Journal of Computational Mathematics
Vol.37, No.6, 2019, 778–812.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1906-m2018-0282

PROXIMAL-PROXIMAL-GRADIENT METHOD*

Ernest K. Ryu and Wotao Yin
Department of Mathematics, University of California, Los Angeles, CA 90095, USA
Email: eryu@math.ucla.edu, wotaoyin@math.ucla.edu

Abstract

In this paper, we present the proximal-proximal-gradient method (PPG), a novel optimization method that is simple to implement and simple to parallelize. PPG generalizes the proximal-gradient method and ADMM and is applicable to minimization problems written as a sum of many differentiable and many non-differentiable convex functions. The non-differentiable functions can be coupled. We furthermore present a related stochastic variation, which we call stochastic PPG (S-PPG). S-PPG can be interpreted as a generalization of Finito and MISO over to the sum of many coupled non-differentiable convex functions.

We present many applications that can benefit from PPG and S-PPG and prove convergence for both methods. We demonstrate the empirical effectiveness of both methods through experiments on a CUDA GPU. A key strength of PPG and S-PPG is, compared to existing methods, their ability to directly handle a large sum of non-differentiable non-separable functions with a constant stepsize independent of the number of functions. Such non-diminishing stepsizes allows them to be fast.

Mathematics subject classification: 47N10, 65K05, 90C06, 90C25, 90C30.
Key words: Proximal-gradient, ADMM, Finito, MISO, SAGA, Operator splitting, First-order methods, Distributed optimization, Stochastic optimization, Almost sure convergence, linear convergence.

1. Introduction

In the past decade, first-order methods like the proximal-gradient method and ADMM have enjoyed wide popularity due to their broad applicability, simplicity, and good empirical performance on problems with large data sizes. However, there are many optimization problems such existing simple first-order methods cannot directly handle. Without a simple and scalable method to solve them such optimization problems have been excluded from machine learning and statistical modeling. In this paper we present the proximal-proximal-gradient method (PPG), a novel method that expands the class of problems that one can solve with a simple and scalable first-order method.

Consider the optimization problem

$$\text{minimize} \quad r(x) + \frac{1}{n}\sum_{i=1}^{n}(f_i(x) + g_i(x)), \tag{1.1}$$

where $x \in \mathbb{R}^d$ is the optimization variable, $f_1, \ldots, f_n$, $g_1, \ldots, g_n$, and r are convex, closed, and proper functions from $\mathbb{R}^d$ to $\mathbb{R} \cup \{\infty\}$. Furthermore, assume $f_1, \ldots, f_n$ are differentiable. We

* Received December 18, 2018 / Accepted June 24, 2019 /
Published online July 31, 2019 /

call the method

$$
\begin{aligned}
x^{k+1/2} &= \mathbf{prox}_{\alpha r}\left(\frac{1}{n}\sum_{i=1}^{n} z_i^k\right),\\
x_i^{k+1} &= \mathbf{prox}_{\alpha g_i}\left(2x^{k+1/2} - z_i^k - \alpha\nabla f_i(x^{k+1/2})\right),\\
z_i^{k+1} &= z_i^k + x_i^{k+1} - x^{k+1/2},
\end{aligned}
\tag{PPG}
$$

the *proximal-proximal-gradient method* (PPG). The x_i^{k+1} and z_i^{k+1} updates are performed for all $i = 1, \ldots, n$ and $\alpha > 0$ is a stepsize parameter. To clarify, x, $x_1, \ldots, x_n$ and $z_1, \ldots, z_n$ are all vectors in $\mathbb{R}^d$ (x_i is not a component of x), $x_1^{k+1}, \ldots, x_n^{k+1}$ and $x^{k+1/2}$ approximates the solution to Problem (1.1).

Throughout this paper we write $\mathbf{prox}_h$ for the *proximal operator* with respect to the function h, defined as

$$
\mathbf{prox}_h(x_0) = \operatorname*{argmin}_x \left\{ h(x) + \frac{1}{2}\|x - x_0\|_2^2 \right\}
$$

for a function $h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. When h is the zero function, $\mathbf{prox}_h$ is the identity operator. When h is convex, closed, and proper, the minimizer that defines $\mathbf{prox}_h$ exists and is unique [40]. For many interesting functions h, the proximal operator $\mathbf{prox}_h$ has a closed or semi-closed form solution and is computationally easy to evaluate [12, 45]. We loosely say such functions are *proximable*.

In general, the proximal-gradient method or ADMM cannot directly solve optimization problems expressed in the form of (1.1). When $f_1, \ldots, f_n$ are not proximable, ADMM either doesn't apply or must run another optimization algorithm to evaluate the proximal operators at each iteration. When $n \geq 2$ and $g_1, \ldots, g_n$ are nondifferentiable nonseparable, so $g_1 + \cdots + g_n$ is not proximable (although each individual $g_1, \ldots, g_n$ is proximable). Hence, proximal-gradient doesn't apply.

One possible approach to solving (1.1) is to smooth the non-smooth parts and applying a (stochastic) gradient method. Sometimes, however, keeping non-smooth part is essential. For example, it is the non-smoothness of total variation penalty that induces sharp edges in image processing. In these situations (PPG) is particularly useful as it can handle a large sum of smooth and non-smooth terms directly without smoothing.

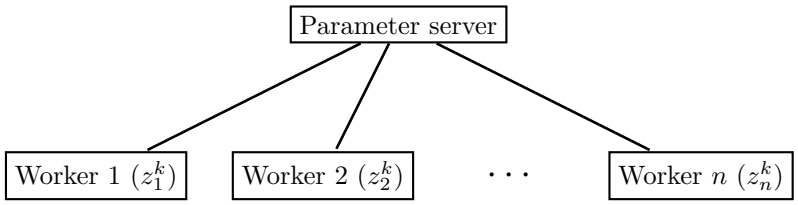

Fig. 1.1. When (PPG) is implemented on a parameter server computing model, the worker nodes communicate (synchronously) with the parameter server but not directly with each other.

Distributed PPG. To understand the algorithmic structure of the method, it is helpful to see how (PPG) is well-suited for a distributed computing network. See Fig. 1.1, which illustrates a parameter server computing model with a master node and n worker nodes.