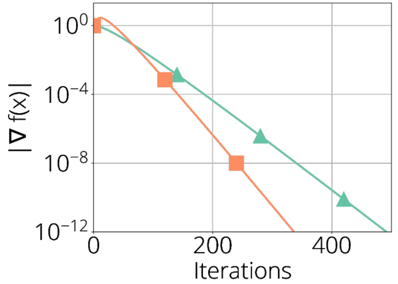
Improving upon Polyak Heavy Ball algorithm in Quadratic Optimization: three directions

Polyak Momentum and Heavy Ball algorithms:

$$
\begin{aligned}
w_{t+1} &= w_t - \gamma_t \nabla f_H(w_t) + \beta_t (w_t - w_{t-1}) \quad \text{(PM)}\\
&= w_t - \underbrace{\gamma}_{\text{step}} \nabla f_H(w_t) + \underbrace{\beta}_{\text{momentum}} (w_t - w_{t-1}) \quad \text{(HB)}
\end{aligned}
$$

Optimal HB tuning on $\mathcal{Q}_{\mu,L}$.

$$\beta^* = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2 \quad \gamma^* = \frac{2}{\mu+L}(1+\beta^*).$$



HB vs Cy-HB on Mnist

1. *Restricting* the class $\mathcal{Q}_{\mu,L}$ 2. *Extending* the class of algorithms

- **Class of functions:** quadratic with a gap in eigenvalues:

$$\mathcal{Q}_{\mu_1,L_1,\mu_2,L_2} = \{f_H, \ \ \mathrm{Sp}(H) \subset [\mu_1, L_1] \cup [\mu_2, L_2]\}\,.$$

$\rightarrow$ **Faster convergence rates with K-cyclical step sizes**

$$w_{t+1} = w_t - \gamma_{t \bmod K} \nabla f_H(w_t) + \beta (w_t - w_{t-1}) \quad \text{(K-Cy-HB)}$$

Super acceleration with cyclical step sizes [a]

$\rightarrow$ **Result:** Super acceleration (beyond $1-\sqrt{\kappa}$!) with (PM) and cyclic steps $\gamma_0, \gamma_1, \gamma_2, \gamma_0, \gamma_1, \gamma_2, \ldots$.

$\rightarrow$ If the gap is symmetric, 2 steps are enough.

♡ Example of class $\mathcal{F}$ over which a frequently-used strategy provably improves.

[a] Goujaud, Scieur, **D**, Taylor, and Pedregosa, "Super-acceleration with cyclical step-sizes"

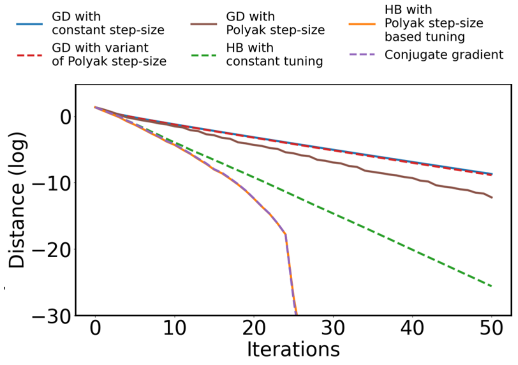
Improving upon Polyak Heavy Ball algorithm in Quadratic Optimization: three directions

Polyak Momentum and Heavy Ball algorithms:

$$\begin{aligned} w_{t+1} &= w_t - \gamma_t \nabla f_H(w_t) + \beta_t (w_t - w_{t-1}) \quad \text{(PM)} \\ &= w_t - \underbrace{\gamma}_{\text{step}} \nabla f_H(w_t) + \underbrace{\beta}_{\text{momentum}} (w_t - w_{t-1}) \quad \text{(HB)} \end{aligned}$$

Optimal HB tuning on $\mathcal{Q}_{\mu,L}$.

$$\beta^* = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2 \quad \gamma^* = \frac{2}{\mu+L}(1+\beta^*).$$



1. *Restricting* the class $\mathcal{Q}_{\mu,L}$ 2. *Extending* the class of algorithms

→ **Adaptive alg. with Polyak step-size and Polyak Momentum**

- **Motivation** Function dependent algorithm → Extends the class of algorithms.
- Classical strategy: *Polyak step size*[2]: step $\propto (f(w_t) - f_\star)/\|\nabla f(w_t)\|^2$
- **New algorithm:** (PM) with $\beta_0 \triangleq 0, \quad \forall t \geq 1,$

$$\beta_t \triangleq \frac{-(f(w_t)-f_\star)\langle \nabla f(w_t), \nabla f(w_{t-1})\rangle}{(f(w_{t-1})-f_\star)\|\nabla f(w_t)\|^2 + (f(w_t)-f_\star)\langle \nabla f(w_t), \nabla f(w_{t-1})\rangle}, \quad \gamma_t \triangleq \frac{2(f(w_t)-f_\star)}{\|\nabla f(w_t)\|^2}(1+\beta_t) \qquad \text{(PSPM)}$$

Theorem 1 (PSPM[3])

(PSPM) *on* $\mathcal{Q}_{\mu,L}$ *is equivalent to a conjugate gradient:*

$$w_{t+1} = \operatorname{argmin}_w \left\{ \|w - w_*\|^2 \ s.t.\ w \in w_0 + \mathrm{Span}\{(\nabla f(w_i))_{i=1}^t\} \right\}$$

(PSPM) algorithm

→ Instance optimality

♡ Theoretically grounded design of PS+PM.

[3]Goujaud, Taylor, and **D**, "Quadratic minimization: from conjugate gradient to an adaptive HB method with Polyak step-sizes"

Improving upon Polyak Heavy Ball algorithm in Quadratic Optimization: three directions

Polyak Momentum and Heavy Ball algorithms:

$$\begin{aligned} w_{t+1} &= w_t - \gamma_t \nabla f_H(w_t) + \beta_t (w_t - w_{t-1}) \qquad \text{(PM)} \\ &= w_t - \underbrace{\gamma}_{\text{step}} \nabla f_H(w_t) + \underbrace{\beta}_{\text{momentum}} (w_t - w_{t-1}) \qquad \text{(HB)} \end{aligned}$$

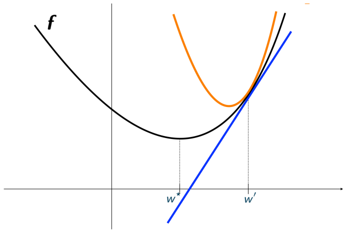
Optimal HB tuning on $\mathcal{Q}_{\mu,L}$.

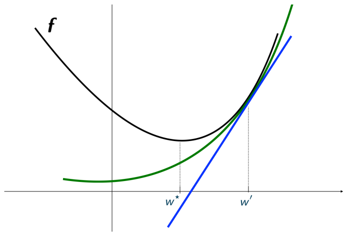
$$\beta^* = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2 \quad \gamma^* = \frac{2}{\mu+L}(1+\beta^*).$$

1. *Restricting* the class $\mathcal{Q}_{\mu,L}$ 2. *Extending* the class of algorithms 3. *Extending* the class (beyond quadratics).

Example: Class of L-smooth and μ-strongly convex $\mathcal{F}_{\mu,L}$ $\rightarrow$ ***Implicit** description of the class!*

- L-smooth functions f,
  $f(w) \leq f(w') + \langle \nabla f(w'), w - w' \rangle + \frac{L}{2}\|w - w'\|^2$
- μ-strongly-convex functions f,
  $f(w) \geq f(w') + \langle \nabla f(w'), w - w' \rangle + \frac{\mu}{2}\|w - w'\|^2$





Challenges:

- Problem: **proofs** rapidly become very complicated - need for a deeper understanding.

Understanding proofs on implicit function classes: Performance estimation problems

$\rightarrow$ **Performance Estimation problems**: rethinking proofs of first-order optimization for implicit classes[4][5].

What is a worst-case guarantee (WCG)?

Example	$\forall f \in \mathcal{Q}_{\mu,L}$	for $w_1 = w_0 - L^{-1}\nabla f(w_0)$	then	$\lVert w_1 - w_\star \rVert^2$	$\leq$	$(1-\kappa)$	$\lVert w_0 - w_\star \rVert^2$
Generically	$\forall f \in \mathcal{F}$	for $w_T = \mathcal{A}(w_0, (\nabla f(w_t))_{t=1}^{T-1})$	then	$\mathrm{Perf}(w_T)$	$\leq$	τ	$\mathrm{Init}(w_0)$
$\rightarrow$	Functional class	Algorithm		Worst-case guarantee			

Equivalently:

$$\tau = \max_{f, w_0, w_T} \frac{\mathrm{Perf}(w_T)}{\mathrm{Init}(w_0)}$$

$$\text{s.t. } f \in \mathcal{F}, w_T = \mathcal{A}(w_0, (\nabla f(w_t))_{t=1}^{T-1})$$

$\rightarrow$ Non-convex optimization problem

$\rightarrow$ Infinite dimensional class of functions.

1. **Sampling** $\rightarrow$ equivalent maximization on $w_i, g_i \in \mathbb{R}^d$, $f_i \in \mathbb{R}$ s.t. **there exists** $f \in \mathcal{F}$, s.t., $\nabla f(w_i) = g_i$, $f(w_i) = f_i$, finite dimension
2. **Interpolation conditions** $\rightarrow$ the existence of f is characterized by simple inequalities: e.g., $\lVert g_i - g_j \rVert^2 \leq L\langle x_i - x_j, g_i - g_j \rangle$.
3. **SDP lifting**: we can recover a convex problem!

Worst-case guarantees with Performance estimation

$\rightarrow$ Finding a WCG rate can be cast a a convex problem

$\rightarrow$ Long derivation $\rightarrow$ Automate the process $\rightarrow$ `Pepit`.

♡ Automatically obtain WCG numerically $\rightarrow$ design, proofchecking

♡♡ Dual $\rightarrow$ Proofs!

[4] Drori and Teboulle, "Performance of first-order methods for smooth convex minimization: a novel approach".

[5] Taylor, Hendrickx, and Glineur, "Smooth strongly convex interpolation and exact worst-case performance of first-order methods".

Computer assisted worst-case analysis : PEPit[8], a performance estimation toolbox in Python

Goals:

- Avoids SDP modeling steps,
- Collaborative and easy-to-use methodology,
- Easy to add new features,
- Code is as close as possible to mathematical specifications.

$\leftrightsquigarrow$ Related Matlab package: `Pesto`[6]

Setup:

- Algorithm: most first-order updates
  $\rightarrow$ **e.g.** Gradient, Prox, Inexact-Gradient, Line Search, ...
- Any class of functions
  (s.t. interpolation constraints expressible linearly in F and G)
  $\rightarrow$ **e.g.** smooth, (strongly-)convex, quadratically upper bounded[7].
- Any performance metric
  (expressible linearly in F and G.)
  $\rightarrow$ **e.g.** $f(w_T) - f_*$, $\|w_T - w_*\|^2$, $\|\nabla f(x_T)\|^2$.

- More than 75+ examples,

- 1. Unconstrained convex minimization
  - 1.1. Gradient descent
  - 1.2. Subgradient method
  - 1.3. Subgradient method under restricted secant inequality and error bound
  - 1.4. Gradient descent with exact line search
  - 1.5. Conjugate gradient
  - 1.6. Heavy Ball momentum
  - 1.7. Accelerated gradient for convex objective
  - 1.8. Accelerated gradient for strongly convex objective
  - 1.9. Optimized gradient
  - 1.10. Optimized gradient for gradient
  - 1.11. Robust momentum
  - 1.12. Triple momentum
  - 1.13. Information theoretic exact method
  - 1.14. Proximal point
  - 1.15. Accelerated proximal point
  - 1.16. Inexact gradient descent
  - 1.17. Inexact gradient descent with exact line search
  - 1.18. Inexact accelerated gradient
  - 1.19. Epsilon-subgradient method
  - 1.20. Gradient descent for quadratically upper bounded convex objective
  - 1.21. Gradient descent with decreasing step sizes for quadratically upper bounded convex objective
  - 1.22. Conjugate gradient for quadratically upper bounded convex objective
  - 1.23. Heavy Ball momentum for quadratically upper bounded convex objective
- 2. Composite convex minimization
  - 2.1. Proximal gradient
  - 2.2. Accelerated proximal gradient
  - 2.3. Bregman proximal point
  - 2.4. Douglas Rachford splitting
  - 2.5. Douglas Rachford splitting contraction
  - 2.6. Accelerated Douglas Rachford splitting
  - 2.7. Frank Wolfe
  - 2.8. Improved interior method
  - 2.9. No Lips in function value
  - 2.10. No Lips in Bregman divergence
  - 2.11. Three operator splitting
- 3. Non-convex optimization
  - 3.1. Gradient Descent
  - 3.2. No Lips 1
  - 3.3. No Lips 2
- 4. Stochastic and randomized convex minimization
  - 4.1. Stochastic gradient descent
  - 4.2. Stochastic gradient descent in overparametrized setting
  - 4.3. SAGA
  - 4.4. Point SAGA
  - 4.5. Randomized coordinate descent for smooth strongly convex functions
  - 4.6. Randomized coordinate descent for smooth convex functions
- 5. Monotone inclusions and variational inequalities
  - 5.1. Proximal point
  - 5.2. Accelerated proximal point
  - 5.3. Optimal Strongly-monotone Proximal Point
  - 5.4. Douglas Rachford Splitting
  - 5.5. Three operator splitting
  - 5.6. Optimistic gradient
  - 5.7. Past extragradient
- 6. Fixed point
  - 6.1. Halpern iteration
  - 6.2. Optimal Contractive Halpern iteration
  - 6.3. Krasnoselskii-Mann with constant step-sizes
  - 6.4. Krasnoselskii-Mann with increasing step-sizes
- 7. Potential functions
  - 7.1. Gradient descent Lyapunov 1
  - 7.2. Gradient descent Lyapunov 2
  - 7.3. Accelerated gradient method
- 8. Inexact proximal methods
  - 8.1. Accelerated inexact forward backward
  - 8.2. Partially inexact Douglas Rachford splitting
  - 8.3. Relatively inexact proximal point
- 9. Adaptive methods
  - 9.1. Polyak steps in distance to optimum
  - 9.2. Polyak steps in function value
- 10. Low dimensional worst-cases scenarios
  - 10.1. Inexact gradient
  - 10.2. Non-convex gradient descent
  - 10.3. Optimized gradient
  - 10.4. Frank Wolfe
  - 10.5. Proximal point
  - 10.6. Halpern iteration
  - 10.7. Alternate projections
  - 10.8. Averaged projections
  - 10.9. Dykstra
- 11. Continuous-time models
  - 11.1. Gradient flow for strongly convex functions
  - 11.2. Gradient flow for convex functions
  - 11.3. Accelerated gradient flow for strongly convex functions
  - 11.4. Accelerated gradient flow for convex functions
- 12. Tutorials
  - 12.1. Contraction rate of gradient descent

- Complete doc, 50$\star$
- Github link

[6] Taylor, Hendrickx, and Glineur, "Performance estimation toolbox (PESTO): automated worst-case analysis of FO optimization methods"
[7] Goujaud, Taylor, and **D**, "Optimal first-order methods for convex functions with a quadratic upper bound"
[8] Goujaud, Moucer, Glineur, Hendrickx, Taylor, and **D**, "PEPit: computer-assisted worst-case analyses of FO optimization methods in Python".

Application - Building Counter-examples to first-order methods

A long standing question: does $(\text{HB})_{\gamma,\beta}$ accelerate on $\mathcal{F}_{\mu,L}$?

What is known?

1. For the optimal tuning γ^*, β^* on $\mathcal{Q}_{\mu,L}$ there exists a function over which (HB) cycles.[9]
2. There exist parameters γ, β for which (HB) converges uniformly on $\mathcal{F}_{\mu,L}$, but without acceleration.[10]

But no general answer... yet, one of the most widely used algorithm in practice!

Searching for cycles[11]

♡ Cycles can be observed after a finite number of iterations.

♡♡ Finding a cycle of length K can be cast as a PEP!

$$\begin{array}{|ll} \underset{d\geq 1, f\in\mathcal{F}, w\in\left(\mathbb{R}^d\right)^{\mathbb{N}}}{\text{minimize}} & \|w_0 - w_K\|^2 \\ \text{subject to} & \begin{cases} w = \mathcal{A}(f, (w_t)_{t\in[\![0,\ell-1]\!]}) \\ \|w_1 - w_0\|^2 \geq 1. \end{cases} \end{array}$$

→ Existence of a cycle ⇒ no worst-case conv. guarantee

→ Application to various classes of algorithm!

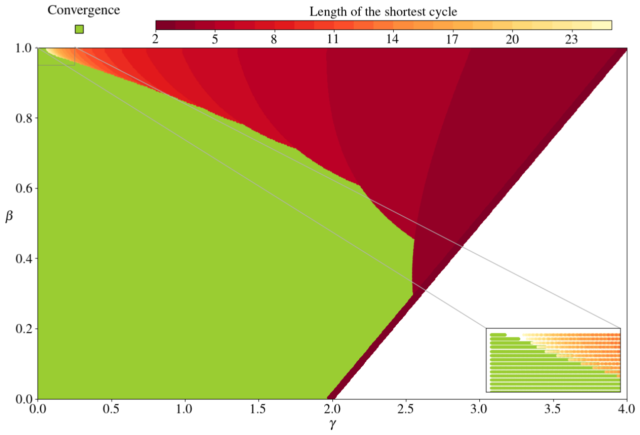


Figure: Cycles for HB

[9] Lessard, Recht, and Packard, "Analysis and design of optimization algorithms via integral quadratic constraints".
[10] Ghadimi, Feyzmahdavian, and Johansson, "Global convergence of the heavy-ball method for convex optimization".
[11] Goujaud, **D**, and Taylor, "Counter-examples in first-order optimization: a constructive approach"

Does HB accelerate on $\mathcal{F}_{\mu,L}$?!

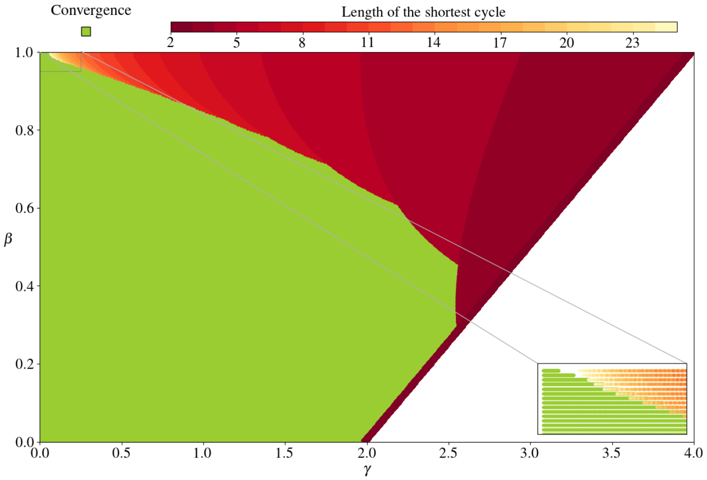


Figure: Cycles for HB

→ For *almost* any set of parameters, either we have a Lyapunov function, or a cycle!

→ so far, computed numerically only...

Theorem 2 (GTD, this week)

For any set of parameters (β, γ) such that $(HB)_{\gamma,\beta}$ admits a worst-case convergence guarantee on $\mathcal{Q}_{\mu,L}$:

1. *Either there exists a function in $\mathcal{F}_{\mu,L}$ and an initialization such that $(HB)_{\gamma,\beta}$ cycles.*
2. *Or the worst-case convergence rate on $\mathcal{Q}_{\mu,L}$ is at best $\Omega(1 - c\kappa)^t$.*

HB does not accelerate on $\mathcal{F}_{\mu,L}$![12]

[12] This may not be a big thing a for you, but was Baptiste's life mission, and quickly became ours...

Wrapping-up – First-order optimization

Worst-case convergence analysis for first-order methods: strong guarantees and a design guide.

On quadratic functions:

- Classically: HB and conjugate gradient algorithms

$\rightarrow$ New HB algorithms (adaptive or cyclical)!

Beyond quadratics...

$\rightarrow$ PEPit can make your life drastically easier.

$\rightarrow$ Fantastic tool to analyze and design new algorithms.

$\rightarrow$ HB does not accelerate!

What's next:

♠ Leveraging Performance Estimation in more complex situations (stochastic, structure)

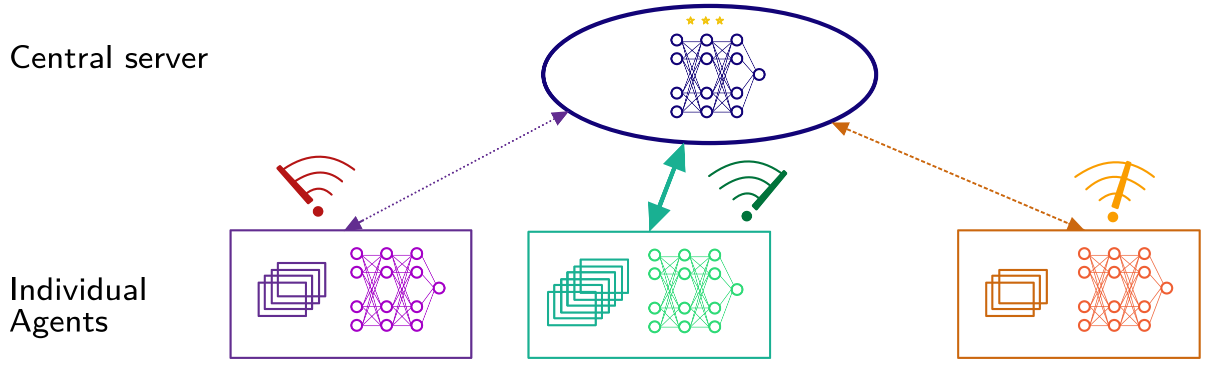
♠♠♠ Automatic formal proofs (beyond numerical).

Federated Learning: a collaborative learning framework

Part 2: Insights on communication constrained Federated Learning with statistical heterogeneity

Objective:

- building better models in Machine Learning
- by enabling multiple participants to participate in training process

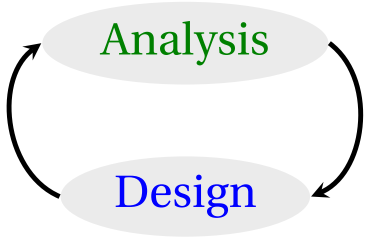


Applications:

- Medical data - multiple hospitals
- Network of devices

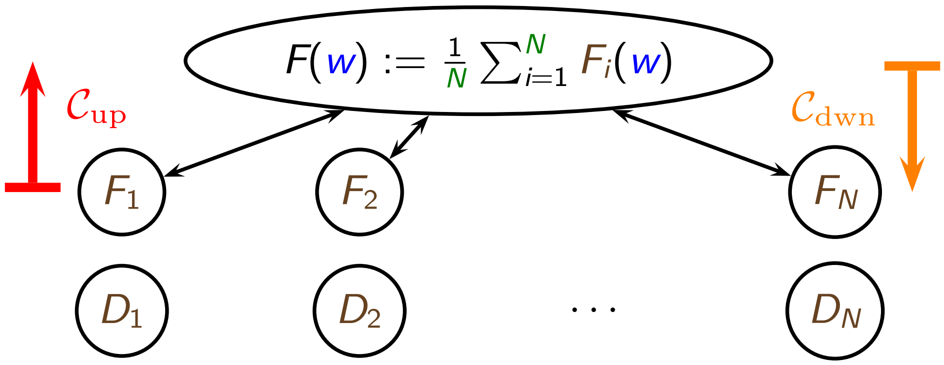
Challenges

- Heterogeneity and Adaptation
- Privacy and trust
- Communication, device availability, (adversaries)



→ Mathematical framework and compressed based approaches

Federated Learning: mathematical framework and communication constraints



$$w_* = \arg\min_{w\in\mathbb{R}^d} \left\{ F(w) := \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{E}_{z\sim\mathcal{D}_i}\left[\ell(z,w)\right]}_{F_i(w)} \right\}.$$

F: global cost function
F_i: local loss
N: workers
d: dimension

w: model
$\mathcal{D}_i$: local data distribution
g_k^i: stochastic oracle on ∇F_i

$\mapsto$ **Optimization based on Stochastic Approximation**

Compressed Distributed SGD:

$$w_k = w_{k-1} - \gamma \left(\frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(g_k^i(w_{k-1})) \right)$$

$\to$ Communication cost $N \times 32d \times k$

$\to$ Communicate with a fraction of workers

$\to$ Communicate a fraction of the weights

$\to$ Communicate low precision updates on weights

$\to$ Perform multiple local iterations before communication

Fedavg (local iterations):

$$w_k = w_{k-1} - \gamma \left(\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_{\mathrm{loc}}} g_{k,t}^i(w_{k-1,t}^i) \right)$$

Motivation for Compression

- Includes multiple natural solutions
- Complementary to local iterations
- $\rightsquigarrow$ quantized models (e.g. binary networks)
- Focus on bi-directional compression

Compression operators - overview and desirable properties

1. Sparsification / projection based
   - $p-$sparsification →Keep each coordinate with probability p

$$\mathcal{C}(x) = p^{-1}((B_i)_{1\leq i\leq d}) \odot x, \qquad (B_i)_{1\leq i\leq d} \sim \mathcal{B}(p)^{\otimes d}$$

   - Partial participation → client sampling

$$\mathcal{C}(x) = p^{-1}(B_0)x, \qquad B_0 \sim \mathcal{B}(p)$$

} Communicate a fraction of the weights

} Communicate with a fraction of workers

2. Quantization on a codebook: Scalar Quantization[13], Delaunay[14]

$$\mathcal{C}(x) = \|x\|\mathrm{sign}(x) \odot ((B_i)_{1\leq i\leq d}), \qquad (B_i)_{1\leq i\leq d} \sim \otimes_{i=1}^{d}\mathcal{B}\left(\frac{|x_i|}{\|x\|}\right).$$

} Communicate low precision updates

Desirable properties → nothing like traditional SP and IT coding!

The compressed signal is stochastic	→	No need for *low-error* compression
Non-stationary unknown distribution	→	no distributional assumption.
Repeated communication: multiple iterations and multiple agents	→	unbiased (random) compression

Assumption U-RBV Compression operators $\mathcal{C}$ is U-RBV: There exists a constant $\omega \in \mathbb{R}_+^*$ s.t. for all Δ in $\mathbb{R}^d$:

$$\mathbb{E}[\mathcal{C}(\Delta)] = \Delta \quad \text{and} \quad \mathbb{E}\left[\|\mathcal{C}(\Delta) - \Delta\|^2\right] \leq \omega \|\Delta\|^2 .$$

[13] Alistarh, Grubic, Li, Tomioka, and Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding"
[14] Leconte, **D**, Oyallon, Moulines, and Pages, "DoStoVoQ: Doubly Stochastic Voronoi Vector Quantization SGD for Federated Learning"

Roadmap

$\rightarrow$ Mathematical framework and compressed based approaches ✓

1 Mitigating heterogeneity for compression based approaches

2 Feedback loops to reduce error

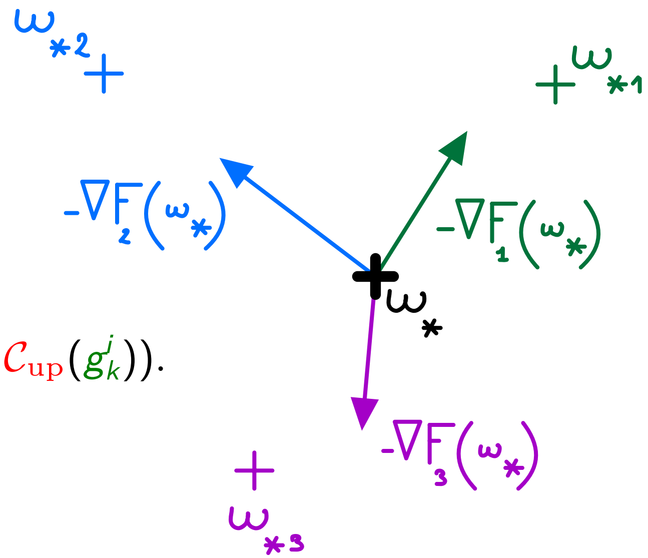
3 Beyond worst case assumption on compression operators

4 An unbiased Random Voronoi compressor.

1. a. Tradeoffs between **heterogeneity** and **communication constraints**

1. Warm up example: distributed gradient descent with client subsampling:

$$w_k = w_{k-1} - \gamma \left(\frac{1}{N} \sum_{i=1}^{N} \frac{B_i}{p} \nabla f_i(w_{k-1}) \right) \quad (B_i) \sim \mathcal{B}(p)^{\otimes d}$$



→ Particular case of compression

→ Heterogeneity: w_* is not a stable point for GD on any F_i

2. General case: SGD with double compression: $w_k = w_{k-1} - \gamma \mathcal{C}_{\mathrm{dwn}}\left(\frac{1}{N}\sum_{i=1}^{N} \mathcal{C}_{\mathrm{up}}(g_k^i)\right)$.

Lemma 3 (Variance increase for bi-directionally compressed SGD[15])

(H1) *Compression operators $\mathcal{C}_{\mathrm{dwn}}$ and $\mathcal{C}_{\mathrm{up}}$ are U-RBV, constants $\omega_{\mathrm{up}}, \omega_{\mathrm{dwn}}$.*

(H2) *Gradient oracles g_k^i are unbiased with variance σ^2. $\mathbb{E}[\|g_k^i - \nabla F_i(w_{k-1})\|^2 | w_{k-1}] \leq \sigma^2$.*

(H3) *Device heterogeneity. $B^2 = N^{-1} \sum_{i=1}^{N} \|\nabla F_i(w_*)\|^2$*

Then $\mathcal{C}_{\mathrm{dwn}}\left(\frac{1}{N}\sum_{i=1}^{N} \mathcal{C}_{\mathrm{up}}(g_k^i)\right)$ is an unbiased stochastic oracle of $\nabla F(w_{k-1})$, with variance bounded by :

$$(1+\omega_{\mathrm{dwn}})(1+\frac{\omega_{\mathrm{up}}}{N})\sigma^2 + \omega_{\mathrm{dwn}}\frac{\omega_{\mathrm{up}}}{N} B^2 + \omega_{\mathrm{dwn}} \|\nabla F(w_{k-1})\|^2$$

[15] Philippenko and **D**, "Artemis: tight convergence guarantees for bidirectional compression in Federated Learning"

[15] **D**, Durmus, and Bach, "Bridging the Gap between Constant Step Size Stochastic Gradient Descent and Markov Chains"

1.b. Mitigating the variance increase with control variate

- **Objective:** recover a convergence similar to the homogeneous case (indep. of B^2)
- **Solution:** Compute (on the server and the worker independently) a **"memory"** h_k^i [16] s.t. $h_k^i \to \nabla F_i(w_*)$.

$$\begin{cases} w_k &= w_{k-1} - \gamma \mathcal{C}_{\mathrm{dwn}} \left(\frac{1}{N} \sum_{i=1}^N \mathcal{C}_{\mathrm{up}} (g_k^i - h_k^i) + h_k^i \right) \\ h_{k+1}^i &= h_k^i + \alpha \mathcal{C}_{\mathrm{up}} (g_k^i - h_k^i) \end{cases} \tag{1}$$

Theorem 4 (Convergence of (1)[17])

*Under regularity assumptions, and **(H1-3)** there exists $\gamma_{\max}$ s.t. for $\gamma \leq \gamma_{\max}$, for $\alpha \in \{0, (\omega_{\mathrm{up}} + 1)^{-1}\}$ and for $k \in \mathbb{N}$, the mean squared distance to w_* decreases at a linear rate up to a constant of the order of E_α:*

$$\mathbb{E}\left[\|w_k - w_*\|^2\right] \leq (1 - \gamma\mu)^k \left(\delta_0^2 + \tau_0^2\right) + \frac{2\gamma E_\alpha}{\mu N}, \qquad \begin{cases} E_0 &= (\omega_{\mathrm{dwn}} + 1)((\omega_{\mathrm{up}} + 1)\sigma_*^2 + \omega_{\mathrm{up}} B^2) \\ E_{(\omega_{\mathrm{up}}+1)^{-1}} &= (\omega_{\mathrm{dwn}} + 1)\ (2\omega_{\mathrm{up}} + 1)\sigma_*^2 \end{cases}$$

♡ Control variates for compression + heterogeneity.

♡ Recover the variance w.o. heterogeneity.

⇝ Client-wise variance reduction scheme[18]

→ Other applications: Langevin[19], EM[20], MM.

→ Duality with local iterations (e.g., *Scaffold*)[21],[22]

[16] Mishchenko, Gorbunov, Takáč, and Richtárik, "Distributed learning with compressed gradient differences"
[17] Philippenko and **D**, "Artemis: tight convergence guarantees for bidirectional compression in Federated Learning"
[18] Schmidt, Le Roux, and Bach, "Minimizing finite sums with the stochastic average gradient"
[19] Vono, Plassier, Durmus, **D**, and Moulines, "QLSD: Quantised Langevin stochastic dynamics for Bayesian federated learning"
[20] **D**, Fort, Moulines, and Robin, "Federated-EM with heterogeneity mitigation and variance reduction"
[21] Karimireddy, Kale, Mohri, Reddi, Stich, and Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning"
[22] Noble, Bellet, and **D**, "Differentially private federated learning on heterogeneous data"

2. Mitigating compression by feedback loops: "non-degraded" update

Feedback loops

When using compression, the worker/server observes **both** the signal and its compressed and transmitted version.
→This can be leveraged to improve convergence.[23]

In bi-directional compression frameworks,

1. Approach (1):

- compress the aggregated update, update the model, broadcast it back.
- The gradient is taken at the point w_k held by the central server.

$$w_k = w_{k-1} - \gamma \mathcal{C}_{\mathrm{dwn}} \left(\frac{1}{N} \sum_{i=1}^{N} \mathcal{C}_{\mathrm{up}}(g_k^i(w_{k-1})) \right)$$

2. MCM [24]

- preserve the model on the central server.
- Gradient measured at $\hat{w}_k$:
  - $\hat{w}_k$ is updated through a compressed update by $\mathcal{C}_{\mathrm{dwn}}$
  - $\mathbb{E}[\hat{w}_k | w_k] = w_k$
  - The variance is controlled

$$w_k = w_{k-1} - \gamma \left(\frac{1}{N} \sum_{i=1}^{N} \mathcal{C}_{\mathrm{up}}(g_k^i(\hat{w}_{k-1})) \right)$$

[24] Karimireddy, Rebjock, Stich, and Jaggi, "Error Feedback Fixes SignSGD and other Gradient Compression Schemes"
[24] Philippenko and **D**, "Preserved central model for faster bidirectional compression in distributed settings"

2.b. Three sequence update and convergence for MCM

MCM. Design of $\hat{w}_k$: *three-sequences update.*

1. Main preserved model w_k on the central server
2. Unbiased model estimator $\hat{w}_k$ on workers
3. Support model for difference compression H_k
4. The *difference* Ω_{k+1} between the model and the support is compressed and exchanged
5. The local model $\hat{w}_k$ is reconstructed from this information

$$\begin{cases} w_k & = w_{k-1} - \gamma \left(\frac{1}{N} \sum_{i=1}^{N} \mathcal{C}_{\mathrm{up}}(g_k^i(\hat{w}_{k-1})) \right) \\ \Omega_{k+1} & = w_{k+1} - H_k \\ \widehat{w}_{k+1} & = H_k + \mathcal{C}_{\mathrm{dwn}}(\Omega_{k+1}) \\ H_{k+1} & = H_k + \alpha_{\mathrm{dwn}} \mathcal{C}_{\mathrm{dwn}}(\Omega_{k+1}). \end{cases} \tag{MCM}$$

$\to$ The third sequence H_k is critical to control the variance of the local model $\widehat{w}_{k+1}$ with unbiased compression.

Theorem 5 (Convergence of MCM, convex case)

*Under **H1-3**, for $K \in \mathbb{N}$, with a step-size $\gamma = \sqrt{\frac{\delta_0^2 Nb}{(1+\omega_{\mathrm{up}})\sigma^2 K}}$, denoting $\bar{w}_K = \frac{1}{K}\sum_{i=0}^{K-1} w_i$, we have:*

$$\mathbb{E}\left[F(\bar{w}_K) - F_*\right] \leq \underbrace{2\sqrt{\frac{\delta_0^2(1+\omega_{\mathrm{up}})\sigma^2}{NbK}}}_{\textit{dominant term}} + \underbrace{O\left(\frac{\omega_{\mathrm{up}}\omega_{\mathrm{dwn}}}{K}\right)}_{\textit{lower order term}} .$$

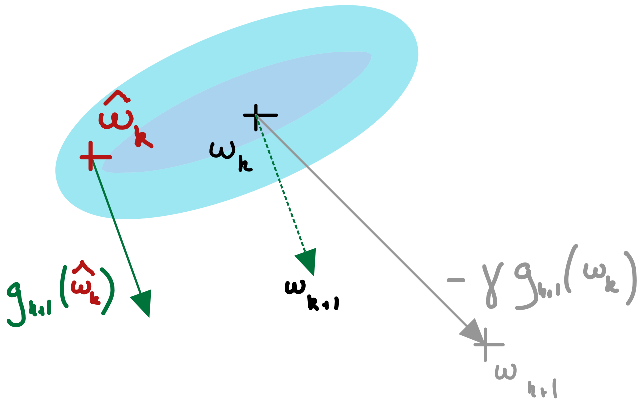
Dominant term:
- independent of ω_{dwn}
- identical to Diana (uni-compression)

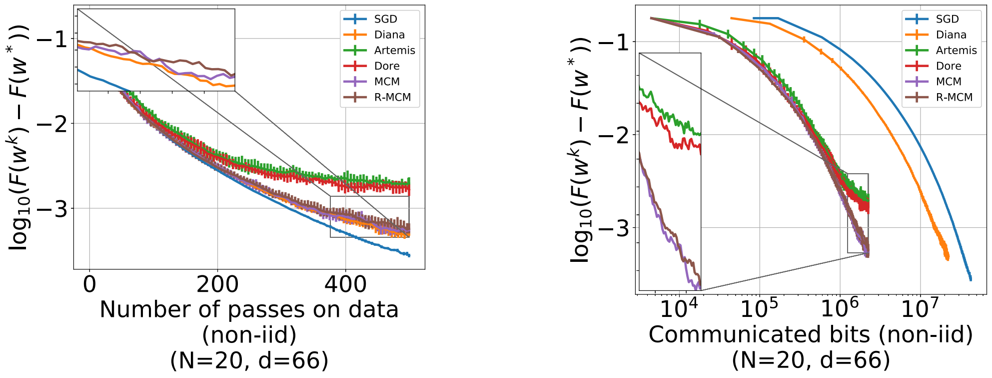
Lower order term:
- depends on ω_{dwn}
- asymptotically negligible

2.c. MCM- summary and experiments

MCM: New algorithm for bi-directional compression with a preserved central model

- ♡ Link with randomized smoothing: unbiased local models.
- ♡ Reduces (nearly cancels) impact of downlink compression
- → Achieves the same asymptotic rate of convergence as unidirectional compression.
- → Extension to worker dependent model on the downlink compression





(a) X axis in # iterations

(b) X axis in # bits

Figure: Quantum with $b = 400$, $\gamma = 1/L$ (LSR).

3.a. Beyond the worst-case assumption on compression[28]

Assumption U-RBV Compression operators $\mathcal{C}$ is U-RBV: There exists a constant $\omega \in \mathbb{R}_+^*$ s.t. for all Δ in $\mathbb{R}^d$:

$$\mathbb{E}[\mathcal{C}(\Delta)] = \Delta \quad \text{and} \quad \mathbb{E}\left[\|\mathcal{C}(\Delta) - \Delta\|^2\right] \leq \omega \|\Delta\|^2 .$$

1. Encompasses all examples cited before
2. Yet, this hides two differences.

1. **Regularity:**
   - Sparsification/projection based are often a.s. linear : $\mathcal{W}_2(C_s(x), C_s(y))^2 \leq \omega\|x-y\|^2$.
   - Quantization based are not $\mathcal{W}_2(C_q(x), C_q(y))^2 \geq \|x-y\|$.
2. **Higher-order moments.** E.g., $p-$client-sampling and $p-$sparsification satisfy the same U-RVB assumption

Idea: consider the **Least-Squares Regression** framework with compression.

- Tight asymptotic[25] and non-asymptotic theory[26][27].
- Typically for a smooth stochastic gradient-field.

---

[25] Polyak and Juditsky, "Acceleration of Stochastic Approximation by Averaging".
[26] Bach and Moulines, "Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n)".
[27] **D** and Bach, "Nonparametric stochastic approximation with large step-sizes".
[28] Philippenko and **D**, "Convergence rates compressed least-square regression: application to Federated Learning".

3.b. Beyond the worst-case assumption on compression[29]

Theorem 6 (For compressed LSR (single worker), Holder compression scheme, $\gamma \propto K^{-\alpha}$)

$$\mathbb{E}\left[F(\bar{w}_{K-1}) - F(w_*)\right] \leq \frac{20}{K}\left(\mathrm{Tr}\left(\mathfrak{C}H^{-1}\right) \underbrace{+ \frac{\left\|H^{-1/2}\eta_0\right\|^2}{K^{(1-2\alpha)}} + \frac{\mathcal{M}_1\sqrt{\mathcal{A}}}{\mu K^{\alpha/2}} + \frac{\mathcal{M}_2\mathcal{A}}{\mu K^{\alpha}}}_{\text{For Pr. Gadat}}\right)$$

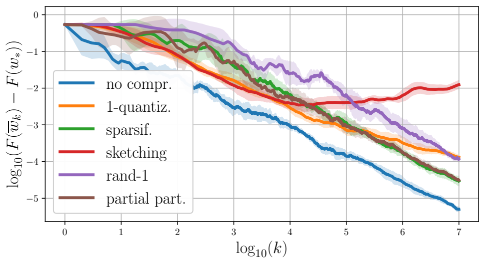
where $\mathfrak{C} = \mathbb{E}[\mathcal{C}(\epsilon)^{\otimes 2}] \to$ **noised induced by the compression near convergence**.

Depending on the compression scheme:

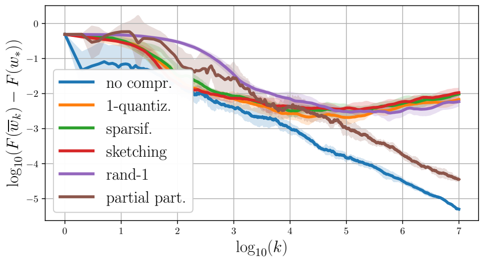
All or nothing: $\mathfrak{C} = aH$

Sparsification: $\mathfrak{C} = a'H + b\ \mathrm{diag}(H)$.

Random projection: $\mathfrak{C} = a''H + b''\ \mathrm{Tr}(H)\ \mathrm{Id}_d$



H diagonal $\uparrow$ or not $\downarrow$

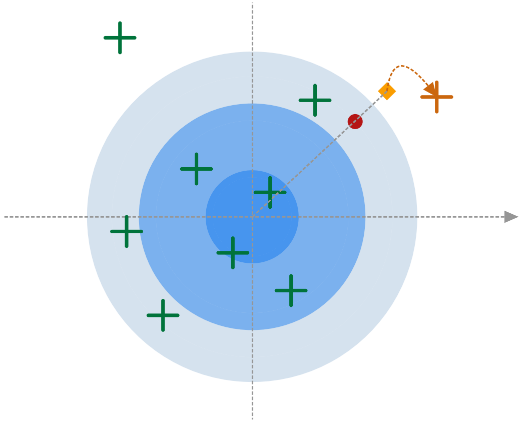


- $\to$ Classical LMS: noise covariance $\to H$
- $\to$ Compression may induce isotropic noise $\to \mathrm{Id}$
- $\to$ Significantly impacts the limit distribution / rate $(\mathrm{Tr}(H^{-1}))$
- $\to$ Same variance but different behaviors!
- ♡ LSR to understand compression.

[29] Philippenko and **D**, "Convergence rates compressed least-square regression: application to Federated Learning".

4. A new Unbiased Voronoi Vector Quantization: StoVoQ Algorithm

→ **Voronoi Vector quantization** The input $x \in \mathbb{R}^d$ is mapped to its nearest neighbor in a codebook $\mathcal{D}_M = \{c_i\}_{i=1}^M$.
→ **Random codebook.** A *new codebook* is sampled *every time* a quantization is performed.
→ **Unitary invariant codewords** The distribution of the codewords p is binvariant under the unitary group
→ **Bias removal: (pre)-compute r_M^p.** and output $\frac{1}{r_M^p(\|x\|)} \mathrm{VQ}(x, \mathcal{D}_M^p)$



Theorem 7 (Quantization bias)

Assume that the codebook distribution is unitary invariant. Then, for all $M \in \mathbb{N}$, there exists a function $r_M^p : \mathbb{R}_+ \mapsto \mathbb{R}_+$ such that for all $x \in \mathbb{R}^d$,

$$\mathbb{E}_{\mathcal{D}_M \sim p}[\mathrm{VQ}(x, \mathcal{D}_M)] = r_M^p(\|x\|)x.$$

- The expectation of the quantized vector is colinear to the vector x, i.e., is directionally unbiased.
- The radial bias only depends on $\|x\|$, M and the distribution p.

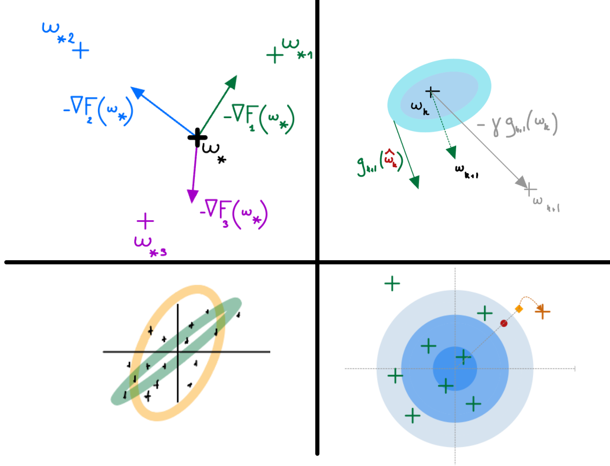
StovoQ[30]

♡ Chosen compression rate (M - vs 2^d atoms for SQ)
♡ Many variants (spherical, rotated grid, Gaussian).
♡♡ Variance $\simeq$ twice smaller than classical SQ

→ randomness on the codebook, not the decomposition.
⋆⋆ Detailed analysis of the debiasing function r_M

[30] Leconte, **D**, Oyallon, Moulines, and Pages, "DoStoVoQ: Doubly Stochastic Voronoi Vector Quantization SGD for Federated Learning"

Wrapping-up – Federated learning and communications constraints



Four examples of algorithm designs or theoretical insights.

→ Mathematical framework and compressed based approaches

✓ Mitigating heterogeneity for compression based approaches

✓ Feedback loops to reduce error

✓ Beyond worst case assumption on compression operators

✓ An unbiased Random Voronoi compressor.

What's next:

♣ Nearly all insights above can extend to any federated task obtained as a Stochastic Approximation: SGD, EM, MM, TD-learning...[a]

♠ Feedback loops and Performance estimation problems.

♠♠ Stability of compressed SGD and generalization (non regular operators)

♠♠ Implicit regularization of compression Schemes (over-parametrized least-squares)

♠♠ Choosing the error distribution in compression to improve convergence (randomized smoothing)

♠♠♠ Compressed models, binary networks, etc.

[a] **D**, Fort, Moulines, and Hoi-To, "Stochastic Approximation Beyond Gradient for Signal Processing and Machine Learning".

Contributions to learning with prediction with missing data

1. Stochastic algorithms for prediction with missing-data[31]
2. Consistency for linear models and worst-case guarantees[32].
3. Impact of imputation: implicit regularization of imputation in high dimension[33]

Contributions to uncertainty quantification with conformal prediction

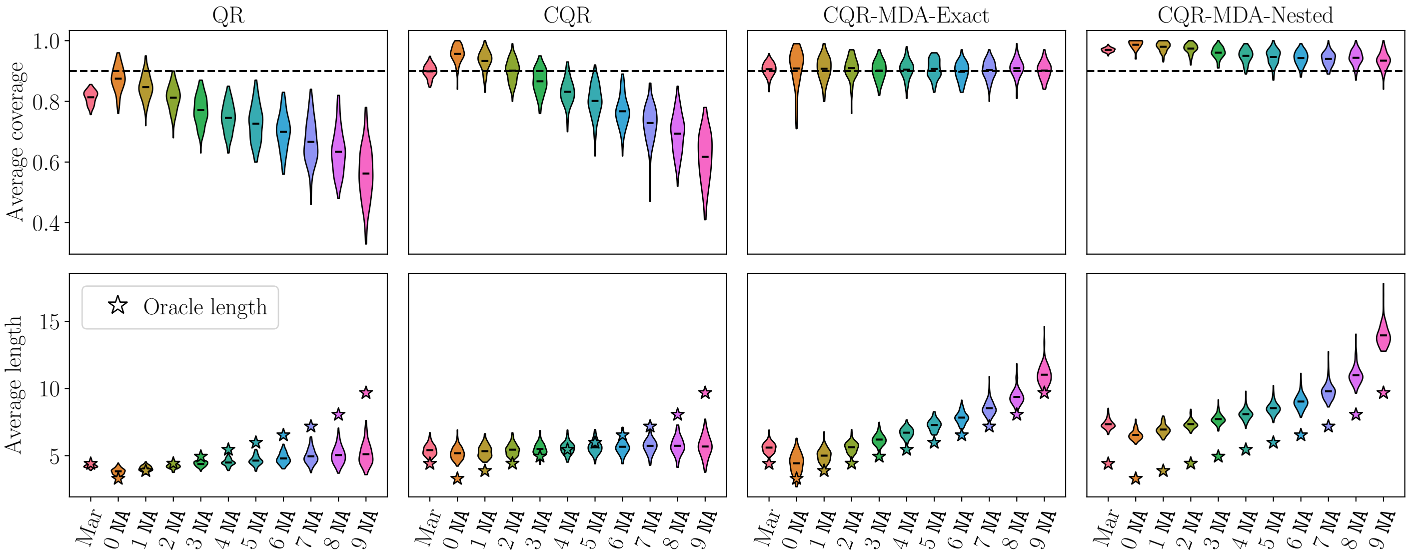
1. For time series[34]
2. With missing data[35]

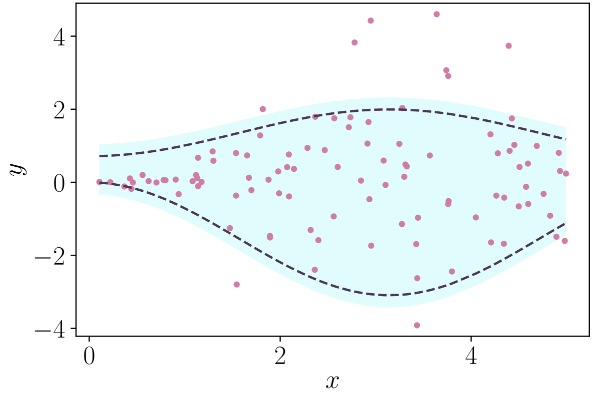
- Link between with multi-task learning and prediction with missing data
- Leverage the links between the tasks
- Link between the classical assumptions (MCAR, MAR, MNAR) and relations between patterns.

[31] Sportisse, Boyer, **D**, and Josse, "Debiasing averaged stochastic gradient descent to handle missing values".
[32] Ayme, Boyer, **D**, and Scornet, "Near-optimal rate of consistency for linear models with missing values".
[33] Ayme, Boyer, **D**, and Scornet, "Naive imputation implicitly regularizes high-dimensional linear models".
[34] Zaffran, Féron, Goude, Josse, and **D**, "Adaptive conformal predictions for time series".
[35] Zaffran, **D**, Josse, and Romano, "Uncertainty quantification in presence of missing values".

Uncertainty quantification for prediction with missing data.

1. The pattern may be informative
2. In most situations, the prediction uncertainty increases with the number of un-observed data





Conformalization recovers marginal coverage

→ **Conformalized quantile regression with missing data.**[36]

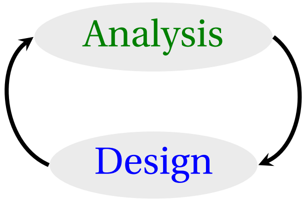
[36] Zaffran, **D**, Josse, and Romano, "Uncertainty quantification in presence of missing values".

An ongoing challenge: Design and Analysis of Optimization Algorithms

Continuous Optimization

$$\min_{w \in \mathcal{W} \subset \mathbb{R}^d} f(w).$$

- **Design:** Build algorithms to minimize a function f
- **Assumption:** f belongs to a class $\mathcal{F}$.
- **Analysis:** guarantee convergence and rates.



$\rightarrow$ Applications: statistics, machine learning, control.

Deterministic Stochastic Multi-agent & Federated

♥♠