Algorithms for Convex Optimization

In the last few years, algorithms for convex optimization have revolutionized algorithm design, both for discrete and continuous optimization problems. For problems such as maximum flow, maximum matching, and submodular function minimization, the fastest algorithms involve essential methods such as gradient descent, mirror descent, interior point methods, and ellipsoid methods. The goal of this self-contained book is to enable researchers and professionals in computer science, operations research, data science, and machine learning to gain an in-depth understanding of these algorithms. The text emphasizes how to derive key algorithms for convex optimization from first principles and how to establish precise running time bounds. This modern text explains the success of these algorithms in problems of discrete optimization, as well as how these methods have significantly pushed the state of the art of convex optimization itself.

NISHEETH K. VISHNOI is A. Bartlett Giamatti Professor of Computer Science at Yale University. His research areas include theoretical computer science, optimization, and machine learning. He is a recipient of the Best Paper Award at IEEE FOCS in 2005, the IBM Research Pat Goldberg Memorial Award in 2006, the Indian National Science Academy Young Scientist Award in 2011, and the Best Paper Award at ACM FAccT in 2019. He was elected an ACM Fellow in 2019. He obtained a bachelor's degree in computer science and engineering from IIT Bombay and a PhD in algorithms, combinatorics, and optimization from Georgia Institute of Technology.

Algorithms for Convex Optimization

NISHEETH K. VISHNOI
Yale University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of
education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108482028
DOI: 10.1017/9781108699211

First published 2021

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Vishnoi, Nisheeth K., 1976– author.
Title: Algorithms for convex optimization / Nisheeth K. Vishnoi.
Description: New York : Cambridge University Press, [2021] |
Includes bibliographical references and index.
Identifiers: LCCN 2020052071 (print) | LCCN 2020052072 (ebook) |
ISBN 9781108482028 (hardback) | ISBN 9781108741774 (paperback) |
ISBN 9781108699211 (epub)
Subjects: LCSH: Mathematical optimization. | Convex functions. |
Convex programming.
Classification: LCC QA402.5 .V57 2021 (print) | LCC QA402.5 (ebook) |
DDC 515/.882–dc23
LC record available at https://lccn.loc.gov/2020052071
LC ebook record available at https://lccn.loc.gov/2020052072

ISBN 978-1-108-48202-8 Hardback
ISBN 978-1-108-74177-4 Paperback

Cambridge University Press
978-1-108-48202-8 — Algorithms for Convex Optimization
Nisheeth K. Vishnoi
Frontmatter
More Information

Dedicated to Maya and Vayu

Contents

Preface

Convex optimization studies the problem of minimizing a convex function over a convex set. Convexity, along with its numerous implications, has been used to come up with efficient algorithms for many classes of convex programs. Consequently, convex optimization has broadly impacted several disciplines of science and engineering.

In the last few years, algorithms for convex optimization have revolutionized algorithm design, both for discrete and continuous optimization problems. The fastest-known algorithms for problems such as maximum flow in graphs, maximum matching in bipartite graphs, and submodular function minimization involve an essential and nontrivial use of algorithms for convex optimization such as gradient descent, mirror descent, interior point methods, and cutting plane methods. Surprisingly, algorithms for convex optimization have also been used to design counting problems over discrete objects such as matroids. Simultaneously, algorithms for convex optimization have become central to many modern machine learning applications. The demand for algorithms for convex optimization, driven by larger and increasingly complex input instances, has also significantly pushed the state of the art of convex optimization itself.

The goal of this book is to enable a reader to gain an in-depth understanding of algorithms for convex optimization. The emphasis is to derive key algorithms for convex optimization from first principles and to establish precise running time bounds in terms of the input length. Given the broad applicability of these methods, it is not possible for a single book to show the applications of these methods to all of them. This book shows applications to fast algorithms for various discrete optimization and counting problems. The applications selected in this book serve the purpose of illustrating a rather surprising bridge between continuous and discrete optimization.

The structure of the book. The book has roughly four parts. Chapters 3, 4, and 5 provide an introduction to convexity, models of computation and notions of efficiency in convex optimization, and duality. Chapters 6, 7, and 8 introduce first-order methods such as gradient descent, mirror descent and the multiplicative weights update method, and accelerated gradient descent, respectively. Chapters 9, 10, and 11 present Newton's method and various interior point methods for linear programming. Chapters 12 and 13 present cutting plane methods such as the ellipsoid method for linear and general convex programs. Chapter 1 summarizes the book via a brief history of the interplay between continuous and discrete optimization: how the search for fast algorithms for discrete problems is leading to improvements in algorithms for convex optimization.

Many chapters contain applications ranging from finding maximum flows, minimum cuts, and perfect matchings in graphs, to linear optimization over 0-1-polytopes, to submodular function minimization, to computing maximum entropy distributions over combinatorial polytopes.

The book is self-contained and starts with a review of calculus, linear algebra, geometry, dynamical systems, and graph theory in Chapter 2. Exercises posed in this book not only play an important role in checking one's understanding; sometimes important methods and concepts are introduced and developed entirely through them. Examples include the Frank-Wolfe method, coordinate descent, stochastic gradient descent, online convex optimization, the min-max theorem for zero-sum games, the Winnow algorithm for classification, bandit optimization, the conjugate gradient method, primal-dual interior point method, and matrix scaling.

How to use this book. This book can be used either as a textbook for a stand-alone advanced undergraduate or beginning graduate-level course, or as a supplement to an introductory course on convex optimization or algorithm design. The intended audience includes advanced undergraduate students, graduate students, and researchers from theoretical computer science, discrete optimization, operations research, statistics, and machine learning. To make this book accessible to a broad audience with different backgrounds, the writing style deliberately emphasizes the intuition, sometimes at the expense of rigor.

A course for a theoretical computer science or discrete optimization audience could cover the entire book. A course on convex optimization can omit the applications to discrete optimization and can, instead, include applications as per the choice of the instructor. Finally, an introductory course on convex optimization for machine learning could include material from Chapters 2 to 7.

Beyond convex optimization? This book should also prepare the reader for working in areas beyond convex optimization, e.g., nonconvex optimization and geodesic convex optimization, which are currently in their formative years.

Nonconvex optimization. One property of convex functions is that a "local" minimum is also a "global" minimum. Thus, algorithms for convex optimization, essentially, find a local minimum. Interestingly, this viewpoint has led to convex optimization methods being very successful for nonconvex optimization problems, especially those that arise in machine learning. Unlike convex programs, some of which can be **NP**-hard to optimize, most interesting classes of nonconvex optimization problems are **NP**-hard. Hence, in many of these applications, we define a suitable notion of local minimum and look for methods that can take us to one. Thus, algorithms for convex optimization are important for nonconvex optimization as well; see the survey by Jain and Kar (2017).

Geodesic convex optimization. Sometimes, a function that is nonconvex in a Euclidean space turns out to be convex if we introduce a suitable Riemannian metric on the underlying space and redefine convexity with respect to the "straight lines" – geodesics – induced by the metric. Such functions are called geodesically convex and arise in optimization problems over Riemannian manifolds such as matrix Lie groups; see the survey by Vishnoi (2018). The theory of efficient algorithms for geodesic convex optimization is under construction, and the paper by Bürgisser et al. (2019) presents some recent progress.

Acknowledgments

The contents of this book have been developed over several courses – for both undergraduate and graduate students – that I have taught, starting in Fall 2014 and is closest to that of a course taught in Fall 2019 at Yale. I am grateful to all the students and other attendees of these courses for their questions and comments that have made me reflect on the topic and improve the presentation. I am thankful to Slobodan Mitrovic, Damian Straszak, Jakub Tarnawski, and George Zakhour for being some of the first to take this course and scribing my initial lectures on this topic. Special thanks to Damian for scribing a significant fraction of my lectures, sometimes adding his own insights. I am indebted to Somenath Biswas, Elisa Celis, Yan Zhong Ding, and Anay Mehrotra for carefully reading a draft of this book and giving numerous valuable comments and suggestions.

Finally, this book has been influenced by several classic works: *Geometric Algorithms and Combinatorial Optimization* by Grötschel et al. (1988), *Convex Optimization* by Boyd and Vandenberghe (2004), *Introductory Lectures on Convex Optimization* by Nesterov (2014), and *The Multiplicative Weights Update Method: A Meta-algorithm and Applications* by Arora et al. (2012).

Notation

Numbers and sets:

- The set of natural numbers, integers, rationals, and real numbers are denoted by $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$, and $\mathbb{R}$, respectively. $\mathbb{Z}_{\geq 0}, \mathbb{Q}_{\geq 0}$, and $\mathbb{R}_{\geq 0}$ denote the set of nonnegative integers, rationals, and reals, respectively.
- For a positive integer n, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$.
- For a set $S \subseteq [n]$, we use $1_S \in \mathbb{R}^n$ to denote the indicator vector of S defined as $1_S(i) = 1$ for all $i \in S$ and $1_S(i) = 0$ otherwise.
- For a set $S \subseteq [n]$ of cardinality k, we sometimes write $\mathbb{R}^S$ to denote $\mathbb{R}^k$.

Vectors, matrices, inner products, and norms:

- Vectors are denoted by x and y. A vector $x \in \mathbb{R}^n$ is a column vector but is usually written as $x = (x_1, \ldots, x_n)$. The transpose of a vector x is denoted by $x^\top$.
- The standard basis vectors in $\mathbb{R}^n$ are denoted by $e_1, \ldots, e_n$, where e_i is the vector whose ith entry is one and the remaining entries are zero.
- For vectors $x, y \in \mathbb{R}^n$, by $x \geq y$, we mean that $x_i \geq y_i$ for all $i \in [n]$.
- For a vector $x \in \mathbb{R}^n$, we use $\mathrm{Diag}(x)$ to denote the $n \times n$ matrix whose (i,i)th entry is x_i for $1 \leq i \leq n$ and is zero on all other entries.
- When it is clear from context, 0 and 1 are also used to denote vectors with all 0 entries and all 1 entries, respectively.
- For vectors x and y, their inner product is denoted by $\langle x, y \rangle$ or $x^\top y$.
- For a vector x, its ℓ_2 or Euclidean norm is denoted by $\|x\|_2 := \sqrt{\langle x, x \rangle}$. We sometimes also refer to the ℓ_1 or Manhattan distance norm $\|x\|_1 := \sum_{i=1}^n |x_i|$. The ℓ_∞-norm is defined as $\|x\|_\infty := \max_{i=1}^n |x_i|$.
- The outer product of a vector x with itself is denoted by $xx^\top$.
- Matrices are denoted by capitals, e.g., A and L. The transpose of A is denoted by $A^\top$.

- The trace of an $n \times n$ matrix A is $\mathrm{Tr}(A) := \sum_{i=1}^{n} A_{ii}$. The determinant of an $n \times n$ matrix A is $\det(A) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} A_{i\sigma(i)}$. Here S_n is the set of all permutations of n elements and $\mathrm{sgn}(\sigma)$ is the number of transpositions in a permutation σ, i.e., the number of pairs $i < j$ such that $\sigma(i) > \sigma(j)$.

Graphs:

- A graph G has a vertex set V and an edge set E. All graphs are assumed to be undirected unless stated otherwise. If the graph is weighted, there is a weight function $w : E \to \mathbb{R}_{\geq 0}$.
- A graph is said to be simple if there is at most one edge between two vertices and there are no edges whose endpoints are the same vertex.
- Typically, n is reserved for the number of vertices $|V|$ and m for the number of edges $|E|$.

Probability:

- $\mathbb{E}_{\mathcal{D}}[\cdot]$ denotes the expectation and $\Pr_{\mathcal{D}}[\cdot]$ denotes the probability over a distribution $\mathcal{D}$. The subscript is dropped when clear from context.

Running times:

- Standard big-O notation is used to describe the limiting behavior of a function. $\tilde{O}$ denotes that potential poly-logarithmic factors have been omitted, i.e., $f = \tilde{O}(g)$ is equivalent to $f = O(g \log^k(g))$ for some constant k.