Learning Optimization Algorithms with Average Case Convergence Rates

Peter Ochs

Mathematical Optimization for Data Science
Saarland University

— 07.05.2026 —

joint work: Michael Sucker

Example:

$$\min_x f(x)\,, \quad f(x) := \frac{1}{2}\|Ax - b\|^2\,.$$

Example:

$$\min_x f(x)\,, \quad f(x) := \frac{1}{2}\|Ax - b\|^2\,.$$

How do we solve the problem?

Example:

$$\min_x f(x)\,, \quad f(x) := \frac{1}{2}\|Ax - b\|^2\,.$$

How do we solve the problem?

- Inspect the properties of the problem.

  Example: Smooth/Quadratic problem with $L = \|A\|^2$-Lipschitz gradient.

Example:

$$\min_x f(x)\,, \quad f(x) := \frac{1}{2}\|Ax - b\|^2\,.$$

How do we solve the problem?

- Inspect the properties of the problem.

  Example: Smooth/Quadratic problem with $L = \|A\|^2$-Lipschitz gradient.

- Embed the problem into a **class of problems** for which algorithms are available.

  Example: Use Gradient Descent with step size $\alpha = 1/L$

$$x^{(t+1)} = x^{(t)} - \alpha \nabla f(x^{(t)})\,.$$

Worst case convergence guarantee:

$$f(x^{(t)}) - \min f \leq O(1/t)\,.$$

Hidden Structures

If we knew ...

If we knew ...

- that for 99% of the problems that we have to solve the matrix A is diagonal

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{pmatrix} \quad ?$$

Hidden Structures

If we knew ...

- that for 99% of the problems that we have to solve the matrix A is diagonal

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{pmatrix} \quad ?$$

- In that case, we could solve 99% of all problems directly

$$x_i^* = b_i / a_i \,, \quad i = 1, \ldots, n \,.$$

If we knew ...

- that for 99% of the problems that we have to solve the matrix A is diagonal

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{pmatrix} \quad ?$$

- In that case, we could solve 99% of all problems directly

$$x_i^* = b_i / a_i \,, \quad i = 1, \ldots, n \,.$$

- The worst-case does not explain what is usually observed in practice.

If we knew ...

- that for 99% of the problems that we have to solve the matrix A is diagonal

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{pmatrix} \quad ?$$

- In that case, we could solve 99% of all problems directly

$$x_i^* = b_i / a_i \,, \quad i = 1, \ldots, n \,.$$

- The worst-case does not explain what is usually observed in practice.
- Sometimes the **"best" class of problems is not obvious!**

Hidden Structures

If we knew ...

- that for 99% of the problems that we have to solve the matrix A is diagonal

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{pmatrix} \quad ?$$

- In that case, we could solve 99% of all problems directly

$$x_i^* = b_i / a_i \,, \quad i = 1, \ldots, n \,.$$

- The worst-case does not explain what is usually observed in practice.
- Sometimes the **"best" class of problems is not obvious!**

Can we construct an algorithm that adapts to hidden problem structures?

Data Driven Approach

Learning alleviates the bounds of analytic tractability
by providing more:

- **Information:** Leverage more structure, *for example, statistical information*.
- **Automation:** Less “hand-crafting”.
- **Possibilities:** More building blocks.

Data Driven Approach

Learning alleviates the bounds of analytic tractability
by providing more:

- **Information:** Leverage more structure, *for example, statistical information*.
- **Automation:** Less “hand-crafting”.
- **Possibilities:** More building blocks.

Our goal: **Go beyond analytic tractability**

- Develop completely new algorithms.
- Define tight classes of problems.
- Break the barrier of worst-case estimates.
- We insist on having theoretical guarantees.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

$$\min_{x \in \mathbb{R}^d} \ \ell(x, \boldsymbol{\theta}) \,, \quad \text{random variable } \boldsymbol{\theta} : \left(\Omega, \mathfrak{A}, \mathbb{P}\right) \to \Theta \text{ defines the } \textbf{class of problems}.$$

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

$$\min_{x \in \mathbb{R}^d} \ \ell(x, \boldsymbol{\theta}) \,, \quad \text{random variable } \boldsymbol{\theta} : \left(\Omega, \mathfrak{A}, \mathbb{P}\right) \to \Theta \text{ defines the } \textbf{class of problems}.$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}) \,, \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}) \,, \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.
- Preconditioned GD: $x^{(t+1)} = x^{(t)} - P\nabla\ell(x^{(t)}, \theta)$, i.e. $\alpha := P$, $\mathcal{H} := \mathbb{R}^{d \times d}$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}), \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

  $$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \,|\, x^{(t)}_{\alpha,\theta} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}}.$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.
- Preconditioned GD: $x^{(t+1)} = x^{(t)} - P\nabla\ell(x^{(t)}, \theta)$, i.e. $\alpha := P$, $\mathcal{H} := \mathbb{R}^{d \times d}$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ \ell(x, \boldsymbol{\theta}) \,, \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

  $$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \,|\, x^{(t)}_{\alpha,\theta} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\} \,, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}} \,.$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.
- Preconditioned GD: $x^{(t+1)} = x^{(t)} - P\nabla\ell(x^{(t)}, \theta)$, i.e. $\alpha := P$, $\mathcal{H} := \mathbb{R}^{d \times d}$.
- Check if $x^{(t)}$ satisfies $\|\nabla\ell(x^{(t)}, \theta)\| \leq \varepsilon$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}), \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

  $$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \,|\, x^{(t)}_{\alpha,\theta} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}}.$$

- For best **expected case complexity** w.r.t. choice of algorithm $\mathcal{A}$, solve

  $$\min_{\alpha \in \mathcal{H}} \Big(\quad \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{x} | \boldsymbol{\alpha}} \{\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x})\} \quad \Big).$$

Example:

- Regularized Inverse Problem: $\ell(x, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \lambda R(x)$, i.e. $\theta := \lambda$, $\Theta = [0, 1]$.
- Preconditioned GD: $x^{(t+1)} = x^{(t)} - P\nabla\ell(x^{(t)}, \theta)$, i.e. $\alpha := P$, $\mathcal{H} := \mathbb{R}^{d \times d}$.
- Check if $x^{(t)}$ satisfies $\|\nabla\ell(x^{(t)}, \theta)\| \leq \varepsilon$.

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

$$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}), \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

$$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

$$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \,|\, x^{(t)}_{\alpha,\theta} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}}.$$

- For best **expected case complexity** w.r.t. choice of algorithm $\mathcal{A}$, solve

$$\min_{\alpha \in \mathcal{H}} \Big(\quad \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{x} | \boldsymbol{\alpha}} \{\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x})\} \quad \Big).$$

Using **expectation is intractable** in general!

Quest for Theoretical Convergence Guarantees

Routine in **Machine Learning**:
Empirical Risk Minimization

Quest for Theoretical Convergence Guarantees

Routine in **Machine Learning**:
Empirical Risk Minimization

- Want to minimize the **risk** $\mathcal{R}(\alpha)$ defined as the **expected loss**:

$$\min_{\alpha \in \mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[f(\alpha, \boldsymbol{\theta})\big]\,.$$

Deep Learning notation:

α	$\sim$	*neural network parameters*
f	$\sim$	*network architecture including loss*
$\boldsymbol{\theta}$	$\sim$	*data distribution*

Quest for Theoretical Convergence Guarantees

Routine in **Machine Learning**:
Empirical Risk Minimization

- Want to minimize the **risk** $\mathcal{R}(\alpha)$ defined as the **expected loss**:

$$\min_{\alpha \in \mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[f(\alpha, \boldsymbol{\theta})\big]\,.$$

This is intractable since the distribution of $\boldsymbol{\theta}$ is **unknown**.

Deep Learning notation:

α	$\sim$	*neural network parameters*
f	$\sim$	*network architecture including loss*
$\boldsymbol{\theta}$	$\sim$	*data distribution*

Quest for Theoretical Convergence Guarantees

Routine in **Machine Learning**:
Empirical Risk Minimization

- Want to minimize the **risk** $\mathcal{R}(\alpha)$ defined as the **expected loss**:

$$\min_{\alpha \in \mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[f(\alpha, \boldsymbol{\theta})\big]\,.$$

This is intractable since the distribution of $\boldsymbol{\theta}$ is **unknown**.

Deep Learning notation:

α	$\sim$	*neural network parameters*
f	$\sim$	*network architecture including loss*
$\boldsymbol{\theta}$	$\sim$	*data distribution*

- Hence, resort to minimizing the **empirical risk** $\hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]})$ over some dataset $\boldsymbol{\theta}_{[N]} := \{\boldsymbol{\theta}_i\}_{i=1}^N$:

$$\min_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]})\,, \quad \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) := \frac{1}{N} \sum_{i=1}^{N} f(\alpha, \boldsymbol{\theta}_i)\,.$$

Quest for Theoretical Convergence Guarantees

Routine in **Machine Learning**:
Empirical Risk Minimization

- Want to minimize the **risk** $\mathcal{R}(\alpha)$ defined as the **expected loss**:

$$\min_{\alpha \in \mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[f(\alpha, \boldsymbol{\theta})\big]\,.$$

This is intractable since the distribution of $\boldsymbol{\theta}$ is **unknown**.

Deep Learning notation:

α	$\sim$	*neural network parameters*
f	$\sim$	*network architecture including loss*
$\boldsymbol{\theta}$	$\sim$	*data distribution*

- Hence, resort to minimizing the **empirical risk** $\hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]})$ over some dataset $\boldsymbol{\theta}_{[N]} := \{\boldsymbol{\theta}_i\}_{i=1}^N$:

$$\min_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]})\,, \quad \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) := \frac{1}{N} \sum_{i=1}^{N} f(\alpha, \boldsymbol{\theta}_i)\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for the overall performance $\mathcal{R}$?

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \leq \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

- Such bounds are called **PAC-bounds**, which is an acronym for:

$$\underbrace{\text{Probably}}_{\text{With high probability,}} \quad \underbrace{\text{Approximately}}_{\text{the empirical risk is close to}} \quad \underbrace{\text{Correct}}_{\text{the true risk.}}\,.$$

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\left\{\mathcal{R}\left(\alpha^*_{\boldsymbol{\theta}_{[N]}}\right) \leq \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\right\} \geq 1 - \epsilon\,.$$

- Such bounds are called **PAC-bounds**, which is an acronym for:

$$\underbrace{\text{Probably}}_{\text{With high probability,}} \quad \underbrace{\text{Approximately}}_{\text{the empirical risk is close to}} \quad \underbrace{\text{Correct}}_{\text{the true risk.}}\,.$$

PAC-Bayes extends this to the Bayes-risk:

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

- Such bounds are called **PAC-bounds**, which is an acronym for:

$$\underbrace{\text{Probably}}_{\text{With high probability,}} \quad \underbrace{\text{Approximately}}_{\text{the empirical risk is close to}} \quad \underbrace{\text{Correct}}_{\text{the true risk.}}\,.$$

PAC-Bayes extends this to the Bayes-risk:

- Such bounds hold for **distributions** $\rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})$:

$$\mathbb{P}\Big\{\rho^*_{\boldsymbol{\theta}_{[N]}}[\mathcal{R}(\boldsymbol{\alpha})] \leq \inf_{\rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})} \rho[\hat{\mathcal{R}}(\boldsymbol{\alpha}, \boldsymbol{\theta}_{[N]})] + K(\rho, N, \epsilon)\Big\} \geq 1 - \epsilon\,,$$

where $\mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})$ denotes some class of (probability) measures on $\mathcal{H}$ that satisfy a certain property w.r.t. the **distribution** $\mathbb{P}_{\boldsymbol{\alpha}}$.

Is the performance on $\hat{\mathcal{R}}$ representative for $\mathcal{R}$?

- Yes, if we have **uniform generalization bounds**, i.e. bounds of the form: $\forall \varepsilon > 0$:

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*_{\boldsymbol{\theta}_{[N]}}\big) \;\leq\; \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \boldsymbol{\theta}_{[N]}) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

- Such bounds are called **PAC-bounds**, which is an acronym for:

$$\underbrace{\text{Probably}}_{\text{With high probability,}} \quad \underbrace{\text{Approximately}}_{\text{the empirical risk is close to}} \quad \underbrace{\text{Correct}}_{\text{the true risk.}}\,.$$

PAC-Bayes extends this to the Bayes-risk:

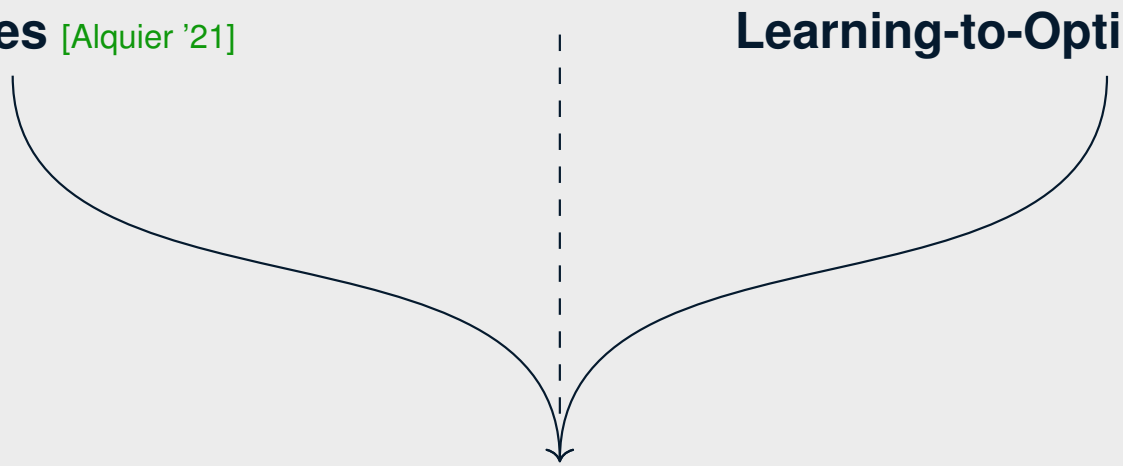
- Such bounds hold for **posterior distributions** $\rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})$:

$$\mathbb{P}\Big\{\rho^*_{\boldsymbol{\theta}_{[N]}}[\mathcal{R}(\boldsymbol{\alpha})] \leq \inf_{\rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})} \rho[\hat{\mathcal{R}}(\boldsymbol{\alpha}, \boldsymbol{\theta}_{[N]})] + K(\rho, N, \epsilon)\Big\} \geq 1 - \epsilon\,,$$

where $\mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}})$ denotes some class of (probability) measures on $\mathcal{H}$ that satisfy a certain property w.r.t. the **prior distribution** $\mathbb{P}_{\boldsymbol{\alpha}}$.

This is a naming convention! Not to be confused with prior and posterior in Bayesian analysis, which are linked by a likelihood.

For good reviews of two long lines of work see...



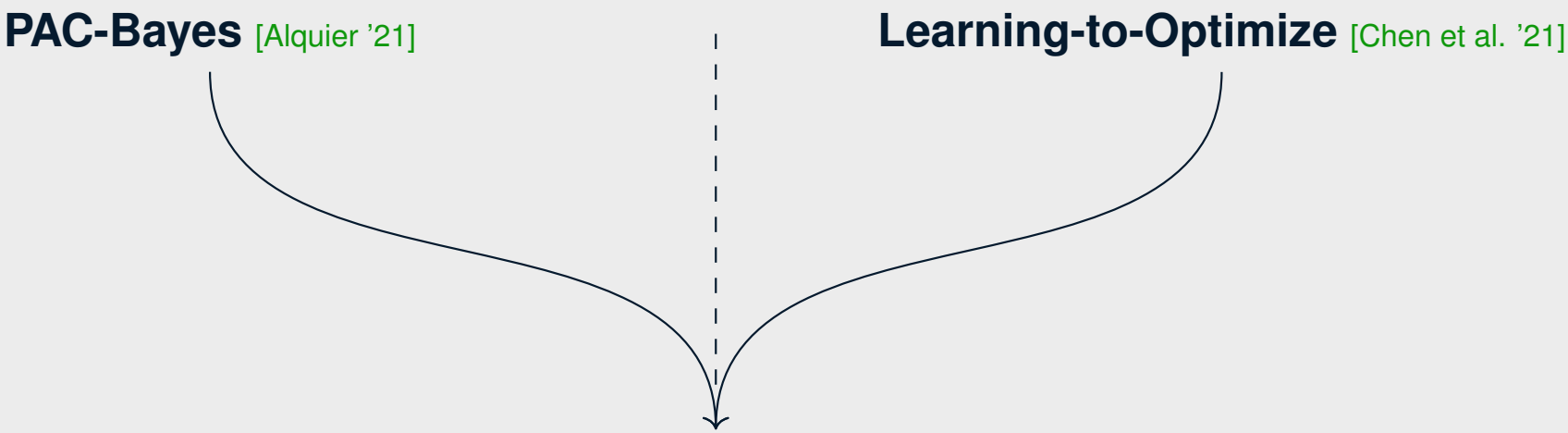
PAC-Bayesian Learning of Optimization Algorithms
[Sucker, O. '22], [Sucker, Fadili, O. '24]

A Markovian Model for Learning-to-Optimize
[Sucker, O. '24]

A Generalization Result for Convergence in Learning-to-Optimize
[Sucker, O. '24]

- [Alquier '21]: "User-friendly introduction to PAC-Bayes bounds", Foundations and Trends in Machine Learning 17(2):174–303, 2024 (arXiv 2021).
- [Chen et al. '22]: "Learning to optimize: A primer and a benchmark", Journal of Machine Learning Research 23(1):8562–8620, 2022 (arXiv 2021).

For good reviews of two long lines of work see...



- [Alquier '21]: "User-friendly introduction to PAC-Bayes bounds", Foundations and Trends in Machine Learning 17(2):174–303, 2024 (arXiv 2021).
- [Chen et al. '22]: "Learning to optimize: A primer and a benchmark", Journal of Machine Learning Research 23(1):8562–8620, 2022 (arXiv 2021).

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}), \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

  $$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \,|\, x^{(t)}_{\alpha,\boldsymbol{\theta}} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\boldsymbol{\theta}})_{t \in \mathbb{N}}.$$

- For best **expected case complexity** w.r.t. choice of algorithm $\mathcal{A}$, solve

  $$\min_{\alpha \in \mathcal{H}} \Big(\quad \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{x} | \boldsymbol{\alpha}} \{\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x})\} \quad \Big).$$

Using the **expectation is intractable** in general!
$\rightsquigarrow$ resolve using PAC-Bayesian bounds

Probabilistic Formulation of the Optimization Procedure

- Random parametric optimization problem with measurable $\ell : \mathbb{R}^d \times \Theta \to \mathbb{R}_{\geq 0}$:

  $$\min_{x \in \mathbb{R}^d} \ell(x, \boldsymbol{\theta}), \quad \text{random variable } \boldsymbol{\theta} : (\Omega, \mathfrak{A}, \mathbb{P}) \to \Theta \text{ defines the } \textbf{class of problems}.$$

- Use a parametric (iterative) optimization algorithm with $t = 0, 1, 2, \ldots$:

  $$\mathcal{A} : \mathcal{H} \times \Theta \times \mathbb{R}^d \to \mathbb{R}^d, \quad (\alpha, \theta, x^{(t)}) \mapsto \mathcal{A}(\alpha, \theta, x^{(t)}) =: x^{(t+1)}$$

- Stop the algorithm, if $x^{(t)}$ satisfies a stopping criterion $\mathsf{C}_\varepsilon(\boldsymbol{\theta})$ and define stopping time

  $$\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \in \mathbb{N} \mid x^{(t)}_{\alpha,\theta} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}, \quad \text{where } \boldsymbol{x} := (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}}.$$

- For best **expected case complexity** w.r.t. choice of algorithm $\mathcal{A}$, solve

  $$\min_{\alpha \in \mathcal{H}} \Big(\quad \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{x} | \boldsymbol{\alpha}} \{\tau_\alpha(\boldsymbol{\theta}, \boldsymbol{x})\} \quad \Big).$$

Using the **expectation is intractable** in general!
$\rightsquigarrow$ resolve using PAC-Bayesian bounds

- **Stopping time needs access to the full trajectory** $\boldsymbol{x} = (x^{(t)}_{\alpha,\theta})_{t \in \mathbb{N}}$
- **Expectation requires the conditional distribution of** $\boldsymbol{x}$ **given** (α, θ)**!**

$\rightsquigarrow$ model the distribution over algorithms/trajectories.

From randomness of the algorithm to the distribution over trajectories

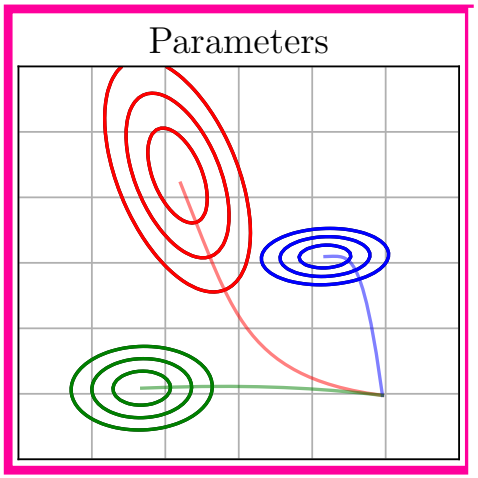
(Iterative) Optimization Algorithm:

$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)})$$

From randomness of the algorithm to the distribution over trajectories

(Iterative) Optimization Algorithm:

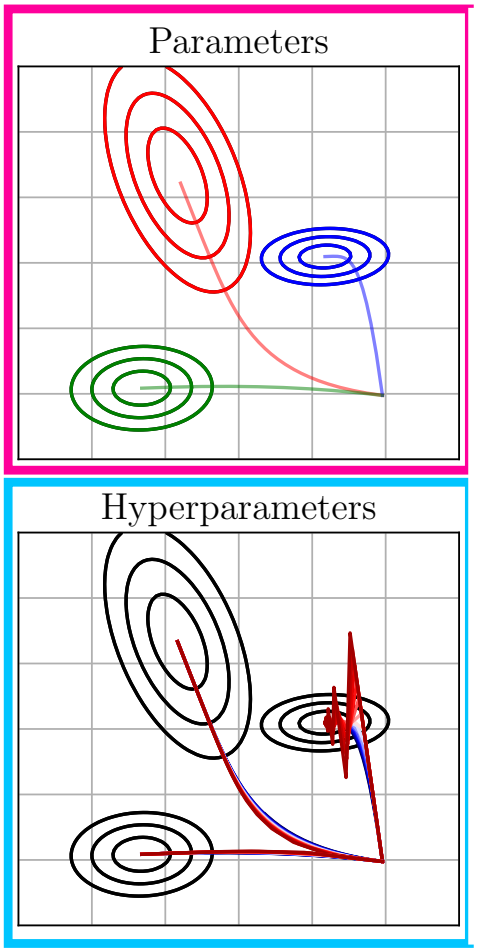
$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)})$$



From randomness of the algorithm to the distribution over trajectories

(Iterative) Optimization Algorithm:

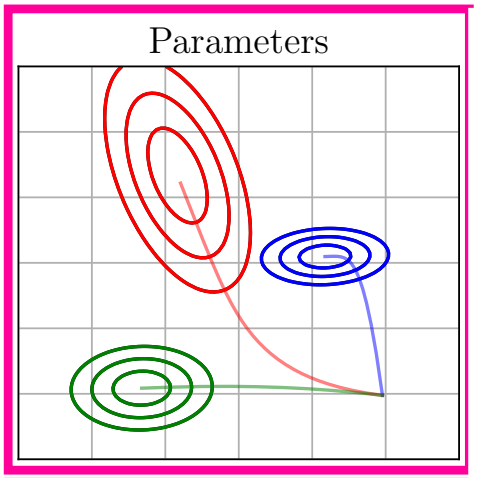
$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)})$$

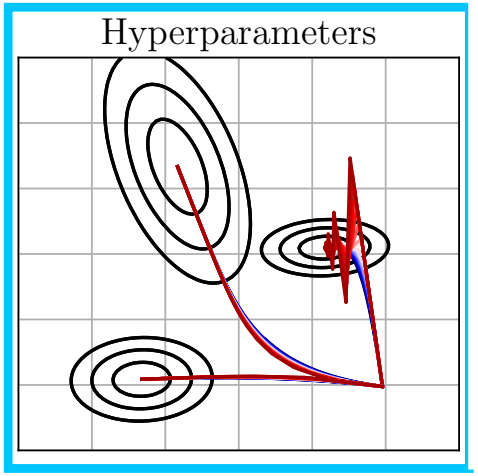


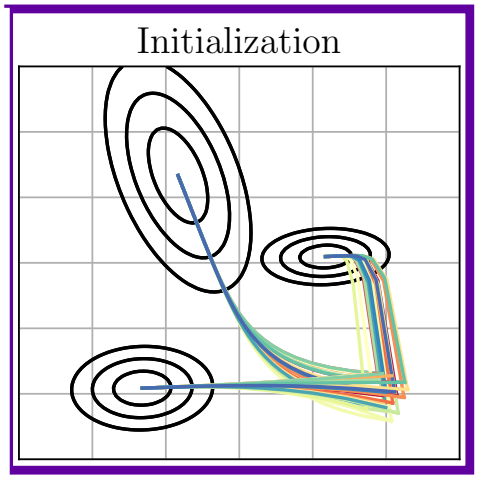
From randomness of the algorithm to the distribution over trajectories

(Iterative) Optimization Algorithm:

$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)})$$



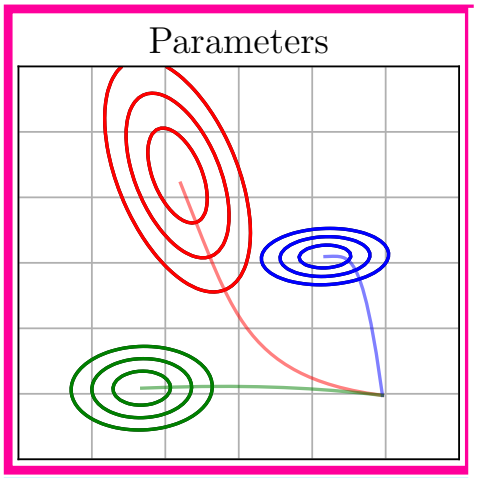


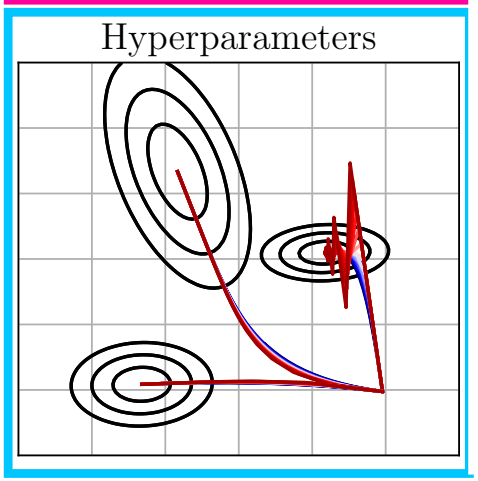


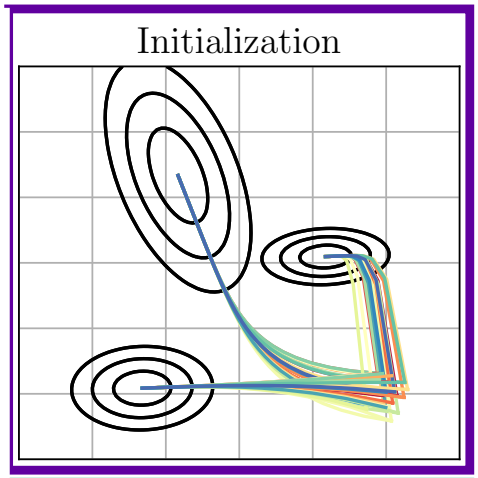
From randomness of the algorithm to the distribution over trajectories

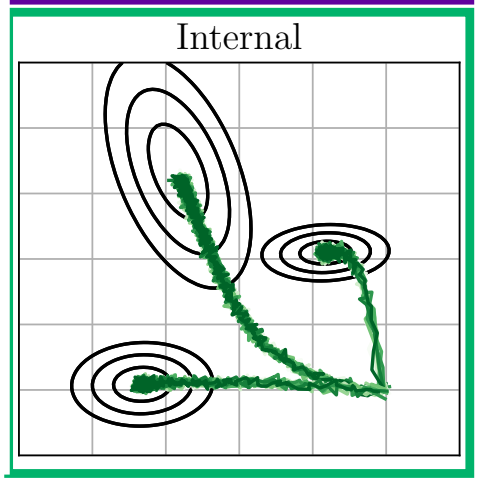
(Iterative) Optimization Algorithm: $\eta^{(t+1)}$ used for stochastic algorithms

$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)}, \eta^{(t+1)})$$





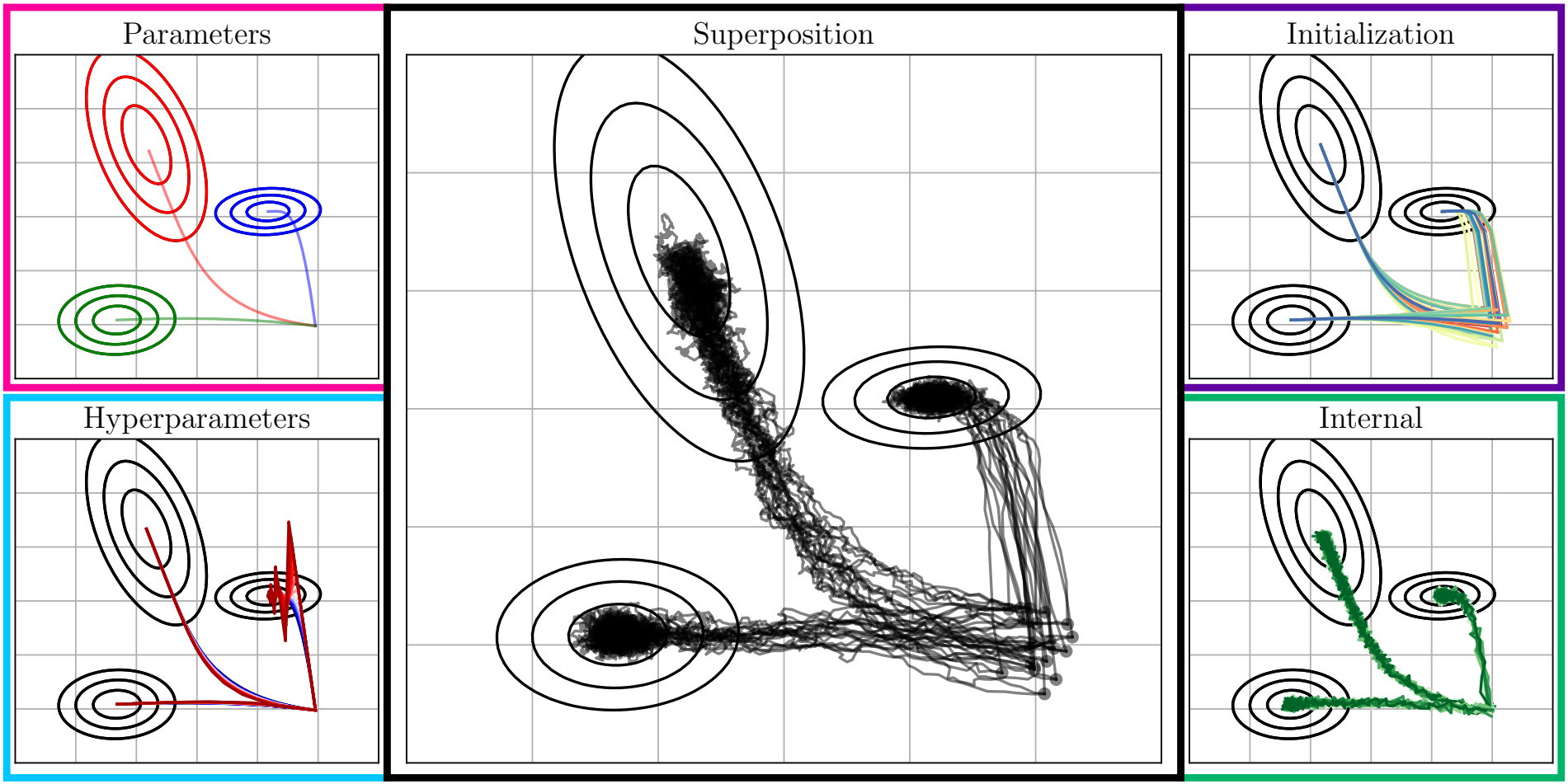




From randomness of the algorithm to the distribution over trajectories

(Iterative) Optimization Algorithm: $\mathcal{A}$ describes a discrete-time stochastic process

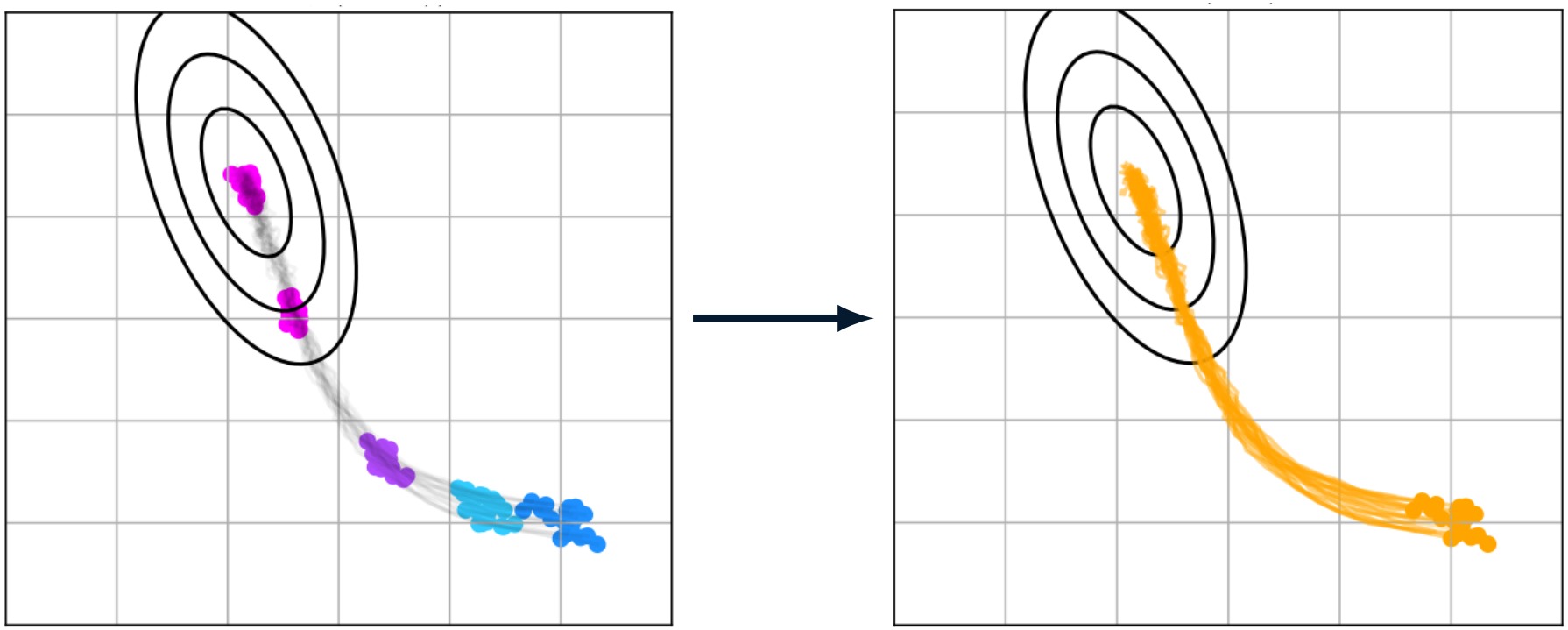
$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)}, \eta^{(t+1)})$$



From randomness of the algorithm to the distribution over trajectories

$$x^{(t+1)} = \mathcal{A}(\alpha, \theta, x^{(t)}, \eta^{(t+1)}) \xrightarrow{\text{Theorem}} \mathbb{P}_{\boldsymbol{x}|\boldsymbol{\alpha},\boldsymbol{\theta}}\{\mathbf{B}\},\ \mathbf{B} \subset (\mathbb{R}^d)^{\mathbb{N}}$$

exists regular conditional distribution over trajectories $\boldsymbol{x}$ given $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$



distribution over iteration maps

distribution over trajectories

PAC-Bayesian Bounds: Generalization of the Stopping Time

Use the measurable stopping time $\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \leq t_{\max} \mid x^{(t)}_{\boldsymbol{\alpha},\boldsymbol{\theta}} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}$ and define

average expected stopping time: $\bar{\tau}_{\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha}} \{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$

n-th expected stopping time: $\bar{\tau}_{n,\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha},\boldsymbol{\theta}_n} \{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$.

PAC-Bayesian Bounds: Generalization of the Stopping Time

Use the measurable stopping time $\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \leq t_{\max} \mid x^{(t)}_{\boldsymbol{\alpha},\boldsymbol{\theta}} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}$ and define

average expected stopping time: $\bar{\tau}_{\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha}}\{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$

n-th expected stopping time: $\bar{\tau}_{n,\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha},\boldsymbol{\theta}_n}\{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$.

Theorem: For every $\lambda \in (0, \infty)$ and $\epsilon > 0$ it holds that:

$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\Big\{\forall \rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}}) \;:\; \rho[\bar{\tau}_{\boldsymbol{\alpha}}] \leq \frac{1}{N}\sum_{n=1}^{N} \rho[\bar{\tau}_{n,\boldsymbol{\alpha}}] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_{\boldsymbol{\alpha}}) + \frac{\lambda^2}{2N} t_{\max}^2 - \log(\epsilon)}{\lambda}\Big\} \geq 1 - \epsilon\,.$$

Reminder: *(Abstract PAC-Bayesian bound from before)*

$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\big\{\forall \rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}}):\; \rho^*_{\boldsymbol{\theta}_{[N]}}[\mathcal{R}(\boldsymbol{\alpha})] \leq \rho[\hat{\mathcal{R}}(\boldsymbol{\alpha}, \boldsymbol{\theta}_{[N]})] + K(\rho, N, \epsilon)\big\} \geq 1 - \epsilon\,,$$

PAC-Bayesian Bounds: Generalization of the Stopping Time

Use the measurable stopping time $\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x}) := \inf\{t \leq t_{\max} \,|\, x^{(t)}_{\boldsymbol{\alpha},\boldsymbol{\theta}} \in \mathsf{C}_\varepsilon(\boldsymbol{\theta})\}$ and define

average expected stopping time: $\bar{\tau}_{\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha}} \{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$

n-th expected stopping time: $\bar{\tau}_{n,\boldsymbol{\alpha}} := \mathbb{E}_{(\boldsymbol{\theta},\boldsymbol{x})|\boldsymbol{\alpha},\boldsymbol{\theta}_n} \{\tau_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, \boldsymbol{x})\}$.

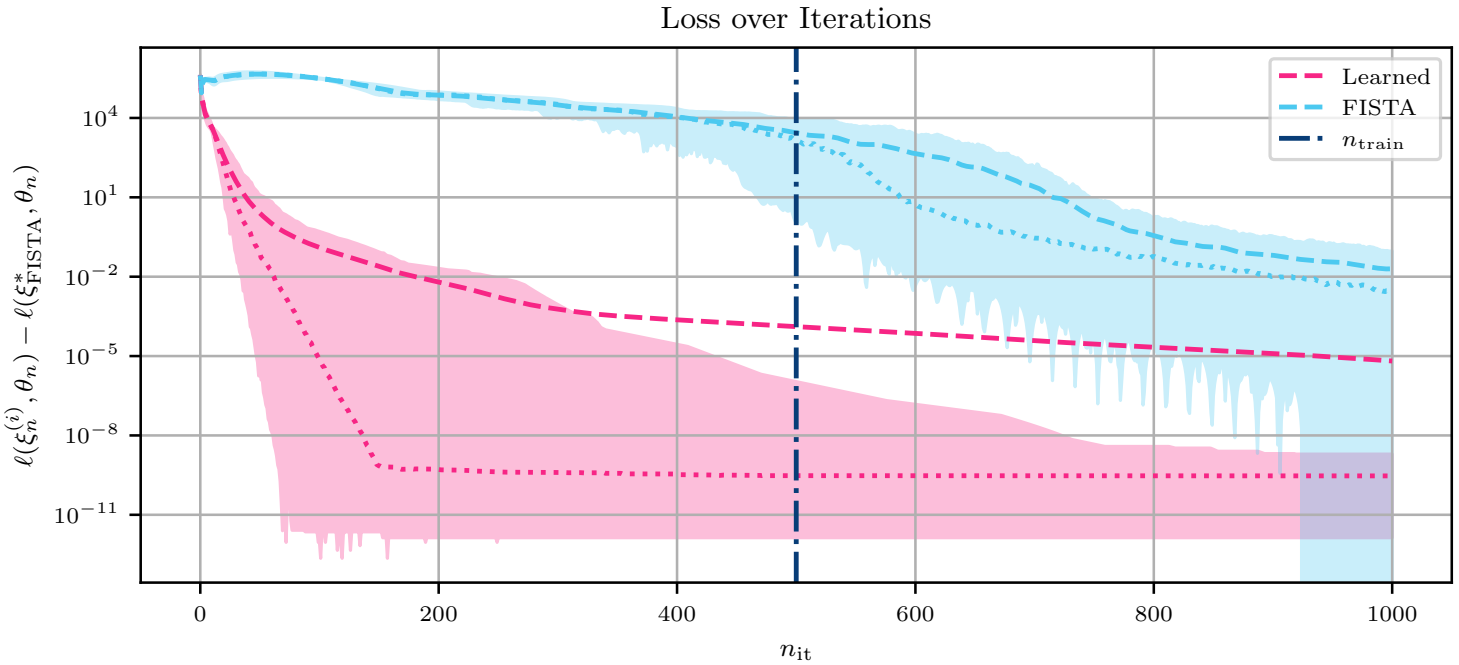
Theorem: For every $\lambda \in (0, \infty)$ and $\epsilon > 0$ it holds that:

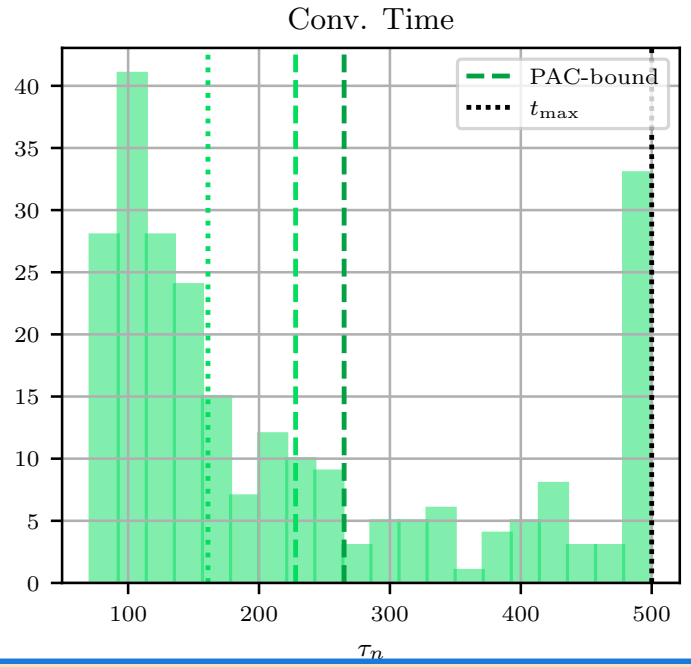
$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\Big\{\forall \rho \in \mathcal{M}(\mathbb{P}_{\boldsymbol{\alpha}}) \;:\; \rho[\bar{\tau}_{\boldsymbol{\alpha}}] \leq \frac{1}{N}\sum_{n=1}^{N} \rho[\bar{\tau}_{n,\boldsymbol{\alpha}}] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_{\boldsymbol{\alpha}}) + \frac{\lambda^2}{2N} t_{\max}^2 - \log(\epsilon)}{\lambda}\Big\} \geq 1 - \epsilon\,.$$

Training / Learning the algorithm: $=$ optimize the right hand side

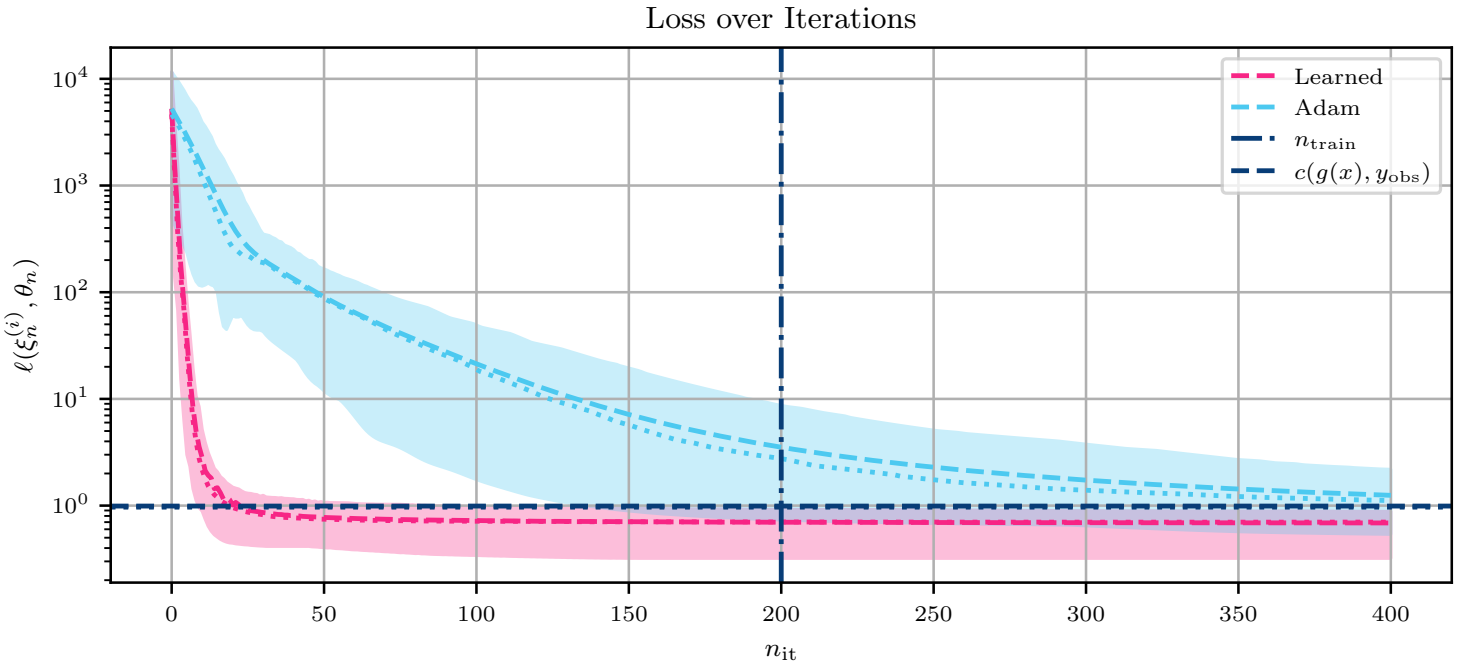
- Closed form solution of right hand side. Only optimize w.r.t. λ.
- Note that stopping times can be translated to **convergence rates**. *(easier to train with)*
- **Criticism:** Prior needs to be constructed. *(usually trained on independent data)*

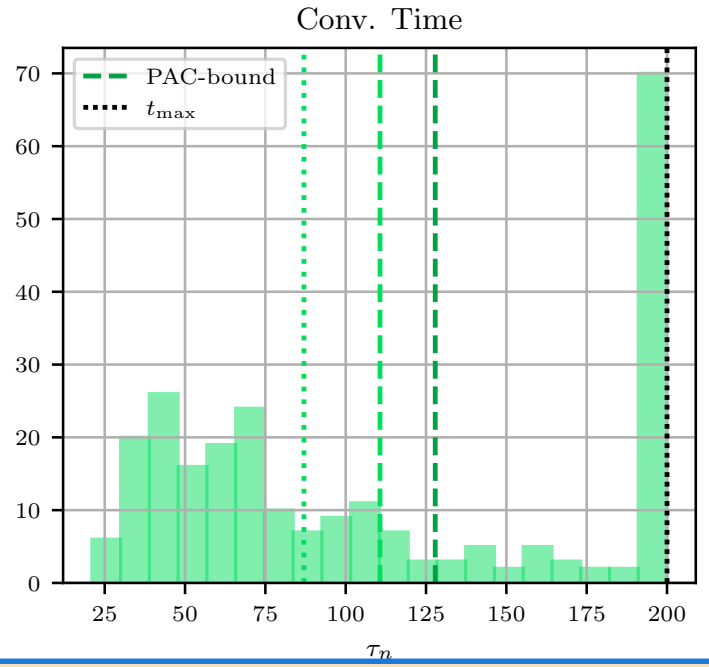
Numerical Result: Lasso Problem



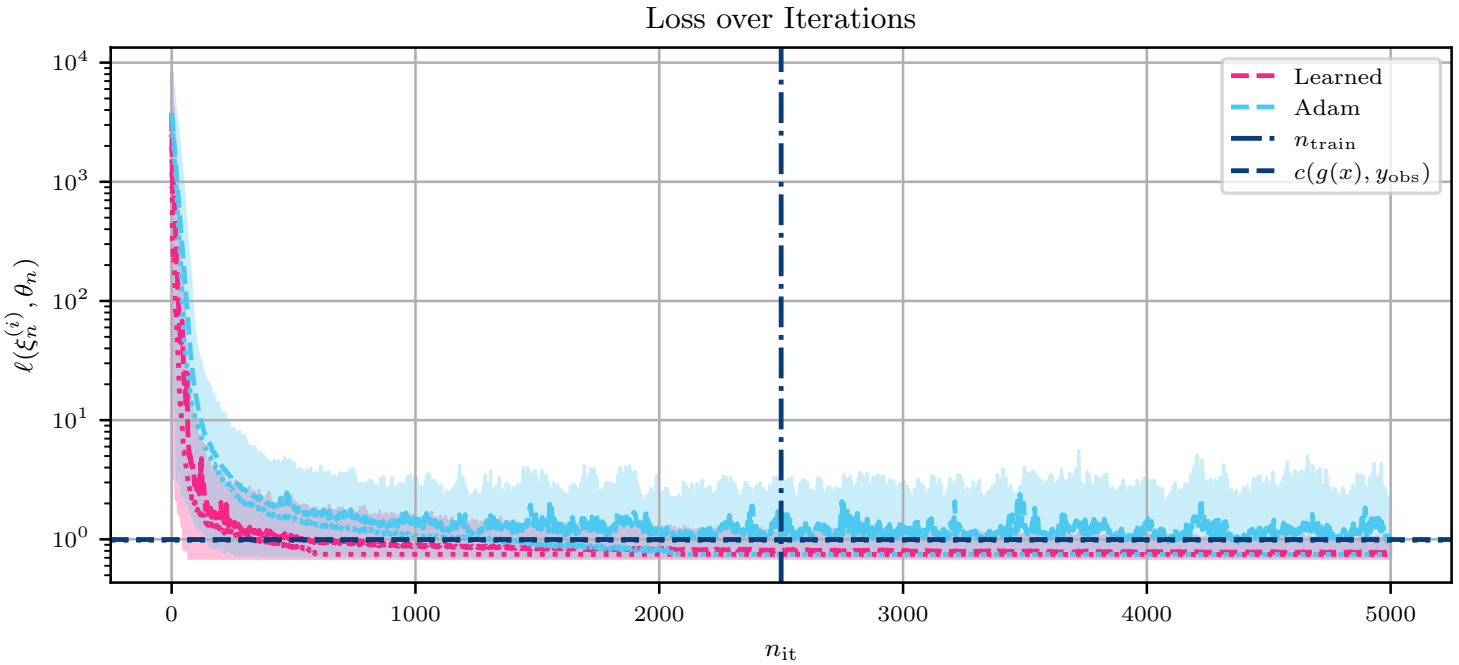


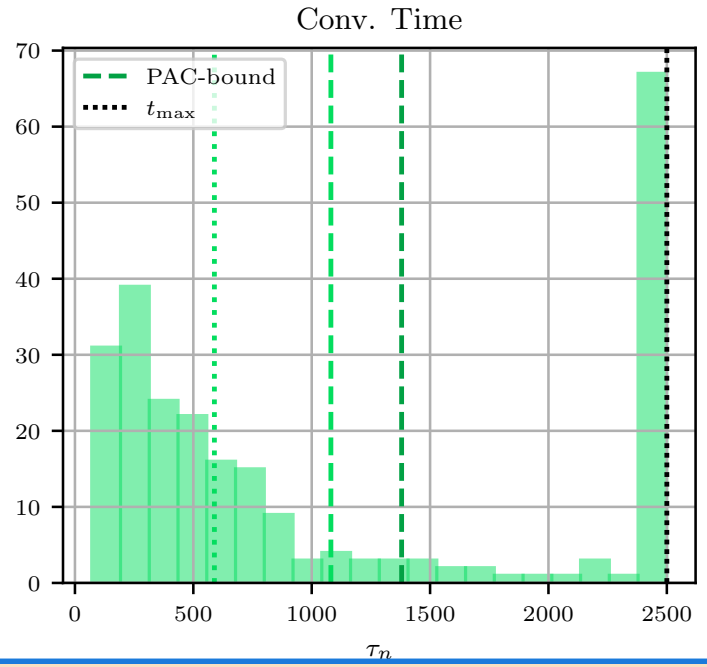
Numerical Results: Training a neural network for regression





Numerical Results: same problem but with mini-batch update





Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Classical strategy:

- Show that the algorithm (or the trajectory $\boldsymbol{x}$) has property a and b.

Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Classical strategy:

- Show that the algorithm (or the trajectory $\boldsymbol{x}$) has property a and b.
- Prove that $a \wedge b$ imply convergence c:

$$\boldsymbol{x} \text{ satisfies } a \wedge b \quad \Longrightarrow \quad \boldsymbol{x} \text{ satisfies } c$$

Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Classical strategy:

- Show that the algorithm (or the trajectory $\boldsymbol{x}$) has property a and b.
- Prove that $a \wedge b$ imply convergence c:

$$\boldsymbol{x} \text{ satisfies } a \wedge b \quad \Longrightarrow \quad \boldsymbol{x} \text{ satisfies } c$$

- **However**, we can only say that $\boldsymbol{x}$ satisfies a property with some probability.

Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Classical strategy:

- Show that the algorithm (or the trajectory $\boldsymbol{x}$) has property a and b.
- Prove that $a \wedge b$ imply convergence c:

$$\boldsymbol{x} \text{ satisfies } a \wedge b \quad \Longrightarrow \quad \boldsymbol{x} \text{ satisfies } c$$

- **However**, we can only say that $\boldsymbol{x}$ satisfies a property with some probability.

Probabilistic / measure theoretic strategy:

- Define $A := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } a\}$, $B := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } b\}$, $C := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } c\}$.
- Then the question of “implication” becomes an “inclusion”

$$A \cap B = \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } a \wedge b\} \subset C$$

Convergence of Learnable Algorithms

How to prove convergence of learned algorithms to a critical point ?

Classical strategy:

- Show that the algorithm (or the trajectory $\boldsymbol{x}$) has property a and b.
- Prove that $a \wedge b$ imply convergence c:

$$\boldsymbol{x} \text{ satisfies } a \wedge b \quad \Longrightarrow \quad \boldsymbol{x} \text{ satisfies } c$$

- **However**, we can only say that $\boldsymbol{x}$ satisfies a property with some probability.

Probabilistic / measure theoretic strategy:

- Define $A := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } a\}$, $B := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } b\}$, $C := \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } c\}$.
- Then the question of "implication" becomes an "inclusion"

$$A \cap B = \{\boldsymbol{x} \,|\, \boldsymbol{x} \text{ satisfies } a \wedge b\} \subset C$$

- Given a probability measure μ over trajectories $\boldsymbol{x}$, we can conclude

$$\mu\{A \cap B\} \leq \mu\{C\}\,.$$

Convergence of Abstract Descent Methods

Theorem: [[Attouch et al. 2013]] Let $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ be a proper lsc Kurdyka-Łojasiewicz function that is continuous on $\operatorname{dom} f$ and $\inf f > -\infty$. Suppose there exists $a, b > 0$ such that $(x^{(t)})_{t\in\mathbb{N}}$ satisfies

- **Sufficient decrease condition:**

$$f(x^{(t+1)}) + a\|x^{(t+1)} - x^{(t)}\|^2 \leq f(x^{(t)})\,, \quad \forall t \in \mathbb{N}\,;$$

- **Relative error condition:**

$$\exists v^{(t+1)} \in \partial f(x^{(t+1)})\colon \ \|v^{(t+1)}\| \leq b\|x^{(t+1)} - x^{(t)}\|\,, \quad \forall t \in \mathbb{N}\,;$$

- **Boundedness condition:**

$$(x^{(t)})_{t\in\mathbb{N}} \text{ is bounded}\,.$$

Then, $(x^{(t)})_{t\in\mathbb{N}}$ converges to a critical point of f.

Abstract Descent Methods from “specific properties”

- Collection of sequences that **converge to a stationary point:**

$$\mathrm{A}_{\mathrm{conv}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \to \tilde{x} \text{ as } t \to \infty \text{ and } 0 \in \partial_x \ell(\tilde{x}, \boldsymbol{\theta})\}$$

Abstract Descent Methods from "specific properties"

- Collection of sequences that **converge to a stationary point:**

$$\mathrm{A}_{\mathrm{conv}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \to \tilde{x} \text{ as } t \to \infty \text{ and } 0 \in \partial_x \ell(\tilde{x}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **sufficient decrease condition:**

$$\mathrm{A}_{\mathrm{desc}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom} \ell \text{ for all } t \in \mathbb{N} \text{ and}$$
$$\exists a > 0 \colon \forall t \in \mathbb{N} \colon \ \ell(x^{(t+1)}, \boldsymbol{\theta}) + a\|x^{(t+1)} - x^{(t)}\|^2 \leq \ell(x^{(t)}, \boldsymbol{\theta})\}$$

Abstract Descent Methods from "specific properties"

- Collection of sequences that **converge to a stationary point:**

$$\mathsf{A}_{\mathrm{conv}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \to \tilde{x} \text{ as } t \to \infty \text{ and } 0 \in \partial_x \ell(\tilde{x}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **sufficient decrease condition:**

$$\mathsf{A}_{\mathrm{desc}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and}$$
$$\exists a > 0 \colon \forall t \in \mathbb{N} \colon\ \ell(x^{(t+1)}, \boldsymbol{\theta}) + a\|x^{(t+1)} - x^{(t)}\|^2 \leq \ell(x^{(t)}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **relative error condition:**

$$\mathsf{A}_{\mathrm{rerr}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and}$$
$$\exists b > 0 \colon \forall t \in \mathbb{N} \colon\ \|v(x^{(t+1)}, \boldsymbol{\theta})\| \leq b\|x^{(t+1)} - x^{(t)}\|\}$$

Abstract Descent Methods from "specific properties"

- Collection of sequences that **converge to a stationary point:**

$$\mathrm{A}_{\mathrm{conv}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \to \tilde{x} \text{ as } t \to \infty \text{ and } 0 \in \partial_x \ell(\tilde{x}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **sufficient decrease condition:**

$$\mathrm{A}_{\mathrm{desc}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and}$$
$$\exists a > 0 \colon \forall t \in \mathbb{N} \colon\ \ell(x^{(t+1)}, \boldsymbol{\theta}) + a\|x^{(t+1)} - x^{(t)}\|^2 \leq \ell(x^{(t)}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **relative error condition:**

$$\mathrm{A}_{\mathrm{rerr}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and}$$
$$\exists b > 0 \colon \forall t \in \mathbb{N} \colon\ \|v(x^{(t+1)}, \boldsymbol{\theta})\| \leq b\|x^{(t+1)} - x^{(t)}\|\}$$

- Collection of sequences that are **bounded:**

$$\mathrm{A}_{\mathrm{bd}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, (x^{(t)})_{t\in\mathbb{N}} \text{ is bounded}\}$$

Abstract Descent Methods from "specific properties"

- Collection of sequences that **converge to a stationary point:**

$$\mathsf{A}_{\mathrm{conv}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \to \tilde{x} \text{ as } t \to \infty \text{ and } 0 \in \partial_x \ell(\tilde{x}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **sufficient decrease condition:**

$$\mathsf{A}_{\mathrm{desc}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and } \exists a > 0 \colon \forall t \in \mathbb{N} \colon\ \ell(x^{(t+1)}, \boldsymbol{\theta}) + a\|x^{(t+1)} - x^{(t)}\|^2 \leq \ell(x^{(t)}, \boldsymbol{\theta})\}$$

- Collection of sequences that satisfy the **relative error condition:** ($(x, \boldsymbol{\theta}) \mapsto v(x, \boldsymbol{\theta})$ is a measurable selection of $\partial_x \ell(x, \boldsymbol{\theta})$:)

$$\mathsf{A}_{\mathrm{rerr}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, x^{(t)} \in \operatorname{dom}\ell \text{ for all } t \in \mathbb{N} \text{ and } \exists b > 0 \colon \forall t \in \mathbb{N} \colon\ \|v(x^{(t+1)}, \boldsymbol{\theta})\| \leq b\|x^{(t+1)} - x^{(t)}\|\}$$

- Collection of sequences that are **bounded:**

$$\mathsf{A}_{\mathrm{bd}} := \{(\boldsymbol{\theta}, (x^{(t)})_{t\in\mathbb{N}}) \,|\, (x^{(t)})_{t\in\mathbb{N}} \text{ is bounded}\}$$

Proposition: All sets listed above are measurable and if $\ell(\cdot, \boldsymbol{\theta})$ is a Kurdyka-Łojasiewicz function for every $\boldsymbol{\theta}$, then we have that

$$\mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{rerr}} \cap \mathsf{A}_{\mathrm{bd}} \subset \mathsf{A}_{\mathrm{conv}} .$$

Convergence of Learned Algorithms

- Recall: $\mathsf{A} := \mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{rerr}} \cap \mathsf{A}_{\mathrm{bd}} \subset \mathsf{A}_{\mathrm{conv}}$
- Using monotonicity

$$\mathbb{P}_{(\boldsymbol{\theta},x)|\boldsymbol{\alpha}}\{\mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{rerr}} \cap \mathsf{A}_{\mathrm{bd}}\} \leq \mathbb{P}_{(\boldsymbol{\theta},x)|\boldsymbol{\alpha}}\{\mathsf{A}_{\mathrm{conv}}\}\,,$$

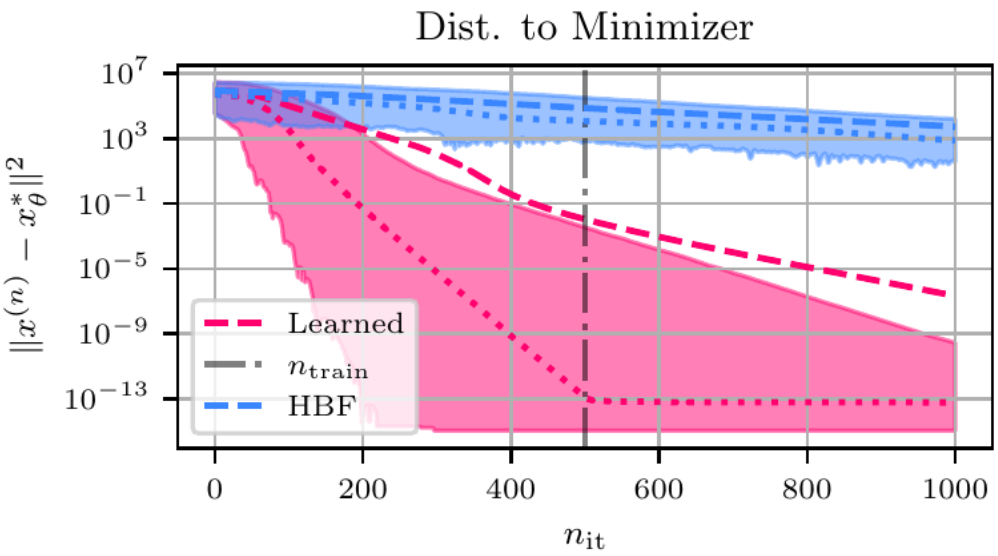
we obtain **generalization of the probability to converge**:

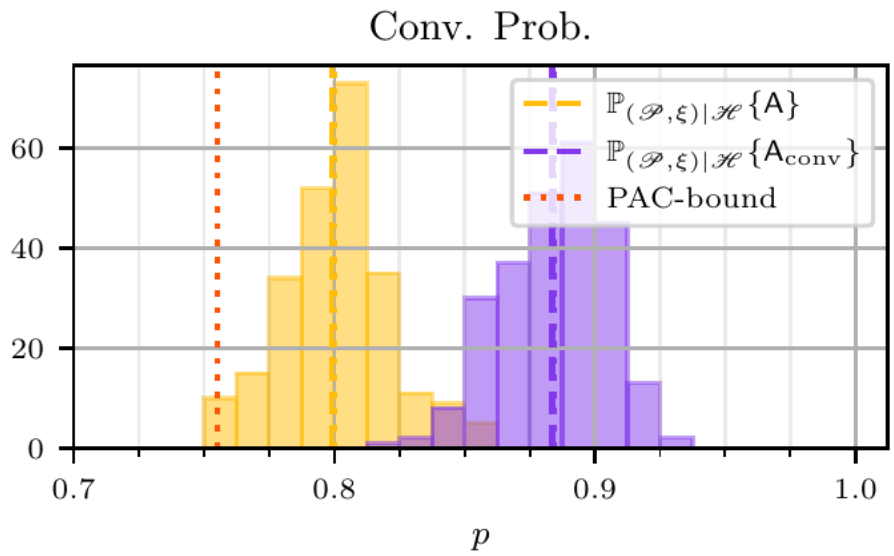
Theorem: [Catoni-like PAC bound] Let $\ell(\cdot,\boldsymbol{\theta})$ is a Kurdyka-Łojasiewicz function for every $\boldsymbol{\theta}$. Then, for any $\lambda \in (0,\infty)$, it holds that:

$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\Big\{\forall \rho \in \mathcal{M}\left(\mathbb{P}_{\boldsymbol{\alpha}}\right) \;:\; \rho\left[\mathbb{P}_{(\boldsymbol{\theta},x)|\boldsymbol{\alpha}}\{\mathsf{A}_{\mathrm{conv}}\}\right]$$
$$\geq 1 - \Phi^{-1}_{\frac{\lambda}{N}}\left(\frac{1}{N}\sum_{n=1}^{N}\rho\left[\mathbb{P}_{(\boldsymbol{\theta}_n,x_n)|\boldsymbol{\alpha},\boldsymbol{\theta}_n}\{\mathsf{A}^{\mathrm{c}}\}\right] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_{\boldsymbol{\alpha}}) - \log(\epsilon)}{\lambda}\right)\Big\} \geq 1-\epsilon\,.$$

Convergence of Learned Algorithms

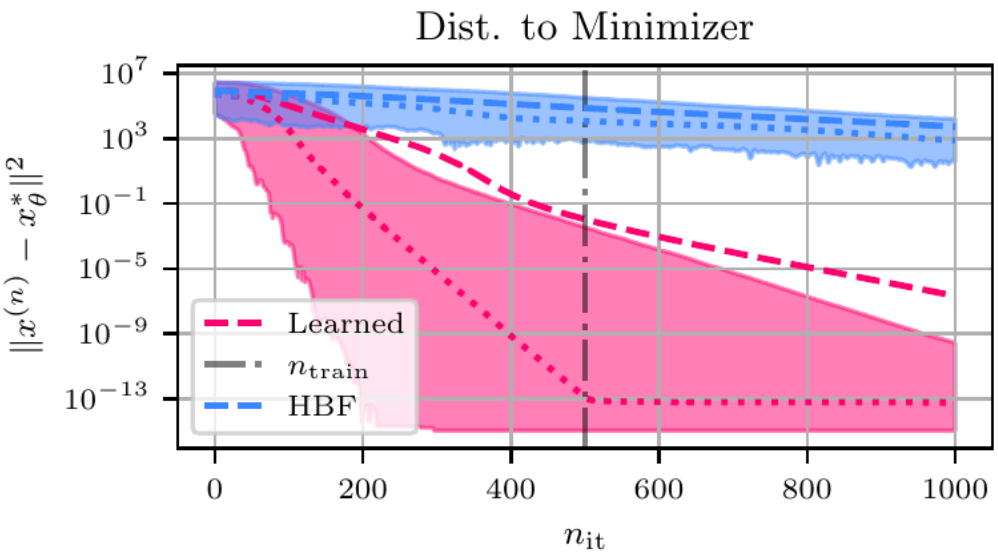
Smooth and strongly convex problem:

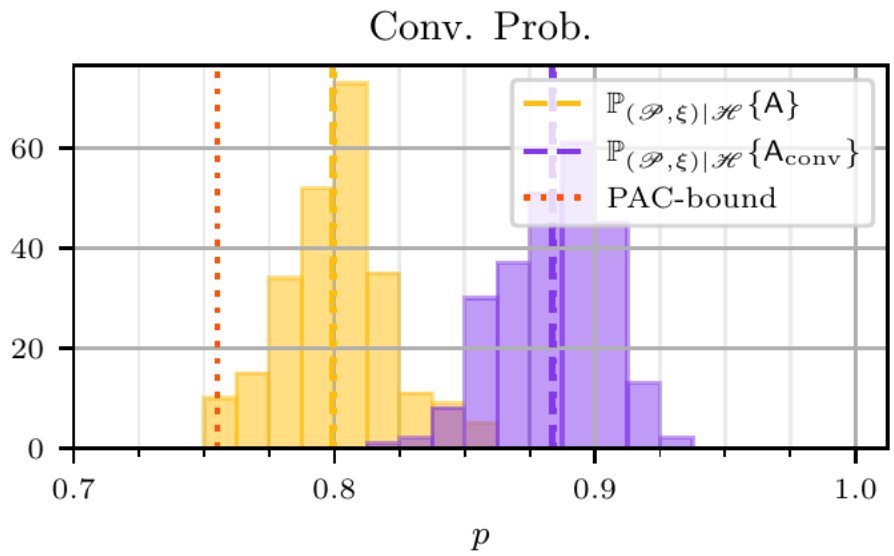




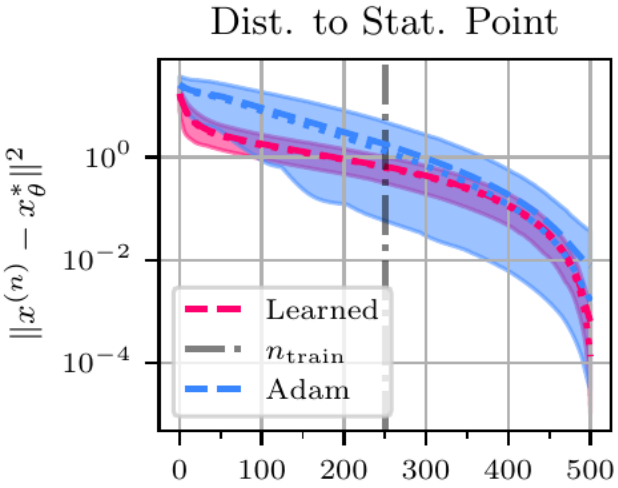
Convergence of Learned Algorithms

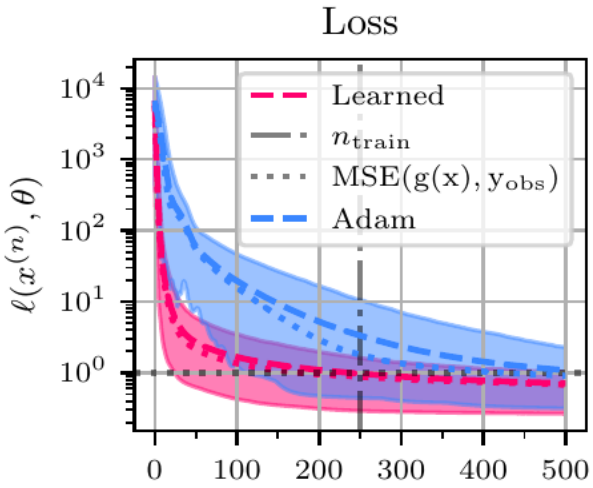
Smooth and strongly convex problem:

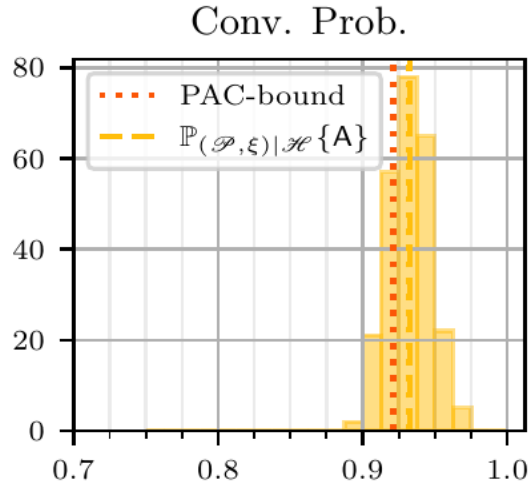




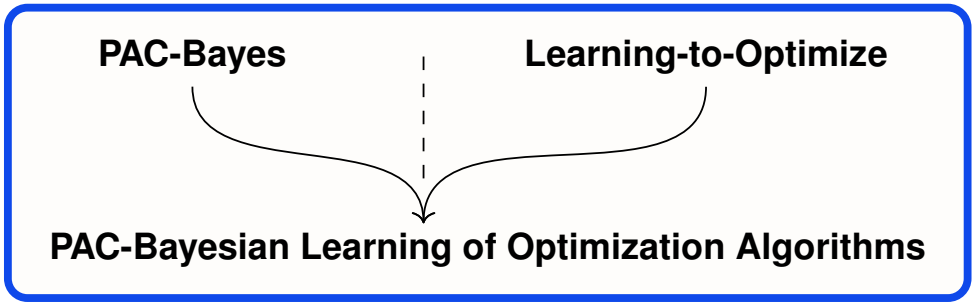
Non-smooth and non-convex problem:



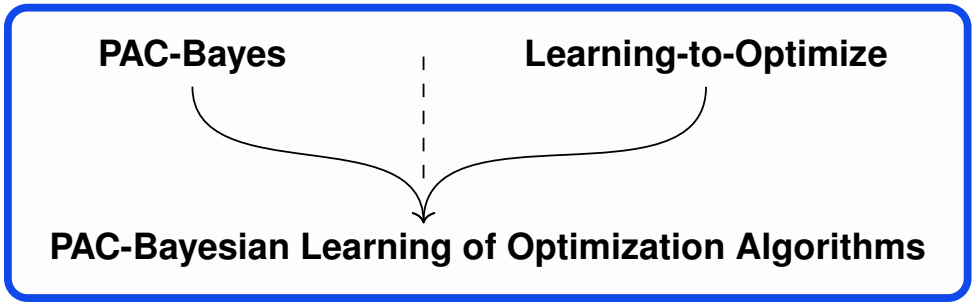




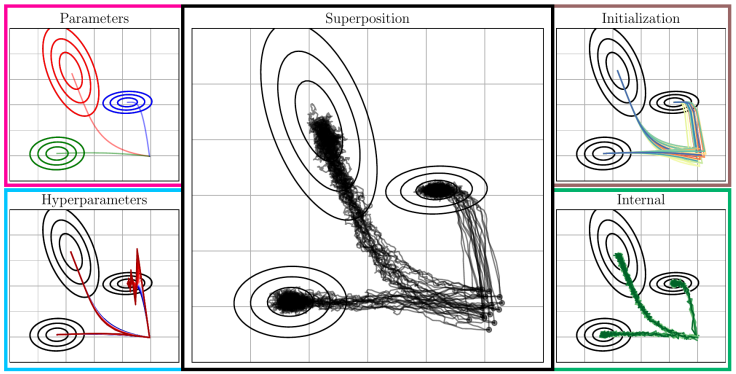
Goal: Learning of optimization algorithms with theoretical guarantees



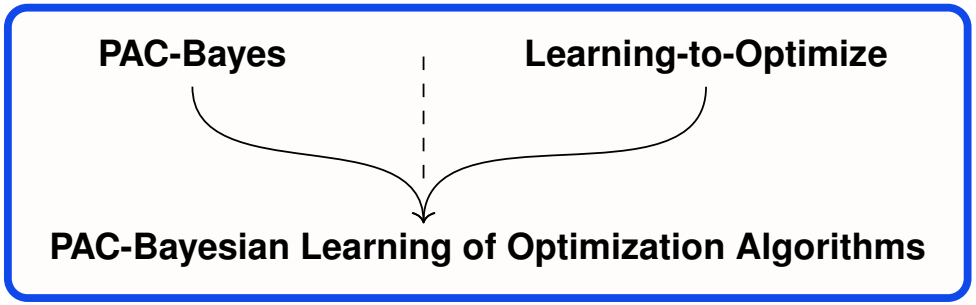
Goal: Learning of optimization algorithms with theoretical guarantees



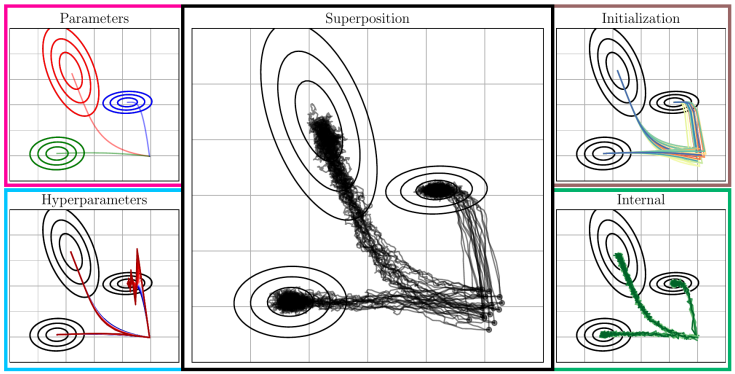
- probabilistic framework for learning algorithms



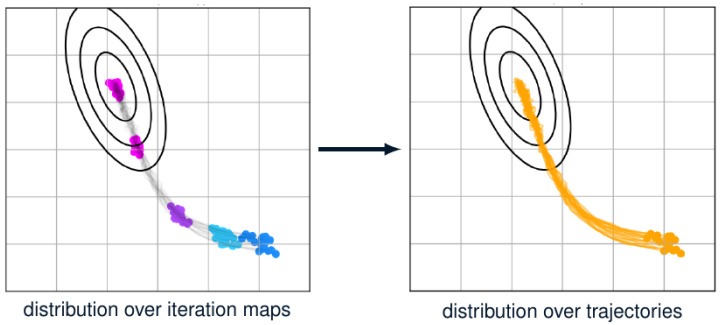
Goal: Learning of optimization algorithms with theoretical guarantees



- probabilistic framework for learning algorithms



- analysis requires distributions over trajectories

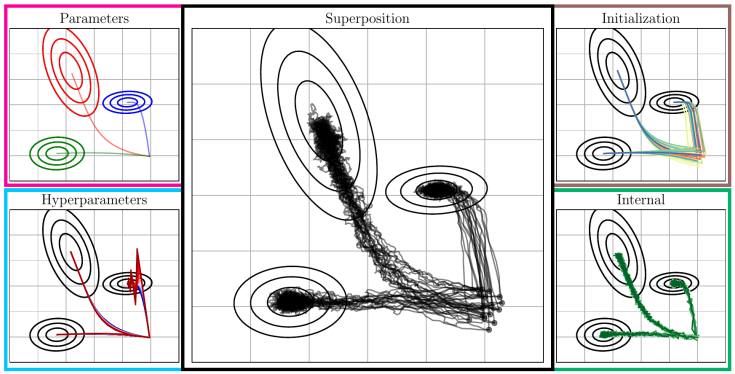


Goal: Learning of optimization algorithms with theoretical guarantees

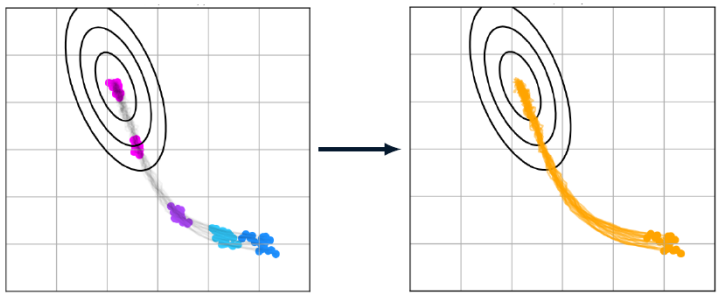
PAC-Bayes | **Learning-to-Optimize**

PAC-Bayesian Learning of Optimization Algorithms

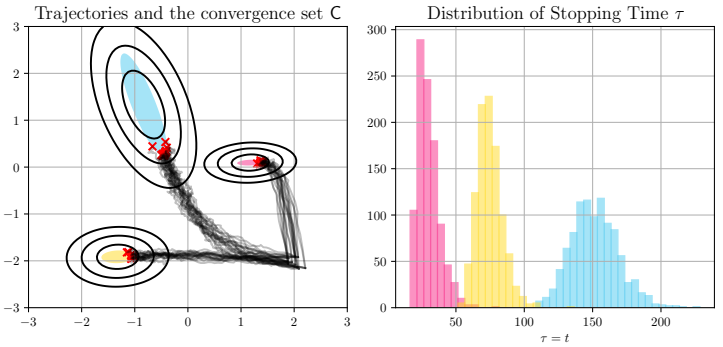
- probabilistic framework for learning algorithms



- analysis requires distributions over trajectories



- learning of **Stopping Times**



with PAC-Bayesian bounds

Corollary: For every $\lambda \in (0, \infty)$ and $\epsilon > 0$ it holds that:

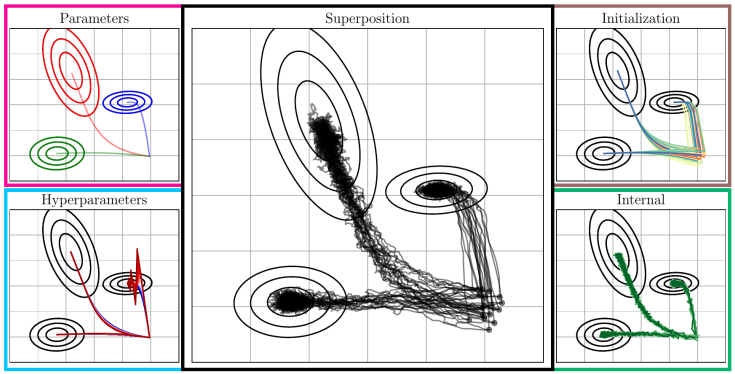
$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\Big\{\forall \rho \in \mathcal{P}(\mathbb{P}_{\boldsymbol{\alpha}}) : \rho[\bar{\tau}] \leq \frac{1}{N}\sum_{n=1}^{N} \rho[\mathbb{E}\{\tau_n \mid \boldsymbol{\alpha}, \boldsymbol{\theta}_n\}] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_{\boldsymbol{\alpha}}) + \frac{\lambda^2}{2N} t_{\max}^2 - \log(\epsilon)}{\lambda}\Big\} \geq 1 - \epsilon\,.$$

Goal: Learning of optimization algorithms with theoretical guarantees

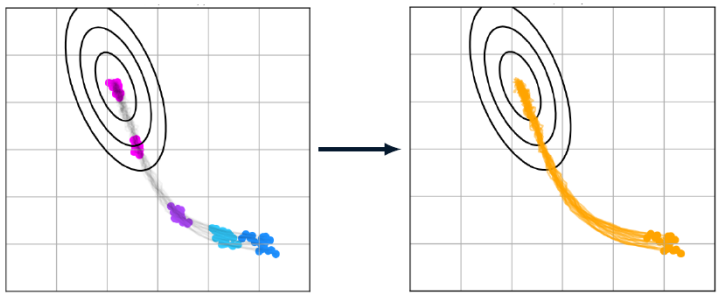
PAC-Bayes | **Learning-to-Optimize**

PAC-Bayesian Learning of Optimization Algorithms

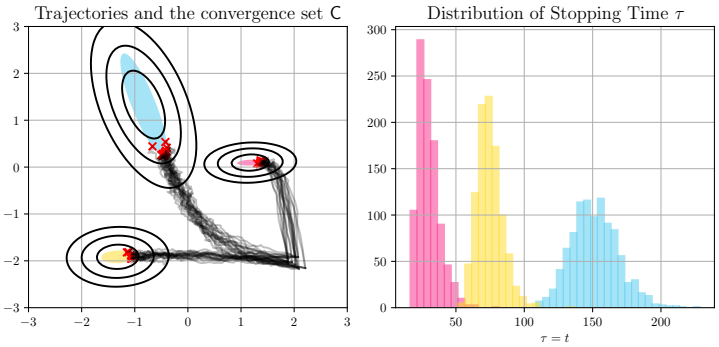
- probabilistic framework for learning algorithms



- analysis requires distributions over trajectories



- learning of **Stopping Times**



with PAC-Bayesian bounds

Corollary: For every $\lambda \in (0, \infty)$ and $\epsilon > 0$ it holds that:

$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\left\{\forall \rho \in \mathcal{P}(\mathbb{P}_\alpha) : \rho[\bar{\tau}] \leq \frac{1}{N}\sum_{n=1}^{N} \rho[\mathbb{E}\{\tau_n \mid \boldsymbol{\alpha}, \boldsymbol{\theta}_n\}] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_\alpha) + \frac{\lambda^2}{2N} t_{\max}^2 - \log(\epsilon)}{\lambda}\right\} \geq 1 - \epsilon.$$

- learning of **algorithms that converge to critical points** by combining [Attouch et al. 2013] with PAC-Bayesian bounds

Theorem: Let $\ell(\cdot, \boldsymbol{\theta})$ is a Kurdyka-Łojasiewicz function for every $\boldsymbol{\theta}$. Then, for any $\lambda \in (0, \infty)$, it holds that:

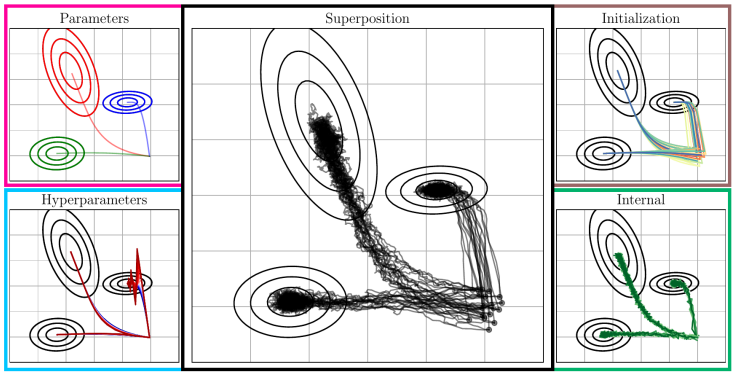
$$\mathbb{P}_{\boldsymbol{\theta}_{[N]}}\left\{\forall \rho \in \mathcal{P}(\mathbb{P}_\alpha) : \rho\left[\mathbb{P}_{(\boldsymbol{\theta},\xi)|\alpha}\{\mathsf{A}_{\mathrm{conv}}\}\right] \geq 1 - \Phi_{\frac{\lambda}{N}}^{-1}\left(\frac{1}{N}\sum_{n=1}^{N} \rho\left[\mathbb{P}_{(\boldsymbol{\theta}_n,\xi_n)|\alpha,\boldsymbol{\theta}_n}\{\mathsf{A}^{\mathrm{c}}\}\right] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_\alpha) - \log(\epsilon)}{\lambda}\right)\right\} \geq 1 - \epsilon.$$

Goal: Learning of optimization algorithms with theoretical guarantees

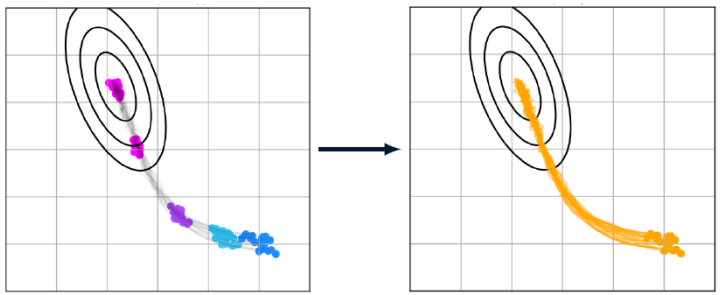
PAC-Bayes | **Learning-to-Optimize**

PAC-Bayesian Learning of Optimization Algorithms

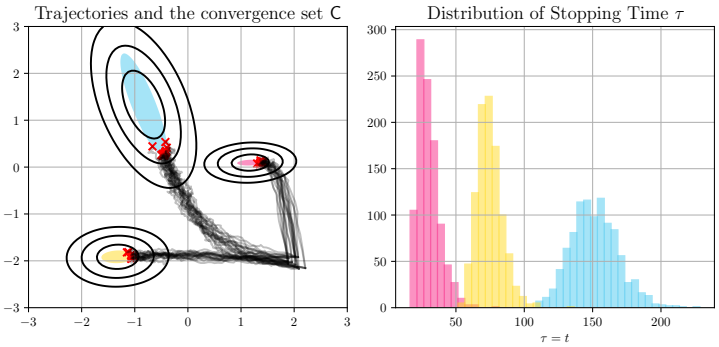
- probabilistic framework for learning algorithms



- analysis requires distributions over trajectories



- learning of **Stopping Times**



with PAC-Bayesian bounds

Corollary: For every $\lambda \in (0, \infty)$ and $\epsilon > 0$ it holds that:

$$\mathbb{P}_{\theta_{[N]}}\Big\{\forall \rho \in \mathcal{P}(\mathbb{P}_\alpha) \; : \; \rho[\bar{\tau}] \leq \frac{1}{N}\sum_{n=1}^{N} \rho[\mathbb{E}\{\tau_n \mid \alpha, \theta_n\}] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_\alpha) + \frac{\lambda^2}{2N} t_{\max}^2 - \log(\epsilon)}{\lambda}\Big\} \geq 1 - \epsilon\,.$$

- learning of **algorithms that converge to critical points** by combining [Attouch et al. 2013] with PAC-Bayesian bounds

Theorem: Let $\ell(\cdot, \theta)$ is a Kurdyka-Łojasiewicz function for every θ. Then, for any $\lambda \in (0, \infty)$, it holds that:

$$\mathbb{P}_{\theta_{[N]}}\Big\{\forall \rho \in \mathcal{P}(\mathbb{P}_\alpha) \; : \; \rho\left[\mathbb{P}_{(\theta,\xi)|\alpha}\{\mathsf{A}_{\mathrm{conv}}\}\right] \geq 1 - \Phi_{\frac{\lambda}{N}}^{-1}\left(\frac{1}{N}\sum_{n=1}^{N} \rho\left[\mathbb{P}_{(\theta_n,\xi_n)|\alpha,\theta_n}\{\mathsf{A}^{\mathrm{c}}\}\right] + \frac{D_{\mathrm{KL}}(\rho \,\|\, \mathbb{P}_\alpha) - \log(\epsilon)}{\lambda}\right)\Big\} \geq 1 - \epsilon\,.$$

What you observe is what you get !