Formal guarantees in machine learning, statistics, and optimization

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

Collège de France - June 30, 2022

Formal guarantees in ML, statistics, and optimization

Outline

1. Classical supervised machine learning

2. A posteriori statistical guarantees

3. A priori statistical guarantees

4. Guarantees for optimization

Classical supervised machine learning pipeline

- **Input**
  - Training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of input/output pairs
  - Prior knowledge (models, hyperparameters)

- **Output**
  - Prediction function $f : \mathcal{X} \to \mathcal{Y}$
  - Often an algorithm itself

Classical supervised machine learning pipeline

- **Input**
  - Training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of input/output pairs
  - Prior knowledge (models, hyperparameters)
- **Output**
  - Prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$
  - Often an algorithm itself
- **Difficulties**
  - Sets $\mathcal{X}$ and $\mathcal{Y}$ can be complex
  - Relationship between x and y not deterministic
  - Relationship between x and y can be complex
  - Unclear performance criteria

Performance criteria

- **Classical supervised machine learning pipeline**
  - Input: Training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
  - Output: Prediction function $f : \mathcal{X} \to \mathcal{Y}$

Performance criteria

- **Classical supervised machine learning pipeline**
  - Input: Training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
  - Output: Prediction function $f : \mathcal{X} \to \mathcal{Y}$

1. **Computational performance of training algorithm and of f**
   - Speed, memory
   - Certification

Performance criteria

- **Classical supervised machine learning pipeline**
  - Input: Training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
  - Output: Prediction function $f : \mathcal{X} \to \mathcal{Y}$

1. **Computational performance of training algorithm and of f**
   - Speed, memory
   - Certification

2. **Statistical performance of f on testing data**
   - Testing data: subset of $\mathcal{X} \times \mathcal{Y}$, or probability distribution
   - Loss function $\ell(y, f(x))$ assumed given

Statistical performance

- **Expected risk:** $\mathcal{R}(f) = \mathbb{E}_{p(x,y)} \ell(y, f(x))$
  - Binary classification ($\mathcal{Y} = \{0, \dots, k-1\}$): average error rate
  - Regression ($\mathcal{Y} = \mathbb{R}$): mean squared error

Statistical performance

- **Expected risk:** $\mathcal{R}(f) = \mathbb{E}_{p(x,y)} \ell(y, f(x))$
  - Binary classification ($\mathcal{Y} = \{0, \dots, k-1\}$): average error rate
  - Regression ($\mathcal{Y} = \mathbb{R}$): mean squared error

- **Optimal statistical performance** (Devroye et al., 1997)
  - Optimal "Bayes" predictor $f^* = \operatorname{argmin} \mathcal{R}(f)$

$$f^*(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \mathbb{E}_{p(y|x)} \ell(y, z)$$

  - Bayes risk $\mathcal{R}(f^*)$ typically not equal to zero
  - Requires full access to testing distribution $p(x, y)$

Statistical performance

- **Expected risk:** $\mathcal{R}(f) = \mathbb{E}_{p(x,y)} \ell(y, f(x))$
  - Binary classification ($\mathcal{Y} = \{0, \dots, k-1\}$): average error rate
  - Regression ($\mathcal{Y} = \mathbb{R}$): mean squared error

- **Optimal statistical performance** (Devroye et al., 1997)
  - Optimal "Bayes" predictor $f^* = \operatorname{argmin} \mathcal{R}(f)$

$$f^*(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \mathbb{E}_{p(y|x)} \ell(y, z)$$

  - Bayes risk $\mathcal{R}(f^*)$ typically not equal to zero
  - Requires full access to testing distribution $p(x, y)$

- **Absolute vs. relative performance**
  - Risk $\mathcal{R}(f)$ vs. excess risk $\mathcal{R}(f) - \mathcal{R}(f^*)$
  - Guarantees for a prediction function vs. for a training algorithm

Machine learning algorithms

- **Goal**: achieve the risk $\mathcal{R}^*$ of the optimal prediction function f^*

Machine learning algorithms

- **Goal**: achieve the risk $\mathcal{R}^*$ of the optimal prediction function f^*

- **Two main principles**

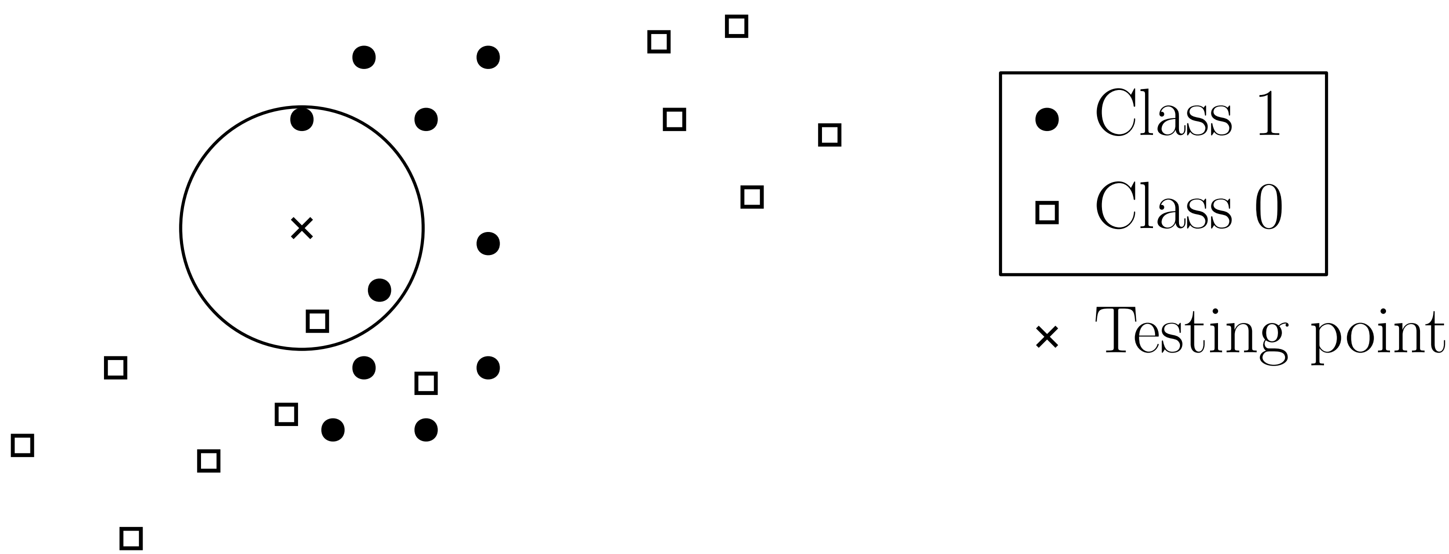
  1. Local averaging
  2. Empirical risk minimization

Local averaging

- **Principle**
  - Estimate conditional distribution $p(y|x)$ and compute $\mathbb{E}(y|x)$

Local averaging

- **Principle**
  - Estimate conditional distribution $p(y|x)$ and compute $\mathbb{E}(y|x)$
- **Examples**
  - k-nearest neighbor
  - "No training", one hyperparameter to determine "locality"



Empirical risk minimization

- **Principle**
  - Minimize the empirical risk $\widehat{\mathcal{R}}(f) = \dfrac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$
  - Parameterized set of functions (e.g., linear models, neural networks)

Empirical risk minimization

- **Principle**
  - Minimize the empirical risk $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$
  - Parameterized set of functions (e.g., linear models, neural networks)

- **Need some "capacity control"**
  - Constrain or penalize some norm on the parameters (with explicit hyperparameter)
  - Algorithmic regularization

Empirical risk minimization

- **Principle**
  - Minimize the empirical risk $\widehat{\mathcal{R}}(f) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \ell(y_i, f(x_i))$
  - Parameterized set of functions (e.g., linear models, neural networks)

- **Need some "capacity control"**
  - Constrain or penalize some norm on the parameters (with explicit hyperparameter)
  - Algorithmic regularization

- **Training = optimization**
  - Can be slow
  - May not converge to the global optimum

Evaluation of statistical performance

- **Given a single prediction function** f
  - From m independent and identically distributed $(x_j, y_j)_{j\in\{1,\dots,m\}}$
  - Hoeffding's inequality: with probability greater than $1-\delta$,

$$\mathbb{E}_{p(x,y)}\ell(y, f(x)) \leqslant \frac{1}{m}\sum_{j=1}^{m}\ell(y_j, f(x_j)) + \frac{\|\ell\|_\infty}{\sqrt{m}}\sqrt{\log\frac{1}{\delta}}$$

Evaluation of statistical performance

- **Given a single prediction function f**
  - From m independent and identically distributed $(x_j, y_j)_{j \in \{1,\dots,m\}}$
  - Hoeffding's inequality: with probability greater than $1-\delta$,

$$\mathbb{E}_{p(x,y)}\ell(y,f(x)) \leqslant \frac{1}{m}\sum_{j=1}^{m}\ell(y_j,f(x_j)) + \frac{\|\ell\|_\infty}{\sqrt{m}}\sqrt{\log\frac{1}{\delta}}$$

- **Multiple tests require "Bonferroni" correction**
  - With T tests, $\log\frac{1}{\delta}$ replaced by $\log\frac{T}{\delta} = \log T + \log\frac{1}{\delta}$

Evaluation of statistical performance

- **Given a single prediction function f**
  - From m independent and identically distributed $(x_j, y_j)_{j\in\{1,\dots,m\}}$
  - Hoeffding's inequality: with probability greater than $1-\delta$,

$$\mathbb{E}_{p(x,y)}\ell(y, f(x)) \leqslant \frac{1}{m}\sum_{j=1}^{m}\ell(y_j, f(x_j)) + \frac{\|\ell\|_\infty}{\sqrt{m}}\sqrt{\log\frac{1}{\delta}}$$

- **Multiple tests require "Bonferroni" correction**
  - With T tests, $\log\frac{1}{\delta}$ replaced by $\log\frac{T}{\delta} = \log T + \log\frac{1}{\delta}$

- **Evaluating performance from training data only?**
  - Training data $(x_i, y_i)_{i\in\{1,\dots,n\}}$ i.i.d. from testing distribution
  - Needs strong (often unverifiable) assumptions

Guarantees from training data

- **Training data $(x_i, y_i)_{i \in \{1,\dots,n\}}$ i.i.d. from testing distribution**

Guarantees from training data

- **Training data $(x_i, y_i)_{i \in \{1,\dots,n\}}$ i.i.d. from testing distribution**
- **Selection of $\hat{f}$ among T functions**: with probability $1-\delta$

$$\mathbb{E}_{p(x,y)} \ell(y, \hat{f}(x)) \leqslant \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{f}(x_i)) + \frac{\|\ell\|_\infty}{\sqrt{n}} \sqrt{\log \frac{T}{\delta}}$$

  – Not adapted to optimization of prediction functions f_θ, $\theta \in \Theta \subset \mathbb{R}^d$

Guarantees from training data

- **Training data** $(x_i, y_i)_{i \in \{1,\dots,n\}}$ **i.i.d. from testing distribution**

- **Selection of** $\hat{f}$ **among** T **functions**: with probability $1-\delta$

$$\mathbb{E}_{p(x,y)} \ell(y, \hat{f}(x)) \ \leqslant \ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{f}(x_i)) + \frac{\|\ell\|_\infty}{\sqrt{n}} \sqrt{\log \frac{T}{\delta}}$$

  – Not adapted to optimization of prediction functions f_θ, $\theta \in \Theta \subset \mathbb{R}^d$

- **Uniform concentration inequalities**: with probability $1-\delta$

$$\forall \theta \in \Theta, \mathbb{E}_{p(x,y)} \ell(y, f_\theta(x)) \ \leqslant \ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \frac{2\|\ell\|_\infty}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}} + C_n$$

  – Capacity of function class C_n
  – Allows optimization of empirical risk *and* a posteriori guarantees

A posteriori guarantees from training data in practice?

- **Many available statistical frameworks**
  - Rademacher complexities (see, e.g., Boucheron et al., 2005)
  - PAC-Bayesian analysis (see, e.g., Alquier, 2021)

A posteriori guarantees from training data in practice?

- **Many available statistical frameworks**
  - Rademacher complexities (see, e.g., Boucheron et al., 2005)
  - PAC-Bayesian analysis (see, e.g., Alquier, 2021)

- **Non-trivial if n sufficiently large and model class well chosen**
  - Based on computable quantities
  - ⚠ Only use the testing distribution at the end
  - ⚠ Based on distributional assumptions

Guarantees for training algorithms

- **Main goal**
  - Given a class of distributions $p(x,y)$
  - Estimator $\hat{f}_n$ obtained from n observations
  - Proof that $\mathcal{R}(\hat{f}_n) - \mathcal{R}(f^*)$ goes to zero when $n \to +\infty$
  - If possible, rate of convergence

- **A priori guarantees**
  - ⚠ Depends on unknown quantities

No free lunch theorems
(Devroye et al., 2013, Theorem 7.2)

- **Assumptions**
  - Binary classification with 0-1 loss, with $\mathcal{X}$ infinite
  - $\mathcal{P}$ = set of all probability distributions on $\mathcal{X} \times \{0, 1\}$
  - $\mathcal{D}_n(p)$ data set of n pairs (x_i, y_i) sampled i.i.d. from $p \in \mathcal{P}$

No free lunch theorems (Devroye et al., 2013, Theorem 7.2)

- **Assumptions**
  - Binary classification with 0-1 loss, with $\mathcal{X}$ infinite
  - $\mathcal{P}$ = set of all probability distributions on $\mathcal{X} \times \{0, 1\}$
  - $\mathcal{D}_n(p)$ data set of n pairs (x_i, y_i) sampled i.i.d. from $p \in \mathcal{P}$

- **Lower-bound**
  - For any decreasing (a_n) tending to zero and such that $a_1 \leqslant 1/16$
  - For any learning algorithm $\mathcal{A}$: datasets $\rightarrow$ prediction functions
  - There exists $p \in \mathcal{P}$, such that for all $n \geqslant 1$:

$$\mathbb{E}\Big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\Big] - \mathcal{R}_p^* \geqslant a_n$$

No free lunch theorems
(Devroye et al., 2013, Theorem 7.2)

- **Assumptions**
  - Binary classification with 0-1 loss, with $\mathcal{X}$ infinite
  - $\mathcal{P}$ = set of all probability distributions on $\mathcal{X} \times \{0,1\}$
  - $\mathcal{D}_n(p)$ data set of n pairs (x_i, y_i) sampled i.i.d. from $p \in \mathcal{P}$

- **Lower-bound**
  - For any decreasing (a_n) tending to zero and such that $a_1 \leqslant 1/16$
  - For any learning algorithm $\mathcal{A}$: datasets $\rightarrow$ prediction functions
  - There exists $p \in \mathcal{P}$, such that for all $n \geqslant 1$:

$$\mathbb{E}\Big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\Big] - \mathcal{R}_p^* \geqslant a_n$$

- **All learning algorithms must have weaknesses**

Curse of dimensionality on $\mathcal{X} = \mathbb{R}^d$

- **Weak assumption**: optimal function f^* is Lipschitz-continuous

$$\exists L,\ \forall x, x' \in \mathcal{X}, \quad |f^*(x) - f^*(x')| \leqslant L\|x - x'\|$$

  – Denote $\mathcal{P}_{\text{Lip.}}$ the corresponding set of probability distributions

Curse of dimensionality on $\mathcal{X} = \mathbb{R}^d$

- **Weak assumption**: optimal function f^* is Lipschitz-continuous

$$\exists L,\ \forall x, x' \in \mathcal{X},\ \ |f^*(x) - f^*(x')| \leqslant L\|x - x'\|$$

  – Denote $\mathcal{P}_{\text{Lip.}}$ the corresponding set of probability distributions

- **Lower bound on worst case performance** (Tsybakov, 2008)

$$\sup_{p \in \mathcal{P}_{\text{Lip.}}} \Big\{ \mathbb{E}\Big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p))) \Big] - \mathcal{R}_p^* \Big\} \geqslant C n^{-2/(d+2)}$$

  – Need $n \geqslant C(1/\varepsilon)^{d/2+1}$ to reach excess risk ε

- **Unavoidable**

Curse of dimensionality on $\mathcal{X} = \mathbb{R}^d$

- **Weak assumption**: optimal function f^* is Lipschitz-continuous

$$\exists L,\ \forall x, x' \in \mathcal{X},\ \ |f^*(x) - f^*(x')| \leqslant L\|x - x'\|$$

  – Denote $\mathcal{P}_{\text{Lip.}}$ the corresponding set of probability distributions

- **Lower bound on worst case performance** (Tsybakov, 2008)

$$\sup_{p \in \mathcal{P}_{\text{Lip.}}} \Big\{ \mathbb{E}\Big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\Big] - \mathcal{R}_p^* \Big\} \geqslant C n^{-2/(d+2)}$$

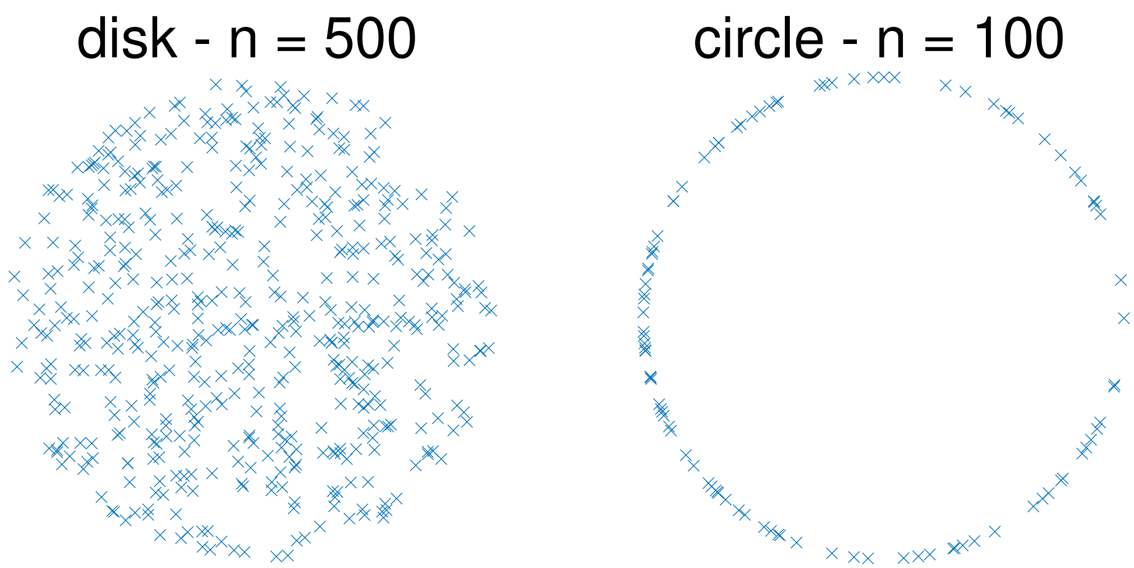
  – Need $n \geqslant C(1/\varepsilon)^{d/2+1}$ to reach excess risk ε

- **Unavoidable without extra assumptions**

  – Examples: support of inputs, smoothness and latent variables

Support of inputs

- **Assumption**
  - Input data only occupy a low-dimensional subspace or manifold
  - Dimension $r < d$



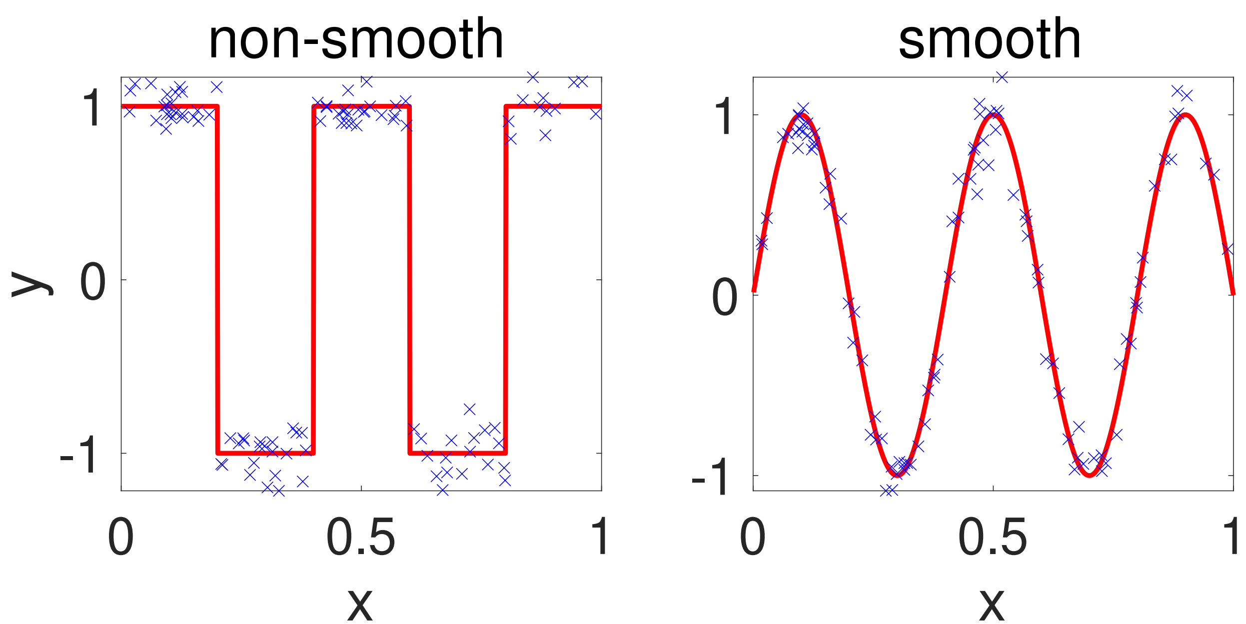
Support of inputs

- **Assumption**
  - Input data only occupy a low-dimensional subspace or manifold
  - Dimension $r < d$

- **Effect on learning algorithms**
  - Replace d by r in rates $\Rightarrow$ replace $n^{-2/(d+2)}$ by $n^{-2/(r+2)}$
  - Can reasonably estimated easily / directly from data
  - Most algorithms automatically adapt to it

Smoothness of the prediction function

- **Assumption**
  - Bounded s-th order derivatives
  - Order $s > 1$



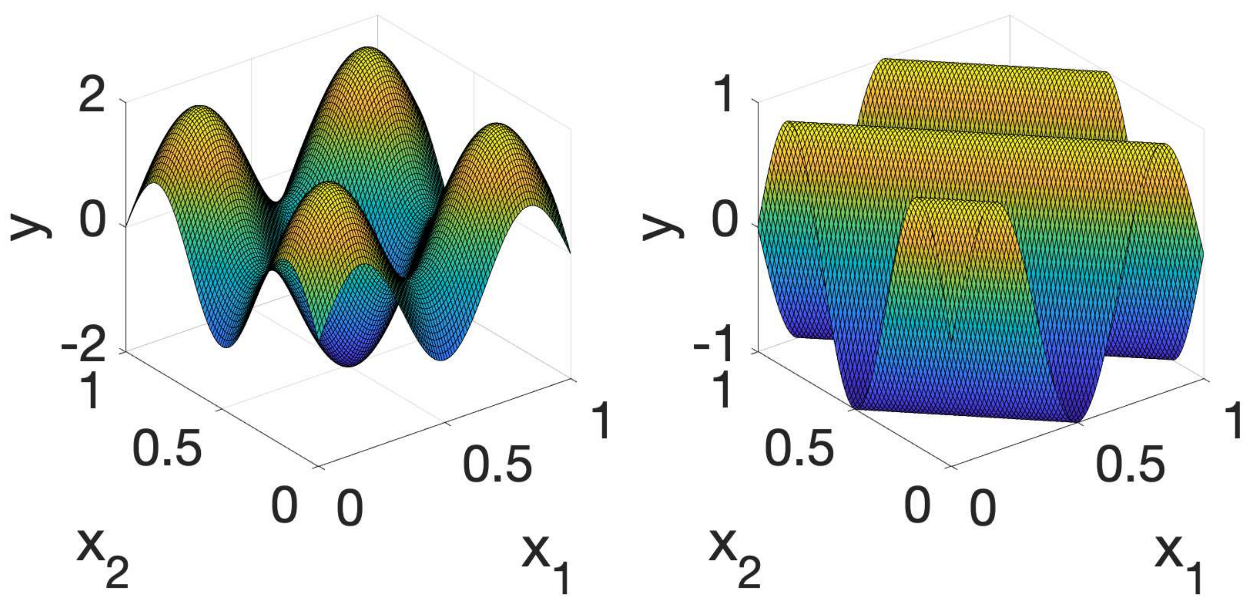
Smoothness of the prediction function

- **Assumption**
  - Bounded s-th order derivatives
  - Order $s > 1$

- **Effect on learning algorithms**
  - Replace d by d/s in rates $\Rightarrow$ replace $n^{-2/(d+2)}$ by $n^{-2/(d/s+2)}$
  - See, e.g., Györfi et al. (2002); Tsybakov (2008)
  - Cannot be easily / directly estimated from data
  - Algorithms may or may not adapt to it

Latent variables

- **Assumption**
  - Dependence only on unknown r-dimensional projection of the data
  - Dimension $r < d$



Latent variables

- **Assumption**
  - Dependence only on unknown r-dimensional projection of the data
  - Dimension $r < d$
- **Effect on learning algorithms**
  - Replace d by r in rates $\Rightarrow$ replace $n^{-2/(d+2)}$ by $n^{-2/(r+2)}$
  - See, e.g., Tong et al. (2002); Fukumizu et al. (2009)
  - Cannot be easily estimated from data
  - Algorithms may or may not adapt to it

Need for adaptivity

- **Unknown properties**
  - Support of inputs, smoothness and latent variables
  - Other (problem-dependent) properties could be considered

Need for adaptivity

- **Unknown properties**
  - Support of inputs, smoothness and latent variables
  - Other (problem-dependent) properties could be considered
- **Adaptivity of a learning algorithm**
  - With the proper choice of hyperparameters
  - Benefit from the assumption
  - Hopefully with a "logarithmic" cost

Need for adaptivity

- **Unknown properties**
  - Support of inputs, smoothness and latent variables
  - Other (problem-dependent) properties could be considered
- **Adaptivity of a learning algorithm**
  - With the proper choice of hyperparameters
  - Benefit from the assumption
  - Hopefully with a "logarithmic" cost
- **Quest for adaptivity: who wins?**
  - Barring computational and optimization issues

$$\text{local averaging} < \text{positive definite kernels} < \text{neural networks}$$

Guarantees for optimization

- **Common way of obtaining estimators**

- **Two different classes of functions**

  1. Convex
  2. Non convex

Convex optimization problems

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \quad \ell\big(y_i, f_\theta(x_i)\big) \quad + \quad \lambda \Omega(\theta)$$

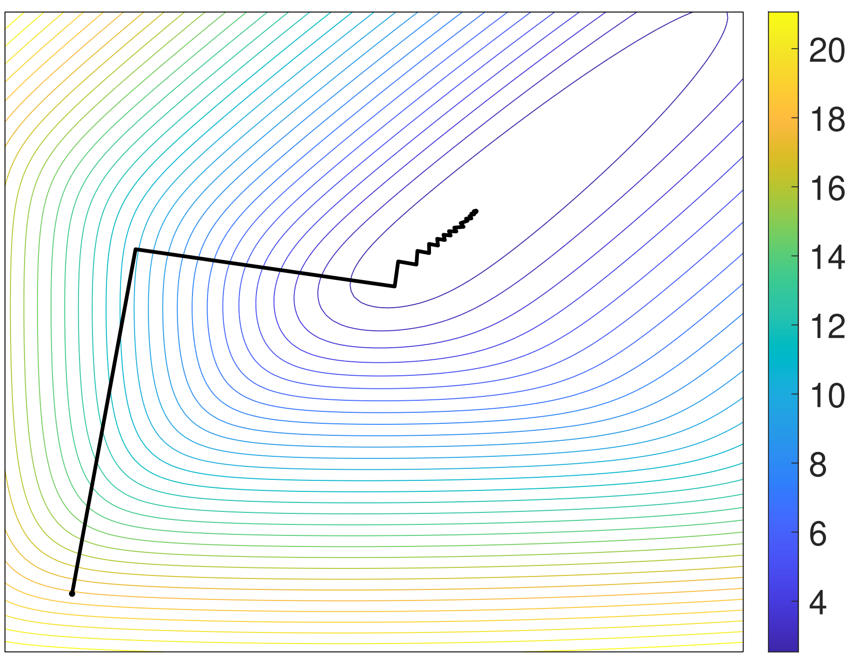
- **Conditions**: Convex loss and "linear" predictions $f_\theta(x) = \theta^\top \Phi(x)$
- **Consequences**
  - Efficient algorithms (typically gradient-based)
  - Quantitative runtime and prediction performance guarantees

Deterministic and stochastic methods

- Minimize $g(\theta) = \dfrac{1}{n}\sum_{i=1}^{n} h_i(\theta)$ with $h_i(\theta) = \ell\big(y_i, f_\theta(x_i)\big) + \lambda\Omega(\theta)$

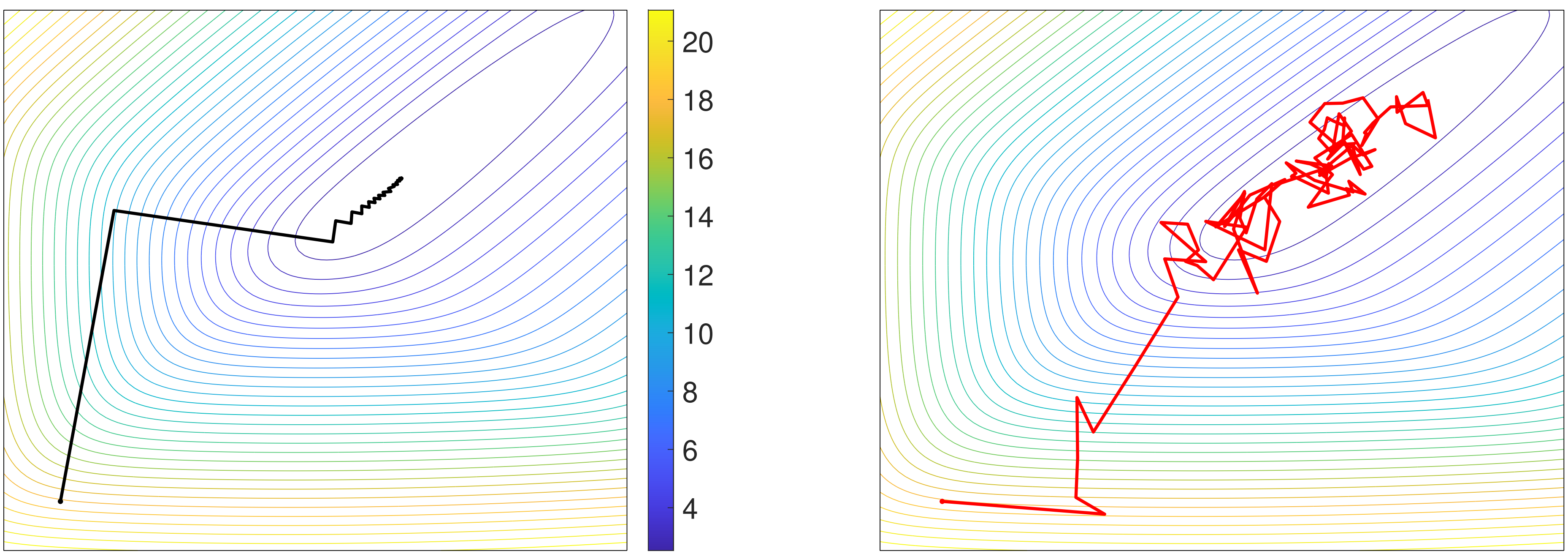
Deterministic and stochastic methods

- Minimize $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} h_i(\theta)$ with $h_i(\theta) = \ell\big(y_i, f_\theta(x_i)\big) + \lambda\Omega(\theta)$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma\nabla g(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma}{n}\sum_{i=1}^{n}\nabla h_i(\theta_{t-1})$ (Cauchy, 1847)



Deterministic and stochastic methods

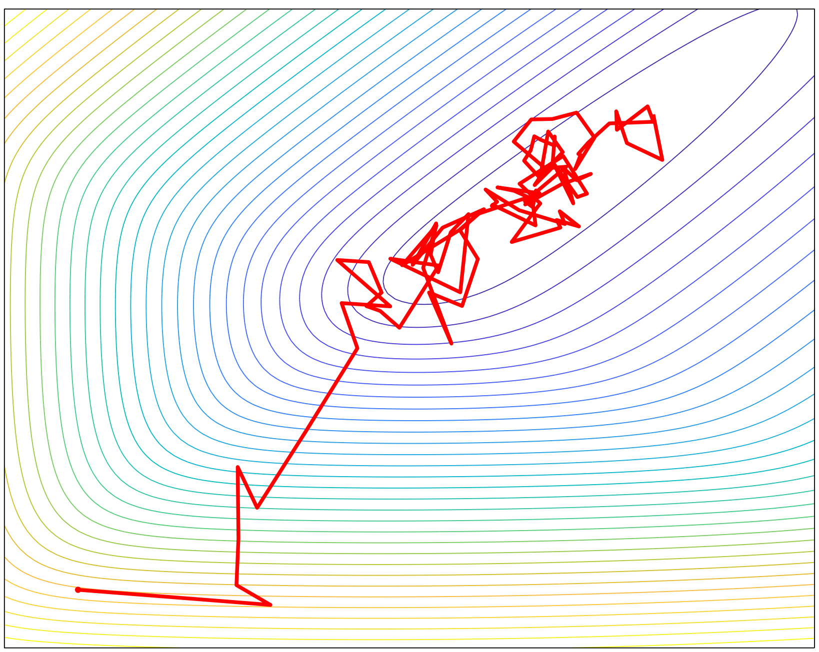
- Minimize $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} h_i(\theta)$ with $h_i(\theta) = \ell\big(y_i, f_\theta(x_i)\big) + \lambda\Omega(\theta)$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma \nabla g(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma}{n} \sum_{i=1}^{n} \nabla h_i(\theta_{t-1})$ (Cauchy, 1847)
- **Stochastic gradient descent**: $\theta_t = \theta_{t-1} - \gamma \nabla h_{i(t)}(\theta_{t-1})$ (Robbins and Monro, 1951)



Stochastic gradient with exponential convergence

- **Variance reduction**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014)

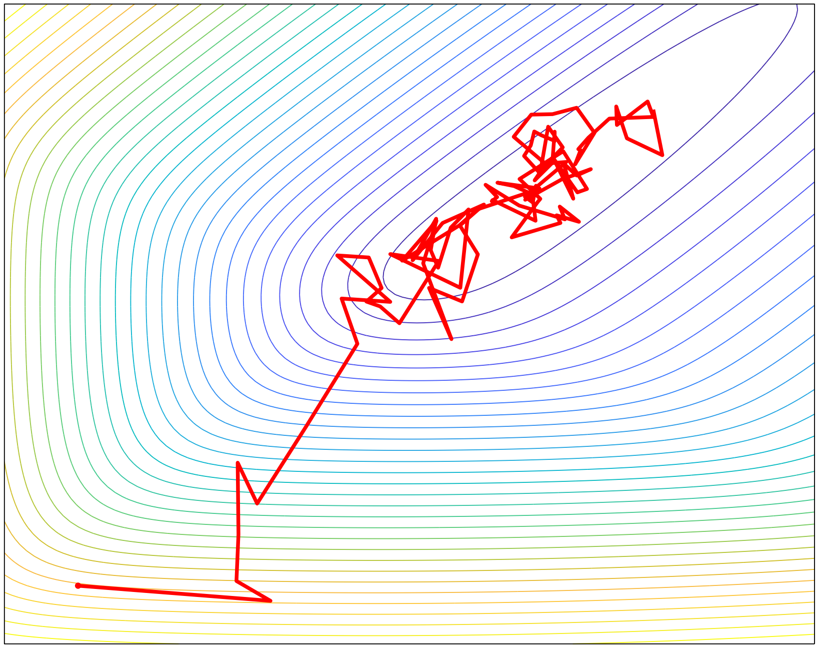
$$\theta_t = \theta_{t-1} - \gamma \Big[\nabla h_{i(t)}(\theta_{t-1}) \qquad\qquad\qquad\qquad \Big]$$

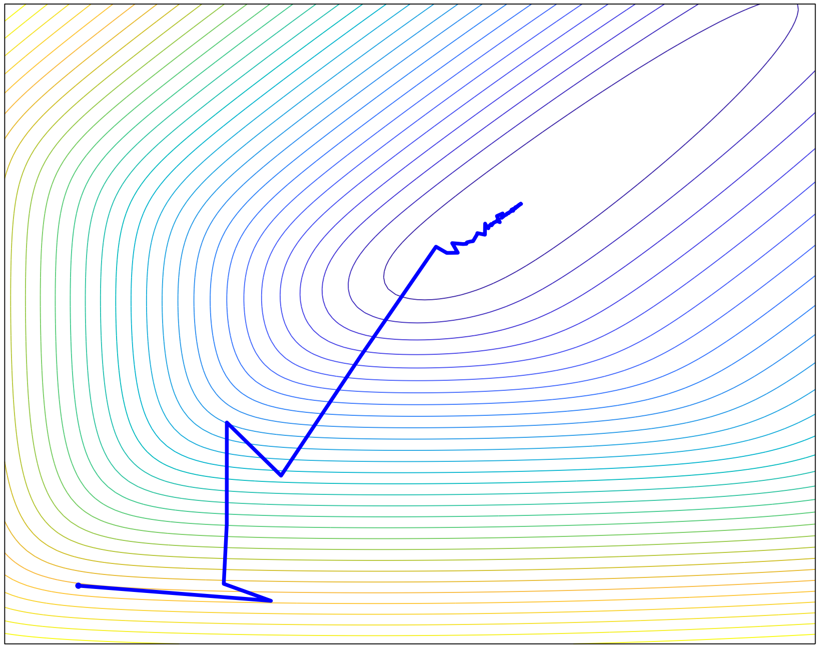


Stochastic gradient with exponential convergence

- **Variance reduction**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014)

$$\theta_t = \theta_{t-1} - \gamma\Big[\nabla h_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^{n} y_i^{t-1} - y_{i(t)}^{t-1}\Big]$$





Stochastic gradient with exponential convergence

- **Variance reduction**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014)

- **Number of individual gradient computations to reach error** ε (convex objectives with condition number κ)

Gradient descent	$n\kappa \quad \times \log \frac{1}{\varepsilon}$
Stochastic gradient descent	$\kappa \quad \times \quad \frac{1}{\varepsilon}$
Variance reduction	$(n+\kappa) \quad \times \log \frac{1}{\varepsilon}$

Stochastic gradient with exponential convergence

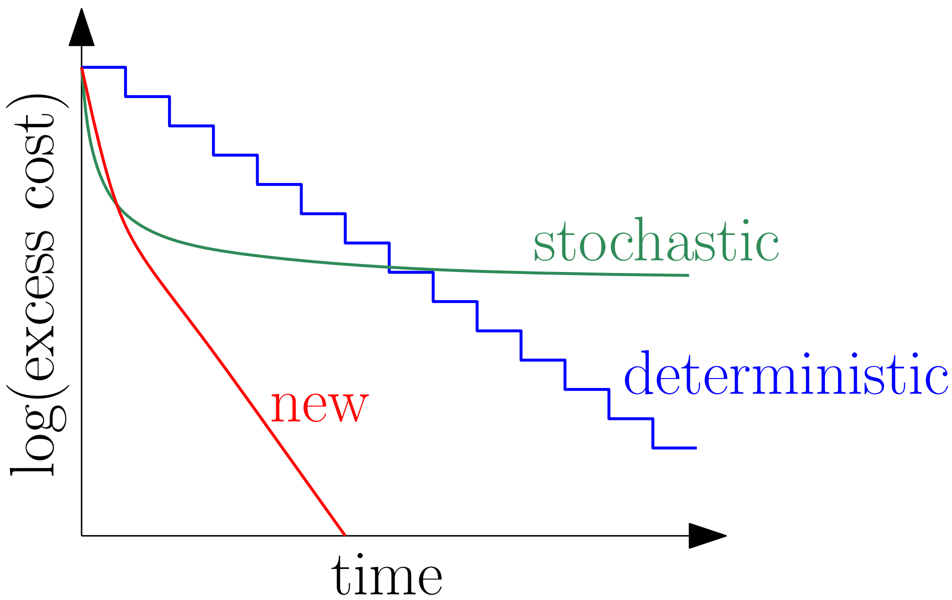
- **Variance reduction**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014)

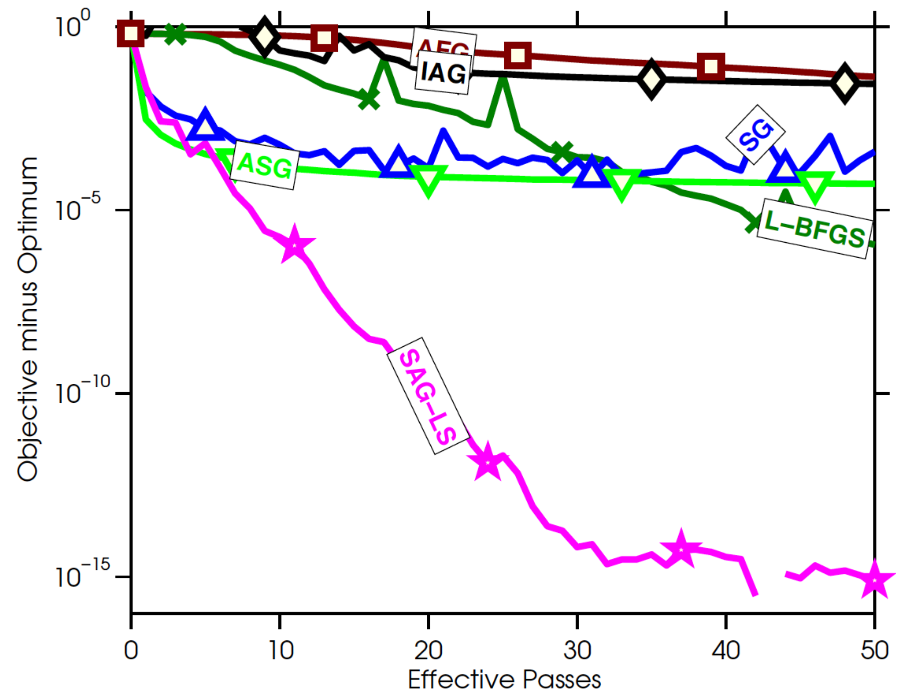
- **Number of individual gradient computations to reach error** ε (convex objectives with condition number κ)

Gradient descent	$n\kappa \quad \times \log \frac{1}{\varepsilon}$
Stochastic gradient descent	$\kappa \quad \times \quad \frac{1}{\varepsilon}$
Variance reduction	$(n+\kappa) \quad \times \log \frac{1}{\varepsilon}$

- **Empirical behavior close to complexity bounds**

Stochastic gradient with exponential convergence
From theory to practice and vice-versa





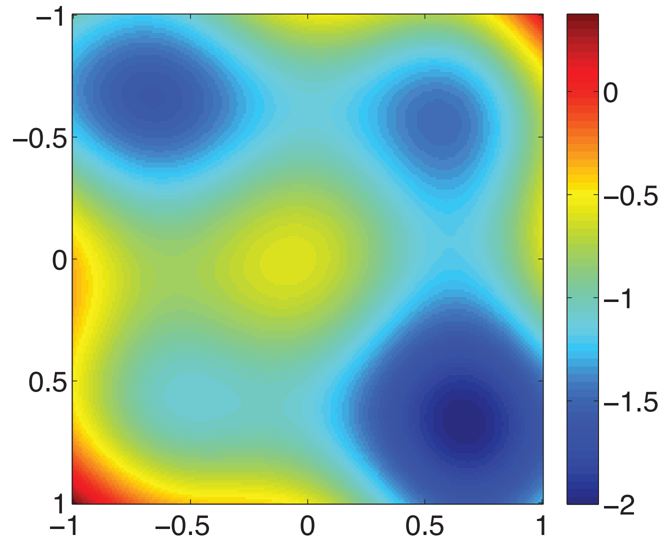
- **Empirical performance "matches" theoretical guarantees**

- **Theoretical analysis suggests practical improvements**
  - Non-uniform sampling, acceleration
  - Matching upper and lower bounds

Beyond convex optimization

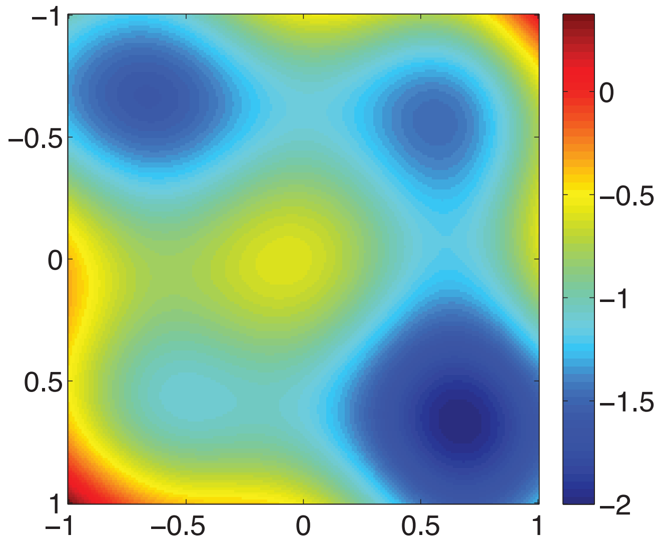
- **What can go wrong with non-convex optimization problems?**

  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...



Beyond convex optimization

- **What can go wrong with non-convex optimization problems?**
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...

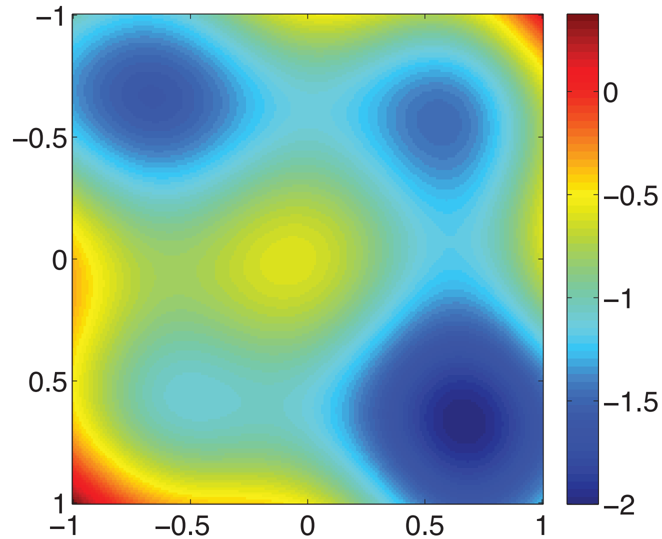


- **Generic local theoretical guarantees**
  - Convergence to stationary points or local minima
  - See, e.g., Lee et al. (2016); Jin et al. (2017)

Beyond convex optimization

- **What can go wrong with non-convex optimization problems?**

  – Local minima
  – Stationary points
  – Plateaux
  – Bad initialization
  – etc...



- **General global performance guarantees impossible to obtain**

Beyond convex optimization

- **Neural networks**
  - No guaranteed polynomial-time training
  - Qualitative benefits of over-parameterization (Chizat and Bach, 2018)

Beyond convex optimization

- **Neural networks**
  - No guaranteed polynomial-time training
  - Qualitative benefits of over-parameterization (Chizat and Bach, 2018)

- **Global optimization**
  - Only access to n evaluations of f
  - Cannot avoid the curse of dimensionality $\varepsilon = \frac{1}{n^{1/d}}$
  - Smooth functions allow $\varepsilon = \frac{1}{n^{s/d}}$
  - Polynomial-time algorithms with "sums-of-squares" (Lasserre, 2001; Rudi, Marteau-Ferey, and Bach, 2020)

Formal guarantees in ML, statistics, and optimization
Conclusion

- **Need for guarantees**
  - Computational vs. statistical guarantees
  - Guarantees of the training algorithms vs. of the prediction function
  - A priori vs. a posteriori guarantees
- **Many open problems within machine learning**
  - Probabilistic inference
  - Robust optimization
  - etc.

References

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

S. Boucheron, O. Bousquet, G. Lugosi, et al. Theory of classification: A survey of some recent advances. *ESAIM Probability and statistics*, 9:323–375, 2005.

M. A. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes rendus des séances de l'Académie des sciences*, 25(1):536–538, 1847.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, pages 3036–3046, 2018.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014.

L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Springer, February 1997.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Kenji Fukumizu, Francis Bach, and Michael I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

László Györfi, Michael Kohler, Adam Krzyzak, Harro Walk, et al. *A distribution-free theory of nonparametric regression*, volume 1. Springer, 2002.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

Jean-Bernard Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22:400–407, 1951.

Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding global minima via kernel approximations. Technical Report 2012.11978, arXiv, 2020.

Howell Tong, Y Xia, and L. Zhu. An adaptive estimation of dimension reduction space, with discussion. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 64(3):363–410, 2002.

A. B. Tsybakov. Introduction to nonparametric estimation. 2008.

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, 2013.