No Author Given

Algorithmic Stochastic Convex Optimization

– Monograph –

July 1, 2022

Springer Nature

This book is dedicated to «three oracles»
Boris Teodorovich Polyak,
Arkadi Semenovich Nemirovski,
Yurii Evgenievich Nesterov
who invited us to modern numerical optimization methods.

Foreword

Use the template *foreword.tex* together with the document class SVMono (monograph-type books) or SVMult (edited books) to style your foreword.

Ask to write Foreword by A.S. Nemirovksi or B.T. Polyak.

The foreword covers introductory remarks preceding the text of a book that are written by a *person other than the author or editor* of the book. If applicable, the foreword precedes the preface which is written by the author or editor of the book.

Place, month year *Firstname Surname*

Preface

Use the template *preface.tex* together with the document class SVMono (monograph-type books) or SVMult (edited books) to style your preface.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s),	*Firstname Surname*
month year	*Firstname Surname*

Acknowledgements

We would like to thank Boris Mordukhovich and Elizabeth Loew. Without their help this book would hardly have been published by Springer.

This book was prepared since 2012. During all this time the authors have wonderful opportunities to consult with S. Chukanov, Yu. Golubev, G. Gidel, O. Granichin, R. Hildebrand, A. Juditsky, A. Nedich, A. Nemirovski, Yu. Nesterov, B. Polyak, V. Protasov, P. Richtarik, G. Scutari, A. Shananin, A. Shapiro, V. Spokoiny, S. Stich, M. Takac, S. Tarasov, A. Taylor, A. Tsybakov. We would like to express our sincere gratitude to all of them.

Contents

Acronyms

Use the template *acronym.tex* together with the document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

ABC	Spelled-out abbreviation and definition
BABI	Spelled-out abbreviation and definition
CABR	Spelled-out abbreviation and definition

Chapter 1
Stochastic optimization and Data Science

Abstract This chapter aims to motivate stochastic optimization problems by statistics and statistical learning theory, where the goal is to maximize log-likelihood or minimize population risk.

In this chapter, we briefly describe two main approaches for solving expectation minimization problems

$$\min_{x \in Q} \left\{ f(x) = \mathbb{E}_{\xi \sim \mathcal{D}} [f(x, \xi)] \right\}. \tag{1.1}$$

The first approach is *offline* (Monte Carlo / Sample Average Approximation) and the second one is *online* (Stochastic Approximation).

1.1 Stochastic optimization motivation

Following [109] one can say that «Optimization problems involving stochastic models occur in almost all areas of science and engineering, so diverse as telecommunication, medicine, or finance, to name just a few. This stimulates interest in rigorous ways of formulating, analyzing, and solving such problems. Due to the presence of random parameters in the model, the theory combines concepts of the optimization theory, the theory of probability and statistics, and functional analysis. Moreover, in recent years the theory and methods of stochastic programming have undergone major advances.»

This «major advances» are strongly stimulated by the explosive growth of interest to *Data science* problems. In the last decade there appear a several good books dedicated to the connection of *Stochastic Optimization* and *Data Science* [109, 106, 8]. In this section we briefly describe two main original sources for appearance of stochastic optimization problems in Data Science: 1) *Statistics* source (*Fisher's theorem*, *maximum likelihood estimation*) and 2) *Machine Learning* source (expected risk minimization).

1.1.1 Statistical motivation

We start with the the most simple situation. Let $x_* \in \mathbb{R}$ be an unknown scalar parameter, $\eta \sim \mathcal{N}\left(0, \sigma^2\right)$ – Gaussian noise. Assume that we can measure

$$\xi^k = x_* + \eta^k,\ k = 1, ..., N,$$

where η^k i.i.d. (independent identically distributed as η). **The goal is to estimate** x_* **from** $\left\{\xi^k\right\}_{k=1}^N$**.**

The main observation is as follows: x_* is a solution of Stochastic optimization problem

$$\min_{x \in \mathbb{R}} \mathbb{E}_\xi \left[f(x,\xi) := (\xi - x)^2\right], \tag{1.2}$$

where $\xi \sim \mathcal{N}\left(x_*, \sigma^2\right)$. Indeed,

$$\mathbb{E}_\xi (\xi - x)^2 = \mathbb{E}_\xi \xi^2 - 2x\mathbb{E}_\xi \xi + x^2 = x_*^2 + \sigma^2 - 2xx_* + x^2 = (x_* - x)^2 + \sigma^2$$

attains minimum at $x = x_*$. But we do not know x_* (and maybe σ^2). How should we solve the problem (1.2)? Since $\left\{\xi^k\right\}_{k=1}^N$ is available one can use Monte Carlo approach. This approach consists in the replacing of problems (1.2) by empirical version

$$\min_{x \in \mathbb{R}} \left[\frac{1}{N}\sum_{k=1}^N (\xi^k - x)^2\right]. \tag{1.3}$$

The problem (1.3) could be easily solved

$$\bar{x}^N = \frac{1}{N}\sum_{k=1}^N \xi^k. \tag{1.4}$$

What is known in Statistics as *Sample Average*, which is the best know estimate of unknown parameter in the described parametric model, see Theorem 1.1 below.

The same solution (1.4) could be obtained by online procedure

$$x^{k+1} = x^k - \frac{1}{2(k+1)}\nabla_x f(x^k, \xi^k) = x^k - \frac{1}{k+1}(x^k - \xi^k), k = 0, ..., N-1, \tag{1.5}$$

where $x^0 = \xi^1$. This procedure corresponds to the *Stochastic Gradient Descent* (SGD) for 2-strongly convex in 2-norm stochastic optimization problem (1.2).

The main question which should have occurred after the reading the text above: By what scheme was $f(x,\xi)$ selected in (1.2)? Maybe there are many ways to choose $f(x,\xi)$. And if so, what is the best way? Below we briefly describe the basics of maximum likelihood theory, which allows us to obtain the answer for these questions.

Assume that some random variable ξ depends on unknown vector of parameters $x_* \in \mathbb{R}^n$. Let $p(x,\xi)$ is probability (probability density function) that we observe ξ if the true vector of parameters is $x \in \mathbb{R}^n$. In the described above example $n = 1$ and probability density function was

$$p(x,\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\xi - x)^2}{2\sigma^2}\right).$$

If i.i.d. samples $\left\{\xi^k\right\}_{k=1}^N$ are available let us introduce likelihood

$$p\left(x, \{\xi^k\}_{k=1}^N\right) = \prod_{k=1}^N p(x,\xi^k).$$

Perhaps one of the most productive ideas in Statistics is to estimate true vector of parameters x_* as a vector that maximize likelihood $p\left(x, \left\{\xi^k\right\}_{k=1}^N\right)$. This problem can be equivalently reformulated as minimization of (normalized) minus log-likelihood

$$\min_{x\in\mathbb{R}^n}\left[-\frac{1}{N}\log p\left(x, \{\xi^k\}_{k=1}^N\right) = -\frac{1}{N}\sum_{k=1}^n \log p(x,\xi^k)\right].$$

This minimization problem can be considered as empirical (Monte Carlo) version of Stochastic optimization problem

$$\min_{x\in\mathbb{R}^n} \mathbb{E}_\xi\left[-\log p(x,\xi)\right]. \tag{1.6}$$

In particular, for considered above Gaussian model this problem looks like

$$\min_{x\in\mathbb{R}} \mathbb{E}_\xi\left[\frac{1}{2\sigma^2}(\xi - x)^2 + \frac{1}{2}\log\left(2\pi\sigma^2\right)\right],$$

which is equivalent to (1.2).

Moreover, the observation that the true value of unknown vector of parameters x_* is a solution of (1.6) holds true in the general case, i.e.

$$x_* \in \operatorname{Arg}\min_{x\in\mathbb{R}^n} \mathbb{E}_\xi\left[-\log p(x,\xi)\right].$$

Indeed,[1]

$$\mathbb{E}_\xi\left[-\log p(x,\xi)\right] = -\int p(x_*,\xi)\log p(x,\xi)d\xi \geq -\int p(x_*,\xi)\log p(x_*,\xi)d\xi$$

since (*Jensen's inequality* for entropy)

$$KL\left(p(x_*,\cdot), p(x,\cdot)\right) = \int p(x_*,\xi)\log\left(\frac{p(x_*,\xi)}{p(x,\xi)}\right)d\xi \geq 0$$

and $KL\left(p(x_*,\cdot), p(x,\cdot)\right) = 0$, when $x = x_*$.

[1] For certainty, here $p(x,\xi)$ is assumed to be a probability density function.

So, we have just explained that in the general case $f(x,\xi) := -\log p(x,\xi)$ in (1.2) and maximum likelihood approach is nothing more than Monte Carlo approach for Stochastic optimization problem (1.6).

Definitely the main gem of Statistics is Fisher's theorem about asymptotic properties of *maximum likelihood estimation* (MLE)

$$\hat{x}^N_{MLE} = \arg\max_{x\in\mathbb{R}^n} p\left(x, \left\{\xi^k\right\}_{k=1}^N\right) = \arg\min_{x\in\mathbb{R}^n}\left[-\log p\left(x, \left\{\xi^k\right\}_{k=1}^N\right)\right]. \tag{1.7}$$

Informal variant of this theorem looks as follows.

Theorem 1.1 *Assume that $p(x,\xi)$ is sufficiently smooth and the set*

$$\{\xi : p(x,\xi) > 0\}$$

does not depend on x.[2] *Then*

1) For all the statistics $\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right)$ with finite second moment Rao–Cramer inequality holds true[3]

$$\mathbb{E}_{\left\{\xi^k\right\}_{k=1}^N}\left[\left(\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right) - x_*\right)\left(\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right) - x_*\right)^T\right] \succcurlyeq \left[NI_{x_*}\right]^{-1},$$

where

$$I_{x_*} = \mathbb{E}_\xi\left[\nabla_x p(x_*,\xi)\left(\nabla_x p(x_*,\xi)\right)^T\right]$$

– Fisher information matrix.[4]

2) MLE $\hat{x}^N_{MLE}\left(\left\{\xi^k\right\}_{k=1}^N\right)$ (see (1.7)*) has asymptotically*[5] *normal (Gaussian) distribution $\mathcal{N}\left(x_*, \left[NI_{x_*,N}\right]^{-1}\right)$ and achieves equality in Rao–Cramer inequality. That is MLE has asymptotically the smallest variance along all the directions and independently of what is x_*.*

As a consequence of this theorem one may expect to build asymptotically the smallest confident set around MLE. The online approach (based on SGD, proper stepsize policy and *Polyak–Juditsky–Ruppert averaging*) leads to a similar asymptotic result.

Unfortunately, asymptotic theory does not fully characterize the real state of affairs when N is not sufficiently large. For example, if we consider the Bernoullie parametric model (coin flips) with likelihood $p(x,\xi) = x^\xi(1-x)^{1-\xi}$ and $x_* > 0$ small enough, then while $N \lesssim 1/x_*$ with positive probability for MLE $\hat{x}^N = 0$ [106]. Hence $\mathbb{E}_\xi\left[-\log p(0,\xi)\right] = \infty$ is not well defined.

[2] That is true for Gaussian noise model $\xi = x + \eta$, but is not true if the noise η is uniformly distributed on $[0,x]$.

[3] $A \succcurlyeq B$ means that for all $z \in \mathbb{R}^n$ $\langle z, (A-B)z\rangle \geq 0$.

[4] Note that $KL\left(p(x_*,\cdot), p(x_*+h,\cdot)\right) \simeq \frac{h^2}{2}I_{x_*}$.

[5] When $N \to \infty$.

Modern offline asymptotic theory of statistics [51] (le Cam's theory) was further developed in partially non asymptotic and misspecification[6] directions [112]. In this book we mainly (except the next section) concentrate on non asymptotic online approaches for (1.6) and more general problems formulations.

At the end of this section we aim to demonstrate the role of regularization in offline approach as a Bayesian prior. Assume that in the general scheme, which described by the parametric model $p(x, \xi)$, we have an additional information about vector of parameters x: x is a random vector that was a priory independently generated from the distribution with density function $\pi(x)$.

A Bayesian estimator is an estimator that minimize the posterior expected value of loss function (we consider quadratic loss), which is coincide with a posterior mean:

$$\hat{x}_B^N = \arg\min_{x\in\mathbb{R}^n} \int\limits_{\mathbb{R}^n} \|x - z\|^2 p\left(z, \left\{\xi^k\right\}_{k=1}^N\right) \pi(z) dz = \int\limits_{\mathbb{R}^n} x p\left(x, \left\{\xi^k\right\}_{k=1}^N\right) \pi(x) dx. \tag{1.8}$$

Informal analogue of Theorem 1.1 in this case looks as follows.

Theorem 1.2 *Assume that $p(x, \xi)$, $\pi(x)$ are sufficiently smooth and the set*

$$\{\xi : p(x, \xi) > 0\}$$

does not depend on x. Then

1) For all the statistics $\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right)$ with finite second moment van Trees inequality holds true

$$\mathbb{E}_{\left(x, \left\{\xi^k\right\}_{k=1}^N\right)}\left[\left(\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right) - x\right)\left(\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right) - x\right)^T\right] \succcurlyeq \left[N I_p + I_\pi\right]^{-1},$$

where

$$I_p = \mathbb{E}_{(x,\xi)}\left[\nabla_x p(x, \xi) \left(\nabla_x p(x, \xi)\right)^T\right]$$

– Fisher information matrix and

$$I_\pi = \mathbb{E}_x\left[\nabla \pi(x) \left(\nabla \pi(x)\right)^T\right].$$

2) Bayesian estimator $\hat{x}_B^N\left(\left\{\xi^k\right\}_{k=1}^N\right)$ (see (1.8)) has conditional (with a priori drawing $x = x_*$) asymptotically normal distribution $\mathcal{N}\left(x_*, \left[N I_{x_*}\right]^{-1}\right)$.

A close result is contained in the *Bernstein–von Mises theorem*: a posterior distribution has asymptotically normal distribution centered at the MLE with covariance matrix $N I_{x_*}$.

In Bayesian statistics a *maximum a posterior estimation* (MAP):

[6] If the parametric model is wrong, MLE could be interpreted as asymptotically the best way to estimate the KL-projection of the true vector of parameters on the parametric model.

$$\hat{x}^N_{MAP} = \arg\max_{x\in\mathbb{R}^n} p\left(x, \left\{\xi^k\right\}_{k=1}^N\right)\pi(x) = \arg\min_{x\in\mathbb{R}^n}\left[-\log p\left(x, \left\{\xi^k\right\}_{k=1}^N\right) - \log\pi(x)\right].$$

plays also an important role. MAP has typically the same asymptotic behavior as Bayesian estimator.

Let us consider several examples. The first example is *Regularized Least Squares*.

Ridge Regression and LASSO

Assume that $x_* \in \mathbb{R}^n$ be an unknown vector of parameters, $\eta \sim \mathcal{N}\left(0, \sigma^2\right)$ Gaussian noise. Assume that we can measure

$$\xi^k = \langle a_k, x_*\rangle + \eta^k,\; k = 1, ..., N,$$

where η^k i.i.d. (independent identically distributed as η) and matrix $A = [a_1, ..., a_N]^T$ is known.[7] **The goal is to estimate x_* from $\xi := \left\{\xi^k\right\}_{k=1}^N$.** Simple calculations lead to the following formulas

$$\hat{x}^N_{MLE} = \arg\min_{x\in\mathbb{R}^n}\left[\frac{1}{2\sigma^2}\|Ax-\xi\|_2^2\right],$$

$$\hat{x}^N_B = \hat{x}^N_{MAP} = \arg\min_{x\in\mathbb{R}^n}\left[\frac{1}{2\sigma^2}\|Ax-\xi\|_2^2 + \frac{1}{2\sigma_\pi^2}\|x-\bar{x}\|_2^2\right],$$

where a priory x_i, $i = 1, ..., n$ assumed to be independent and identically distributed according to $\mathcal{N}\left(\bar{x}, \sigma_\pi^2\right)$ (Ridge Regression) and

$$\hat{x}^N_{MAP} = \arg\min_{x\in\mathbb{R}^n}\left[\frac{1}{2\sigma^2}\|Ax-\xi\|_2^2 + \lambda\|x\|_1\right],$$

where the prior probability density is (LASSO):

$$\pi(x) = \prod_{i=1}^n \frac{\lambda}{2}\exp(-\lambda|x_i|) = \left(\frac{\lambda}{2}\right)^n \exp\left(-\lambda\|x\|_1\right).$$

It is obvious that Bayesian estimator and MAP asymptotically ($N \to \infty$) coincide with MLE. Another important observation that Bayesian estimator and MAP asymptotically coincide with MLE when $\sigma_\pi^2 \to \infty$. Both of these observations take place in the general case. So *Bayesian prior* can be interpreted as regularizer in Bayesian version of maximum likelihood optimization problem.

The second example goes back to Vadim V. Mottl.

Soft-SVM

[7] Note that a_k could also be generated randomly. In this case for the results to be preserved it is enough to require that $\{a_k\}_{k=1}^n$ and $\left\{\eta^k\right\}_{k=1}^n$ are independent.

In this example *Soft-Support-Vector Machine* (Soft-SVM) is derived based on Bayesian inference with

$$p\left(x, \xi^k := \left(y^k, a_k\right)\right) \propto \begin{cases} 1, & \text{if } y^k\langle x, a_k\rangle \geq 1 \\ \exp\left(-\left(1 - y^k\langle x, a_k\rangle\right)\right), & \text{else,} \end{cases}$$

where $y^k \in \{-1, 1\}$ and a priory x_i, $i = 1, ..., n$ assumed to be independent and identically distributed according to $\mathcal{N}\left(0, \sigma_\pi^2\right)$. Improper probability density function $p\left(x, \xi^k\right)$ has a natural interpretation: there exists «true» hyperplane (determined by the vector x_*) such that the data points with $y^k = 1$ lie mostly from the one side of this hyperplane and the data points with $y^k = -1$ lie mostly from the other side. The goal is to recognize this hyperplane from the data points having a prior information about x_*. Simple calculations lead to the following formula

$$\hat{x}_{MAP}^N = \arg\min_{x \in \mathbb{R}^n} \left[\sum_{k=1}^N \max\left\{0, 1 - y^k\langle x, a_k\rangle\right\} + \frac{1}{2\sigma_\pi^2}\|x\|_2^2\right].$$

1.1.2 Machine Learning motivation

In the statistical approach the loss function is $f(x, \xi) := -\log p(x, \xi)$. It means that we required parametric model $p(x, \xi)$. In many practical situations $p(x, \xi)$ is not available. However in *Regression problems* we can introduce *least square loss function* $f\left(x, \xi := (y, a)\right) = (y - \langle a, x\rangle)^2$. And without any knowledge of probability nature of ξ we can consider expected loss minimization problem (stochastic optimization problem):

$$\min_{x \in \mathbb{R}^n} \mathbb{E}_{(y,a)}\left[(y - \langle a, x\rangle)^2\right].$$

In offline approach this problem has a form:

$$\min_{x \in \mathbb{R}^n} \|Y - Ax\|_2^2,$$

where $Y = \left(y^1, ..., y^N\right)^T$, $A = [a_1, ..., a_N]^T$. Similarly, in *Classification problems* we can introduce *hinge-loss function* $f\left(x, \xi := (y, a)\right) = \max\{0, 1 - y\langle x, a\rangle\}$ and corresponding stochastic optimization problems has a form:

$$\min_{x \in \mathbb{R}^n} \mathbb{E}_{(y,a)}\left[\max\{0, 1 - y\langle x, a\rangle\}\right].$$

In many real world applications we have some prior information about how much could be (should be) x_*. Typically, this information formalize as a constraint of the type $x \in Q$, where Q is often chosen as a ball $B_p^n(R_p)$ in p-norm ($p \geq 1$) centered at 0 with radius R_p or another convex compact set with simple structure, e.g. unit simplex $S_n(1)$. So the final stochastic optimization problem in general has a form[8]

$$\min_{x\in Q\subseteq\mathbb{R}^n} F(x) := \mathbb{E}_\xi f(x,\xi). \tag{1.9}$$

For $Q = B_2^n(R_2)$ (or $Q = B_1^n(R_1)$) if the constraint is reached it could be replaced by $\|x\|_2^2$-regularization (or $\|x\|_1$-regularization) with Lagrange multiplayer as a regularization parameter.

All the considered above concrete problems (Regression and Classification) have two things in common. The target functions:

1) are convex: for all ξ and $x, z \in Q$

$$f(z,\xi) \geq f(x,\xi) + \langle\nabla_x f(x,\xi), z - x\rangle$$

and *M-Lipschitz continuous* in x in 2-norm: for all ξ and $x, z \in Q$

$$|f(z,\xi) - f(x,\xi)| \leq M\|z - x\|_2.$$

2) have *generalized linear* structure:

$$f(x,\xi) := g\left(y(\xi), \langle x, a(\xi)\rangle\right).$$

The first common thing guarantees the effectiveness of *online* approach. Both of them guarantee the effectiveness of offline approach.

Let us start with *offline* approach. We introduce the *empirical loss*

$$\bar{F}(x) := \bar{F}\left(x, \{\xi^k\}_{k=1}^N\right) = \frac{1}{N}\sum_{k=1}^N f(x,\xi^k)$$

and minimizer of the empirical loss

$$\hat{x}^N \in \operatorname{Arg}\min_{x\in Q} \bar{F}\left(x, \{\xi^k\}_{k=1}^N\right).$$

Theorem 1.3 (Learnability for generalized linear models) *Consider the stochastic optimization problem* (1.9) *with $f(x,\xi)$ satisfies 1) and 2) and convex $Q \subseteq B_2^n(R)$. With probability at least* $1-\beta$*:*

$$\sup_{x\in Q}\left|\bar{F}(x) - F(x)\right| = O\left(MR\sqrt{\frac{\log(1/\beta)}{N}}\right),$$

hence with probability at least $1-\beta$*:*

[8] Here and everywhere below we will denote the solution of this problem as x_*. If the solution is not unique x_* means one of the solutions, e.g. such that is the closest to the starting point (initial guess).

$$F(x) - F(x_*) \le \bar{F}(x) - \bar{F}(\hat{x}^N) + O\left(MR\sqrt{\frac{\log(1/\beta)}{N}}\right). \tag{1.10}$$

If additionally for all ξ and $x, z \in Q$

$$f(z,\xi) \ge f(x,\xi) + \langle \nabla_x f(x,\xi), z - x\rangle + \frac{\mu}{2}\|z-x\|_2^2,$$

i.e. $f(x,\xi)$ is μ-strongly convex in x in 2-norm, then with probability at least $1-\beta$:

$$F(x) - F(x_*) \le 2\left(\bar{F}(x) - \bar{F}(\hat{x}^N)\right) + O\left(\frac{M^2\log(1/\beta)}{\mu N}\right). \tag{1.11}$$

If the property 2) is no longer met, then (1.11) *should be rewritten as follows: with probability at least* $1-\beta$*:*

$$F(x) - F(x_*) \le \sqrt{\frac{2M^2}{\mu}\left(\bar{F}(x) - \bar{F}(\hat{x}^N)\right)} + \tilde{O}\left(\frac{M^2\log(1/\beta)}{\mu N}\right). \tag{1.12}$$

Moreover, all these inequalities are optimal up to a constant factor.

This theorem reduces stochastic optimization problem to the empirical loss (risk) minimization problem

$$\min_{x\in Q}\frac{1}{N}\sum_{k=1}^{N} f(x,\xi^k) \tag{1.13}$$

with proper choice of N, see the next section.

Now we move to *online* approach and explain why it is so called. In the core of online approach lies standard SGD:

$$x^{k+1} = \pi_Q\left(x^k - h_k\nabla_x f(x^k,\xi^k)\right), \tag{1.14}$$

where π_Q is a euclidean projection onto Q. Note that

$$\begin{aligned}\|x^{k+1}-x_*\|_2^2 = \left\|\pi_Q\left(x^k - h_k\nabla_x f(x^k,\xi^k) - x_*\right)\right\|_2^2 \le\\ \le \|x^k - h_k\nabla_x f(x^k,\xi^k) - x_*\|_2^2 = \|x^{k+1}-x_*\|_2^2 -\\ -2h_k\langle\nabla_x f(x^k,\xi^k), x^k - x_*\rangle + h_k^2\|\nabla_x f(x^k,\xi^k)\|_2^2 \le\\ \le \|x^{k+1}-x_*\|_2^2 - 2h_k\langle\nabla_x f(x^k,\xi^k), x^k - x_*\rangle + h_k^2 M^2.\end{aligned}$$

The last inequality holds true since $f(x,\xi)$ is M-Lipschitz continuous in x in 2-norm and therefore, $\|\nabla_x f(x^k,\xi^k)\|_2 \le M$. From the convexity of $f(x,\xi)$ on x:

$$\begin{aligned}f(x^k,\xi^k) - f(x_*,\xi^k) \le \langle\nabla_x f(x^k,\xi^k), x^k - x_*\rangle \le\\ \le \frac{1}{2h_k}\left(\|x^k-x_*\|_2^2 - \|x^{k+1}-x_*\|_2^2\right) + \frac{h_k M^2}{2}.\end{aligned}$$

From the μ-strong convexity of $f(x,\xi)$ on x:

$$f(x^k,\xi^k) - f(x_*,\xi^k) \le \langle \nabla_x f(x^k,\xi^k), x^k - x_* \rangle - \frac{\mu}{2}\|x^k - x_*\|_2^2 \le$$
$$\le \frac{1}{2}\left(\frac{1}{h_k} - \mu\right)\|x^k - x_*\|_2^2 - \frac{1}{2h_k}\|x^{k+1} - x_*\|_2^2 + \frac{h_k M^2}{2}.$$

Summing for $k = 1,...,N$ «convex» inequality with $h_k \equiv \frac{R}{M\sqrt{N}}$ and «strongly convex» inequality with[9] $h_k = \frac{1}{\mu k}$ we obtain after normalization (multiplication on N^{-1}):

$$\frac{1}{N}\sum_{k=1}^{N} f(x^k,\xi^k) \le \frac{1}{N}\sum_{k=1}^{N} f(x_*,\xi^k) + \frac{M\|x^1 - x_*\|_2}{\sqrt{N}}, \tag{1.15}$$

$$\frac{1}{N}\sum_{k=1}^{N} f(x^k,\xi^k) \le \frac{1}{N}\sum_{k=1}^{N} f(x_*,\xi^k) + \frac{M^2(1+\log N)}{2\mu N}. \tag{1.16}$$

Note that in (1.15), (1.16) $x_* \in Q$ can be chosen in an arbitrary manner, say such that minimize RHS, i.e.

$$\frac{1}{N}\sum_{k=1}^{N} f(x^k,\xi^k) \le \min_{x\in Q}\frac{1}{N}\sum_{k=1}^{N} f(x,\xi^k) + \frac{M\|x^1 - x_*\|_2}{\sqrt{N}},$$

$$\frac{1}{N}\sum_{k=1}^{N} f(x^k,\xi^k) \le \min_{x\in Q}\frac{1}{N}\sum_{k=1}^{N} f(x,\xi^k) + \frac{M^2(1+\log N)}{2\mu N}.$$

Since we do not still use the probability nature of ξ^k, it follows that the last two inequalities characterize SGD (1.14) as online learning procedure in the standard online sense [19].

If we remember now about i.i.d. nature of $\{\xi^k\}_{k=1}^N$, remember that: $\mathbb{E}_\xi f(x^k,\xi) \equiv F(x)$, $f(x,\xi)$ is M-Lipschitz continuous in x in 2-norm and $F(x)$ is convex, than (1.15), (1.16) could be further simplify (*online to batch conversion*).

Theorem 1.4 *Consider stochastic optimization problems* (1.9) *with $f(x,\xi)$ satisfies 1). Then for x^k generated by* (1.14) *with probability at least* $1-\beta$*:*

$$F(\bar{x}^N) - F(x_*) = O\left(\frac{M\|x^1 - x_*\|_2 \log(1/\beta)}{\sqrt{N}}\right), \tag{1.17}$$

where

$$\bar{x}^N = \frac{1}{N}\sum_{k=1}^{N} x^k. \tag{1.18}$$

If additionally $f(x,\xi)$ is μ-strongly convex in x in 2*-norm, then with probability at least* $1-\beta$*:*

[9] In this case we have the telescopic property: $\frac{1}{h_{k+1}} - \mu = \frac{1}{h_k}$.

$$F(\bar{x}^N) - F(x_*) = O\left(\frac{M^2 \log(N/\beta)}{\mu N}\right). \tag{1.19}$$

Since $\|x^1 - x_*\|_2 \leq 2R$, it follows that (1.17) and (1.19) correspond to (1.10) and (1.11), (1.12) in *sample complexity* – the required number of samples N. However, online approach does not require to solve an auxiliary empirical problem (1.13) and was justified under weaker assumptions. More detailed comparison online and offline approaches is given in the next section.

To conclude this section, remind the main observation: statistical approach for data science problems is a particular case of the general machine learning (ML) approach, where the loss function has a specific form determined by log-likelihood functions. So further we will consider mainly ML approach, which characterize stochastic optimization problem (1.9).

1.2 Sample Average Approximation vs Stochastic Approximation

In this section we consider stochastic optimization problem (1.9)

$$\min_{x \in Q \subseteq \mathbb{R}^n} F(x) := \mathbb{E}_\xi \left[f(x,\xi)\right].$$

We are mainly interested in the sample complexity of offline (also called *Sample Average Approximation*) and online (also called *Stochastic Approximation*) procedures, which generate $\tilde{x}^N\left(\left\{\xi^k\right\}_{k=1}^N\right)$ from the solution of the empirical problem (1.13) or from the procedure of type (1.14). More precisely, we are interested to estimate such $N := N(\varepsilon, \beta)$ that

$$\mathbb{P}\left(F\left(\tilde{x}\left(\{\xi^k\}_{k=1}^N\right)\right) - F(x_*) \leq \varepsilon\right) \geq 1 - \beta.$$

Assume that $Q \subseteq B_p^n(R_p)$ $(p \geq 1)$ and for all ξ and $x, y \in Q$:

$$|f(y,\xi) - f(x,\xi)| \leq M_p \|y - x\|_p. \tag{1.20}$$

Let $\bar{x}_{\delta,\gamma}^N := \bar{x}_{\delta,\gamma}^N\left(\{\xi^k\}_{k=1}^N\right)$ be the (δ,γ)-solution of the empirical problem (1.13)

$$\min_{x \in Q}\left[\bar{F}(x) := \frac{1}{N}\sum_{k=1}^N f(x,\xi^k)\right],$$

that is, with probability at least $1 - \gamma$:

$$\bar{F}\left(\bar{x}_{\delta,\gamma}^N\right) - \min_{x \in Q} \bar{F}(x) = \bar{F}\left(\bar{x}_{\delta,\gamma}^N\right) - \bar{F}\left(\hat{x}^N\right) \leq \delta.$$

1.2.1 Non-convex case and convex case

One of the first and quite unexpected results about offline approach looks as follows.

Theorem 1.5 *Assume that* (1.20) *holds true. Then for* $\bar{x}^N_{\varepsilon/2,\beta/2}\left(\{\xi^k\}_{k=1}^N\right)$*:*

$$N = O\left(\frac{M_p^2 R_p^2}{\varepsilon^2}\left(n\log\left(\frac{M_p R_p}{\varepsilon}\right) + \log\left(\frac{1}{\beta}\right)\right)\right). \tag{1.21}$$

This bound is optimal up to a logarithmic factor. Moreover, if we additionally assume that $f(x,\xi)$ *is convex and smooth in* x*,* (1.21) *would be still an optimal bound.*

Proof Nazary, please add the proof of the Theorem based on [109] (`https://cpn-us-w2.wpmucdn.com/sites.gatech.edu/dist/4/1470/files/2021/03/SPbook.pdf`) Sections 5.3.1 and 5.3.2. The proof also contains the following assumption: For any $x, x' \in X$ there exists constant $\sigma_{x,x'} > 0$ such that the moment generating function $M_{x,x'}(t) = \mathbb{E}_\xi\left[\exp tY(x,x')\right]$ of random variable $Y(x,x') := [f(x,\xi) - F(x)] - [f(x,\xi) - F(x)]$ satisfies $M_{x,x'}(t) \leq \exp\frac{\sigma^2_{x,x'}t^2}{2}$, for every $t \in \mathbb{R}$. Assumptions

- **(M1)** The expectation function $F(x)$ is well defined and finite valued for all $x \in Q$. □

For $\varepsilon \geq 0$ denote by

$$S^\varepsilon := \left\{x \in Q : F(x) \leq \min_{x\in Q} F(x) + \varepsilon\right\}, \quad \hat{S}^\varepsilon_N := \left\{x \in Q : \bar{F}(x) \leq \min_{x\in Q} \bar{F}(x) + \varepsilon\right\}$$

the sets of ε-optimal solutions of the true and the SAA problems, respectively.

In the close setting online approach gives a better result, see also (1.17) for $p = 2$.

Theorem 1.6 *Assume that* (1.20) *holds true and* $f(x,\xi)$ *is convex in* x *in* Q*. Then for* $\bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$ *(see* (1.18)*) generated by the proper modification of* (1.14)*:*[10]

$$N = \tilde{O}\left(\frac{M_p^2 R_p^2}{\varepsilon^2}\ln\left(\frac{1}{\beta}\right)\right), \text{ if } p \in [1,2],$$

$$N = O\left(n^{1-2/p}\frac{M_2^2 R_p^2}{\varepsilon^2}\log\left(\frac{1}{\beta}\right)\right), \text{ if } p > 2.$$

[10] We will talk about «proper» (*Mirror Descent*) modification in the next chapter in more details. Note that for $p \geq 2$ it is proper to use (1.14). The factor $n^{1-2/p}$ appears since the diameter of $B_p^n(R_p)$ in 2-norm is $O\left(n^{1/2-1/p}R_p\right)$.

These bounds are optimal up to logarithmic factors in the wide class of all reasonable ways to generate $\bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$.[11]

It seems that online setting (e.g. for $p = 2$) is better than offline in the sample complexity for convex $f(x,\xi)$ in x. In the next section we show that the gap factor n in the sample complexity bounds between online and offline approaches can be eliminated by the proper regularization.

1.2.2 Strongly convex case and regularization

If $f(x,\xi)$ is μ_p*-strongly convex* in x in p-norm ($p \geq 1$), that is for all ξ and $x, y \in Q$:

$$f(y,\xi) \geq f(x,\xi) + \langle \nabla_x f(x,\xi), y - x\rangle + \frac{\mu_p}{2}\|y - x\|_p^2, \tag{1.22}$$

then Theorem 1.5 can be improved.

Theorem 1.7 *Assume that* (1.20) *and* (1.22) *hold true. Then for*

$$\bar{x}^N_{\delta,\beta/2}\left(\{\xi^k\}_{k=1}^N\right),\ \delta = \tfrac{\varepsilon^2\mu_p}{8M_p^2} \text{ and } \bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$$

– generated by the proper (restarted[12] *Mirror Descent) modification of* (1.14)*:*

$$N = \tilde{O}\left(\frac{M_p^2}{\mu_p\varepsilon}\log\left(\frac{\log\left(M_p^2/(\mu_p\varepsilon)\right)}{\beta}\right)\right), \quad p \in [1,2]. \tag{1.23}$$

This bound is optimal to up a logarithmic factor in the wide class of all reasonable ways to generate $\bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$. *Moreover, this bound corresponds* (1.12) *and* (1.19) *when* $p = 2$ *and the bound on* δ *derived from the condition that the first term in RHS of* (1.12) *equals* $\varepsilon/2$. *The bound on* δ *also cannot be improved up to a numerical constant.*

Proof For simplicity, we prove (1.23) only in terms of expectation, rather that high probability bounds.

Nazary, please add the proof of the Theorem based on [107] (`https://home.ttic.edu/~nati/Publications/nonlinearTR.pdf`) Section 4 and `https://www.jmlr.org/papers/volume2/bousquet02a/bousquet02a.pdf` p. 508. Note that the proof of the theorem it's sufficient to describe in terms of expectation, rather that high probability bounds. □

[11] We discuss it also in more details in the next chapter. Also in the next chapter we mention that in the non-convex case online approach gives much worse results $N \propto \varepsilon^{-(n+1)}$, which is also optimal bound for non-convex class of $f(x,\xi)$. Note that the bound on $N \propto n^{1-2/p}M_2^2R_p^2\varepsilon^{-2}$ in the regime $p > 2$ can be refined in the dimension-free case $N \lesssim n : N \propto M_p^pR_p^p\varepsilon^{-p}$ [82].

[12] See the proof of Theorem 1.10 for $p = 2$ and the next chapter in the general case.

We emphasis that in Theorem 1.5 $\delta \simeq \varepsilon$, but in Theorem 1.7 $\delta \simeq \frac{\varepsilon^2\mu_p}{M_p^2}$ and the last bound cannot be weakened!

Based on Theorem 1.7 one can derive the result that improve Theorem 1.5 in the convex case ($\mu_p \simeq 0$). Assume for the clarity that $p = 2$.

Lemma 1.1 (Tiknonov's regularization) *Consider regularized stochastic optimization problem*

$$\min_{x\in Q}\left[F_\mu(x) := \mathbb{E}_\xi f(x,\xi) + \frac{\mu}{2}\|x\|_2^2\right] \tag{1.24}$$

with $\mu = \varepsilon/R_2^2$. *Assume that*

$$F_\mu(\tilde{x}) - \min_{x\in Q} F_\mu(x) \le \frac{\varepsilon}{2}.$$

Then

$$F(\tilde{x}) - \min_{x\in Q} F(x) = F(\tilde{x}) - F(x_*) \le \varepsilon.$$

Proof Indeed,

$$F(\tilde{x}) - F(x_*) \le F_\mu(\tilde{x}) - \left(F_\mu(x_*) - \frac{\mu}{2}\|x_*\|_2^2\right) \le$$
$$\le F_\mu(\tilde{x}) - \min_{x\in Q} F_\mu(x) + \frac{\mu}{2}R_2^2 \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

□

The combination of Theorem 1.7 and Lemma 1.1 allow to improve the result of Theorem 1.5 in the convex case.

Theorem 1.8 (the role of the regularization) *Assume that* (1.20) *holds true and* $f(x,\xi)$ *is convex in* x *in* Q. *Then for* $\bar{x}^N_{\delta,\beta/2}\left(\{\xi^k\}_{k=1}^N\right)$ *to be a* $\left(\delta = \frac{\varepsilon^3}{8M_2^2R_2^2}, \frac{\beta}{2}\right)$*-solution of the empirical version of* (1.24)*:*

$$\min_{x\in Q}\left[\frac{1}{N}\sum_{k=1}^N f(x,\xi^k) + \frac{\varepsilon}{2R_2^2}\|x\|_2^2\right], \tag{1.25}$$

we have:

$$N = \tilde{O}\left(\frac{M_2^2R_2^2}{\varepsilon^2}\log\left(\frac{\log(M_2R_2/\varepsilon)}{\beta}\right)\right).$$

Moreover, in the general case $p \in [1,2]$ *the described above technique (with proper regularization) allows to obtain the bounds on* N *that correspond to the bounds in Theorem 1.6 up to logarithmic factors.*

To conclude, from the Theorem 1.8 we derive that in the sample complexity bounds online approach and offline approach (with proper regularization in the convex case) are equivalent up to a logarithmic factors.

1.2.3 *r*-growth condition

We say that $F(x) := \mathbb{E}_\xi f(x,\xi)$ satisfies *r-growth condition* ($r \geq 1$) on $Q_{2\varepsilon}$ if for all

$$x \in Q_{2\varepsilon} := \{x \in Q : \quad F(x) \leq F(x_*) + 2\varepsilon\} :$$

$$F(x) - F(x_*) \geq \mu_{p,r} \|x - x_*\|_p^r, \tag{1.26}$$

where x_* is a projection of x (in p-norm) on the set of the solutions of (1.9).

We relax the condition (1.20) as follows: for all $x, y \in Q$ *sub-Gaussian variance* of $f(y,\xi) - f(x,\xi) - (F(y) - F(x))$ bounded from above by $\lambda^2\|y-x\|_p^2$, i.e. for all $t \in \mathbb{R}$:

$$\mathbb{E}_\xi \left[\exp\left(t \cdot (f(y,\xi) - f(x,\xi) - (F(y) - F(x)))\right)\right] \leq \exp\left(t^2\lambda^2\|y-x\|_p^2/2\right). \tag{1.27}$$

Note that if (1.20) holds true, then $\lambda^2 \leq 2M^2$.

Theorem 1.9 *Assume that $f(x,\xi)$ is convex in x in Q and* (1.26), (1.27) *hold true. Then for $\bar{x}^N_{\varepsilon/2,\beta/2}\left(\{\xi^k\}_{k=1}^N\right)$:*

$$N = O\left(\frac{\lambda_p^2}{\mu_{p,r}^{2/r}\varepsilon^{2(r-1)/r}}\left(n\log\left(\frac{\bar{M}_p R_{p,\varepsilon}}{\varepsilon}\right) + \log\left(\frac{1}{\beta}\right)\right)\right), \tag{1.28}$$

where $\bar{M}_p = \mathbb{E}_\xi\left[M_p(\xi)\right]$ and for all ξ and $x, y \in Q$:

$$|f(y,\xi) - f(x,\xi)| \leq M_p(\xi)\|y-x\|_p; \tag{1.29}$$

$R_{p,\varepsilon}$ is the diameter of $Q_{2\varepsilon}$ in p-norm. In particular, for $r = 1$ $R_{p,\varepsilon} \leq 4\varepsilon/\mu_{p,1}$. Thus in the case of «sharp minimum» ($r = 1$) N does not depend on ε at all.

The bound (1.31) *is optimal up to a logarithmic factor.*

Proof Nazary, please add the proof of the Theorem based on [109] (https://cpn-us-w2.wpmucdn.com/sites.gatech.edu/dist/4/1470/files/2021/03/SPbook.pdf) Section 5.3.2. □

Optimality of (1.31)

Consider a simple example for $p = 2$, $Q = B_2^n(1)$:

$$f(x,\xi) = \|x\|_2^r - r\langle\xi, x\rangle,$$

$\xi \in \mathcal{N}(0,\sigma^2 I_n)$, where I_n – identity $n \times n$ matrix. Hence $F(x) = \|x\|_2^r$, $x_* = 0$, $\mu_{2,r} = 1$ in (1.26) and

$$f(y,\xi) - f(x,\xi) - (F(y) - F(x)) = r\langle\xi, y - x\rangle$$

has $\mathcal{N}\left(0, r^2\sigma^2\|y-x\|_2^2\right)$-distribution. Therefore $\lambda^2 = r^2\sigma^2$ in (1.27).

Note also that

$$\bar{F}(x) = \|x\|_2^r - r\langle\bar{\xi}_N, x\rangle,$$

where $\bar{\xi}_N \in \mathcal{N}(0, \sigma^2 N^{-1} I_n)$. For this problem we can explicitly find the minimizer of the empirical loss

$$\hat{x}^N \in \arg\min_{x\in Q} \bar{F}\,(x) = \frac{\bar{\xi}_N}{\|\bar{\xi}_N\|_2^b},$$

where

$$b = \begin{cases} 1, & \text{if } \|\bar{\xi}_N\|_2 > 1 \\ \frac{r-2}{r-1}, & \text{else.} \end{cases}$$

Since $F(x) = \|x\|_2^r$, it follows that

$$F(\hat{x}^N) - F(x_*) \le \varepsilon$$

is equivalent to

$$\|\bar{\xi}_N\|_2^{\frac{r}{r-1}} \le \varepsilon$$

for sufficiently small ε. Combining this with $\bar{\xi}_N \in \mathcal{N}(0, \sigma^2 N^{-1} I_n)$ we can get that for

$$\mathbb{P}\left(F\left(\hat{x}^N\right) - F(x_*) \le \varepsilon\right) = \mathbb{P}_{\bar{\xi}_N \sim \mathcal{N}(0,\sigma^2 N^{-1} I_n)}\left(\left\|\bar{\xi}_N\right\|_2^{\frac{r}{r-1}} \le \varepsilon\right) \ge 0.7$$

it is required that

$$N > \frac{n\sigma^2}{\varepsilon^{2(r-1)/r}}. \tag{1.30}$$

The lower bound (1.30) corresponds to (1.31) when $\mu_{2,r} = 1$ and r is finite. Note that when $r = 2$ and $\mu_{2,2} = \mu_2 \neq 1$ (1.30) can be clarified as follows

$$N \ge \frac{n\sigma^2}{\mu_2 \varepsilon}.$$

The last lower bound seems to be strange enough (n-factor in the lower bound looks wrong) due to the upper bound from (1.23). But there is no contradiction here even with the strengthened upper bound from (1.23)

$$N = \tilde{O}\left(\frac{\tilde{M}_2^2}{\mu_2 \varepsilon}\right),$$

since $\tilde{M}_2^2 := \mathbb{E}_\xi\left[M_2(\xi)^2\right] = nr^2\sigma^2$, where $M_2(\xi)$ was defined in (1.29).[13]

[13] In (1.23) it is assumed that there exists such M_2 that $M_2(\xi) \le M_2$. Here we relax the notion of M_2 to $\tilde{M}_2$.

Theorem 1.10 *Assume that $f(x,\xi)$ is convex in x in Q, $F(x) := \mathbb{E}_\xi f(x,\xi)$ satisfies r-growth condition in Q and* (1.20) *holds true. Then for $\bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$ generated by the proper (restarted Mirror Descent) modification of* (1.14)*:*

$$N = \tilde{O}\left(\frac{M_p^2}{\mu_{p,r}^{2/r}\varepsilon^{2(r-1)/r}}\right), \quad p \in [1,2]. \tag{1.31}$$

This bound is optimal up to logarithmic factor in the wide class of all reasonable ways to generate $\bar{x}^N\left(\{\xi^k\}_{k=1}^N\right)$.

Proof For clarity we consider only the euclidean case $p = 2$. Since $F(x) := \mathbb{E}_\xi f(x,\xi)$ satisfies r-growth condition in Q, it follows from (1.17) that with probability at least $1-\beta/\kappa$

$$\mu_{2,r}\|\bar{x}^N - x_*\|_2^r \le F(\bar{x}^N) - F(x_*) = O\left(\frac{M_2\|x^1 - x_*\|_2 \log(\kappa/\beta)}{\sqrt{N}}\right),$$

where $\bar{x}^N$ is calculated according to (1.18) based on (1.14). If we choose

$$N = O\left(\frac{M_2^2\log^2(\kappa/\beta)}{\mu_{2,r}^2\|x^1 - x_*\|_2^{2(r-1)}}\right),$$

then

$$\|\bar{x}^N - x_*\|_2^r = \frac{1}{2}\|x^1 - x_*\|_2^r.$$

The idea of the *restart technique* is to put

$$x^1 := \bar{x}^N$$

and to restart algorithm (1.14). By denoting $R_{2,l}$ the distance between the starting point and the solution x_* at l-th restart, we could guarantee that $R_{2,l+1}^r = R_{2,1}^r 2^{-l}$. Similarly, N_l is a number of iteration at l-th restart. Since we would like to solve the problem with probability at least $1-\beta$ and with accuracy ε, the number of the restarts κ is determined from

$$\frac{M_2R_{2,\kappa+1}\log(\kappa/\beta)}{\sqrt{N_{\kappa+1}}} \simeq \mu_{2,r}R_{2,\kappa+1}^r = \mu_{2,r}R_{2,1}^r 2^{-(\kappa+1)} \simeq \varepsilon.$$

Therefore the total number of samples (iterations) is

$$\sum_{l=1}^{\kappa} O\left(\frac{M_2^2 \log^2(\kappa/\beta)}{\mu_{2,r}^2 R_{2,l}^{2(r-1)}}\right) = \frac{M_2^2 \log^2(\kappa/\beta)}{\mu_{2,r}^2 R_{2,1}^{2(r-1)}} \sum_{l=1}^{\kappa} O\left(2^{\frac{2(r-1)}{r}l}\right) =$$

$$= O\left(\frac{M_2^2 \log^2(\kappa/\beta)}{\mu_{2,r}^2 R_{2,1}^{2(r-1)}} 2^{\frac{2(r-1)}{r}(\kappa+1)}\right) = O\left(\frac{M_2^2 \log^2(\kappa/\beta)}{\mu_{2,r}^2 R_{2,1}^{2(r-1)}} \left(\frac{\mu_{2,r} R_{2,1}^r}{\varepsilon}\right)^{\frac{2(r-1)}{r}}\right) =$$

$$= O\left(\frac{M_2^2 \log^2(\kappa/\beta)}{\mu_{2,r}^{2/r} \varepsilon^{2(r-1)/r}}\right).$$

□

1.3 Concluding remarks

For a better structure of this chapter we have collected various comments that clarify the results given above (but have not a primal interest) in this (separate) section at the end of the chapter. In more details most of this comments will be further developed in the next chapters.

1.3.1 Weakening of uniform Lipschitz condition in online approach

An important remark concerns online approach is that we can significantly relax uniform Lipschitz continuity property (1.20), assuming that $M_p(\xi)$ in (1.29) has a finite second moment $\mathbb{E}_\xi\left[M_p(\xi)^2\right] < \infty$. In this case, all the bound remain the same up to a logarithmic factor, see [78, 44] for $p = 2$, and [80, 54] for $p \geq 1$, but for the convergence in expectation. If we have only $\mathbb{E}_\xi\left[M_p(\xi)^{1+\alpha}\right] < \infty$, where $\alpha > 0$ then in the dimension-free case ($N \lesssim n$) the expected $N \sim \varepsilon^{-\max\{2,p\}}$ will get worse $N \sim \varepsilon^{-(1+\alpha_p)/\alpha_p}$, where $\alpha_p = \min\left\{1, \alpha, (p-1)^{-1}\right\}$ [82, 123]. Similarly, in the strongly convex case (r-growth condition) and in the case $N \gtrsim n$. Note that high-probability bound analysis has been developed in this generality mainly for euclidean proximal setup with $\mathbb{E}_\xi\left[M_2(\xi)^2\right] < \infty$.

For offline approach some particular results in this direction are also known, see the references in [109].

1.3.2 Weakening of the convexity condition

The principal difference between online and offline approach is that for optimal results in offline approach the convexity of $f(x, \xi)$ in x for all ξ is typically required. It was shown in [105] that for any regularizer there is a stochastic optimization problem with convex $F(x)$ such that regularized empirical loss minimization approach fails to learn. But for online approach the convexity of $F(x)$ is enough for the same rates of convergences in terms of convergence in expectation [80, 54].

In Section 1.2.1 we have observed that offline approach in non-convex case required $N \propto n\varepsilon^{-2}$ samples despite the fact that online approach in non-convex case required $N \propto \varepsilon^{-(n+1)}$ samples. Moreover, under different additional assumptions [106, 97, 8] (finite VC-dimension e.t.c.) the dependence of n in offline approach $N \propto n\varepsilon^{-2}$ can be relaxed.[14] So it seems that offline approach is much better than online. In terms of the sample complexity (number of different samples of ξ) it really is. But at the end in offline approach we should solve empirical loss (risk) minimization problem that would be non-convex. To solve this problem we required $N \propto n\varepsilon^{-(n+1)}$ stochastic gradient oracle calls[15] that corresponds (up to a factor n) to online approach.

Some results that were mentioned in the previous sections can be generalized if we replace (strong) convexity assumption by quasi-convexity or some growth condition [79] or Polyak–Lojasiewicz(–Lezansky) condition [57, 11]. For example, online and offline approaches under Polyak–Lojasiewicz condition are considered in [3] and [70].

1.3.3 How to make online approach adaptive?

To answer for this question we goes back to SGD (1.14)

$$x^{k+1} = \pi_Q \left(x^k - h_k \nabla_x f(x^k, \xi^k) \right),$$

with

$$h_k \equiv \frac{R}{M\sqrt{N}}.$$

The problem is that the stepsize policy requires the knowledge of R and M parameters. And this stepsize policy in not adaptive in N. We should know the desired N in advance. The last problem was solved in [80] by changing

$$h_k \equiv \frac{R}{M\sqrt{k}}.$$

This stepsize policy leads to the same convergence rate up to a logarithmic factor. This factor can be eliminated by the Nesterov's dual extrapolations scheme [85]. The problem of unknown M-parameter was further solved in [23, 24], where it was proved that for

[14] Factor n is replaced by the «efficient» dimension, which could be much smaller.

[15] This bound can be improved a little bit by using the fact that all the terms in the sum (the empirical loss) have the same distribution. But this improvement will have a minor effect on the total oracle complexity.

$$h_k \equiv \frac{R}{\sqrt{\sum_{j=1}^{k} \|\nabla_x f(x^j, \xi^j)\|_2^2}}$$

the rate convergence does not change up to a numerical constant factor. SGD with this stepsize policy is known as AdaGrad. The works [80, 23] largely determined the development of modern stochastic optimization. For example, one of the most cited stochastic optimization algorithm after SGD is Adam [58, 98], which is based on AdaGrad. This algorithm and its variations are one of the main tools to train Deep Neural Networks [67, 118].

Although in practice different adaptive algorithms show themselves well in the theory typically they converge in the worst case not better than non-adaptive analogues [9, 33].

1.3.4 Overparametrization

In practice for the strongly convex problems ($F(x)$ is μ-strongly convex in 2-norm):

$$\min_{x \in \mathbb{R}^n} F(x) := \mathbb{E}_\xi f(x, \xi)$$

with uniformly Lipschitz continuous gradient: for all $x, y \in \mathbb{R}^n$:[16]

$$\|\nabla_x f(y, \xi) - \nabla_x f(x, \xi)\|_2 \le L\|y - x\|_2 \tag{1.32}$$

it was observed that simple stochastic gradient method (SGD):

$$x^{k+1} = x^k - h\nabla_x f(x^k, \xi^k)$$

converges with linear rate in the vicinity of the solution x_* [76]. That was also confirmed in the theory

$$\mathbb{E}_{\{\xi^k\}_{k=1}^N} \left[\|x^{N+1} - x_*\|_2^2\right] \le \|x^1 - x_*\|_2^2 (1 - h\mu)^N + \frac{2h\sigma_*^2}{\mu}, \tag{1.33}$$

where the stepsize $h \le 1/(2L)$ and

$$\sigma_*^2 = \mathbb{E}_\xi \left[\|\nabla_x f(x_*, \xi) - \nabla F(x_*)\|_2^2\right] = \mathbb{E}_\xi \left[\|\nabla_x f(x_*, \xi)\|_2^2\right],$$

since $\nabla F(x_*) = 0$.

Indeed, from Section 1.1.2:

$$\|x^{k+1} - x_*\|_2^2 \le \|x^k - x_*\|_2^2 - 2h\langle \nabla_x f(x^k, \xi^k), x^k - x_*\rangle + h^2\|\nabla_x f(x^k, \xi^k)\|_2^2.$$

[16] As we will see in the next chapters it sufficiently to consider Lipschitz-type conditions only in some balls centered at starting point and radius determined (up to a logarithmic factor) by the distance between starting point and the closest to this point solution.

Taking the conditional expectation on ξ^k under fixed x^k and using that

$$\begin{aligned}\mathbb{E}_\xi\left[\|\nabla_x f(x,\xi)\|_2^2\right] \leq 2\mathbb{E}_\xi\left[\|\nabla_x f(x,\xi)-\nabla_x f(x_*,\xi)\|_2^2\right]+2\mathbb{E}_\xi\left[\|\nabla_x f(x_*,\xi)\|_2^2\right] \leq\\ \leq 4L\mathbb{E}_\xi\left[f(x,\xi)-f(x_*,\xi)-\langle\nabla_x f(x_*,\xi),x-x_*\rangle\right]+2\sigma_*^2 =\\ = 4L\left(F(x)-F(x_*)\right)+2\sigma_*^2,\end{aligned}$$

we obtain[17]

$$\begin{aligned}\mathbb{E}_{\xi^k}\left[\|x^{k+1}-x_*\|_2^2|x^k\right] \leq \|x^k-x_*\|_2^2-2h\langle\nabla F(x^k),x^k-x_*\rangle+\\ +h^2\left(4L\left(F(x)-F(x_*)\right)+2\sigma_*^2\right) \leq \|x^k-x_*\|_2^2-\\ -2h\left(F(x^k)-F(x_*)+\frac{\mu}{2}\|x^k-x_*\|_2^2\right)+\\ +4Lh^2\left(F(x)-F(x_*)\right)+2h^2\sigma_*^2.\end{aligned}$$

Rearranging the terms in RHS and taking mathematical expectation on x^k we come to the following:

$$\begin{aligned}\mathbb{E}_{\{\xi^j\}_{j=1}^k}\left[\|x^{k+1}-x_*\|_2^2\right] \leq (1-h\mu)\,\mathbb{E}_{\{\xi^j\}_{j=1}^{k-1}}\left[\|x^k-x_*\|_2^2\right]+\\ +2h\left(1-2Lh\right)\left(\mathbb{E}_{\{\xi^j\}_{j=1}^{k-1}}\left[F(x^k)\right]-F(x_*)\right)+2h^2\sigma_*^2 \leq\\ \leq (1-h\mu)\,\mathbb{E}_{\{\xi^j\}_{j=1}^{k-1}}\left[\|x^k-x_*\|_2^2\right]+2h^2\sigma_*^2,\end{aligned}$$

if $h \leq 1/(2L)$. So we come to (1.33) by induction.

The overparametrization effect appears if σ_*^2 is small, that is $\nabla_x f(x_*,\xi) \simeq 0$ for almost all ξ.

For example if we consider offline approach

$$\min_{x\in\mathbb{R}^n} \bar{F}(x) := \frac{1}{N}\sum_{k=1}^N f(x,\xi^k)$$

and reformulate this problems as

$$\min_{x\in\mathbb{R}^n} \bar{F}(x) := \mathbb{E}_k f(x,\xi^k), \tag{1.34}$$

where $\mathbb{P}\left(k=l\right)=1/N$ for $l=1,...,N$. In this case $L=\max_{l=1,...,N} L_l$, where L_l is Lipschitz gradient constant of $f(x,\xi^l)$ in x. The variance is

[17] The last inequality uses the weaker variant of μ-strong convexity assumption of $F(x)$: for all $x\in\mathbb{R}^n$

$$F(x_*) \geq F(x)+\langle\nabla F(x),x_*-x\rangle+\frac{\mu}{2}\|x_*-x\|_2^2.$$

$$\sigma_*^2 = \frac{1}{N}\sum_{k=1}^{N} \|\nabla_x f(x_*, \xi^k)\|_2^2.$$

If $\nabla_x f(x_*, \xi^k) \simeq 0$, which could be possible due to the same stochastic nature of all the terms $f(x, \xi^k)$, then for all $k = 1, ..., N$ we have overparametrization and effect of linear convergence of SGD to a small vicinity of the solution.

Although overparameterized problems have attracted considerable attention in recent years, the results available here are still far away from the theory we have described in the previous sections. For example, in offline approach with $\sigma_*^2 \simeq 0$ we have only [70]:

$$\mathbb{E}_{\{\xi^k\}_{k=1}^N} \left[\|\hat{x}^N - x_*\|_2^2\right] \propto \frac{1}{\mu^2 N^2},$$

rather than we have in online approach with proper stepsize policy $h = 1/(2L)$:

$$\mathbb{E}_{\{\xi^k\}_{k=1}^N} \left[\|x^{N+1} - x_*\|_2^2\right] \propto \left(1 - \frac{\mu}{2L}\right)^N .$$

Little is known about overparameterization in a non-euclidean proximal setup.

1.3.5 Acceleration and batching for smooth convex optimization problems in online approach

Consider smooth convex optimization problem

$$\min_{x \in Q} F(x), \tag{1.35}$$

where for all $x, y \in Q$:

$$\|\nabla F(x) - \nabla F(y)\|_2 \le L\|y - x\|_2. \tag{1.36}$$

Accelerated method [86, 66, 73] allows to solve smooth convex optimization problems with the rate

$$F(x^N) - F(x_*) \lesssim \frac{LR^2}{N^2},$$

where $R^2 = \|x^1 - x_*\|_2^2$ and x_* is the closest solution (in 2-norm) to x^1 if the set of the solutions contains more than one point. Below we describe how to build accelerated batch-parallelized algorithm that significantly outperform SGD in the number of subsequent iterations.

First of all, following [21, 30, 28] we introduce the notion of (δ_1, δ_2, L)-oracle. We say that for the problem (1.35) we have an access to (δ_1, δ_2, L)-oracle at a point x if we can evaluate a vector $\nabla_\delta F(x)$ such that, for all $x, y \in Q$,

$$-\delta_1 \le F(y) - F(x) - \langle \nabla_\delta F(x), y - x\rangle \le \frac{L}{2}\|y - x\|_2^2 + \delta_2,$$

where $\mathbb{E}\delta_1 = 0$ (δ_1 is independently taken at each oracle call), $\mathbb{E}\delta_2 \le \delta$. Note that the left inequality corresponds to the definition of δ_1-(sub)gradient [92] and reduces to the convexity property in the case $\delta_1 = 0$. In this case the LHS holds with $\nabla_\delta F(x) = \nabla F(x)$. The right inequality in the case when $\delta_2 = 0$ is a consequence[18] of (1.36). Let us consider an algorithm $\mathbf{A}(L, \delta_1, \delta_2)$ that converges with the rate[19]

$$\mathbb{E}F(x^N) - F(x_*) = O\left(\frac{LR^2}{N^\alpha} + N^\beta \delta\right). \tag{1.37}$$

The *batching technique*, applied to the problem (1.35) with L-Lipschitz gradient (in 2-norm), is based on the use of the mini-batch stochastic approximation of the gradient

$$\nabla_\delta F(x) = \frac{1}{B}\sum_{j=1}^{B} \nabla_x f(x, \xi^j)$$

in $\mathbf{A}(L, \delta_1, \delta_2)$, where $\{\xi^j\}_{j=1}^B$ are sampled independently and B is an appropriate batch size. The choice of B is based on the following relations

$$\langle \nabla_\delta F(x) - \nabla F(x), y - x\rangle \le \frac{1}{2L}\|\nabla_\delta F(x) - \nabla F(x)\|_2^2 + \frac{L}{2}\|y - x\|_2^2,$$

$$\mathbb{E}_{\{\xi^j\}_{j=1}^B}\left[\|\nabla_\delta F(x) - \nabla F(x)\|_2^2\right] \le \frac{\sigma^2}{B},$$

where σ^2 is the variance of unbiased stochastic gradient $\nabla_x f(x, \xi)$, which is available. Hence, if

$$\delta \le \frac{1}{2L}\max_{x\in Q}\mathbb{E}_{\{\xi^j\}_{j=1}^B}\left[\|\nabla_\delta F(x) - \nabla F(x)\|_2^2\right],$$

i.e. $\delta = \sigma^2/(2LB)$, we have that $\mathbf{A}(2L, \delta_1, \delta_2)$ converges with the rate given in (1.37). From (1.37) we see that to obtain

$$\mathbb{E}F(x^N) - F(x_*) \le \varepsilon$$

it suffices to take

$$N = O\left(\left(\frac{LR^2}{\varepsilon}\right)^{1/\alpha}\right) \quad \text{and} \quad B = O\left(\frac{\sigma^2 N^\beta}{L\varepsilon}\right).$$

[18] Note, that the right inequality in the case when $\delta_2 = 0$ is not equivalent to (1.36), but is typically sufficient to obtain optimal (up to constant factors) bounds on the rate of convergence of different methods [121].

[19] N is a number of iterations which up to a constant factor is equal to the number of (δ_1, δ_2, L)-oracle calls. We can consider more specific rates of convergence for problems with additional structure and develop *batching technique* in a similar way.

In particular, for all known Accelerated gradient methods we have that $\alpha = 2, \beta = 1$ [21, 30]. In this case, we obtain the complexity bounds for batched Accelerated gradient methods (assume that σ^2 is such that $T \geq N$, otherwise we put $T := N$):

$$N = O\left(\sqrt{LR^2/\varepsilon}\right), B = O\left(\sigma^2 R/\left(\sqrt{L}\varepsilon^{3/2}\right)\right), T = N \cdot B = O\left(\sigma^2 R^2/\varepsilon^2\right).$$

It is obvious that we can calculate batch in a parallel manner. This reduces the number of subsequent iterations from $N \propto \varepsilon^{-2}$ for standard SGD with small stepsize (see Section 1.1.2) and $N \propto \varepsilon^{-1}$ for SGD with special stepsize policy $h \simeq \min\{1/L, 1/(\mu N)\}$ [119] (see Section 1.3.4) to the optimal rate $N \propto \varepsilon^{-1/2}$ [82, 128]. Recently [126] this result was generalized to overparametrized problems, see Section 1.3.4.

The described above batching technique is very important and universal technique, which allows to build (optimal) stochastic algorithms based on the (optimal) deterministic algorithms and their analysis of convergence with inexact oracle. We mention here only the two most recent examples. In [39] batching technique was successfully applied in gradient-free optimization. In [74] batching technique was successfully applied for distributed strongly convex-concave saddle-point problems with different constants of strong convexity and strong concavity.

Note that the described technique can be further generalized to strongly convex problems (problems with r-growth condition) and non-euclidean proximal setup [31, 43].

1.3.6 Sum-type problems and offline approach

At the very end offline approach we should solve the empirical loss (risk) minimization problem

$$\min_{x \in Q} \left[\bar{F}(x) := \frac{1}{N}\sum_{k=1}^{N} f(x, \xi^k)\right]. \tag{1.38}$$

For clarity, we assume that $f(x, \xi)$ is μ-strongly convex and M-Lipscitz continuous in x in 2-norm, see (1.22), (1.20). According to Theorem 1.7 $N = \tilde{O}\left(M^2/(\mu\varepsilon)\right)$ and we should solve (1.38) with the accuracy $\delta \simeq \varepsilon^2\mu/M^2$. Unfortunately, without additional assumptions on the smoothness of $f(x, \xi)$ the complexity of this problem (the number of $\nabla_x f(x, \xi^k)$ calculations) is $\tilde{O}\left(M^2/(\mu\delta)\right)$ [82]. That is much worse than N. If we additionally assume that $f(x, \xi)$ has L-Lipscitz continuous gradient in x in 2-norm, see (1.32), then we can apply batch-parallelization and acceleration in the number of subsequent iterations, see Section 1.3.5. But this trick does not solve the problem of oracle complexity. We still required in $\tilde{O}\left(M^2/(\mu\delta)\right)$ calculations of $\nabla_x f(x, \xi^k)$. It seems that we come to some contradiction. Offline approach seems to be worse everytime than online one in terms of the oracle complexity. Fortunately, this is not the case. There exist randomized Variance Reduced (VR) algorithms (see, e.g. [127, 66, 73]) that allow to solve (1.38) (with accuracy δ) with the complexity:[20]

$$\tilde{O}\left(\left(N+\sqrt{N\frac{L}{\mu}}\right)\log\left(\frac{\Delta f}{\delta}\right)\right). \tag{1.39}$$

Under the natural assumption $L/\mu \lesssim N \simeq M^2/(\mu\varepsilon)$, i.e.[21] $L \lesssim M^2/\varepsilon$ this complexity coincide with N up to a logarithmic factor.

Moreover, for many concrete problems (e.g. Soft-SVM, see Section 1.1.1) we can efficiently reduce originally non-smooth problems to smooth one [5] and apply the VR algorithms.

The modern theory of VR methods is well developed, see e.g. [66]. For example, it include non-euclidean proximal setup.

In the core of VR methods lies a very simple idea, which goes back to Monte Carlo theory. Instead of stochastic gradient $\nabla_x f(x,\xi)$ it is proposed to consider the reduced one

$$\tilde{\nabla}_x f(x,\xi) = \nabla_x f(x,\xi) - \nabla_x f(\hat{x}^N,\xi),$$

where $\hat{x}^N$ is the solution of (1.38). Note that with stochastic gradient we have overparametization effect $\tilde{\nabla}_x f(\hat{x}^N,\xi) = 0$ (for all ξ) and therefore we can expect a linear convergence. Unfortunately in this form VR trick is not practical, since it is required to know $\hat{x}^N$. The proper correction of the trick consist in replacing $\nabla_x f(x^k,\xi^{t(k)})$ (where $t(k)$ is an index that equally likely and independently selected among $1,...,N$ at k-th iteration) with

$$\tilde{\nabla}_x f(x^k,\xi^{t(k)}) := \nabla_x f(x^k,\xi^{t(k)}) - \nabla_x f(\bar{x}^k,\xi^{t(k)}) + \nabla\bar{F}(\bar{x}^k),$$

where $\bar{x}^k$ periodically updated as $\bar{x}^k := x^k$ according to the different policies [66, 64]. With this stochastic gradient we may also expect overparametrization along with the convergence $x^k \to \hat{x}^N$. Indeed,

$$\mathbb{E}_{\xi^{t(k)}}\left[\|\tilde{\nabla}_x f(x^k,\xi^{t(k)})\|_2^2\right] \lesssim L\left(\bar{F}(\bar{x}^k) - \bar{F}(\hat{x}^N)\right) \to 0 \tag{1.40}$$

along with $\bar{x}^k \to \hat{x}^N$.

1.3.7 Composite optimization

From the previous sections we have known that regularizers in the empirical loss (risk) minimization approach play an important role. Sometimes this regularizers spoil the properties of the problem, e.g. $\|x\|_1$-regularizer in LASSO makes the

[20] This bound is optimal [127, 66], i.e. there are no algorithms that work only with $\nabla_x f\,(x,\,\xi^k)$ and has a better complexity.

[21] One can always achieve this condition by smoothing a non-smooth problem. In this case $L \simeq M^2/\varepsilon$ [84, 127].

problem non-smooth, see Section 1.1.1. We can solve this issue by using composite optimization approach. Let us remind that standard SGD (1.14) has a following structure:

$$x^{k+1} = \pi_Q \left(x^k - h_k \nabla_x f(x^k, \xi^k) \right) =$$
$$= \arg\min_{x \in Q} \left\{ \langle \nabla_x f(x^k, \xi^k), x - x^k \rangle + \frac{1}{2h} \|x - x^k\|_2^2 \right\}.$$

If the stochastic optimization problem is regularized (i.e. has composite term):

$$\min_{x \in Q} \left[\mathbb{E}_\xi \left[f(x, \xi) \right] + r(x) \right].$$

we could correct the described procedure as follows [131]:

$$x^{k+1} = \arg\min_{x \in Q} \left\{ \langle \nabla_x f(x^k, \xi^k), x - x^k \rangle + \frac{1}{2h} \|x - x^k\|_2^2 + r(x) \right\}.$$

The iteration complexity does not change. But the auxiliary (projection) problem becomes more difficult. Fortunately, for some concrete examples (e.g. LASSO) the auxiliary problem almost retains its complexity. In this case composite term is called «proximal-friendly». The same holds true for Accelerated batched algorithms [22] and VR algorithms [66].

In the case of non proximal-friendly composite terms it happens that we can split the oracle complexities for two terms [66, 61], see also Section 1.3.9. This turned out to be an extremely useful option in distributed optimization [66, 45, 103, 61].

Composite optimization was firstly developed in deterministic setup [10, 25, 83]. Moreover, in [81, 120] it was considered more general «model setup» with $F(x) := \min\{F_1(x), ..., F_m(x)\}$ and composite optimization as particular cases. Under some assumptions this model setup can be further developed on stochastic optimization problems [30].

1.3.8 Overfitting and early stopping for offline approach

Let us return to the empirical problem (1.38):

$$\min_{x \in Q} \left[\bar{F}(x) := \frac{1}{N} \sum_{k=1}^{N} f(x, \xi^k) \right].$$

In Section 1.2.2 we describe regularization trick, that allows to align sample complexities for offline and online approaches for convex, but non-strongly convex problems. Another (a little artificial) way to align sample complexities in both of the approaches is to change the way of obtaining $\bar{x}^N_{\delta,\beta/2}\left(\{\xi^k\}_{k=1}^N\right)$, which is based on sufficiently accurate solution of (1.38) (or its regularized version). The idea is trivial: to «solve»

(1.38) by using SGD with samples $\{\xi^k\}_{k=1}^N$ without repeating. So the first N iteration of this algorithm is completely coincide with standard SGD iterations (1.14). An interesting phenomena is that further iterations of SGD based on the same sample set $\{\xi^k\}_{k=1}^N$ not only improve the quality of the solution (this quality is measured in terms of initial stochastic optimization problem!), but can also provably lead to a decrease in quality (overfitting).

This idea was further developed in seminal work [47], where it was shown that for the standard SGD (with output $\bar{x}^T$ after T iterations, see (1.18)) applied to smooth convex (but not strongly convex!) empirical problem (without any regularization!) in the expectation form (1.34):

$$F(\bar{x}^T) - F(x_*) \propto N^{-1/2} \text{ if } T \propto N.$$

This phenomenon sometimes called «early stopping» [40]. The work [47] initiated a lot of activity around overfitting properties of SGD applied to the empirical problems, see the survey in [70]. In particular, for smooth convex (but not strongly convex!) problems in [72] it was shown that

$$F(\bar{x}^T) - F(x_*) \propto N^{-\eta/(1+\eta)} \text{ if } T \propto N^{2/(1+\eta)}, \eta \in (0,1].$$

It means that too many iteration lead to overfitting. For smooth strongly convex problems it was shown [70] that

$$F(\bar{x}^T) - F(x_*) \propto (\mu N)^{-1} \text{ if } T \propto (N/\mu)^2,$$

which corresponds to (1.12). So in the strongly convex case we do not expect the early stopping effect (this effect was described above as an alternative to regularization) and overfitting effect.

More stronger overfitting effect can be observed if one replace SGD with Gradient Descent (GD) [6, 105]:

$$x^{k+1} = x^k - h\nabla \bar{F}(x^k).$$

In particular, for smooth convex empirical loss minimization problems the better rate of convergence, than

$$F(\bar{x}^T) - F(x_*) \propto N^{-5/12}$$

is impossible (without additional assumptions) independently of what is T and h [105]. Remind that at the same assumptions for SGD we have $F(\bar{x}^T) - F(x_*) \propto N^{-1/2}$ if $T \propto N$. This rate is better, since $1/2 = 6/12 > 5/12$.

Despite all this in practice one can often met that (1.38) with proper regularization is solved by fast deterministic algorithms, say, LBFGS or even by using high-order schemes, see Section 1.3.10. It works due to proper regularization!

1.3.9 Distributed optimization

In Section 1.3.5 we met with batch-parallelization consist in possibility to parallelize the batch calculation:

$$\frac{1}{B}\sum_{j=1}^{B}\nabla_x f(x,\xi^j).$$

If we assume that we have the number of nodes that is a division of B, then we can fully parallelize on these nodes batch calculation. But at each subsequent iteration of considered accelerated algorithm (after the batch calculation) all the nodes are required to share theirs sub-batches. In distributed optimization this is called – communication. So one iteration assume one communication. The natural question appears: does such number of communications is also optimal like the number of subsequent iterations? In general the answer is affirmative [125]. It means that without additional assumptions batched accelerated methods are the best ones in Federated Learning (FL) setup from the theoretical point of view [55]. This conclusion looks somewhat discouraging since from the practice it is well known that local steps (the main ingredient of FL) works good. To explain this contradiction let us consider unconstrained convex quadratic stochastic optimization problem. An important property of accelerated dynamics is its linearity (on average) in terms of x. This linearity generates superposition principle: instead of communication at each iteration we can to run independently at each node accelerated algorithm with reduced (to the number of nodes) batch-size and we communicate only one time at the very end (at the last iteration) by calculating an average of the outputs at all the nodes (this procedure is called «one shoot»). The total output of this approach will have the same quality as the approach we started with [124] (with many communications).

It means that for quadratic stochastic optimization problems (and close to quadratic ones) local steps give tangible benefits. Since quadratic problems are naturally appears as a local approximation of real problem in the vicinity of the solution or at each iteration as an inner problem (for example, iteration of Newton method [16]) we can still exploit local steps. One such example we consider at the very end of this section.

It is interesting to note, that rather than for deterministic distributed convex optimization problems for stochastic convex optimization problems there is a significant difference between the class of quadratic problems and convex ones [82, 125].

More naturally distributed setup appears when dealing with offline approach. For example, if we have m nodes (such that $N = m \cdot s$ for some natural s) we can rewrite the empirical loss minimization problem (1.38) as follows:

$$\min_{x\in Q}\left[\bar{F}(x) := \frac{1}{m}\sum_{k=1}^{m}\bar{f}_k(x) := \frac{1}{m}\sum_{k=1}^{m}\frac{1}{s}\sum_{j=1}^{s} f(x,\xi^{k,j})\right].$$

If we apply standard accelerated method [86] assuming that $\bar{F}(x)$ is μ-strongly convex in 2-norm and has L-Lipscitz gradient, then the number of iterations (communications) will be $\tilde{O}\left(\sqrt{L/\mu}\right)$ (here and below in this section we skip all the logarithmic for for a better visibility) and the number of incremental gradient oracle calls at each node will be $\tilde{O}\left(s\sqrt{L/\mu}\right)$ (the number of $\nabla_x f(x,\xi^{k,j})$ calculation).

Section 1.3.6 gives a hope that this bound can be further improved due to the sum-type structure of the functions stored at each node. Indeed, there exist a distributed accelerated VR scheme [69] with $\tilde{O}\left(\sqrt{L/\mu}\right)$ communication complexity and $\tilde{O}\left(s + \sqrt{sL/\mu}\right)$ oracle complexity in each node, where L in the last formula is a maximal Lipschitz gradient constant in x in 2-norm of functions $f(x, \xi^{k,j})$. This bound is optimal [49] if we do not use that $\left\{\xi^{k,j}\right\}$ are i.i.d. or do not use that among $\bar{f}_k(x)$ there exists some kind of similarity. In more details, if Lipschitz gradient constants of $\bar{F}(x) - \bar{f}_k(x)$ are bounded in 2-norm by l ($l \ll L$) than we may expect better communication complexity $\tilde{O}\left(\sqrt{l/\mu}\right)$, which corresponds to the lower bound under similarity [7].

To use similarity we describe **Accelerated gradient sliding** for unconstrained composite optimization problem:

$$\min_{x \in \mathbb{R}^n} \left[\bar{F}(x) := g(x) + r(x)\right],$$

where $g(x)$ has L_g-Lipschitz continuous gradient, $r(x)$ is convex and has L_r-Lipschitz continuous gradient ($L_g \le L_r$); $\bar{F}(x)$ is μ-strongly convex function in 2-norm. Note that we do not assume $g(x)$ to be convex! The algorithm looks as follows [61]:

$$\tilde{x}^t = \tau x^t + (1-\tau) x_f^t,$$

$$x_f^{t+1} \approx \operatorname*{argmin}_{x \in \mathbb{R}^n} \left[A^t(x) := g(\tilde{x}^t) + \langle \nabla g(\tilde{x}^t), x - \tilde{x}^t \rangle + L_g \|x - \tilde{x}^t\|_2^2 + r(x)\right], \quad (1.41)$$

which means

$$\|\nabla A^t(x_f^{t+1})\|_2^2 \le \frac{L_g^2}{3} \left\|\tilde{x}^t - \arg\min_{x \in \mathbb{R}^n} A^t(x)\right\|_2^2, \quad (1.42)$$

$$x^{t+1} = x^t + \eta\mu(x_f^{t+1} - x^t) - \eta \nabla \bar{F}(x_f^{t+1}),$$

where

$$\tau = \min\left\{1, \frac{\sqrt{\mu}}{2\sqrt{L_g}}\right\}, \quad \eta = \min\left\{\frac{1}{2\mu}, \frac{1}{2\sqrt{\mu L_g}}\right\}.$$

This algorithm (with output point x^N) has an iteration complexity

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right)$$

and solves several tasks at once:

- **(simple acceleration)** If $r(x) \equiv 0$ this algorithm becomes an ordinary accelerated method with
$$x_f^{t+1} = \tilde{x}^t - \frac{1}{2L_p} \nabla g(\tilde{x}^t);$$

- **(Catalyst)** If $g(x) \equiv 0$ this algorithm becomes a Catalyst-type proximal envelop [71], but less sensitive to the accuracy of the solution of (1.41);[22]
- **(Sliding)** If we apply to (1.41) Accelerated gradient sliding with $g(x) := r(x)$ then obtain the total complexity of $\nabla r(x)$ oracle as

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \cdot O\left(\sqrt{\frac{L_g + L_r}{L_g}}\right) = \tilde{O}\left(\sqrt{\frac{L_r}{\mu}}\right).$$

That is, we have split the complexity of considered composite problem to the complexities correspond to the separate problems:

$$\tilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right) \quad \text{for } \#\nabla g(x) \quad \text{and} \quad \tilde{O}\left(\sqrt{\frac{L_r}{\mu}}\right) \quad \text{for } \#\nabla r(x).$$

Let us rewrite the empirical problem as follows

$$\min_{x \in \mathbb{R}^n} \left[\bar{F}(x) := \left(\bar{F}(x) - \bar{f}_1(x)\right) + \bar{f}_1(x)\right]. \tag{1.43}$$

Denoting the first sum as $g(x)$ and the second one as $r(x)$ we can use Sliding trick to split the complexities. Note that we significantly use the fact that in this scheme $g(x)$ is not necessarily convex! So it remains only to notice that described Accelerated gradient sliding under this choice of $g(x)$ and $r(x)$ has a natural distributed interpretation.[23] It gives at the end a distributed algorithm that works according to the lower bounds for communications and oracle calls per node complexities under similarity [7]. Due to the statistical (i.i.d.) nature of $\left\{\xi^{k,j}\right\}$ (statistical similarity) one may expect that [50]: $L_g \propto s^{-1/2}$.

Thus, the number of communications for the developed algorithm is proportional to $\propto \sqrt{1/\sqrt{s}\mu}$ and the number of incremental gradient oracle calls at each node remains the same as for ordinary accelerated method $\tilde{O}\left(s\sqrt{L/\mu}\right)$. It means that we indeed improve the communication complexity by using statistical similarity. But at the same time we have worsened the oracle complexity per node in comparison with VR accelerated method, which uses sum-type structure of the terms stored in each nodes. An open question is to build an «intermediate» algorithm – some kind of convex combination of VR and Sliding with statistical similarity. The parameter

[22] From Catalyst technique one can obtain (1.39) based on (1.40), restarts (see the proof of Theorem 1.10) and accelerated batched algorithm, see Section 1.3.5. Note also the paper [18], where the authors independently proposed stable version (to the accuracy of the solution of (1.41)) of Catalyst. Both of these versions are «logarithmic-free» (do not introduce additional logarithmic multipliers compared to direct acceleration), rather than initial one [71].

[23] Indeed, we can assign the node number 1 to be a master node that minimize at each iteration (1.41) with $g(x) := \bar{F}(x) - \bar{f}_1(x)$ and $r(x) := \bar{f}_1(x)$. It is obvious, that $r(x)$ is available to the master node and $\nabla g(\tilde{x}^t)$ can be available due to communications of the master node with the other ones. At each round of communications k-th node sends $\nabla \bar{f}_k(\tilde{x}^t)$ to the master node and receive in return x_f^{t+1}, which is calculated at the master node.

of this convex combination is determined in practice by the ratio of arithmetic complexities of one oracle call to one communication.

Note that in (1.43) instead of $\bar{f}_1(x)$ we can take an arbitrary smooth convex functions. In particular we can take Taylor series expansion

$$\tilde{f}_1(x) := \bar{f}_1(\tilde{x}^t) + \langle \nabla \bar{f}_1(\tilde{x}^t), x - x^t \rangle + \frac{1}{2!}\langle \nabla^2 \bar{f}_1(\tilde{x}^t)(x - \tilde{x}^t), x - x^t \rangle.$$

Note that $\tilde{f}_1(x)$ – convex function, rather than $\bar{F}(x) - \tilde{f}_1(x)$. Under the third-order smoothness assumption one may expect that $\tilde{f}_1(x)$ has a close hessian to the hessian of $\bar{f}_1(x)$ in the vicinity of $\tilde{x}^t$. Thus we may expect this method to be required only few communication steps when the number of iteration t is large. Note that in this approach we not only have similarity on higher iterations, but also have a quadratic structure for auxiliary problem (1.41). In case of stochastic (randomized) oracle this structure allows to use accelerated one-shoot local methods for (1.41), which strength the effect of communications saving.

In this section we consider distributed centralized algorithms. Some of the results mentioned above have analogues also in decentralized setup, see [45] and references there in.

1.3.10 Accelerated tensor methods

Starting with the work [88] the interest in tensor methods (i.e. the methods that used high-order derivatives) in convex optimization began to grow steadily. In particular, an optimal[24] (up to a logarithmic complexity factor for line-search procedure) second-order method was proposed in [75] and an optimal (also, up to a logarithmic factor) high-order method was proposed in [38]. In [87] it was shown that second and third-order tensor methods are implementable – complexity of each iteration is roughly the same as for Newton method. Optimal methods without line-search (that work according to the lower bounds up to a constant factor) were recently proposed in [63, 17]. Thus the deterministic theory of tensor methods for convex (unconstrained) problems seems to be close to the final point. In Section 1.3.5 we have demonstrated the profit of acceleration in online approach for smooth problems. Fortunately, we can additionally improve the results of Section 1.3.5 by using accelerated tensor methods. For that we need to develop sensitivity analysis of these methods. Such an analysis was made in [1] for accelerated tensor methods according to Nesterov-type of acceleration under high-order smoothness assumption [87]. This acceleration is a little bit worse than the best one Monteiro–Svaiter acceleration [75, 38, 63]. By using the results of [1] and batching technique one can improve the number of subsequent iterations in online approach from Section 1.3.5. If n is not too big then such improvement can be valuable also in terms of arithmetic complexity.

[24] See [60, 37] and references there in for lower bounds.

For offline approach the main motivation to use tensor methods is coming from the similarity approach, see Section 1.3.9. Where the reduced auxiliary problem (1.41)

$$\min_{x\in\mathbb{R}^n}\left[\langle\nabla\bar{F}(\tilde{x}^t)-\nabla\bar{f}_1(\tilde{x}^t),x-\tilde{x}^t\rangle+l\|x-\tilde{x}^t\|_2^2+\frac{1}{s}\sum_{j=1}^{s}f(x,\xi^{1,j})\right]$$

is a sum type problem with the reduced number of terms s ($s \ll N$). If $s \simeq n$ we have that for Newton-type methods the complexity of one iteration is upper bounded by the Hessian-matrix inversion, rather than the complexity of Hessian calculation by itself. In other words, in this case second and third-order tensor methods do not «feel» the sum-type structure of the problem and work with almost the same complexities as if $s = 1$. This idea reduces the number of subsequent iterations of second and third-order methods for inner (auxiliary) problems and simultaneously alleviates the main drawback of tensor methods related with expansive iterations [32].[25]

1.3.11 Saddle-point problems and Variational inequalities

Offline approach to the stochastic Saddle-point problems (SPP) developed in [68, 132, 29, 90], see also [62] for distributed approach. Online approach to the stochastic Variational Inequalities (VI) – and as a consequence for saddle-point problems – developed in [53, 41, 42].

Roughly speaking, all the results for both of the approaches look very similar to the results mentioned in the previous sections for the stochastic optimization problems except absence of acceleration. But there still exist open problems for SPP and VI that were closed for optimization problems. For example, randomized VR algorithms for (strongly) convex problems match the lower complexity bound (see Section 1.3.6), rather than its SPP and VI analogues [4, 46].

1.3.12 Wasserstein barycenter example

Wasserstein barycenter (WB) problem and its dual entropy-smoothing version is an extremely interesting example in many ways at once. First of all, stochastic optimization (population) WB problem formulation comes from Statistics, but is not due to the principle of maximum likelihood [14, 12]. So we may consider this example to be intermediate in terms of Section 1.1.1 and Section 1.1.2. Secondly, the empirical WB problem as a convex optimization problem has an efficient saddle-

[25] Since we have to calculate the sum the iteration must be expensive independently of the order of the method we use. This observation opens up the possibility to increase the order of the method by conserving the complexity of iteration.

point and dual representations [26]. For example, when WB problem solved on the space of finite-support measures (on n points) the complexity of primal gradient oracle is $\tilde{O}\left(n^3\right)$ a.o. (arithmetic operations) and the complexity of dual gradient oracle is $O\left(n^2\right)$ a.o. Moreover, dual gradient oracle has a natural stochastic unbiased estimation with the complexity $O\left(n\right)$ a.o. For some real-world applications $n \simeq 10^6$. Hence mentioned above computational observations play an important role [26]. Thirdly, the possibility to use dual oracle appears only in offline approach. To make this approach correct we need proper regularization [27], see also Theorem 1.8 for euclidean case. This regularization should be non-euclidean, since we have simplex constraint – barycenter is a measure, that is an element of probability simplex $S_n(1)$. Fourthly, WB problem is non-smooth, but strongly convex in 2-norm on $S_n(1)$ if we consider dual entropy-smoothing version [26]. Since the problem is non-smooth it is impossible to use batch-parallelization in online approach, see Section 1.3.5. But due to the strong convexity (comes from regularization or/and from dual entropy-smoothing) the dual problem (in offline approach) is smooth [100] and we can apply distributed (batched-parallelized) accelerated methods to solve it [26]. To conclude, WB problem is an interesting example of the problem for which offline approach motivated not only the ability to distribute calculations across nodes (what is typical of the offline approach in general), but also the possibility to solve dual problem with better properties: cheaper oracle and better iteration-complexities bounds, since smoothness without strong convexity (for dual problem) is better than strong convexity without smoothness (for primal problem).

At the end we mentioned that the empirical WB problem is not well suited for modern distributed Variance Reduced (VR) schemes and algorithms that use similarity. The reason is a simplex constraint. Although for euclidean proximal setup distributed VR is well developed [62] as well as similarity [61], but for non-euclidean proximal setup (generated by the simplex constraint) the results are absent.

With this remark, we wanted to demonstrate that despite the huge progress made in the last decade in convex stochastic programming, there are still a lot of open problems that looks like a minor generalization of already solved ones. Apparently, solving such problems will require the involvement of new ideas.

1.4 Historical Notes

Stochastic optimization has began to take shape in an independent field of knowledge for about 70 years ago starting with the seminal paper of H. Robbins and S. Monro [99]. This field was actively developed along with the usual optimization. In particular, in an outstanding book of A.S. Nemirovski and D.B. Yudin [82] (original version of the book was dated by 1979) the complexity theory of modern convex optimization was build. This theory included stochastic gradient oracle. So we may consider 1979 as a second (theoretical) birth of stochastic optimization. The third significant wave of the interest happened for about 20 years ago in accordance with Data Science applications. It is already impossible to imagine modern data analysis without stochastic optimization. For the moment many books were written around Stochastic optimization [34, 13, 104, 96, 111, 109]. In some books and surveys one can find Data Science applications of Stochastic Optimization [77, 115, 116, 106, 97, 129, 24, 15, 8, 130].

The results of Section 1.1.1 are rather standard and can be mainly find in [51, 114]. An example of Vadim V. Mottl was taken from [65]. Non-asymptotic results can be found in [112, 113]. Polyak–Juditsky–Ruppert averaging was separately proposed in [102] and [93, 91]. Online analogue of Fisher's theorem was developed in [94, 95].

The results of Section 1.1.2 were motivated by the papers [56, 117, 107, 108]. Online learning is well presented in [19, 48, 89, 20]. Note that for the convex case (not strongly convex) the described results can be generalized to non-euclidean proximal setup. The most interesting applications related with unit simplex $Q = S_n(1)$ [19]. Note that in this section we started to use the notion of (unbiased) stochastic subgradient $\nabla_x f(x, \xi)$ without accurate definition of this subject in the non-smooth case. The problems appear when the subgradient is not unique. In this case we understand under $\nabla_x f(x, \xi)$ some kind of measurable selector (no matter what kind of selector). More accurate definitions and properties of stochastic gradient one can find in [109].

The results of Section 1.2.1 were taken from [110, 80, 109]. The tight lower bound for online case was obtained in [2]. The tight lower bound for offline case (for smooth convex problems) was obtained in [35].

Online results of Section 1.2.2 corresponds to [54]. Offline results of Section 1.2.2 corresponds to [107, 106]. High-probability bounds investigated in [36, 59]. Tikhonov's regularization was accurately developed in [122]. For non-euclidean case offline results were generalized in [27, 29].

Online results of Section 1.2.3 were taken from [110, 109]. Offline results of Section 1.2.3 were taken from [54] for the case $r = 2$. For the case $r = 1$ this result was obtained earlier in a different manner [52]. The idea of restarts for strongly convex problems goes back to [82, 81]. For the stochastic optimization problems it was developed in [54]. For a sharp minimum and deterministic optimization convex optimization problems restarts was developed in [101].

References

1. Artem Agafonov, Dmitry Kamzolov, Pavel Dvurechensky, Alexander Gasnikov, and Martin Takac. Inexact tensor methods and their application to stochastic convex optimization, 2020.
2. Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.
3. Ahmad Ajalloeian and Sebastian U. Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.
4. Ahmet Alacaoglu and Yura Malitsky. Stochastic variance reduction for variational inequality methods. *arXiv preprint arXiv:2102.08352*, 2021.
5. Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. *Advances in Neural Information Processing Systems*, 29, 2016.

6. Idan Amir, Yair Carmon, Tomer Koren, and Roi Livni. Never go full batch (in stochastic convex optimization). In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25033–25043. Curran Associates, Inc., 2021.
7. Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
8. Francis Bach. Learning theory from first principles. *e-print,* `https://www.di.ens.fr/~fbach/ltfp_book.pdf`, 2021.
9. Francis Bach and Kfir Y. Levy. A universal algorithm for variational inequalities adaptive to smoothness and noise. In *Conference on learning theory*, pages 164–194. PMLR, 2019.
10. Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
11. Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.
12. Jérémie Bigot and Thierry Klein. Characterization of barycenters in the wasserstein space by averaging optimal transport maps. *ESAIM: Probability and Statistics*, 22:35–57, 2018.
13. John R. Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
14. Emmanuel Boissard, Thibaut Le Gouic, and Jean-Michel Loubes. Distribution's template estimate with wasserstein metrics. *Bernoulli*, 21(2):740–759, 2015.
15. Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.
16. Brian Bullins, Kshitij Patel, Ohad Shamir, Nathan Srebro, and Blake E. Woodworth. A stochastic newton algorithm for distributed convex optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26818–26830. Curran Associates, Inc., 2021.
17. Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive monteiro-svaiter acceleration. *arXiv preprint arXiv:2205.15371*, 2022.
18. Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Recapp: Crafting a more efficient catalyst for convex optimization. *arXiv preprint arXiv:2206.08627*, 2022.
19. Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
20. Nicolo Cesa-Bianchi and Francesco Orabona. Online learning algorithms. *Annual review of statistics and its application*, 2021.
21. Olivier Devolder. *Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization*. PhD thesis, PhD thesis, 2013.
22. Benjamin Dubois-Taine, Francis Bach, Quentin Berthet, and Adrien Taylor. Fast stochastic composite minimization and an accelerated frank-wolfe algorithm under parallelization. *arXiv preprint arXiv:2205.12751*, 2022.
23. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
24. John C. Duchi. Introductory lectures on stochastic optimization. *The mathematics of data*, 25:99–186, 2018.
25. John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, volume 10, pages 14–26. Citeseer, 2010.
26. Darina Dvinskikh. Decentralized algorithms for wasserstein barycenters. *arXiv preprint arXiv:2105.01587*, 2021.
27. Darina Dvinskikh. Stochastic approximation versus sample average approximation for wasserstein barycenters. *Optimization Methods and Software*, pages 1–33, 2021.
28. Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallel primal and dual accelerated methods for stochastic convex programming problems. *Journal of Inverse and Ill-posed Problems*, 29(3):385–405, 2021.

29. Darina Dvinskikh, Vitali Pirau, and Alexander Gasnikov. On the relations of stochastic convex optimization problems with empirical risk minimization problems on p-norm balls. *arXiv preprint arXiv:2202.01805*, 2022.
30. Darina Dvinskikh, Alexander Tyurin, Alexander Gasnikov, and Sergey Omelchenko. Accelerated and nonaccelerated stochastic gradient descent with model conception. *Math. Notes*, 108(4):511–522, 2020.
31. Pavel Dvurechensky and Alexander Gasnikov. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.
32. Pavel Dvurechensky, Dmitry Kamzolov, Aleksandr Lukashevich, Soomin Lee, Erik Ordentlich, Cesar A Uribe, and Alexander Gasnikov. Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization. *arXiv preprint arXiv:2102.08246*, 2021.
33. Alina Ene, Huy L. Nguyen, and Adrian Vladu. Adaptive gradient methods for constrained convex optimization and variational inequalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7314–7321, 2021.
34. Yu.M. Ermoliev and R.J.-B. Wets. *Numerical techniques for stochastic optimization*. Springer-Verlag, 1988.
35. Vitaly Feldman. Generalization of erm in stochastic convex optimization: The dimension strikes back. *Advances in Neural Information Processing Systems*, 29:3576–3584, 2016.
36. Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pages 1270–1279. PMLR, 2019.
37. Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif. Near-optimal lower bounds for convex optimization for all orders of smoothness. *Advances in Neural Information Processing Systems*, 34:29874–29884, 2021.
38. Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, César A Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, et al. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In *Conference on Learning Theory*, pages 1392–1393. PMLR, 2019.
39. Alexander Gasnikov, Anton Novitskii, Vasilii Novitskii, Farshed Abdukhakimov, Dmitry Kamzolov, Aleksandr Beznosikov, Martin Takáč, Pavel Dvurechensky, and Bin Gu. The power of first-order smooth optimization for black-box non-smooth problems. *arXiv preprint arXiv:2201.12289*, 2022.
40. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
41. Eduard Gorbunov, Hugo Berard, Gauthier Gidel, and Nicolas Loizou. Stochastic extragradient: General analysis and improved rates. In *International Conference on Artificial Intelligence and Statistics*, pages 7865–7901. PMLR, 2022.
42. Eduard Gorbunov, Marina Danilova, David Dobre, Pavel Dvurechensky, Alexander Gasnikov, and Gauthier Gidel. Clipped stochastic methods for variational inequalities with heavy-tailed noise. *arXiv preprint arXiv:2206.01095*, 2022.
43. Eduard Gorbunov, Marina Danilova, and Alexander Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *Advances in Neural Information Processing Systems*, 33:15042–15053, 2020.
44. Eduard Gorbunov, Marina Danilova, Innokentiy Shibaev, Pavel Dvurechensky, and Alexander Gasnikov. Near-optimal high probability complexity bounds for non-smooth stochastic optimization with heavy-tailed noise. *arXiv preprint arXiv:2106.05958*, 2021.
45. Eduard Gorbunov, Alexander Rogozin, Aleksandr Beznosikov, Darina Dvinskikh, and Alexander Gasnikov. Recent theoretical advances in decentralized distributed convex optimization. *arXiv preprint arXiv:2011.13259*, 2020.
46. Yuze Han, Guangzeng Xie, and Zhihua Zhang. Lower complexity bounds of finite-sum optimization problems: The results and construction. *arXiv preprint arXiv:2103.08280*, 2021.

47. Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1225–1234, New York, New York, USA, 20–22 Jun 2016. PMLR.
48. Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
49. Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An optimal algorithm for decentralized finite-sum optimization. *SIAM Journal on Optimization*, 31(4):2753–2783, 2021.
50. Hadrien Hendrikx, Lin Xiao, Sebastien Bubeck, Francis Bach, and Laurent Massoulie. Statistically preconditioned accelerated gradient method for distributed optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4203–4227. PMLR, 13–18 Jul 2020.
51. Ildar Abdulovich Ibragimov and Rafail Zalmanovich HasMinskii. *Statistical estimation: asymptotic theory*, volume 16. Springer Science & Business Media, 2013.
52. Anatoli Juditsky. A stochastic estimation algorithm with observation averaging. *IEEE transactions on automatic control*, 38(5):794–798, 1993.
53. Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.
54. Anatoli Juditsky and Yuri Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stochastic Systems*, 4(1):44–80, 2014.
55. Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.
56. Sham M. Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. *Advances in Neural Information Processing Systems*, 21, 2008.
57. Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak–lojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 795–811. Springer, 2016.
58. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
59. Yegor Klochkov and Nikita Zhivotovskiy. Stability and deviation optimal risk bounds with convergence rate $O(1/n)$. *arXiv preprint arXiv:2103.12024*, 2021.
60. Guy Kornowski and Ohad Shamir. High-order oracle complexity of smooth and strongly convex optimization. *arXiv preprint arXiv:2010.06642*, 2020.
61. Dmitry Kovalev, Aleksandr Beznosikov, Ekaterina Borodich, Alexander Gasnikov, and Gesualdo Scutari. Optimal gradient sliding and its application to distributed optimization under similarity. *arXiv preprint arXiv:2205.15136*, 2022.
62. Dmitry Kovalev, Aleksandr Beznosikov, Abdurakhmon Sadiev, Michael Persiianov, Peter Richtárik, and Alexander Gasnikov. Optimal algorithms for decentralized stochastic variational inequalities. *arXiv preprint arXiv:2202.02771*, 2022.
63. Dmitry Kovalev and Alexander Gasnikov. The first optimal acceleration of high-order methods in smooth convex optimization. *arXiv preprint arXiv:2205.09647*, 2022.
64. Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In Aryeh Kontorovich

and Gergely Neu, editors, *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pages 451–467. PMLR, 08 Feb–11 Feb 2020.

65. Olga Krasotkina and Vadim Mottl. A bayesian approach to sparse learning-to-rank for search engine optimization. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 382–394. Springer, 2015.
66. Guanghui Lan. *First-order and stochastic optimization methods for machine learning*. Springer, 2020.
67. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
68. Yunwen Lei, Zhenhuan Yang, Tianbao Yang, and Yiming Ying. Stability and generalization of stochastic gradient methods for minimax problems. In *International Conference on Machine Learning*, pages 6175–6186. PMLR, 2021.
69. Huan Li, Zhouchen Lin, and Yongchun Fang. Variance reduced extra and diging and their optimal acceleration for strongly convex decentralized optimization, 2020.
70. Shaojie Li and Yong Liu. Improved learning rates for stochastic optimization: Two theoretical viewpoints. *arXiv preprint arXiv:2107.08686*, 2021.
71. Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
72. Junhong Lin, Raffaello Camoriano, and Lorenzo Rosasco. Generalization properties and implicit regularization for multiple passes sgm. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2340–2348, New York, New York, USA, 20–22 Jun 2016. PMLR.
73. Zhouchen Lin, Huan Li, and Cong Fang. Accelerated optimization for machine learning. *Nature Singapore: Springer*, 2020.
74. Dmitriy Metelev, Alexander Rogozin, Alexander Gasnikov, and Dmitry Kovalev. Decentralized saddle-point problems with different constants of strong convexity and strong concavity. *arXiv preprint arXiv:2206.00090*, 2022.
75. Renato DC Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.
76. Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.
77. Fomin Vladimir N. *Recurrent estimation and adaptive filtration*. M.: Nauka (in Russian), 1984.
78. Alexander V. Nazin, Arkadi S. Nemirovsky, Alexandre B. Tsybakov, and Anatoli B. Juditsky. Algorithms of robust stochastic optimization based on mirror descent method. *Automation and Remote Control*, 80(9):1607–1627, 2019.
79. Ion. Necoara, Yu. Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019.
80. Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.
81. Arkadi Semenovich Nemirovski and Yurii Evgenievich Nesterov. Optimal methods of smooth convex minimization. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 25(3):356–369, 1985.
82. A.S. Nemirovski and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983.
83. Yu. Nesterov. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161, 2013.
84. Yu. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404, 2015.

85. Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.
86. Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.
87. Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186(1):157–183, 2021.
88. Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
89. Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.
90. Asuman Ozdaglar, Sarath Pattathil, Jiawei Zhang, and Kaiqing Zhang. What is a good metric to study generalization of minimax learners? *arXiv preprint arXiv:2206.04502*, 2022.
91. Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.
92. Boris Teodorovich Polyak. Introduction to optimization. optimization software. *Inc., Publications Division, New York*, 1, 1987.
93. Boris Teodorovich Polyak. New method of stochastic approximation type. *Automation and remote control*, 51(7 pt 2):937–946, 1990.
94. Boris Teodorovich Polyak and Yakov Zalmanovich Tsypkin. Adaptive estimation algorithms: convergence, optimality, stability. *Avtomatika i telemekhanika*, (3):71–84, 1979.
95. Boris Teodorovich Polyak and Yakov Zalmanovich Tsypkin. Optimal pseudogradient adaptation algorithms. *Avtomatika i Telemekhanika*, (8):74–84, 1980.
96. András Prékopa. *Stochastic programming*, volume 324. Springer Science & Business Media, 2013.
97. Alexander Rakhlin and Karthik Sridharan. Statistical learning and sequential prediction. *e-print,* `http://www.mit.edu/~rakhlin/courses/stat928/stat928_notes.pdf`, 2014.
98. Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.
99. Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
100. R. Tyrrell Rockafellar. *Convex analysis*, volume 18. Princeton university press, 1970.
101. Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart and acceleration. *Advances in Neural Information Processing Systems*, 30, 2017.
102. David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.
103. Abdurakhmon Sadiev, Darina Dvinskikh, Aleksandr Beznosikov, and Alexander Gasnikov. Decentralized and personalized federated learning. *arXiv preprint arXiv:2107.07190*, 2021.
104. Johannes Schneider and Scott Kirkpatrick. *Stochastic optimization*. Springer Science & Business Media, 2007.
105. Ayush Sekhari, Karthik Sridharan, and Satyen Kale. Sgd: The role of implicit regularization, batch-size and multiple-epochs. *Advances in Neural Information Processing Systems*, 34, 2021.
106. Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
107. Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, volume 2, page 5, 2009.
108. Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.
109. Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.
110. Alexander Shapiro and Arkadi Nemirovski. On complexity of stochastic programming problems. In *Continuous optimization*, pages 111–146. Springer, 2005.
111. James C. Spall. Stochastic optimization. In *Handbook of computational statistics*, pages 173–201. Springer, 2012.

112. Vladimir Spokoiny. Parametric estimation. finite sample theory. *The Annals of Statistics*, 40(6):2877–2909, 2012.
113. Vladimir Spokoiny. Bernstein-von mises theorem for growing parameter dimension. *arXiv preprint arXiv:1302.3430*, 2013.
114. Vladimir Spokoiny and Thorsten Dickhaus. *Basics of modern mathematical statistics*. Springer, 2015.
115. Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright. *Optimization for machine learning*. Mit Press, 2012.
116. Karthik Sridharan. Learning from an optimization viewpoint. *arXiv preprint arXiv:1204.4145*, 2012.
117. Karthik Sridharan, S Shalev-Shwartz, and N Srebro. Fast convergence rates for excess regularized risk with application to svm, 2008.
118. Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep learning with PyTorch*. Manning Publications, 2020.
119. Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.
120. Fedor Stonyakin, Alexander Tyurin, Alexander Gasnikov, Pavel Dvurechensky, Artem Agafonov, Darina Dvinskikh, Mohammad Alkousa, Dmitry Pasechnyuk, Sergei Artamonov, and Victorya Piskunova. Inexact model: A framework for optimization and variational inequalities. *Optimization Methods and Software*, pages 1–47, 2021.
121. Adrien B Taylor, Julien M Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1):307–345, 2017.
122. Andrey Nikolaevich Tikhonov and Vasilii Iakovlevich Arsenin. *Solutions of Ill-posed Problems: Andrey N. Tikhonov and Vasiliy Y. Arsenin. Translation Editor Fritz John*. Wiley, 1977.
123. Nuri Mert Vural, Lu Yu, Krishnakumar Balasubramanian, Stanislav Volgushev, and Murat A Erdogdu. Mirror descent strikes again: Optimal stochastic convex optimization under infinite noise variance. *arXiv preprint arXiv:2202.11632*, 2022.
124. Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local SGD better than minibatch SGD? In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10334–10343. PMLR, 13–18 Jul 2020.
125. Blake E. Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pages 4386–4437. PMLR, 2021.
126. Blake E Woodworth and Nathan Srebro. An even more optimal stochastic optimization algorithm: Minibatching and interpolation learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7333–7345. Curran Associates, Inc., 2021.
127. Blake E. Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. *Advances in neural information processing systems*, 29, 2016.
128. Blake E. Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. *Advances in neural information processing systems*, 31, 2018.
129. Stephen J. Wright. Optimization algorithms for data science. *IAS/Park City Math. Ser.–2016.–* `http://www.optimization-online.org/DB_FILE/2016/12/5748.pdf`, 7:816–824, 2016.
130. Stephen J. Wright and Benjamin Recht. *Optimization for data analysis*. Cambridge University Press, 2022.
131. Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22, 2009.
132. Junyu Zhang, Mingyi Hong, Mengdi Wang, and Shuzhong Zhang. Generalization bounds for stochastic saddle point problems. In *International Conference on Artificial Intelligence and Statistics*, pages 568–576. PMLR, 2021.

Glossary

Use the template *glossary.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your glossary in the Springer layout.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

Solutions

Problems of Chapter ??

?? The solution is revealed here.

?? Problem Heading
(a) The solution of first part is revealed here.
(b) The solution of second part is revealed here.

Index