Northwestern University

Bayesian Statistics & Machine Learning Working Group

Gradient-Based Optimization Algorithms in Machine Learning II

October 27, 2021

Tim Tsz-Kit Lau
4th Year Ph.D. Candidate
Department of Statistics
Northwestern University
timlautk@u.northwestern.edu; https://timlautk.github.io

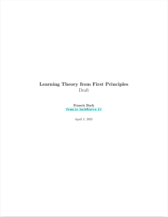
Today's Roadmap

1. Gradient methods on nonsmooth problems
2. Mirror descent
3. Nonconvex optimization for machine learning
4. *More differentiable programming*:
   Implicit differentiation of optimization problems

Materials

Content mainly taken from:

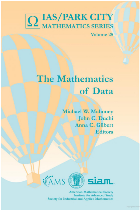
- Learning Theory from First Principles by *Francis Bach* (Chapter 5) (Bach, 2021)
- The Mathematics of Data: Chapter—Introductory Lectures on Stochastic Optimization by *John C. Duchi* (Duchi, 2018)
- On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points *(J. ACM)* by *Jin, Netrapalli, Ge, Kakade, Jordan* (Jin et al., 2021)
- Optimization Methods for Large-Scale Machine Learning *(SIAM Review)* by *Bottou, Curtis, Nocedal* (Bottou et al., 2018)



Learning Theory from First Principles
Draft

Francis Bach
francis.bach@inria.fr

April 5, 2021



IAS/PARK CITY MATHEMATICS SERIES
Volume 25

The Mathematics of Data

Michael W. Mahoney
John C. Duchi
Anna C. Gilbert
Editors

AMS siam

American Mathematical Society
Institute for Advanced Study
Society for Industrial and Applied Mathematics

On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points

CHI JIN, University of California, Berkeley
PRANEETH NETRAPALLI, Microsoft Research, India
RONG GE, Duke University
SHAM M. KAKADE, University of Washington, Seattle
MICHAEL I. JORDAN, University of California, Berkeley

Gradient descent (GD) and stochastic gradient descent (SGD) are the workhorses of large-scale machine learning. While classical theory focused on analyzing the performance of these methods in convex optimization problems, the most notable successes in machine learning have involved *nonconvex* optimization, and a gap has arisen between theory and practice. Indeed, traditional analyses of GD and SGD show that both algorithms converge to stationary points efficiently. But these analyses do not take into account the possibility of converging to saddle points. More recent theory has shown that GD and SGD can avoid saddle points, but the dependence on dimension in these analyses is polynomial. For modern machine learning, where the dimension can be in the millions, such dependence would be catastrophic. We analyze perturbed versions of GD and SGD and show that they are truly efficient—their dimension dependence is only polylogarithmic. Indeed, these algorithms converge to second-order stationary points in essentially the same time as they take to converge to classical first-order stationary points.

11

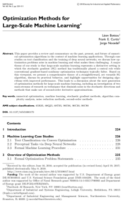
CCS Concepts: • **Theory of computation → Nonconvex optimization; Machine learning theory;**

Additional Key Words and Phrases: Saddle points, (stochastic) gradient descent, perturbations, efficiency

ACM Reference format:
Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. 2021. On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points. J. *ACM* 68, 2, Article 11 (February 2021), 29 pages.
https://doi.org/10.1145/3418526

A preliminary version of this article, with a subset of the results that are presented here, was presented at ICML 2017 and appeared in the proceedings as Jin et al. [27].
We thank Yair Carmon, Tongyang Li and Quanquan Gu for valuable discussions. This work was supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-18-1-2764. Rong Ge also acknowledges the funding support from NSF CCF-1704656, NSF CCF-1845171 (CAREER), Sloan Fellowship and Google Faculty Research Award.
Authors' addresses: C. Jin and M. I. Jordan, University of California, 2578 Hearst Ave, Berkeley, California, USA; emails: {chijin, jordan}@cs.berkeley.edu; P. Netrapalli, Microsoft Research, No. 9, Lavelle Road, Bengaluru, Karnataka, India; email: praneeth@microsoft.com; R. Ge, Duke University, 308 Research Dr, Durham, North Carolina, USA; email: rongge@cs.duke.edu; S. M. Kakade, University of Washington, 185 E Stevens Way NE, Seattle, Washington, USA; email: sham@cs.washington.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
0004-5411/2021/02-ART11 $15.00
https://doi.org/10.1145/3418526

Journal of the ACM, Vol. 68, No. 2, Article 11. Publication date: February 2021.



SIAM REVIEW
Vol. 60, No. 2, pp. 223–311

© 2018 Society for Industrial and Applied Mathematics

Optimization Methods for Large-Scale Machine Learning*

Léon Bottou[†]
Frank E. Curtis[‡]
Jorge Nocedal[§]

Abstract. This paper provides a review and commentary on the past, present, and future of numerical optimization algorithms in the context of machine learning applications. Through case studies on text classification and the training of deep neural networks, we discuss how optimization problems arise in machine learning and what makes them challenging. A major theme of our study is that large-scale machine learning represents a distinctive setting in which the stochastic gradient (SG) method has traditionally played a central role while conventional gradient-based nonlinear optimization techniques typically falter. Based on this viewpoint, we present a comprehensive theory of a straightforward, yet versatile SG algorithm, discuss its practical behavior, and highlight opportunities for designing algorithms with improved performance. This leads to a discussion about the next generation of optimization methods for large-scale machine learning, including an investigation of two main streams of research on techniques that diminish noise in the stochastic directions and methods that make use of second-order derivative approximations.

Key words. numerical optimization, machine learning, stochastic gradient methods, algorithm complexity analysis, noise reduction methods, second-order methods

AMS subject classifications. 65K05, 68Q25, 68T05, 90C06, 90C30, 90C90

DOI. 10.1137/16M1080173

Contents

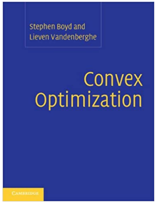
1	**Introduction**	**224**
2	**Machine Learning Case Studies**	**226**
	2.1 Text Classification via Convex Optimization	226
	2.2 Perceptual Tasks via Deep Neural Networks	228
	2.3 Formal Machine Learning Procedure	231
3	**Overview of Optimization Methods**	**235**
	3.1 Formal Optimization Problem Statements	235

[*] Received by the editors June 16, 2016; accepted for publication (in revised form) April 19, 2017; published electronically May 8, 2018.
http://www.siam.org/journals/sirev/60-2/M108017.html
Funding: The work of the second author was supported by U.S. Department of Energy grant DE-SC0010615 and U.S. National Science Foundation grant DMS-1016291. The work of the third author was supported by Office of Naval Research grant N00014-14-1-0313 P00003 and Department of Energy grant DE-FG02-87ER25047s.
[†] Facebook AI Research, New York, NY 10003 (leon@bottou.org).
[‡] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (frank.e.curtis@gmail.com).
[§] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60201 (j-nocedal@northwestern.edu).

Additional Materials

If you are interested in going deeper into optimization (mostly convex and some nonconvex), you might refer to

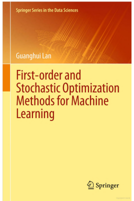
- Convex Optimization by *Stephen Boyd* and *Lieven Vandenberghe*
- First-order and Stochastic Optimization Methods for Machine Learning by *Guanghui Lan*
- First-Order Methods in Optimization by *Amir Beck*
- Convex Optimization Algorithm and Nonlinear Programming (3rd edition) by *Dimitri P. Bertsekas*



Stephen Boyd and Lieven Vandenberghe

Convex Optimization

CAMBRIDGE

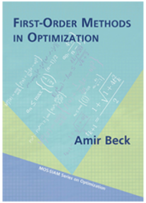


Springer Series in the Data Sciences

Guanghui Lan

First-order and Stochastic Optimization Methods for Machine Learning

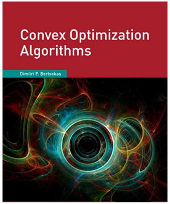
Springer



First-Order Methods in Optimization

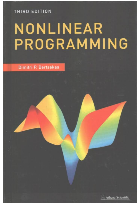
Amir Beck

MOS-SIAM Series on Optimization



Convex Optimization Algorithms

Dimitri P. Bertsekas



THIRD EDITION

NONLINEAR PROGRAMMING

Dimitri P. Bertsekas

Athena Scientific

Why Gradient-Based Optimization?

- In large-scale machine learning (esp. deep learning), complex algorithms (e.g., second-order methods) are generally *infeasible*
- Automatic differentiation libraries (TensorFlow, PyTorch, JAX)
- The workhorse first-order algorithm for optimization (if no higher-order info is used)

Optimization in Machine Learning

- Supervised machine learning: observed samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where each couple of random variables $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ follows an unknown distribution P
- **Goal**: find a predictor $f \colon \mathcal{X} \to \mathbb{R}$ which minimizes the risk

$$\mathcal{R}(f) := \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}))]$$

  where $\ell \colon \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a loss function (usually convex in the second argument)
- In *empirical risk minimization (ERM)*, we minimize the *empirical risk* over a *parameterized* set of predictors $\{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^d}$ with a regularizer $h \colon \mathbb{R}^d \to \mathbb{R}$, i.e.,

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\text{minimize}} \;\; F(\boldsymbol{\theta}) := \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))}_{\text{empirical risk } \widehat{\mathcal{R}}(f_{\boldsymbol{\theta}})} + \underbrace{h(\boldsymbol{\theta})}_{\text{regularizer}}$$

Examples

- Regularized linear regression:

$$F(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}(\langle\boldsymbol{\theta}, \boldsymbol{x}_i\rangle - y_i)^2 + h(\boldsymbol{\theta})$$

- Regularized logistic regression: $y_i \in \{\pm 1\}$,

$$F(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i\langle\boldsymbol{\theta}, \boldsymbol{x}_i\rangle)) + \underbrace{\frac{\mu}{2}\|\boldsymbol{\theta}\|^2}_{\text{Default in sklearn. Why?}} + h(\boldsymbol{\theta})$$

- Two-layer neural networks with weight decay: $m = \#$ of neurons, $\Theta_1 \in \mathbb{R}^{m\times d}$, $\Theta_2 \in \mathbb{R}^{1\times m}$, $\boldsymbol{\theta} = \mathrm{vec}(\Theta_1, \Theta_2)$, ρ (nonlinear) activation function

$$F(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\Theta_2\rho(\Theta_1\boldsymbol{x}_i) - y_i)^2 + \frac{\mu}{2}\|\boldsymbol{\theta}\|^2 + h(\boldsymbol{\theta})$$

Accuracy of Iterative Algorithms

- Let $\boldsymbol{\theta}_\star \in \operatorname{Argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} \mathcal{R}(f_{\boldsymbol{\theta}})$ be a minimizer of the risk
- We can decompose the difference between the risk of the estimated predictor and the smallest risk by

$$\begin{aligned}\mathcal{R}(f_{\widehat{\boldsymbol{\theta}}}) - \inf_{\boldsymbol{\theta}\in\mathbb{R}^d} \mathcal{R}(f_{\boldsymbol{\theta}}) \\ = \underbrace{\left\{\mathcal{R}(f_{\widehat{\boldsymbol{\theta}}}) - \widehat{\mathcal{R}}(f_{\widehat{\boldsymbol{\theta}}})\right\}}_{\leqslant \text{ estimation error}} + \underbrace{\left\{\widehat{\mathcal{R}}(f_{\widehat{\boldsymbol{\theta}}}) - \widehat{\mathcal{R}}(f_{\boldsymbol{\theta}_\star})\right\}}_{\leqslant \text{ optimization error}} + \underbrace{\left\{\widehat{\mathcal{R}}(f_{\boldsymbol{\theta}_\star}) - \mathcal{R}(f_{\boldsymbol{\theta}_\star})\right\}}_{\leqslant \text{ estimation error}}\end{aligned}$$

- Suffice to reach an optimization accuracy of the order of the *estimation error*
- Estimation error usually of the order $O(1/\sqrt{n})$ or $O(1/n)$ (see Bach, 2021, Ch. 4)

Preliminaries

Convex Functions

Definition (Convex function)

A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is said to be *convex* if for any $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

$$f(\alpha\boldsymbol{\theta} + (1-\alpha)\boldsymbol{\eta}) \leqslant \alpha f(\boldsymbol{\theta}) + (1-\alpha) f(\boldsymbol{\eta}).$$

Equivalent definition

If f is differentiable, convexity of f is equivalent to: For any $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathbb{R}^d$,

$$f(\boldsymbol{\eta}) \geqslant f(\boldsymbol{\theta}) + \langle \nabla f(\boldsymbol{\theta}), \boldsymbol{\eta} - \boldsymbol{\theta} \rangle.$$

If f is twice differentiable, convexity of f is equivalent to the Hessian of f, denoted by $\nabla^2 f$, being positive semidefinite: For any $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\nabla^2 f(\boldsymbol{\theta}) \succcurlyeq \mathbf{0}_{d\times d} \quad \text{or} \quad \nabla^2 f(\boldsymbol{\theta}) \in \mathcal{S}_d^+.$$

Example of Convex Functions

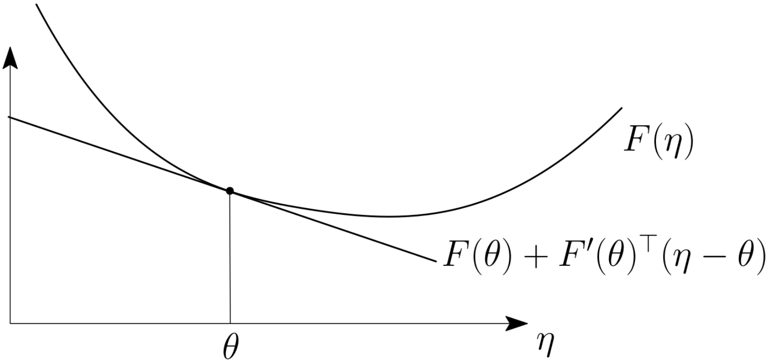


Figure: a convex function *F* (Bach, 2021)

Strong Convexity

Definition (Strong convexity)

A differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is said to be μ-*strongly convex* ($\mu > 0$) if for any $(\boldsymbol{\theta}, \boldsymbol{\eta}) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$f(\boldsymbol{\eta}) \geqslant f(\boldsymbol{\theta}) + \langle \nabla f(\boldsymbol{\theta}), \boldsymbol{\eta} - \boldsymbol{\theta} \rangle + \frac{\mu}{2} \| \boldsymbol{\eta} - \boldsymbol{\theta} \|^2 .$$

Equivalently, f is μ-strongly convex if $f + \frac{\mu}{2} \| \cdot \|^2$ is convex.

Equivalent definition

If f is twice differentiable, μ-strong convexity of f is equivalent to $\nabla^2 f$ having a μ-lower bounded spectrum (i.e., the smallest eigenvalue of $\nabla^2 f$ is lower bounded by μ): For any $\boldsymbol{\theta} \in \mathbb{R}^d$, $\sigma_{\min}(\nabla^2 f) \geqslant \mu \iff \nabla^2 f(\boldsymbol{\theta}) \succcurlyeq \mu \boldsymbol{I}_d$.

Example of Strongly Convex Functions

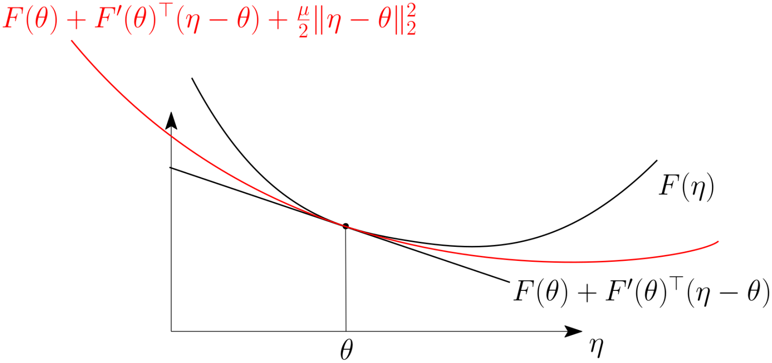
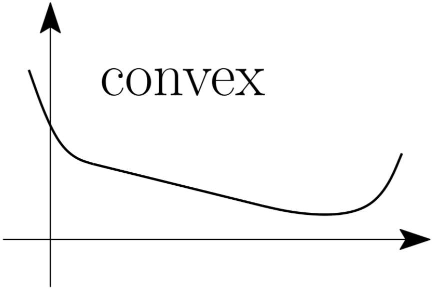
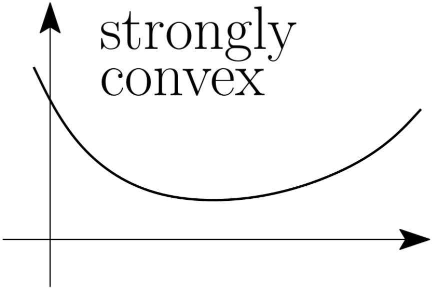

$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
θ
η

Figure: a strongly convex function *F* (Bach, 2021)

Convex Function vs. Strongly Convex Function





Lipschitz Continuity

Definition (Lipschitz continuity)

A function $f\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ is said to be *L-Lipschitz continuous* ($L > 0$) if for any $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathbb{R}^{d_1}$,

$$\|f(\boldsymbol{\theta}) - f(\boldsymbol{\eta})\| \leqslant L\|\boldsymbol{\theta} - \boldsymbol{\eta}\|.$$

Equivalent definition

If f is differentiable, L-Lipschitz continuity of f is equivalent to f having a bounded gradient by L: For any $\boldsymbol{\theta} \in \mathbb{R}^{d}$,

$$\|\nabla f(\boldsymbol{\theta})\| \leqslant L.$$

Smoothness

Definition (Smoothness)

A differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is said to be *L-smooth* ($L > 0$) if for any $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathbb{R}^d$,

$$|f(\boldsymbol{\eta}) - f(\boldsymbol{\theta}) - \langle \nabla f(\boldsymbol{\theta}), \boldsymbol{\eta} - \boldsymbol{\theta} \rangle| \leqslant \frac{L}{2} \|\boldsymbol{\theta} - \boldsymbol{\eta}\|^2.$$

Equivalent definition

L-smoothness of f is equivalent to f having a L-Lipschitz continuous gradient:
For any $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathbb{R}^d$, $\|\nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\eta})\| \leqslant L\|\boldsymbol{\theta} - \boldsymbol{\eta}\|$.
Moreover, if f is twice differentiable, then this is equivalent to $\nabla^2 f$ having an bounded spectrum: For any $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$-L\boldsymbol{I}_d \preccurlyeq \nabla^2 f(\boldsymbol{\theta}) \preccurlyeq L\boldsymbol{I}_d \iff -L \leqslant \sigma_{\min}(\nabla^2 f) \leqslant \sigma_{\max}(\nabla^2 f) \leqslant L.$$

Example of Smooth Functions

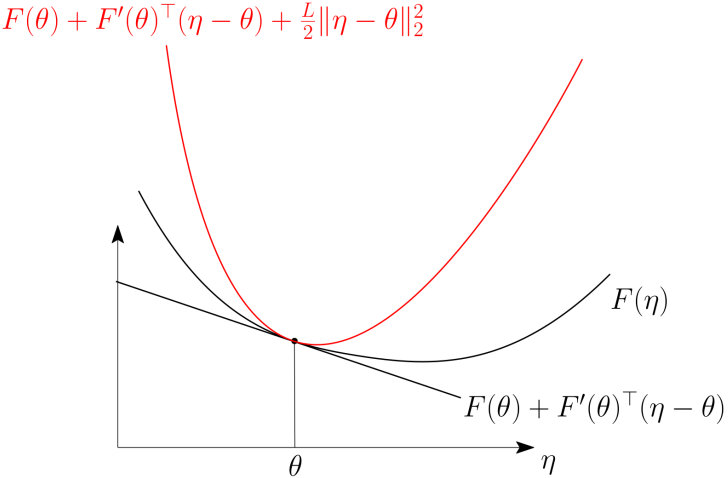

$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{L}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
θ
η

Figure: a smooth function *F* (Bach, 2021)

Some Remarks

- *(Strong) convexity* and *smoothness* are necessary conditions for **gradient descent** to converge to the *global minimum* at fast rates
  (Recall: *quadratic* upper and lower bounds of F)
- We will later discuss the cases without **convexity** and without **smoothness** respectively, but not the *nonconvex nonsmooth* case
- The **nonconvex nonsmooth** case has *relatively few results* in the literature, yet is the realistic case in deep learning (e.g., deep ReLU networks)

Gradient Descent

Gradient Descent

- Let $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ and for $t = 0, 1, 2, \ldots$,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_{t+1} \nabla F(\boldsymbol{\theta}_t),$$

  where $(\alpha_t)_{t \geqslant 1}$ is a well chosen step size sequence
- Focus on the case where the step sizes depend *explicitly* on **problem constants** and sometimes on the **iteration number**

Convergence Analysis of Gradient Descent

How fast does gradient descent converge to the global minimum for (strongly) convex and smooth objective functions?

Simplest Example: Multivariate Linear Regression

- Response vector: $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$
- Design matrix: $\boldsymbol{X} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top \in \mathbb{R}^{n \times d}$

$$F(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|^2 = \frac{1}{2n}\sum_{i=1}^{n}(\langle \boldsymbol{x}_i, \boldsymbol{\theta}\rangle - y_i)^2$$

$$\nabla F(\boldsymbol{\theta}) = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})$$

- $\boldsymbol{H} \coloneqq \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} \in \mathbb{R}^{d \times d}$ is the *Hessian* matrix of F
- A minimizer $\boldsymbol{\theta}^\star$ always exists, but is unique only if $\boldsymbol{H}$ is invertible
- Note that a minimizer $\boldsymbol{\theta}^\star$ satisfies $\nabla F(\boldsymbol{\theta}^\star) = \mathbf{0}$, i.e.,

$$\frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\theta}^\star - \boldsymbol{y}) = \mathbf{0} \iff \boldsymbol{H}\boldsymbol{\theta}^\star = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{y}$$

Simplest Example: Multivariate Linear Regression

- Gradient descent with constant step sizes $\alpha_t = \alpha$:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha \nabla F(\boldsymbol{\theta}_{t-1}) = \boldsymbol{\theta}_{t-1} - \frac{\alpha}{n} \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{\theta}_{t-1} - \boldsymbol{y}) = \boldsymbol{\theta}_{t-1} - \alpha \boldsymbol{H}(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^\star)$$

which implies

$$\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star = \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^\star - \alpha \boldsymbol{H}(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^\star) = (\boldsymbol{I} - \alpha \boldsymbol{H})(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^\star)$$

- Recursively,

$$\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star = (\boldsymbol{I} - \alpha \boldsymbol{H})^t (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)$$

- Measures of performance:

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|^2 = (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)^\top (\boldsymbol{I} - \alpha \boldsymbol{H})^{2t} (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)$$

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) = \frac{1}{2} (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)^\top (\boldsymbol{I} - \alpha \boldsymbol{H})^{2t} \boldsymbol{H} (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)$$

Convergence in Distance to Minimizer

- For $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|^2 \to 0$, need a unique minimizer $\boldsymbol{\theta}^\star$ ($\boldsymbol{H}$ has to be *invertible*)
- For $\boldsymbol{H}$ to be invertible, need $\sigma_{\min}(\boldsymbol{H}) > 0$ (F is then $\sigma_{\min}(\boldsymbol{H})$-strongly convex)
- Since

  $$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|^2 = (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star)^\top (\boldsymbol{I} - \alpha \boldsymbol{H})^{2t} (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star),$$

  we bound the eigenvalues of $(\boldsymbol{I} - \alpha \boldsymbol{H})^{2t}$, which are $(1 - \alpha\sigma)^{2t}$ for σ an eigenvalue of $\boldsymbol{H}$
- Hence

  $$(1 - \alpha\sigma)^{2t} \leqslant \left(\max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} |1 - \alpha\sigma| \right)^{2t}$$

- We can find a constant step size $\alpha > 0$ minimizing the above, which might depend on both $\sigma_{\min}(\boldsymbol{H})$ and $\sigma_{\max}(\boldsymbol{H})$

Convergence in Distance to Minimizer

- To make a simplier choice $\alpha = 1/\sigma_{\max}(\boldsymbol{H})$, then

$$\max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} |1 - \alpha\sigma| = 1 - \frac{\sigma_{\min}(\boldsymbol{H})}{\sigma_{\max}(\boldsymbol{H})} = 1 - \frac{1}{\kappa},$$

  where $\kappa := \sigma_{\max}(\boldsymbol{H})/\sigma_{\min}(\boldsymbol{H})$ is the *condition number* of $\boldsymbol{H}$

- Then

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|^2 \leqslant \left(1 - \frac{1}{\kappa}\right)^{2t} \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star\|^2 \leqslant \mathrm{e}^{-2t/\kappa} \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star\|^2,$$

  which is often referred to as *exponential*, *geometric*, or also *linear* convergence (misleading: *linear* means $\log \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|^2$ decays linearly in t)

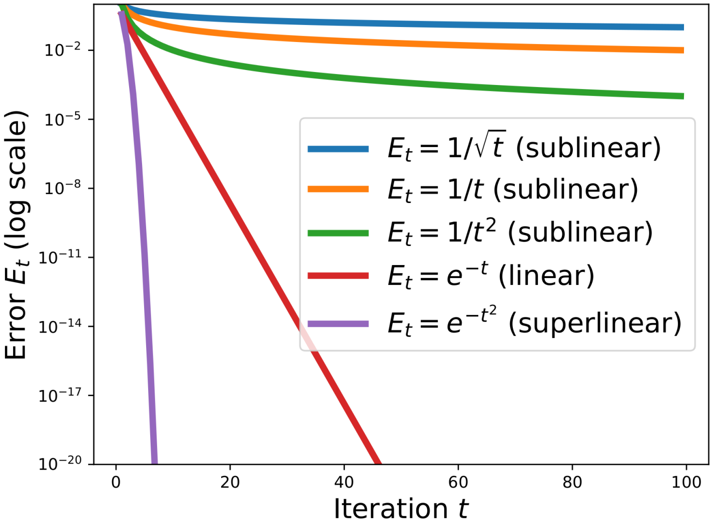
Convergence in Function Values

- Similarly, with $\alpha = 1/\sigma_{\text{max}}(\boldsymbol{H})$,

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) \leqslant \left(1 - \frac{1}{\kappa}\right)^{2t} [F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^\star)] \leqslant \mathrm{e}^{-2t/\kappa} [F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^\star)]$$

- Also *linear* convergence

Convergence Rates ($E_t = F(\theta_t) - F(\theta^\star)$ or $E_t = \|\theta_t - \theta^\star\|^2$)

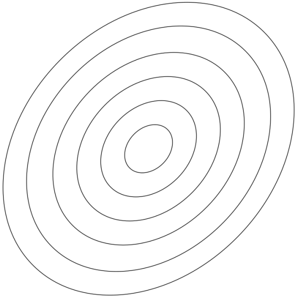


Analysis of GD for Strongly Convex and Smooth Functions

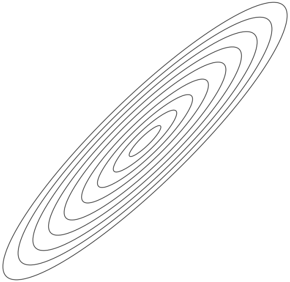
- Recall that for a μ-strongly convex and L-smooth function,

$$(\forall \boldsymbol{\theta} \in \mathbb{R}^d) \quad \mu \boldsymbol{I}_d \preccurlyeq \nabla^2 f(\boldsymbol{\theta}) \preccurlyeq L \boldsymbol{I}_d$$

- Define the *condition number* $\kappa := L/\mu \geqslant 1$ (i.e., $L \geqslant \mu$ is required)

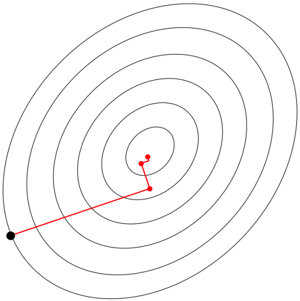


$(\text{small } \kappa = L/\mu)$

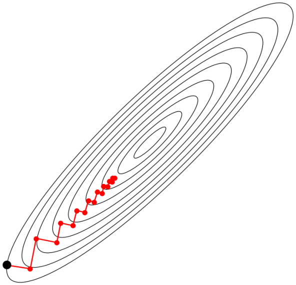


$(\text{large } \kappa = L/\mu)$

Analysis of GD for Strongly Convex and Smooth Functions



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

- *Small* $\kappa \implies$ fast convergence
- *Large* $\kappa \implies$ oscillations
- Recall the group meeting by Prof. Liu on Feb 4:
  Data normalization (for the design matrix $\boldsymbol{X}$)—reduces κ

Analysis of GD for Strongly Convex and Smooth Functions

- Gradient descent converges *exponentially* for *strongly convex* and *smooth* problems

Theorem

Assume that F is μ-strongly convex and L-smooth. Let $\alpha_t = 1/L$ for all $t \geqslant 0$ and $\theta^\star = \mathrm{argmin}_{\theta \in \mathbb{R}^d} F(\theta)$, then the iterates $(\theta_t)_{t \geqslant 0}$ of GD on F satisfy

$$F(\theta_t) - F(\theta^\star) \leqslant \left(1 - \frac{\mu}{L}\right)^t [F(\theta_0) - F(\theta^\star)] \leqslant \exp(-t\mu/L)[F(\theta_0) - F(\theta^\star)].$$

Analysis of GD for Convex and Smooth Functions

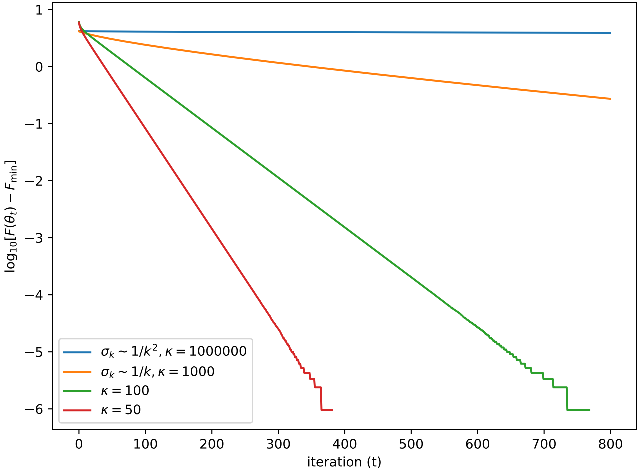
- Only assume *convexity* but not strong convexity of the function (i.e., $\mu = 0$)
- Gradient descent converges at an $O(1/t)$ rate for *convex* and *smooth* problems

Theorem

Assume that F is convex and L-smooth. Let $\alpha_t = 1/L$ for all $t \geqslant 0$ and $\boldsymbol{\theta}^\star = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} F(\boldsymbol{\theta})$, then the iterates $(\boldsymbol{\theta}_t)_{t \geqslant 0}$ of GD on F satisfy

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) \leqslant \frac{L}{2t} \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^\star\|^2.$$

Experiments (Google Colab)



Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent

- Recall the regularized empirical risk minimization problem

$$F(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))) + h(\boldsymbol{\theta})$$

- If n is large, it is costly to compute the full gradient $\nabla F(\boldsymbol{\theta}_t)$
- Instead, only compute *unbiased stochastic estimations of the gradient* $\boldsymbol{g}_t$
- Let $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ and for $t = 0, 1, 2, \ldots$,

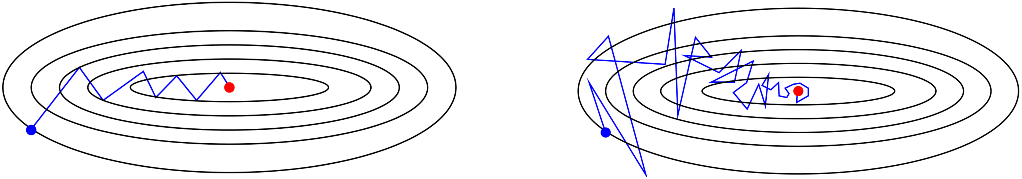
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_{t+1}\boldsymbol{g}_t(\boldsymbol{\theta}_t),$$

with the step size sequence $(\alpha_t)_{t \geqslant 1}$

Stochastic Gradient ~~Descent~~ Method

Remark

Stochastic gradient descent is **NOT** a descent method:
The function values often go up .



(a) Gradient descent

(b) Stochastic gradient descent

Convergence Analysis of Stochastic Gradient Method

Extra assumptions:

- *Unbiased* gradient:

$$\mathbb{E}[\boldsymbol{g}_t(\boldsymbol{\theta}_t) \mid \boldsymbol{\theta}_t] = \nabla F(\boldsymbol{\theta}_t) \quad \text{for all } t \geqslant 0$$

- *Bounded* gradient:

$$\|\boldsymbol{g}_t(\boldsymbol{\theta}_t)\|^2 \leqslant B^2 \quad \text{almost surely, for all } t \geqslant 0$$

Analysis of SGD for Convex and Smooth Functions

- SGD converges at an $O(1/\sqrt{t})$ rate for *convex* and *smooth* problems, with $\alpha_t \propto 1/\sqrt{t}$

Theorem

Assume that F is convex, B-Lipschitz and admits a minimizer $\theta^\star = \operatorname{argmin}_{\theta \in \mathbb{R}^d} F(\theta)$ *which satisfies* $\|\theta^\star - \theta_0\| \leqslant D$. *Assume that the stochastic gradients* $(\boldsymbol{g}_t)_{t \geqslant 0}$ *are unbiased and bounded. Choosing* $\alpha_t = D/(B\sqrt{t})$, *the iterates* $(\theta_t)_{t \geqslant 0}$ *of SGD on F satisfy*

$$\mathbb{E}\big[F(\overline{\theta}_t) - F(\theta^\star)\big] \leqslant DB\frac{2 + \log t}{\sqrt{t}},$$

where $\overline{\theta}_t := \sum_{s=1}^t \alpha_s \theta_{s-1} / \sum_{s=1}^t \alpha_s$ *(the average iterate).*

Analysis of SGD for Convex and Smooth Functions

Facts:

- The bound in $O(BD/\sqrt{t})$ is optimal for this class of problems (impossible to have a better convergence rate)
- The number of iterations to reach a given precision will be larger for SGD (vs. GD), but n times faster in terms of running time complexity
- High precision $\implies$ GD
- Low precision and large n $\implies$ SGD

Analysis of SGD for Strongly Convex and Smooth Functions

- $G(\boldsymbol{\theta}) := F(\boldsymbol{\theta}) + \frac{\mu}{2}\|\boldsymbol{\theta}\|^2$ is μ-strongly convex if F is only convex
- Let $\boldsymbol{\theta}_0 = \mathbf{0}$ and for $t = 0, 1, 2, \ldots$,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_{t+1}[\boldsymbol{g}_t(\boldsymbol{\theta}_t) + \mu\boldsymbol{\theta}_t],$$

with the step size sequence $(\alpha_t)_{t\geqslant 1}$

Analysis of SGD for Strongly Convex and Smooth Functions

- SGD converges at an $O(1/t)$ rate for *strongly convex* and *smooth* problems, with $\alpha_t \propto 1/t$

Theorem

Assume that F is convex, B-Lipschitz and that $G := F + \frac{\mu}{2}\|\cdot\|^2$ admits a (necessarily unique) minimizer $\boldsymbol{\theta}^\star = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} G(\boldsymbol{\theta})$. Assume that the stochastic gradients $(\boldsymbol{g}_t)_{t\geqslant 0}$ are unbiased and bounded. Choosing $\alpha_t = 1/(\mu t)$, the iterates $(\boldsymbol{\theta}_t)_{t\geqslant 0}$ of SGD on F satisfy

$$\mathbb{E}\big[F(\overline{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)\big] \leqslant \frac{2B^2(1+\log t)}{\mu t},$$

where $\overline{\boldsymbol{\theta}}_t := \frac{1}{t}\sum_{s=1}^t \boldsymbol{\theta}_{s-1}$.

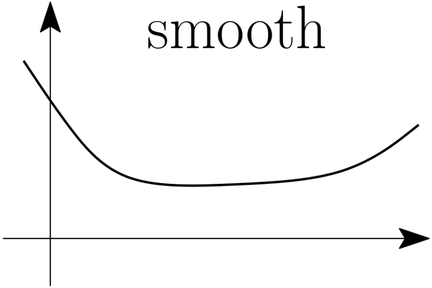
Analysis of SGD for Strongly Convex and Smooth Functions

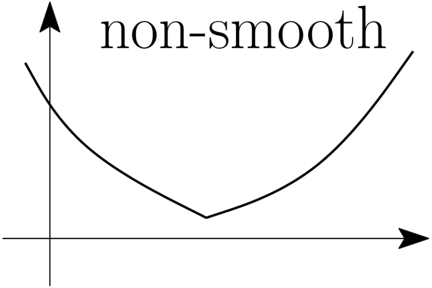
Facts:

- The bound in $O(B^2/\mu t)$ is optimal for this class of problems
- **Loss of adaptivity**: the step-size now depends on the difficulty of the problem

Gradient Methods on Nonsmooth Problems

Nonsmooth Functions



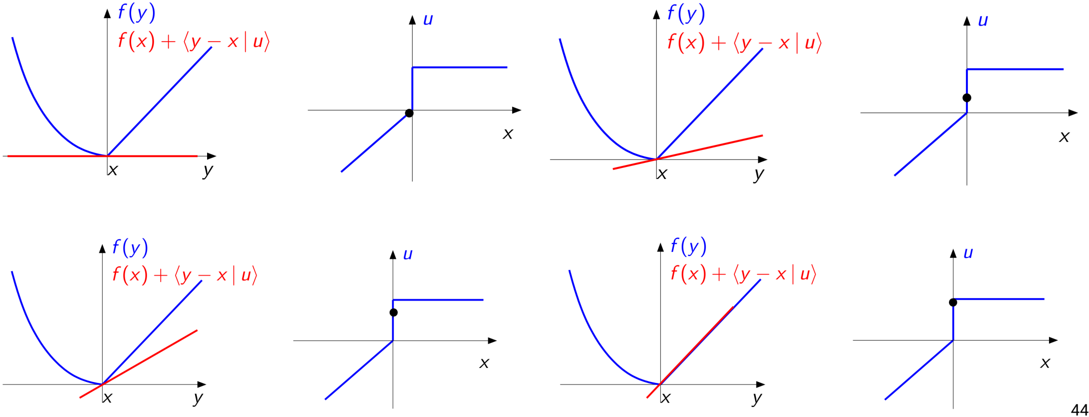


Subgradient Methods

- Assume that F is convex and Lipschitz continuous
- Although F is nonsmooth, it is still almost everywhere differentiable
- For such points, we define the **set** of slopes of lower-bounding tangents as the *subdifferential*, denoted by ∂F
- Any element of it is called a *subgradient*
- Use any subgradient of F in place of ∇F
- The subgradient method is **NOT** a descent method as well

Subdifferential

$$\partial f(x) := \{u \in \mathbb{R}^d : (\forall y \in \mathbb{R}^d)\; f(x) + \langle y - x, u\rangle \leqslant f(y)\}$$



$f(y)$
$f(x) + \langle y - x \,|\, u\rangle$
x
y
u
x
$f(y)$
$f(x) + \langle y - x \,|\, u\rangle$
x
y
u
x
$f(y)$
$f(x) + \langle y - x \,|\, u\rangle$
x
y
u
x
$f(y)$
$f(x) + \langle y - x \,|\, u\rangle$
x
y
u
x

Analysis of Subgradient Method for Convex, Lipschitz and Nonsmooth Functions

Theorem

Assume that F is convex, B-Lipschitz and admits a minimizer $\theta^\star = \operatorname{argmin}_{\theta \in \mathbb{R}^d} F(\theta)$ which satisfies $\|\theta^\star - \theta_0\| \leqslant D$. By choosing $\alpha_t = D/(B\sqrt{t})$, the iterates $(\theta_t)_{t \geqslant 0}$ of GD on F satisfy

$$\min_{0 \leqslant s \leqslant t-1} F(\theta_s) - F(\theta^\star) \leqslant DB \frac{2 + \log t}{\sqrt{t}}.$$

Mirror Descent

Mirror Descent

- For $\boldsymbol{\theta}$ constrained on a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, *projected* GD (PGD) has an interesting reformulation:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \mathrm{proj}_{\mathcal{C}}(\boldsymbol{\theta}_t - \alpha_t \nabla F(\boldsymbol{\theta}_t)) \\ \Leftrightarrow \quad \boldsymbol{\theta}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{C}} \left\{ F(\boldsymbol{\theta}_t) + \langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle + \frac{1}{2\alpha_t} \|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2 \right\} \end{aligned}$$

- Use distance-measuring functions other than the squared Euclidean norm
- Bregman divergence associated with a strictly convex and differentiable φ, defined by

$$D_\varphi(\boldsymbol{\theta}, \boldsymbol{\eta}) := \varphi(\boldsymbol{\theta}) - \varphi(\boldsymbol{\eta}) - \langle \nabla \varphi(\boldsymbol{\eta}), \boldsymbol{\theta} - \boldsymbol{\eta} \rangle$$

- Bregman divergence is **NOT** a distance (in *mathematical terms*) since it is **NOT** *symmetric* and might not satisfy the **triangle inequality**

Bregman Divergence

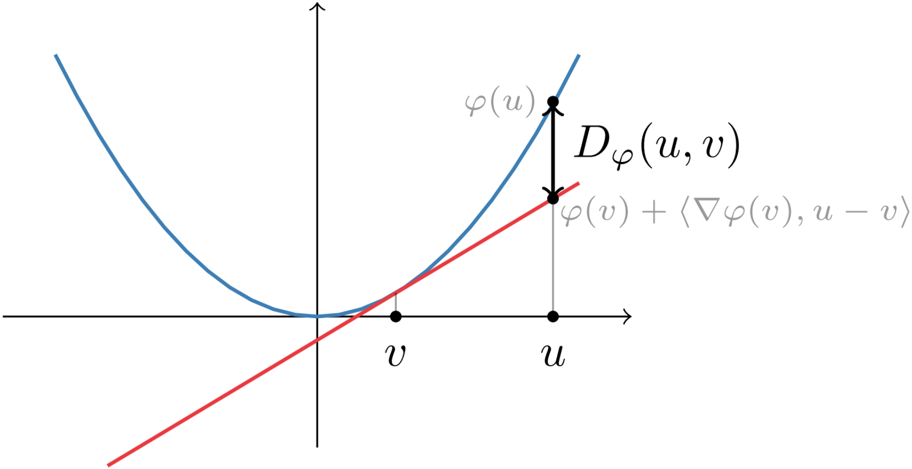


Figure: Bregman divergence $D_\varphi(u, v)$ (see e.g., Duchi, 2018, Figure 4.2.1)

Mirror Descent

- For $\boldsymbol{\theta}$ constrained on a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, the mirror descent method takes the form

$$\boldsymbol{\theta}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{C}} \left\{ F(\boldsymbol{\theta}_t) + \langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle + \frac{1}{\alpha_t} D_\varphi(\boldsymbol{\theta}, \boldsymbol{\theta}_t) \right\}$$

- When using the Shannon entropy $\varphi(\boldsymbol{\theta}) := \sum_{i=1}^d (\theta_i \log \theta_i - \theta_i)$ with $\theta_i \geqslant 0$ and $0 \log 0 := 0$, its induced Bregman divergence is the *Kullback−Leibler* (KL) divergence

$$D_{\mathrm{KL}}(\boldsymbol{\theta} \,\|\, \boldsymbol{\eta}) := \sum_{i=1}^d \left[\theta_i \log \frac{\theta_i}{\eta_i} - \theta_i + \eta_i \right]$$

- Restricting $\boldsymbol{\theta}$ to the probability simplex $\triangle^d := \{\boldsymbol{\theta} \in \mathbb{R}_+^d : \sum_{i=1}^d \theta_i = 1\}$,

$$D_{\mathrm{KL}}(\boldsymbol{\theta} \,\|\, \boldsymbol{\eta}) = \sum_{i=1}^d \theta_i \log \frac{\theta_i}{\eta_i}$$

Exponentiated Gradient Method or Entropic Mirror Descent

- Consider the constrained optimization where $\boldsymbol{\theta}$ lies in the probability simplex
- Each step of mirror descent updates solve the following subproblem

$$\underset{\boldsymbol{\theta} \in \triangle^d}{\text{minimize}} \left\{ \langle \nabla F(\boldsymbol{\theta}'), \boldsymbol{\theta} \rangle + \frac{1}{\alpha} D_{\mathrm{KL}}(\boldsymbol{\theta} \,\|\, \boldsymbol{\theta}') \right\}$$

- Some algebraic manipulations show that this has a closed form solution

$$\theta_i = \frac{\theta'_i \exp(-\alpha \nabla F(\theta'_i))}{\sum_{j=1}^{d} \theta'_j \exp(-\alpha \nabla F(\theta'_j))}$$

- This scheme is also called *exponentiated gradient method* or *entropic descent*

$$\theta_{t+1,i} = \frac{\theta'_{t,i} \exp(-\alpha_t \nabla F(\theta'_{t,i}))}{\sum_{j=1}^{d} \theta'_{t,j} \exp(-\alpha_t \nabla F(\theta'_{t,j}))}$$

Convergence Analysis of Entropic Mirror Descent

Theorem

Let $\alpha_t = \alpha > 0$ be a fixed step size, $\mathcal{C} = \triangle^d$ and φ be the Shannon entropy. Let $\boldsymbol{\theta}_0 = \mathbf{1}/d$. Then, the iterates $(\boldsymbol{\theta}_t)_{t \geqslant 0}$ of EMD on F satisfy

$$F(\overline{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) \leqslant \frac{\log d}{\alpha t} + \frac{\alpha}{2t} \sum_{s=1}^{t} \|\boldsymbol{g}_s\|_\infty^2,$$

where $\boldsymbol{g}_s \in \partial F(\boldsymbol{\theta}_s)$ if F is not differentiable and $\boldsymbol{g}_s = \nabla F(\boldsymbol{\theta}_s)$ otherwise.

- Better convergence guarantee than the standard Euclidean (projected) gradient method
- For more about mirror descent, e.g., in online learning, watch Five Miracles of Mirror Descent on YouTube by Sebastien Bubeck

Experiment: Robust Regression (Google Colab)

- $\boldsymbol{X} = (\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_n^\top)^\top \in \mathbb{R}^{n \times d}$, entries drawn i.i.d. N(0, 1)
- $y_i = \frac{1}{2}(x_{i,1} + x_{i,2}) + \varepsilon_i$ with $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, 10^{-2})$

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}} \; F(\boldsymbol{\theta}) := \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_1 = \sum_{i=1}^{n} |\langle \boldsymbol{x}_i, \boldsymbol{\theta} \rangle - y_i|$$

- Nonsmooth problem; use **subgradient** $\boldsymbol{g}_t \in \partial F(\boldsymbol{\theta}_t)$
- Compare with projected (sub)gradient descent (PGD):

$$\boldsymbol{\theta}_{t+1} = \mathrm{proj}_{\triangle^d}(\boldsymbol{\theta}_t - \alpha_t \boldsymbol{g}_t)$$

- Both mirror descent and PGD are sensitive to the choice of step sizes to meet performance close to theoretical guarantees **(but how? adaptive step sizes)**

Nonconvex Optimization for Machine Learning

Nonconvex Optimization for Machine Learning

- **Nonconvexity** is ubiquitous in modern machine learning, notably in deep learning
- For convex problems, the number of iterations of algorithms like gradient descent are provably *independent* of **dimension**
- For nonconvex problems, studying iteration complexity as a function of **dimension** is key
- Nonconvex optimization problems are *intractable* in general
- **Local minima** might suffice in ML
  - No *spurious* local minima
  - Local minima found be gradient-based algorithms are effective for generalization

Key Messages and Takeaways

- **Goal**: avoid *saddle points*
- Characterize the **iteration complexity** of *avoiding saddle points*, as a function of *target accuracy* and *dimension*
- GD, under random initialization or with perturbations (adding *Gaussian noise* to the stochastic gradients), asymptotically avoids saddle points with probability one
- Suitably-perturbed verions of GD and SGD escape saddle points in a number of iterations that is only *polylogarithmic* in **dimension**
- For details, refer to Jin et al. (2021)

More Differentiable Programming:

Implicit Differentiation of Optimization Problems

Motivation

- Recall the regularized ERM problem in supervised machine learning:

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\text{minimize}}\ F(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + h(\boldsymbol{\theta})$$

- Introduce a hyperparameter $\lambda > 0$ to h (denoted by h_λ)
- E.g., the relative weight of the empirical risk and the regularizer if $h_\lambda = \lambda h$
- Choosing an appriopriate value of λ is however tricky
- Many standard methods proposed in the statistics and ML literature if h is a sparsity-inducing function (e.g., cross-validation, regularization paths)

Implicit Differentiation of Optimization Problems

- If the dimension of the hyperparameter(s) grows, the existing methods cannot handle it (e.g., $\boldsymbol{\lambda} \in \mathbb{R}^r$, with $r \gg 1$)
- Instead, solve the bilevel optimization problem

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}^r}{\text{minimize}} \left\{ \mathscr{L}(\boldsymbol{\lambda}) := \mathscr{C}(\widehat{\boldsymbol{\theta}}_{\boldsymbol{\lambda}}) \right\} \quad \text{subject to} \quad \widehat{\boldsymbol{\theta}}_{\boldsymbol{\lambda}} \in \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{Argmin}}\ F_{\boldsymbol{\lambda}}(\boldsymbol{\theta}),$$

  where $\mathscr{C}$ is some criterion to ensure good generalization, e.g., the *hold-out (test) loss*, *cross-validation loss*

- Require the computation of the (sub)gradients w.r.t. the parameter $\boldsymbol{\theta}$ and the hyperparameter $\boldsymbol{\lambda}$
- See recent work by Bertrand et al. (2021); Blondel et al. (2021); Bolte et al. (2021) for both computational and theoretical frameworks for $F_{\boldsymbol{\lambda}}$ is **nonsmooth** (e.g., Lasso)

What We DID NOT Cover Today (Yet Important)

1. Accelerated (stochastic) gradient descent (e.g., Nesterov’s acceleration)
2. Variance reduced stochastic gradient methods (e.g., SVRG, SAGA) (Gower et al., 2020)
3. Variants of SGD widely used in deep learning (e.g., Adam, Adagrad)
4. Second-order methods (e.g., Newton’s method)
5. Nonconvex nonsmooth optimization problems (*relatively untouched* in the literature)
6. Stochastic optimization problems in which data distributions P also depend on θ (Perdomo et al., 2020; Mendler-Dünner et al., 2020; Drusvyatskiy and Xiao, 2020)

Some of the above are briefly discussed in Chapter 8 of the PML book by Murphy

Reference I

F. Bach. *Learning Theory from First Principles*. Draft, 2021. URL https://www.di.ens.fr/~fbach/ltfp_book.pdf.

Q. Bertrand, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021.

M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*, 2021.

J. Bolte, T. Le, E. Pauwels, and A. Silveti-Falls. Nonsmooth implicit differentiation for machine learning and optimization. *arXiv preprint arXiv:2106.04350*, 2021.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Reference II

D. Drusvyatskiy and L. Xiao. Stochastic optimization with decision-dependent distributions. *arXiv preprint arXiv:2011.11173*, 2020. URL https://arxiv.org/abs/2011.11173.

J. C. Duchi. Introductory lectures on stochastic optimization. In M. W. Mahoney, J. C. Duchi, and A. C. Gilbert, editors, *The Mathematics of Data*, volume 25 of *IAS/Park City Mathematics Series*, pages 99–185. The American Mathematical Society, 2018.

R. M. Gower, M. Schmidt, F. Bach, and P. Richtarik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

C. Mendler-Dünner, J. C. Perdomo, T. Zrnic, and M. Hardt. Stochastic optimization for performative prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Reference III

J. C. Perdomo, T. Zrnic, C. Mendler-Dünner, and M. Hardt. Performative prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

The End

Thank you!