PAC-Bayesian Learning of Optimization Algorithms

Peter Ochs

Mathematical Optimization for Data Science
Saarland University

— 07.09.2023 —

joint work: Michael Sucker

Inverse Problems are often Modelled as an Optimization Problem

Example 1:

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

Inverse Problems are often Modelled as an Optimization Problem

Example 1:

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

How do we solve the problem?

Inverse Problems are often Modelled as an Optimization Problem

Example 1:

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

How do we solve the problem?

- Inspect the properties of the problem.

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

Inverse Problems are often Modelled as an Optimization Problem

Example 1:

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

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 $\alpha = 1/L$

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

Worst case convergence guarantee:

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

If we knew ...

If we knew ...

- that A is actually of the form

$$A = \begin{pmatrix} 10 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad ?$$

Hidden Structures

If we knew ...

- that A is actually of the form

$$A = \begin{pmatrix} 10 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad ?$$

- We would write down a different algorithm (that directly returns the solution).

> - **Game-changer**, if many such problems for different b need to be solved.
> - Sometimes the **“best” class of problems is not obvious!**

Hidden Structures

If we knew ...

- that A is actually of the form

$$A = \begin{pmatrix} 10 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad ?$$

- We would write down a different algorithm (that directly returns the solution).

> - **Game-changer**, if many such problems for different b need to be solved.
> - Sometimes the **“best” class of problems is not obvious!**

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

Example with Noisy Operator A

Example 2: Solve many problems of the form

$$\min_x f_A(x)\,, \quad f_A(x) := \frac{1}{2}\|Ax - b\|^2 \quad \text{where} \quad A = \bar{A} + \text{noise}\,.$$

Example with Noisy Operator A

Example 2: Solve many problems of the form

$$\min_x f_A(x)\,, \quad f_A(x) := \frac{1}{2}\|Ax - b\|^2 \quad \text{where} \quad A = \bar{A} + \text{noise}\,.$$

Using Gradient Descent:

- For **each problem** f_A, we need to compute $L = \|A\|^2$,
- and run Gradient Descent with $\alpha = 1/L$ to solve the problem.

> - Computation of $\|A\|$ can be expensive.

Example with Noisy Operator A

Example 2: Solve many problems of the form

$$\min_x f_A(x)\,, \quad f_A(x) := \frac{1}{2}\|Ax - b\|^2 \quad \text{where} \quad A = \bar{A} + \text{noise}\,.$$

Using Gradient Descent:

- For **each problem** f_A, we need to compute $L = \|A\|^2$,
- and run Gradient Descent with $\alpha = 1/L$ to solve the problem.

> - Computation of $\|A\|$ can be expensive.

Or ...

- if the noise is bounded, we can use a worst case estimate for L.

> - Results in **small step sizes**.
> - Upper bound may be **too pessimistic for most problems** in practice.

Example with Noisy Operator A

Example 2: Solve many problems of the form

$$\min_x f_A(x)\,, \quad f_A(x) := \frac{1}{2}\|Ax - b\|^2 \quad \text{where} \quad A = \bar{A} + \text{noise}\,.$$

Using Gradient Descent:

- For **each problem** f_A, we need to compute $L = \|A\|^2$,
- and run Gradient Descent with $\alpha = 1/L$ to solve the problem.

> - Computation of $\|A\|$ can be expensive.

Or ...

- if the noise is bounded, we can use a worst case estimate for L.

> - Results in **small step sizes**.
> - Upper bound may be **too pessimistic for most problems** in practice.

Can we construct an algorithm with good performance for more likely problems?

Data Driven Approach

Yes, using data driven approaches / learning !

Data Driven Approach

Yes, using data driven approaches / learning !

Learning alleviates the bounds of analytical tractability
by providing more:

- **Information:** Leverage more structure.
- **Automation:** Less “hand-crafting”.
- **Possibilities:** More building blocks.

Data Driven Approach

Yes, using data driven approaches / learning !

Learning alleviates the bounds of analytical tractability
by providing more:

- **Information:** Leverage more structure.
- **Automation:** Less “hand-crafting”.
- **Possibilities:** More building blocks.

Our goals:

- Breaking the barrier of worst-case estimates.
- Adapt algorithms to hidden problem structures.
- Define tight classes of problems.
- We insist on having some theoretical guarantees.

Formalize the Problem

- Consider the random parametric optimization problem:

$$\min_{x\in\mathbb{R}^n} \; \ell(x, \mathfrak{S})$$

  - $\ell : \mathbb{R}^n \times \Theta \to \mathbb{R}_{\geq 0}$ is a given measurable loss-function.
  - $\mathfrak{S} : (\Omega, \mathcal{F}, \mathbb{P}) \to \Theta$ is a random variable.

Formalize the Problem

- Consider the random parametric optimization problem:

$$\min_{x\in\mathbb{R}^n} \ \ell(x,\mathfrak{S})$$

  - $\ell : \mathbb{R}^n \times \Theta \to \mathbb{R}_{\geq 0}$ is a given measurable loss-function.
  - $\mathfrak{S} : (\Omega, \mathcal{F}, \mathbb{P}) \to \Theta$ is a random variable.

Example:

- Regularized Inverse Problem:

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

Formalize the Problem

- Consider the random parametric optimization problem:

$$\min_{x\in\mathbb{R}^n} \ \ell(x, \mathfrak{S})$$

  - $\ell : \mathbb{R}^n \times \Theta \to \mathbb{R}_{\geq 0}$ is a given measurable loss-function.
  - $\mathfrak{S} : (\Omega, \mathcal{F}, \mathbb{P}) \to \Theta$ is a random variable.

- Use a parametric optimization algorithm, i.e. a measurable function:

$$\mathcal{A} : \mathcal{H} \times \mathbb{R}^n \times \Theta \to \mathbb{R}^n, \quad (\alpha, x^{(0)}, \theta) \mapsto \mathcal{A}(\alpha, x^{(0)}, \theta)$$

Example:

- Regularized Inverse Problem:

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

Formalize the Problem

- Consider the random parametric optimization problem:

$$\min_{x \in \mathbb{R}^n} \ \ell(x, \mathfrak{S})$$

  - $\ell : \mathbb{R}^n \times \Theta \to \mathbb{R}_{\geq 0}$ is a given measurable loss-function.
  - $\mathfrak{S} : (\Omega, \mathcal{F}, \mathbb{P}) \to \Theta$ is a random variable.

- Use a parametric optimization algorithm, i.e. a measurable function:

$$\mathcal{A} : \mathcal{H} \times \mathbb{R}^n \times \Theta \to \mathbb{R}^n, \quad (\alpha, x^{(0)}, \theta) \mapsto \mathcal{A}(\alpha, x^{(0)}, \theta)$$

Example:

- Regularized Inverse Problem:

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

- Concatenation of a fixed number of Preconditioned Gradient Descent steps:

$$x^{(k+1)} = x^{(k)} - P\nabla\ell(x^{(k)}, \theta), \qquad \text{i.e. } \alpha := P, \mathcal{H} := \mathbb{R}^{n \times n} .$$

Formalize the Problem

- Consider the random parametric optimization problem:

$$\min_{x \in \mathbb{R}^n} \ \ell(x, \mathfrak{S})$$

  - $\ell : \mathbb{R}^n \times \Theta \to \mathbb{R}_{\geq 0}$ is a given measurable loss-function.
  - $\mathfrak{S} : (\Omega, \mathcal{F}, \mathbb{P}) \to \Theta$ is a random variable.

- Use a parametric optimization algorithm, i.e. a measurable function:

$$\mathcal{A} : \mathcal{H} \times \mathbb{R}^n \times \Theta \to \mathbb{R}^n, \quad (\alpha, x^{(0)}, \theta) \mapsto \mathcal{A}(\alpha, x^{(0)}, \theta)$$

Example:

- Regularized Inverse Problem:

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

- Concatenation of a fixed number of Preconditioned Gradient Descent steps:

$$x^{(k+1)} = x^{(k)} - P\nabla\ell(x^{(k)}, \theta), \qquad \text{i.e. } \alpha := P,\ \mathcal{H} := \mathbb{R}^{n\times n}\,.$$

$\rightsquigarrow$ Learning boils down to **hyperparameter optimization**, i.e. how to choose $\alpha \in \mathcal{H}$.

Quest for Theoretical Convergence Guarantees

Deterministic/Analytic Approach: Worst case performance

$$\min_{\alpha \in \mathcal{H}} \sup_{\theta \in \Theta} \ell\big(\mathcal{A}(\alpha, \theta), \theta\big) .$$

Only possible for certain classes of problems.

Quest for Theoretical Convergence Guarantees

Deterministic/Analytic Approach: Worst case performance

$$\min_{\alpha\in\mathcal{H}} \sup_{\theta\in\Theta} \ell\big(\mathcal{A}(\alpha,\theta),\theta\big)\,.$$

Only possible for certain classes of problems.

Learning Based Approach: Expected case performance:

- Minimize the **risk** $\mathcal{R}(\alpha)$, defined as the **expected loss**:

$$\min_{\alpha\in\mathcal{H}} \mathcal{R}(\alpha)\,,\quad \mathcal{R}(\alpha) := \mathbb{E}\big[\ell\big(\mathcal{A}(\alpha,\mathfrak{S}),\mathfrak{S}\big)\big]\,.$$

Quest for Theoretical Convergence Guarantees

Deterministic/Analytic Approach: Worst case performance

$$\min_{\alpha\in\mathcal{H}} \sup_{\theta\in\Theta} \ell\big(\mathcal{A}(\alpha,\theta),\theta\big)\,.$$

Only possible for certain classes of problems.

Learning Based Approach: Expected case performance:

- Minimize the **risk** $\mathcal{R}(\alpha)$, defined as the **expected loss**:

$$\min_{\alpha\in\mathcal{H}} \mathcal{R}(\alpha)\,,\quad \mathcal{R}(\alpha) := \mathbb{E}\big[\ell\big(\mathcal{A}(\alpha,\mathfrak{S}),\mathfrak{S}\big)\big]\,.$$

This is intractable, since the distribution $\mathbb{P}_{\mathfrak{S}}$ is **unknown**.

Quest for Theoretical Convergence Guarantees

Deterministic/Analytic Approach: Worst case performance

$$\min_{\alpha \in \mathcal{H}} \sup_{\theta \in \Theta} \ell\big(\mathcal{A}(\alpha, \theta), \theta\big) \,.$$

Only possible for certain classes of problems.

Learning Based Approach: Expected case performance:

- Minimize the **risk** $\mathcal{R}(\alpha)$, defined as the **expected loss**:

$$\min_{\alpha \in \mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[\ell\big(\mathcal{A}(\alpha, \mathfrak{S}), \mathfrak{S}\big)\big] \,.$$

  This is intractable, since the distribution $\mathbb{P}_{\mathfrak{S}}$ is **unknown**.

- Hence, resort to minimizing the **empirical risk** $\hat{\mathcal{R}}(\alpha, \mathfrak{D}_N)$ over some dataset $\mathfrak{D}_N := \{\mathfrak{S}_i\}_{i=1}^N$:

$$\min_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N)\,, \quad \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) := \frac{1}{N} \sum_{i=1}^{N} \ell(\mathcal{A}(\alpha, \mathfrak{S}_i), \mathfrak{S}_i) \,.$$

Why do we need Generalization Guarantees?

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

Why do we need Generalization Guarantees?

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

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

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*(\mathfrak{D}_N)\big) \;\leq\; \inf_{\alpha\in\mathcal{H}} \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Why do we need Generalization Guarantees?

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

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

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*(\mathfrak{D}_N)\big) \;\leq\; \inf_{\alpha\in\mathcal{H}} \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Why do we need Generalization Guarantees?

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

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

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*(\mathfrak{D}_N)\big) \;\leq\; \inf_{\alpha\in\mathcal{H}} \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Why do we need Generalization Guarantees?

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

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

$$\mathbb{P}\Big\{\mathcal{R}\big(\alpha^*(\mathfrak{D}_N)\big) \;\leq\; \inf_{\alpha\in\mathcal{H}}\; \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) + K(N, \alpha, \epsilon)\Big\} \geq 1 - \epsilon\,.$$

Why do we need Generalization Guarantees?

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

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

$$\mathbb{P}\{\mathcal{R}(\alpha^*(\mathfrak{D}_N)) \leq \inf_{\alpha \in \mathcal{H}} \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N) + K(N, \alpha, \epsilon)\} \geq 1 - \epsilon\,.$$

- 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** $\mathbb{Q} \in \mathcal{M}(\mathbb{P}_{\mathfrak{H}})$:

$$\mathbb{P}\Big\{\mathbb{E}_{\mathbb{Q}^*(\mathfrak{D}_N)}[\mathcal{R}] \leq \inf_{\mathbb{Q}\in\mathcal{M}(\mathbb{P}_{\mathfrak{H}})} \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}(\mathfrak{D}_N)] + K(\mathbb{Q}, N, \epsilon)\Big\} \geq 1 - \epsilon\,,$$

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

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-Bayes [Alquier '21]

Learning-to-Optimize [Chen et al. '22]

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

- [Alquier '21]: "User-friendly introduction to PAC-Bayes bounds", arXiv:2110.11216 (2021).
- [Chen et al. '22]: "Learning to optimize: A primer and a benchmark", Journal of Machine Learning Research (2022), pp. 8562–8620.

A Form of the Donsker–Varadhan Variational Formulation

Lemma: Consider an exponential family $(\mathbb{Q}_\lambda)_{\lambda\in\Lambda}$ w.r.t. the prior $\mathbb{P}_{\mathfrak{H}}$, i.e. distributions of the form:

$$\mathbb{Q}_\lambda \propto \exp(\langle \eta(\lambda), T\rangle)\cdot \mathbb{P}_{\mathfrak{H}}, \qquad \lambda \in \Lambda$$

and denote $c(\lambda) := \mathbb{E}_{\mathbb{P}_{\mathfrak{H}}}\big[\exp(\langle \eta(\lambda), T\rangle)\big]$. Then it holds:

$$\log\big(c(\lambda)\big) = \sup_{\mathbb{Q}\ll\mathbb{P}_{\mathfrak{H}}} \mathbb{E}_{\mathbb{Q}}[\langle \eta(\lambda), T\rangle] - D_{KL}(\mathbb{Q}\,\|\,\mathbb{P}_{\mathfrak{H}})$$

Furthermore, for every $\lambda\in\Lambda$, the supremum is attained at $\mathbb{Q}_\lambda$.

A Form of the Donsker–Varadhan Variational Formulation

Lemma: Consider an exponential family $(\mathbb{Q}_\lambda)_{\lambda\in\Lambda}$ w.r.t. the prior $\mathbb{P}_{\mathfrak{H}}$, i.e. distributions of the form:

$$\mathbb{Q}_\lambda \propto \exp(\langle \eta(\lambda), T\rangle) \cdot \mathbb{P}_{\mathfrak{H}}, \qquad \lambda \in \Lambda$$

and denote $c(\lambda) := \mathbb{E}_{\mathbb{P}_{\mathfrak{H}}}\big[\exp(\langle \eta(\lambda), T\rangle)\big]$. Then it holds:

$$\log\big(c(\lambda)\big) = \sup_{\mathbb{Q}\ll\mathbb{P}_{\mathfrak{H}}} \mathbb{E}_{\mathbb{Q}}[\langle \eta(\lambda), T\rangle] - D_{KL}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}})$$

Furthermore, for every $\lambda \in \Lambda$, the supremum is attained at $\mathbb{Q}_\lambda$.

Towards a PAC-Bayes Theorem for Exponential Families

A Form of the Donsker–Varadhan Variational Formulation

Lemma: Consider an exponential family $(\mathbb{Q}_\lambda)_{\lambda\in\Lambda}$ w.r.t. the prior $\mathbb{P}_{\mathfrak{H}}$, i.e. distributions of the form:

$$\mathbb{Q}_\lambda \propto \exp(\langle \eta(\lambda), T\rangle) \cdot \mathbb{P}_{\mathfrak{H}}, \qquad \lambda \in \Lambda$$

and denote $c(\lambda) := \mathbb{E}_{\mathbb{P}_{\mathfrak{H}}}\big[\exp(\langle \eta(\lambda), T\rangle)\big]$. Then it holds:

$$\log\big(c(\lambda)\big) = \sup_{\mathbb{Q}\ll\mathbb{P}_{\mathfrak{H}}} \mathbb{E}_{\mathbb{Q}}[\langle \eta(\lambda), T\rangle] - D_{KL}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}})$$

Furthermore, for every $\lambda \in \Lambda$, the supremum is attained at $\mathbb{Q}_\lambda$.

A Form of the Donsker–Varadhan Variational Formulation

Lemma: Consider an exponential family $(\mathbb{Q}_\lambda)_{\lambda\in\Lambda}$ w.r.t. the prior $\mathbb{P}_{\mathfrak{H}}$, i.e. distributions of the form:

$$\mathbb{Q}_\lambda \propto \exp(\langle \eta(\lambda), T\rangle) \cdot \mathbb{P}_{\mathfrak{H}}, \qquad \lambda \in \Lambda$$

and denote $c(\lambda) := \mathbb{E}_{\mathbb{P}_{\mathfrak{H}}}\big[\exp(\langle \eta(\lambda), T\rangle)\big]$. Then it holds:

$$\log\big(c(\lambda)\big) = \sup_{\mathbb{Q}\ll\mathbb{P}_{\mathfrak{H}}} \mathbb{E}_{\mathbb{Q}}[\langle \eta(\lambda), T\rangle] - D_{KL}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}})$$

Furthermore, for every $\lambda \in \Lambda$, the supremum is attained at $\mathbb{Q}_\lambda$.

Theorem: If $\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big] \leq 1$, then for all $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\Big\{\forall\lambda \in \Lambda\colon \forall\mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} : \mathbb{E}_{\mathbb{Q}}\big[\langle \eta(\lambda), T\rangle\big] \leq D_{\mathrm{KL}}(\mathbb{Q} \parallel \mathbb{P}_{\mathfrak{H}}) + \log(|\Lambda|/\varepsilon)\Big\} \geq 1 - \varepsilon$$

Towards a PAC-Bayes Theorem for Exponential Families

Theorem: If $\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big] \leq 1$, then for all $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\Big\{\forall\lambda \in \Lambda\colon \forall\mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} : \mathbb{E}_{\mathbb{Q}}\big[\langle\eta(\lambda), T\rangle\big] \leq D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}}) + \log(|\Lambda|/\varepsilon)\Big\} \geq 1 - \varepsilon$$

Sketch of Proof.

- Use Markov's inequality

$$\mathbb{P}_{\mathfrak{D}_N}\big\{c(\lambda) \geq \exp(s)\big\} \leq \frac{\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big]}{\exp(s)} \leq 1/\exp(s) =: 1/s' \,.$$

Theorem: If $\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big] \leq 1$, then for all $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\big\{\forall\lambda \in \Lambda\colon \forall\mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} : \mathbb{E}_{\mathbb{Q}}\big[\langle\eta(\lambda), T\rangle\big] \leq D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}}) + \log(|\Lambda|/\varepsilon)\big\} \geq 1 - \varepsilon$$

Sketch of Proof.

- Use Markov's inequality

$$\mathbb{P}_{\mathfrak{D}_N}\big\{c(\lambda) \geq \exp(s)\big\} \leq \frac{\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big]}{\exp(s)} \leq 1/\exp(s) =: 1/s'\,.$$

- Union-bound argument: (use covering argument for compact continuous Λ)

$$\mathbb{P}_{\mathfrak{D}_N}\big\{\sup_{\lambda\in\Lambda} c(\lambda) > s'\big\} = \mathbb{P}_{\mathfrak{D}_N}\big\{\bigcup_{\lambda\in\Lambda}\{c(\lambda) > s'\}\big\} \leq \sum_{\lambda\in\Lambda} \mathbb{P}_{\mathfrak{D}_N}\big\{\{c(\lambda) > s'\}\big\} \leq |\Lambda|/s' =: \varepsilon$$

Towards a PAC-Bayes Theorem for Exponential Families

Theorem: If $\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big] \leq 1$, then for all $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\Big\{\forall\lambda \in \Lambda\colon \forall \mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} : \mathbb{E}_{\mathbb{Q}}\big[\langle \eta(\lambda), T\rangle\big] \leq D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}_{\mathfrak{H}}) + \log(|\Lambda|/\varepsilon)\Big\} \geq 1 - \varepsilon$$

Sketch of Proof.

- Use Markov's inequality

$$\mathbb{P}_{\mathfrak{D}_N}\big\{c(\lambda) \geq \exp(s)\big\} \leq \frac{\mathbb{E}_{\mathfrak{D}_N}\big[c(\lambda)\big]}{\exp(s)} \leq 1/\exp(s) =: 1/s' \,.$$

- Union-bound argument: (use covering argument for compact continuous Λ)

$$\mathbb{P}_{\mathfrak{D}_N}\big\{\sup_{\lambda\in\Lambda} c(\lambda) > s'\big\} = \mathbb{P}_{\mathfrak{D}_N}\Big\{\bigcup_{\lambda\in\Lambda}\{c(\lambda) > s'\}\Big\} \leq \sum_{\lambda\in\Lambda} \mathbb{P}_{\mathfrak{D}_N}\big\{\{c(\lambda) > s'\}\big\} \leq |\Lambda|/s' =: \varepsilon$$

- Apply Donsker–Varadhan variational formulation in

$$\mathbb{P}_{\mathfrak{D}_N}\big\{\sup_{\lambda\in\Lambda} \log(c(\lambda)) \leq \log(|\Lambda|/\varepsilon)\big\} \geq 1 - \varepsilon \,.$$

□

Specify η and T accordingly to **construct** a PAC-Bayesian generalization bound:

$$\eta(\lambda) = (\eta_1(\lambda), \eta'(\lambda))$$

and

$$T(\alpha, \mathfrak{D}_N) = (\mathcal{R}(\alpha) - \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N), T'(\alpha, \mathfrak{D}_N)) \, .$$

Specify η and T accordingly to **construct** a PAC-Bayesian generalization bound:

$$\eta(\lambda) = (\eta_1(\lambda), \eta'(\lambda))$$

and

$$T(\alpha, \mathfrak{D}_N) = (\mathcal{R}(\alpha) - \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N), T'(\alpha, \mathfrak{D}_N)) \,.$$

This provides a bound of the following form:

Theorem: Under mild assumptions, it holds for $\epsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\Big\{\forall \lambda \in \Lambda,\ \forall \mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}}\ :\ \mathbb{E}_{\mathbb{Q}}[\mathcal{R}] \leq \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}] + G(N, \lambda, \mathbb{Q}, \epsilon)\Big\} \geq 1 - \epsilon \,.$$

Note:

(i) By the definition of the risk and the algorithm, this bound gives a guarantee for the **function value of the algorithm's output**.

(ii) This is a statement about **relative** values, **not absolute** ones.

Specify η and T accordingly to **construct** a PAC-Bayesian generalization bound:

$$\eta(\lambda) = (\eta_1(\lambda), \eta'(\lambda))$$

and

$$T(\alpha, \mathfrak{D}_N) = (\mathcal{R}(\alpha) - \hat{\mathcal{R}}(\alpha, \mathfrak{D}_N), T'(\alpha, \mathfrak{D}_N))\,.$$

This provides a bound of the following form:

Theorem: Under mild assumptions, it holds for $\epsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\left\{\forall \lambda \in \Lambda,\ \forall \mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}}\ :\ \mathbb{E}_{\mathbb{Q}}[\mathcal{R}] \leq \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}] + G(N, \lambda, \mathbb{Q}, \epsilon)\right\} \geq 1 - \epsilon\,.$$

Note:

(i) By the definition of the risk and the algorithm, this bound gives a guarantee for the **function value of the algorithm's output**.

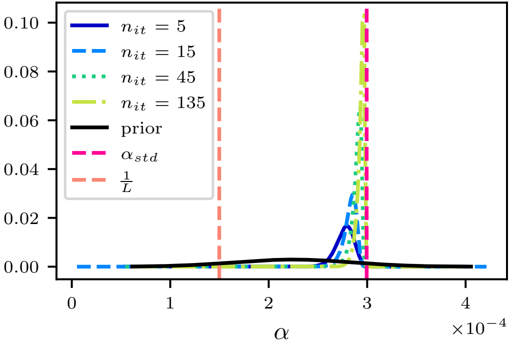
(ii) This is a statement about **relative** values, **not absolute** ones.

⇝ Since supremum is attained at $\mathbb{Q}_\lambda$, learning can be phrased as an **optimization in** λ (possibly very low-dimensional).

Simple Case Study: Gradient Descent

Issue: A bad performance on a single problem dominates the average.

- Sometimes, analytic worst-case bounds are sharp.
- Gradient Descent on quadratics diverges for $\alpha > 2/L$.
- Trying to learn the step size (without this bound) yields an extremely large loss for $\alpha > 2/L$, which dominates the cost of the "average performance" (the empirical risk). Therefore, learnable step sizes obey the deterministic step size rule $\alpha \in (0, 2/L)$.



In this case, the analytically best known step size is recoverd by learning.

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

> We develop a variant that allows to **Trade-Off Guarantees and Speed**.

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

We develop a variant that allows to **Trade-Off Guarantees and Speed**.

The algorithm may diverge (for extreme cases), if this happens in rare cases and the Trade-Off can be controlled.

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

We develop a variant that allows to **Trade-Off Guarantees and Speed**.

The algorithm may diverge (for extreme cases), if this happens in rare cases and the Trade-Off can be controlled.

- Account for **likelihood** of e.g. the worst-case.

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

We develop a variant that allows to **Trade-Off Guarantees and Speed**.

The algorithm may diverge (for extreme cases), if this happens in rare cases and the Trade-Off can be controlled.

- Account for **likelihood** of e.g. the worst-case.
- **Encode** properties of the algorithm in the convergence set $\mathrm{C} \subset \mathcal{H} \times \Theta$, e.g.,

$$\mathrm{C}_\alpha := \{\theta \in \Theta \ : \ \ell(\mathcal{A}(\alpha, \theta), \theta) \leq \ell(x^{(0)}, \theta)\} \, .$$

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

We develop a variant that allows to **Trade-Off Guarantees and Speed**.

The algorithm may diverge (for extreme cases), if this happens in rare cases and the Trade-Off can be controlled.

- Account for **likelihood** of e.g. the worst-case.
- **Encode** properties of the algorithm in the convergence set $\mathrm{C} \subset \mathcal{H} \times \Theta$, e.g.,

$$\mathrm{C}_\alpha := \{\theta \in \Theta \;:\; \ell(\mathcal{A}(\alpha, \theta), \theta) \leq \ell(x^{(0)}, \theta)\} \,.$$

- **Condition** on it to get the convergence risk $\mathcal{R}_c \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$:

$$\mathcal{R}_c(\alpha) := \mathbb{E}\big[\ell(\mathcal{A}(\alpha, \mathfrak{S}), \mathfrak{S}) \mid \mathrm{C}_\alpha\big] \,.$$

Trade-Off Guarantees and Speed

However this might correspond to unlikely special/extreme cases:

We develop a variant that allows to **Trade-Off Guarantees and Speed**.

The algorithm may diverge (for extreme cases), if this happens in rare cases and the Trade-Off can be controlled.

- Account for **likelihood** of e.g. the worst-case.
- **Encode** properties of the algorithm in the convergence set $\mathrm{C} \subset \mathcal{H} \times \Theta$, e.g.,

$$\mathrm{C}_\alpha := \{\theta \in \Theta \ : \ \ell(\mathcal{A}(\alpha, \theta), \theta) \leq \ell(x^{(0)}, \theta)\} \,.$$

- **Condition** on it to get the convergence risk $\mathcal{R}_c \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$:

$$\mathcal{R}_c(\alpha) := \mathbb{E}\big[\ell(\mathcal{A}(\alpha, \mathfrak{S}), \mathfrak{S}) \mid \mathrm{C}_\alpha\big] \,.$$

- **Guarantees** in form of the convergence probability $\mathbb{P}_{\mathfrak{S}}[\mathrm{C}_\alpha]$ instead of convergence for every sample.

Trade-Off Guarantees and Speed

Applying the same machinery again yields the following **generalization**:

Theorem: Under mild assumptions, it holds for $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\left\{\forall\lambda \in \Lambda, \forall\mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} \;:\; \mathbb{E}_{\mathbb{Q}}[\mathcal{R}_c] \leq \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}_c] + G(N, \lambda, \mathbb{Q}, \varepsilon)\right\} \geq 1 - \varepsilon\,.$$

Trade-Off Guarantees and Speed

Applying the same machinery again yields the following **generalization**:

Theorem: Under mild assumptions, it holds for $\varepsilon > 0$:

$$\mathbb{P}_{\mathfrak{D}_N}\left\{\forall \lambda \in \Lambda, \forall \mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} \ : \ \mathbb{E}_{\mathbb{Q}}[\mathcal{R}_c] \leq \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}_c] + G(N, \lambda, \mathbb{Q}, \varepsilon)\right\} \geq 1 - \varepsilon \, .$$

- Learning must achieve that $\mathcal{A}$ converges for "sufficiently many problems" (according to the convergence probability).
- Therefore, the algorithm can focus on quickly solving the remaining problems.

Trade-Off Guarantees and Speed

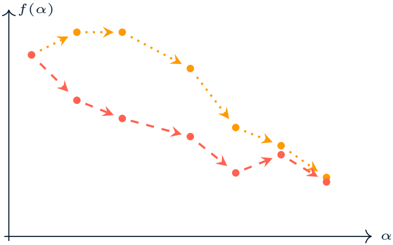
Applying the same machinery again yields the following **generalization**:

Theorem: Under mild assumptions, it holds for $\varepsilon > 0$:

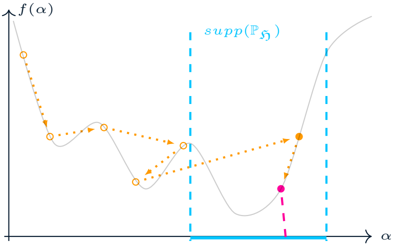
$$\mathbb{P}_{\mathfrak{D}_N}\left\{\forall \lambda \in \Lambda, \forall \mathbb{Q} \ll \mathbb{P}_{\mathfrak{H}} \;:\; \mathbb{E}_{\mathbb{Q}}[\mathcal{R}_c] \leq \mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}_c] + G(N, \lambda, \mathbb{Q}, \varepsilon)\right\} \geq 1 - \varepsilon\,.$$

- Learning must achieve that $\mathcal{A}$ converges for “sufficiently many problems” (according to the convergence probability).
- Therefore, the algorithm can focus on quickly solving the remaining problems.
- **Example Statement:** With high probability, the algorithm that is trained to optimize 95% of all problems in $\mathfrak{D}_N$ quickly, will optimize 95% of all problems quickly.

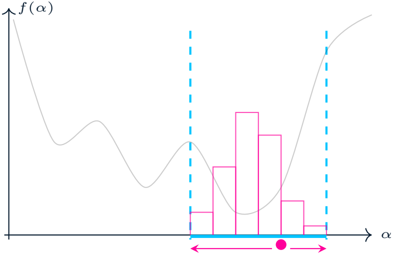
The Whole Training Process



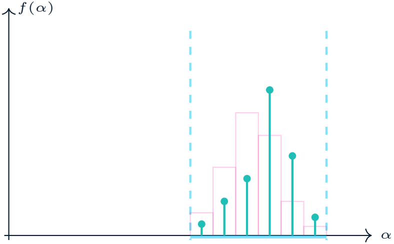
1) Find a “trainable” initialization by following another algorithm.



2) Find a point inside the constraint with small empirical risk.

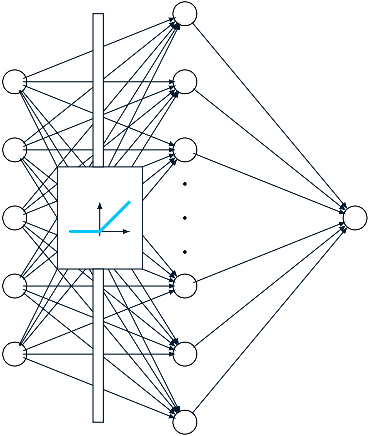


3) Run a specifically constrained sampling procedure.

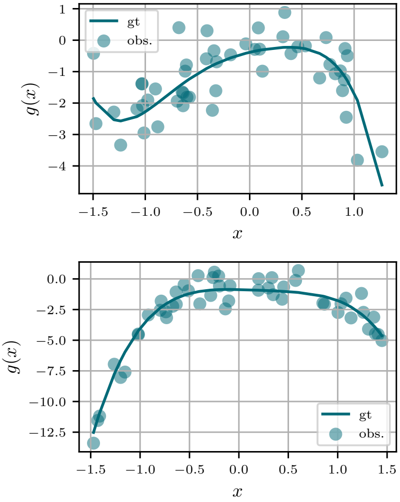


4) Find λ^* and perform a reweighting based on closed-form of the posterior.

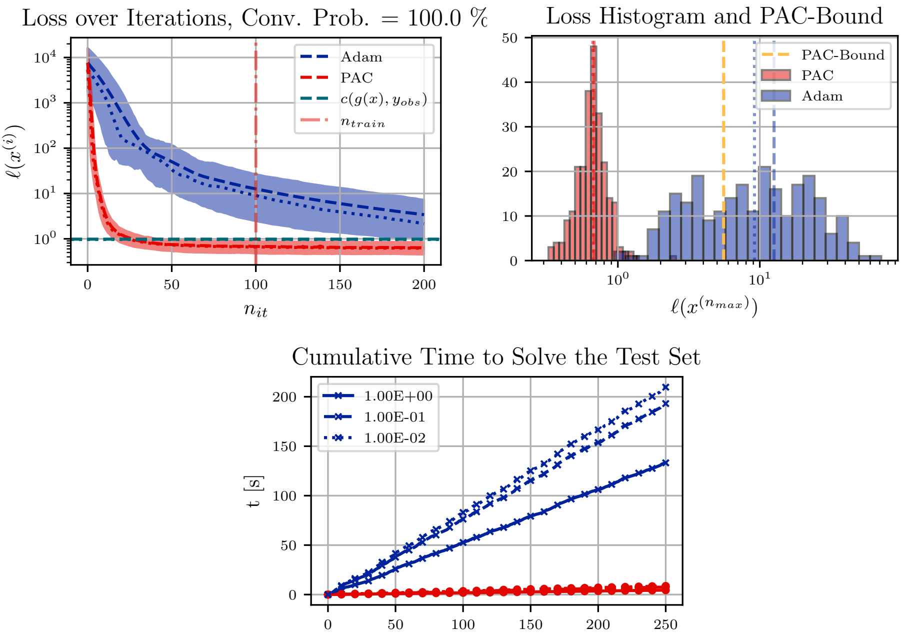
Learn an Algorithm to Train 2-Layer Regression Networks



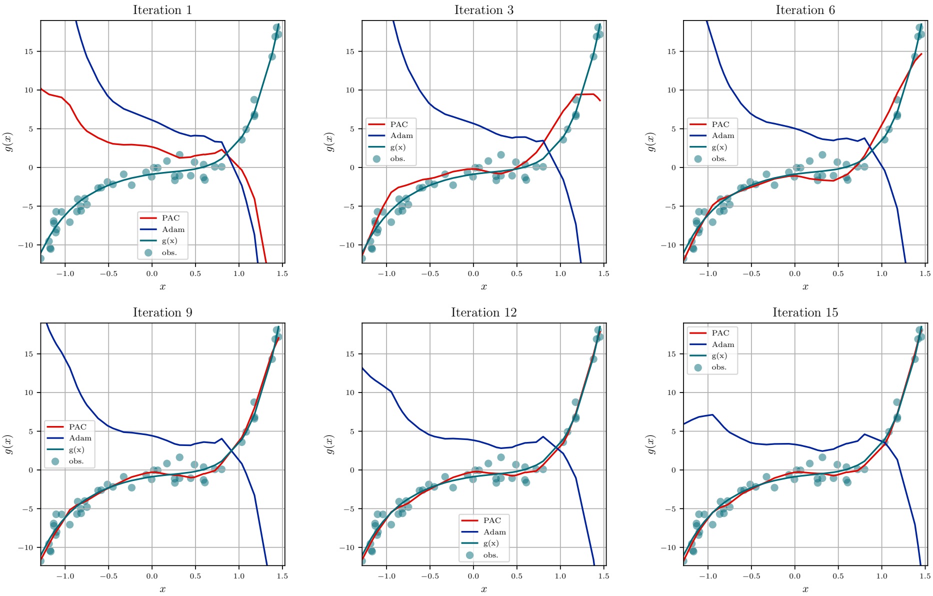
Training a 2-layer neural network with ReLU-activations ...



... to perform regression.


Loss over Iterations, Conv. Prob. = 100.0 %
Adam
PAC
$c(g(x), y_{obs})$
n_{train}
$\ell(x^{(i)})$
n_{it}
Loss Histogram and PAC-Bound
PAC-Bound
PAC
Adam
$\ell(x^{(n_{max})})$
Cumulative Time to Solve the Test Set
1.00E+00
1.00E-01
1.00E-02
t [s]

Learning gets faster...



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

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

- Breaking the barrier of worst-case estimates

$$\min_{\alpha\in\mathcal{H}} \mathcal{R}(\alpha)\,, \quad \mathcal{R}(\alpha) := \mathbb{E}\big[\ell\big(\mathcal{A}(\alpha,\mathfrak{S}),\mathfrak{S}\big)\big]\,.$$

- by learning spezialized optimization algorithms

$$\min_{\alpha\in\mathcal{H}} \hat{\mathcal{R}}(\alpha,\mathfrak{D}_N)\,, \quad \hat{\mathcal{R}}(\alpha,\mathfrak{D}_N) := \frac{1}{N}\sum_{i=1}^{N}\ell(\mathcal{A}(\alpha,\mathfrak{S}_i),\mathfrak{S}_i)\,.$$

- with theoretical guarantees via PAC-Bayes generalization bounds:

$$\mathbb{P}_{\mathfrak{D}_N}\Big\{\forall\lambda\in\Lambda,\ \forall\mathbb{Q}\ll\mathbb{P}_{\mathfrak{H}}\ :\ \mathbb{E}_{\mathbb{Q}}[\mathcal{R}]\le\mathbb{E}_{\mathbb{Q}}[\hat{\mathcal{R}}]+G(N,\lambda,\mathbb{Q},\epsilon)\Big\}\ge 1-\epsilon\,.$$