Proximal Algorithms

Neal Parikh
Department of Computer Science
Stanford University
npparikh@cs.stanford.edu

Stephen Boyd
Department of Electrical Engineering
Stanford University
boyd@stanford.edu
# Foundations and Trends® in Optimization

## Volume 1, Issue 3, 2014

### Editorial Board

#### Editors-in-Chief

<table>
<thead>
<tr>
<th>Stephen Boyd</th>
<th>Yinyu Ye</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford University</td>
<td>Stanford University</td>
</tr>
<tr>
<td>United States</td>
<td>United States</td>
</tr>
</tbody>
</table>

#### Editors

<table>
<thead>
<tr>
<th>Dimitris Bertsimas</th>
<th>Yuri Nesterov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Massachusetts Institute of Technology</td>
<td>UC Louvain</td>
</tr>
<tr>
<td>John R. Birge</td>
<td>Jorge Nocedal</td>
</tr>
<tr>
<td>University of Chicago</td>
<td>Northwestern University</td>
</tr>
<tr>
<td>Robert E. Bixby</td>
<td>Pablo A. Parrilo</td>
</tr>
<tr>
<td>Rice University</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>Emmanuel Candès</td>
<td>Roman Polyak</td>
</tr>
<tr>
<td>Stanford University</td>
<td>George Mason University</td>
</tr>
<tr>
<td>David Donoho</td>
<td>Tamás Terlaky</td>
</tr>
<tr>
<td>Stanford University</td>
<td>Lehigh University</td>
</tr>
<tr>
<td>Laurent El Ghaoui</td>
<td>Michael J. Todd</td>
</tr>
<tr>
<td>University of California, Berkeley</td>
<td>Cornell University</td>
</tr>
<tr>
<td>Donald Goldfarb</td>
<td>Kim-Chuan Toh</td>
</tr>
<tr>
<td>Columbia University</td>
<td>National University of Singapore</td>
</tr>
<tr>
<td>Michael I. Jordan</td>
<td>John N. Tsitsiklis</td>
</tr>
<tr>
<td>University of California, Berkeley</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>Zhi-Quan (Tom) Luo</td>
<td>Lieven Vandenberghe</td>
</tr>
<tr>
<td>University of Minnesota, Twin Cities</td>
<td>University of California, Los Angeles</td>
</tr>
<tr>
<td>George L. Nemhauser</td>
<td>Robert J. Vanderbei</td>
</tr>
<tr>
<td>Georgia Institute of Technology</td>
<td>Princeton University</td>
</tr>
<tr>
<td>Arkadi Nemirovski</td>
<td>Stephen J. Wright</td>
</tr>
<tr>
<td>Georgia Institute of Technology</td>
<td>University of Wisconsin</td>
</tr>
</tbody>
</table>
Editorial Scope

Topics

Foundations and Trends® in Optimization publishes survey and tutorial articles on methods for and applications of mathematical optimization, including the following topics:

- Algorithm design, analysis, and implementation (especially on modern computing platforms)
- Models and modeling systems
- New optimization formulations for practical problems
- Applications of optimization in:
  - Machine learning
  - Statistics
  - Data analysis
  - Signal and image processing
  - Computational economics and finance
  - Engineering design
  - Scheduling and resource allocation
  - and other areas

Information for Librarians

Foundations and Trends® in Optimization, 2013, Volume XX, 4 issues. ISSN paper version 2167-3888. ISSN online version 2167-3918. Also available as a combined paper and online subscription.
Proximal Algorithms

Neal Parikh
Department of Computer Science
Stanford University
nparikh@cs.stanford.edu

Stephen Boyd
Department of Electrical Engineering
Stanford University
boyd@stanford.edu
## Contents

1 Introduction .......................... 2  
1.1 Definition  .................................. 3  
1.2 Interpretations .............................. 3  
1.3 Proximal algorithms ....................... 5  
1.4 What this paper is about .................... 6  
1.5 Related work ............................... 7  
1.6 Outline ...................................... 7  

2 Properties ................................. 8  
2.1 Separable sum ............................... 8  
2.2 Basic operations ............................. 9  
2.3 Fixed points ................................. 9  
2.4 Proximal average ............................ 12  
2.5 Moreau decomposition ...................... 12  

3 Interpretations ............................. 14  
3.1 Moreau-Yosida regularization ............... 14  
3.2 Resolvent of subdifferential operator .......... 16  
3.3 Modified gradient step ..................... 17  
3.4 Trust region problem ....................... 18  
3.5 Notes and references ...................... 19
4 Proximal Algorithms

4.1 Proximal minimization .................................. 21
4.2 Proximal gradient method ................................ 27
4.3 Accelerated proximal gradient method ................... 31
4.4 Alternating direction method of multipliers ............ 32
4.5 Notes and references .................................... 38

5 Parallel and Distributed Algorithms .......................... 40

5.1 Problem structure ........................................ 40
5.2 Consensus .................................................. 42
5.3 Exchange .................................................... 46
5.4 Allocation ..................................................... 49
5.5 Notes and references ...................................... 50

6 Evaluating Proximal Operators ................................ 51

6.1 Generic methods ........................................... 52
6.2 Polyhedra .................................................... 58
6.3 Cones ......................................................... 62
6.4 Pointwise maximum and supremum ....................... 64
6.5 Norms and norm balls .................................... 66
6.6 Sublevel set and epigraph ................................ 69
6.7 Matrix functions ........................................... 70
6.8 Notes and references ...................................... 74

7 Examples and Applications ................................... 75

7.1 Lasso ......................................................... 75
7.2 Matrix decomposition ..................................... 79
7.3 Multi-period portfolio optimization ....................... 83
7.4 Stochastic optimization .................................. 88
7.5 Robust and risk-averse optimization ..................... 89
7.6 Stochastic control ......................................... 90

8 Conclusions .................................................. 95

References ....................................................... 98
Abstract

This monograph is about a class of optimization algorithms called *proximal algorithms*. Much like Newton’s method is a standard tool for solving unconstrained smooth optimization problems of modest size, proximal algorithms can be viewed as an analogous tool for nonsmooth, constrained, large-scale, or distributed versions of these problems. They are very generally applicable, but are especially well-suited to problems of substantial recent interest involving large or high-dimensional datasets. Proximal methods sit at a higher level of abstraction than classical algorithms like Newton’s method: the base operation is evaluating the *proximal operator* of a function, which itself involves solving a small convex optimization problem. These subproblems, which generalize the problem of projecting a point onto a convex set, often admit closed-form solutions or can be solved very quickly with standard or simple specialized methods. Here, we discuss the many different interpretations of proximal operators and algorithms, describe their connections to many other topics in optimization and applied mathematics, survey some popular algorithms, and provide a large number of examples of proximal operators that commonly arise in practice.


DOI: 10.1561/2400000003.
This monograph is about a class of algorithms, called proximal algorithms, for solving convex optimization problems. Much like Newton’s method is a standard tool for solving unconstrained smooth minimization problems of modest size, proximal algorithms can be viewed as an analogous tool for nonsmooth, constrained, large-scale, or distributed versions of these problems. They are very generally applicable, but they turn out to be especially well-suited to problems of recent and widespread interest involving large or high-dimensional datasets.

Proximal methods sit at a higher level of abstraction than classical optimization algorithms like Newton’s method. In the latter, the base operations are low-level, consisting of linear algebra operations and the computation of gradients and Hessians. In proximal algorithms, the base operation is evaluating the proximal operator of a function, which involves solving a small convex optimization problem. These subproblems can be solved with standard methods, but they often admit closed-form solutions or can be solved very quickly with simple specialized methods. We will also see that proximal operators and proximal algorithms have a number of interesting interpretations and are connected to many different topics in optimization and applied mathematics.
1.1 Definition

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function, which means that its epigraph

$$\text{epi } f = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}$$

is a nonempty closed convex set. The effective domain of $f$ is

$$\text{dom } f = \{x \in \mathbb{R}^n \mid f(x) < +\infty\},$$

i.e., the set of points for which $f$ takes on finite values.

The proximal operator $\text{prox}_f : \mathbb{R}^n \to \mathbb{R}^n$ of $f$ is defined by

$$\text{prox}_f(v) = \arg\min_x \left(f(x) + \frac{1}{2}\|x - v\|^2\right), \quad (1.1)$$

where $\|\cdot\|_2$ is the usual Euclidean norm. The function minimized on the righthand side is strongly convex and not everywhere infinite, so it has a unique minimizer for every $v \in \mathbb{R}^n$ (even when $\text{dom } f \subsetneq \mathbb{R}^n$).

We will often encounter the proximal operator of the scaled function $\lambda f$, where $\lambda > 0$, which can be expressed as

$$\text{prox}_{\lambda f}(v) = \arg\min_x \left(f(x) + \frac{1}{2\lambda}\|x - v\|^2\right). \quad (1.2)$$

This is also called the proximal operator of $f$ with parameter $\lambda$. (To keep notation light, we write $(1/2\lambda)$ rather than $(1/(2\lambda))$.)

Throughout this monograph, when we refer to the proximal operator of a function, the function will be assumed to be closed proper convex, and it may take on the extended value $+\infty$.

1.2 Interpretations

Figure 1.1 depicts what a proximal operator does. The thin black lines are level curves of a convex function $f$; the thicker black line indicates the boundary of its domain. Evaluating $\text{prox}_f$ at the blue points moves them to the corresponding red points. The three points in the domain of the function stay in the domain and move towards the minimum of the function, while the other two move to the boundary of the domain and towards the minimum of the function. The parameter $\lambda$ controls
the extent to which the proximal operator maps points towards the minimum of $f$, with larger values of $\lambda$ associated with mapped points near the minimum, and smaller values giving a smaller movement towards the minimum. It may be useful to keep this figure in mind when reading about the subsequent interpretations.

We now briefly describe some basic interpretations of (1.1) that we will revisit in more detail later. The definition indicates that $\text{prox}_f(v)$ is a point that compromises between minimizing $f$ and being near to $v$. For this reason, $\text{prox}_f(v)$ is sometimes called a *proximal point* of $v$ with respect to $f$. In $\text{prox}_{\lambda f}$, the parameter $\lambda$ can be interpreted as a relative weight or trade-off parameter between these terms.

When $f$ is the *indicator function*

$$I_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C \end{cases},$$
where $\mathcal{C}$ is a closed nonempty convex set, the proximal operator of $f$ reduces to Euclidean projection onto $\mathcal{C}$, which we denote

$$
\Pi_\mathcal{C}(v) = \arg\min_{x \in \mathcal{C}} \|x - v\|_2.
$$

(1.3)

Proximal operators can thus be viewed as generalized projections, and this perspective suggests various properties that we expect proximal operators to obey.

The proximal operator of $f$ can also be interpreted as a kind of gradient step for the function $f$. In particular, we have (under some assumptions described later) that

$$
\text{prox}_{\lambda f}(v) \approx v - \lambda \nabla f(v)
$$

when $\lambda$ is small and $f$ is differentiable. This suggests a close connection between proximal operators and gradient methods, and also hints that the proximal operator may be useful in optimization. It also suggests that $\lambda$ will play a role similar to a step size in a gradient method.

Finally, the fixed points of the proximal operator of $f$ are precisely the minimizers of $f$ (we will show this in §2.3). In other words, $\text{prox}_{\lambda f}(x^*) = x^*$ if and only if $x^*$ minimizes $f$. This implies a close connection between proximal operators and fixed point theory, and suggests that proximal algorithms can be interpreted as solving optimization problems by finding fixed points of appropriate operators.

### 1.3 Proximal algorithms

A **proximal algorithm** is an algorithm for solving a convex optimization problem that uses the proximal operators of the objective terms. For example, the **proximal minimization algorithm**, discussed in more detail in §4.1, minimizes a convex function $f$ by repeatedly applying $\text{prox}_f$ to some initial point $x^0$. The interpretations of $\text{prox}_f$ above suggest several potential perspectives on this algorithm, such as an approximate gradient method or a fixed point iteration. In Chapters 4 and 5 we will encounter less trivial and far more useful proximal algorithms.

Proximal algorithms are most useful when all the relevant proximal operators can be evaluated sufficiently quickly. In Chapter 6, we discuss how to evaluate proximal operators and provide many examples.
There are many reasons to study proximal algorithms. First, they work under extremely general conditions, including cases where the functions are nonsmooth and extended real-valued (so they contain implicit constraints). Second, they can be fast, since there can be simple proximal operators for functions that are otherwise challenging to handle in an optimization problem. Third, they are amenable to distributed optimization, so they can be used to solve very large scale problems. Finally, they are often conceptually and mathematically simple, so they are easy to understand, derive, and implement for a particular problem. Indeed, many proximal algorithms can be interpreted as generalizations of other well-known and widely used algorithms, like the projected gradient method, so they are a natural addition to the basic optimization toolbox for anyone who uses convex optimization.

1.4 What this paper is about

We aim to provide a readable reference on proximal operators and proximal algorithms for a wide audience. There are several novel aspects.

First, we discuss a large number of different perspectives on proximal operators, some of which have not previously appeared in the literature, and many of which have not been collected in one place. These include interpretations based on projection operators, smoothing and regularization, resolvent operators, and differential equations. Second, we place strong emphasis on practical use, so we provide many examples of proximal operators that are efficient to evaluate. Third, we have a more detailed discussion of distributed optimization algorithms than most previous references on proximal operators.

To keep the treatment accessible, we have omitted a few more advanced topics, such as the connection to monotone operator theory.

We also include source code for all examples, as well as a library of implementations of proximal operators, at

http://www.stanford.edu/~boyd/papers/prox_algs.html

We provide links to other libraries of proximal operators, such as those by Becker et al. and Vaiter, in the documentation for our own library.
1.5  Related work

We emphasize that proximal operators are not new and that there have been other surveys written on various aspects of this topic over the years. Lemaire [123] surveys the literature on the proximal point algorithm up to 1989. Iusem [110] reviews the proximal point method and its connection to augmented Lagrangians. An excellent recent reference by Combettes and Pesquet [63] discusses proximal operators and proximal algorithms in the context of signal processing problems. The lecture notes for Vandenberghe’s EE 236C course [196] covers proximal algorithms in detail. Finally, the recent monograph by Boyd et al. [33] is about a particular algorithm (ADMM), but also discusses connections to proximal operators. We will discuss more of the history of proximal operators in the sequel.

1.6  Outline

In Chapter 2, we give some basic properties of proximal operators. In Chapter 3, we discuss a variety of interpretations of proximal operators. Chapter 4 covers some core proximal algorithms for solving convex optimization problems. In Chapter 5, we discuss how to use these algorithms to solve problems in a parallel or distributed fashion. Chapter 6 presents a large number of examples of different projection and proximal operators that can be evaluated efficiently. In Chapter 7, we illustrate these ideas with some examples and applications.
References


References


References


References


References


References


References


References


References


References


References


References


