Proximal-Proximal-Gradient Method

Ernest K. Ryu and Wotao Yin

October 17, 2017

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.

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

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

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

$f_1, \dots, f_n$ are differentiable. We 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 $i = 1, \dots, n$ and $\alpha > 0$ is a stepsize parameter. To clarify, x, $x_1, \dots, x_n$ and $z_1, \dots, z_n$ are all vectors in $\mathbb{R}^d$ (x_i is not a component of x), $x_1^{k+1}, \dots, x_n^{k+1}$ and $x^{k+1/2}$ approximates the solution to Problem (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 [39]. 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 [11, 44]. 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). When $f_1, \dots, f_n$ 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 $g_1, \dots, g_n$ are nondifferentiable nonseparable, so $g_1 + \cdots + g_n$ is not proximable (although each individual $g_1, \dots, g_n$ is proximable). Hence, proximal-gradient doesn't apply.

One possible approach to solving (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.

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 Figure 1, which illustrates a parameter server computing model with a master node and n worker nodes.

At each iteration, the workers send their z_i^k to the parameter server, the parameter server computes $x^{k+1/2}$ and broadcasts it to the workers, and each worker i computes z_i^{k+1} with access to f_i, g_i and z_i^k for $i = 1, \dots, n$. The workers maintain their private copy of z_i^k and do not directly communicate with each other.

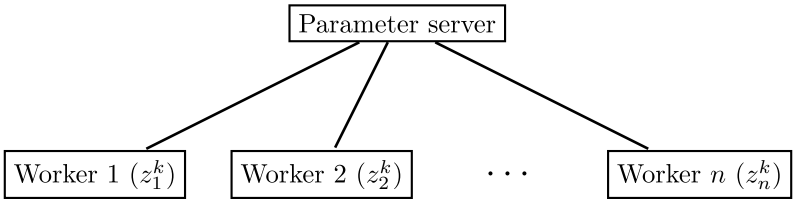

Figure 1: When (PPG) is implemented on a parameter server computing model, the worker nodes communicate (synchronously) with the parameter server but not directly with each other.

In other words, a distributed implementation performs step 1 of (PPG), the $x^{k+1/2}$ update, with an all-reduce operation. It performs step 2 and step 3, the x_i^{k+1} and z_i^{k+1} updates, in parallel.

Time complexity, space complexity, and parallelization. At each iteration, (PPG) evaluates the proximal operators with respect to r and $g_1, \dots, g_n$ and computes the gradients of $f_1, \dots, f_n$. So based on the iteration cost alone, we can predict (PPG) to be useful when r and $g_1, \dots, g_n$ are individually proximable and the gradients of $f_1, \dots, f_n$ are easy to evaluate. If, furthermore, the number of iterations required to reach necessary accuracy is not exorbitant, (PPG) is actually useful.

Let's say the computational costs of evaluating $\mathbf{prox}_{\alpha r}$ is c_r, $\mathbf{prox}_{\alpha g_i}$ is c_g for $i = 1, \dots, n$, and ∇f_i is c_f for $i = 1, \dots, n$. Then the time complexity of (PPG) is $\mathcal{O}(nd+c_r+nc_g+nc_f)$ per iteration (recall $x \in \mathbb{R}^d$). As discussed, this cost can be reduced with parallelization. Computing $x^{k+1/2}$ involves computing an average (a reduce operation), computing $\mathbf{prox}_{\alpha r}$, and a broadcast, and computing z_i^{k+1} for $i = 1, \dots, n$ is embarrassingly parallel. The space complexity of (PPG) is $\mathcal{O}(nd)$ since it must store $z_1^k, \dots z_n^k$. ($x_1^k, \dots, x_n^k$ need not be stored.)

When the problem has sparse structure, the computation time and storage can be further reduced. For example, if $f_i + g_i$ does not depend on $(x_i)_1$ for some i, then $(z_i)_1$ and $(x_i)_1$ can be eliminated from the algorithm since $(z_i)_1^{k+1} = (x^{k+1/2})_1$.

The storage requirement of $O(nd)$ is fundamentally difficult to improve upon due to the n non-differentiable terms. Consider the case where $r = f_1 = \cdots = f_n = 0$:

$$\text{minimize} \quad g_1(x) + \cdots + g_n(x).$$

If $g_1, \dots, g_n$ were differentiable, then $\nabla g_1(x^*) + \cdots + \nabla g_n(x^*) = 0$ would certify x^* is optimal. However, we allow $g_1, \dots, g_n$ to be non-differentiable, so one must find a particular set of subgradients $u_i \in \partial g_i(x^*)$ for $i = 1, \dots, n$ such that $u_1 + \cdots + u_n = 0$ to certify x^* is optimal. The choices of subgradients, $u_1, \dots, u_n$, depend on each other and cannot be found independently. In other words, certifying optimality requires $\mathcal{O}(nd)$ information, and that is what PPG uses. For comparison, ADMM also uses $\mathcal{O}(nd)$ storage when used to minimize a sum of n non-smooth functions.

Stochastic PPG. Each iteration of (PPG) updates z_i^{k+1} for all $i = 1, \dots, n$, which takes at least $\mathcal{O}(nd)$ time per iteration. In Section 3 we present the method (S-PPG) which can be considered a stochastic variation of (PPG). Each iteration of (S-PPG) only updates one z_i^{k+1} for some i and therefore can take as little as $\mathcal{O}(d)$ time per iteration. When compared epoch by epoch, (S-PPG) can be faster than (PPG).

Convergence. Assume Problem (1) has a solution (not necessarily unique) and meets a certain regularity condition. Furthermore, assume each f_i in has L-Lipschitz gradient for $i = 1, \dots, n$, so $\|\nabla f_i(x) - \nabla f_i(y)\|_2 \le L\|x - y\|_2$ for all $x, y \in \mathbb{R}^d$ and $i = 1, \dots, n$. Then (PPG) converges to a solution of Problem (1) for $0 < \alpha < 3/(2L)$. In particular, we do not need strong convexity to establish convegence.

In section 4 we discuss convergence in more detail. In particular, we prove both that the iterates converge to a solution and that the objective values converge to the optimal value.

Contribution of this work. The methods of this paper, (PPG) and (S-PPG), are novel methods that can directly handle a sum of many differentiable and non-differentiable but proximable functions. To solve such problems, exising first-order methods like ADMM must evaluate proximal operators of differentiable functions, which may hinder computational performance if said operator has no closed form solution. Furthermore, the simplicity of our methods allows simple and efficient parallelization, a point we discuss briefly.

The theoretical analysis of (PPG) and (S-PPG), especially that of (S-PPG), is novel. As we discuss later, (S-PPG) can be interpreted as a generalization to varianced reduced gradient methods like Finito/MISO or SAGA. The techniques we use to analyze (S-PPG) is different from those used to analyze other varianced reduced gradient methods, and we show more general (albeit not faster) convergence guarantees. In particular, we establish almost sure convergence of iterates, and we do so without any strong convexity assumptions. To the best of our knowledge, the existing varianced reduced gradient method literature does not prove such results.

Finally, our method is the first work to establish a clear connection between operator splitting methods and varianced reduced gradient methods. As the name implies, existing varianced reduced gradient methods view the method as improving, by reducing variance, stochastic gradient methdos. Our work identifies Finito/MISO as stochastic block-coordinate update applied to an operator splitting method. It is this observation that allows us to analyze (S-PPG), which generalizes Finito/MISO to handle a sum of non-differentiable but proximable functions.

2 Relationship to other methods

In this section, we discuss how (PPG) generalizes certain known methods. To clarify, PPG is a proper generalization of these existing methods and cannot be analyzed as a special case of one of these methods.

Proximal-gradient. When $g_i = 0$ for $i = 1, \ldots, n$ in (1), (PPG) simplifies to

$$x^{k+1} = \mathbf{prox}_{\alpha r} \left(x^k - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(x^k) \right).$$

This method is the called proximal-gradient method or forward-backward splitting [45, 13].

ADMM. When $f = r = 0$ in (1), (PPG) simplifies to

$$\begin{aligned} x_i^{k+1} &= \operatorname*{argmin}_x \left\{ g_i(x) - (y_i^k)^T x + \frac{1}{2\alpha} \|x - x^k\|_2^2 \right\} \\ y_i^{k+1} &= y_i^k + \alpha(\bar{x}^{k+1} - x_i^{k+1}) \end{aligned}$$

where

$$\bar{x}^{k+1} = \frac{1}{n} \sum_{i=1}^{n} x_i^{k+1}.$$

This is also an instance of ADMM [21, 20]. See §7.1 of [8].

Generalized forward-backward splitting. When $r = 0$ and $f_1 = f_2 = \cdots = f_n = f$ in (1), (PPG) simplifies to

$$\begin{aligned} z_i^{k+1} &= z_i^k - x^k + \mathbf{prox}_{\alpha g_i} \left(2x^k - z_i^k - \alpha \nabla f(x^k) \right) \\ x^{k+1} &= \frac{1}{n} \sum_{i=1}^{n} z_i^{k+1}. \end{aligned}$$

This is an instance of generalized forward-backward splitting [46].

Davis-Yin splitting. When $n = 1$ in (1), (PPG) reduces to Davis-Yin splitting, and much of convergence analysis for (PPG) is inspired by that of Davis-Yin splitting [15]. However, (PPG) is more general and parallelizable. Furthermore, (1) has a stochastic variation (S-PPG).

3 Stochastic PPG

Each iteration of (PPG) updates z_i^{k+1} for all $i = 1, \ldots, n$, which requires at least $\mathcal{O}(nd)$ time (without parallelization or sparse structure). Often, the data that specifies Problem (1) is of size $\mathcal{O}(nd)$, and, roughly speaking, (PPG) must process the entire dataset every iteration. This cost of $\mathcal{O}(nd)$ time per iteration may be inefficient in certain applications.

The following method, which we call the *stochastic proximal-proximal-gradient* method (S-PPG), overcomes this issue:

$$
\begin{aligned}
x^{k+1/2} &= \mathbf{prox}_{\alpha r}\left(\frac{1}{n}\sum_{i=1}^{n} z_i^k\right)\\
i(k) &\sim \mathrm{Uniform}(\{1,\dots,n\})\\
x_{i(k)}^{k+1} &= \mathbf{prox}_{\alpha g_{i(k)}}\left(2x^{k+1/2} - z_{i(k)}^k - \alpha\nabla f_{i(k)}(x^{k+1/2})\right)\\
z_j^{k+1} &= \begin{cases} z_{i(k)}^k + x_{i(k)}^{k+1} - x^{k+1/2} & \text{for } j = i(k),\\ z_j^k & \text{for } j \neq i(k).\end{cases}
\end{aligned}
\tag{S-PPG}
$$

At each iteration, only $z_{i(k)}^{k+1}$ is updated, where the index $i(k)$ is chosen uniformly at random from $\{1,\dots,n\}$. We can interpret (S-PPG) as a stochastic or coordinate update version of (PPG).

Time and space complexity. The space requirement of (S-PPG) is no different from that of (PPG); both methods use $\mathcal{O}(nd)$ space to store $z_1^k,\dots,z_n^k$. However, the cost per iteration of (S-PPG) can be as low as $\mathcal{O}(d)$. This is achieved with the following simple trick: maintain the quantity

$$\bar{z}^k = \frac{1}{n}\sum_{i=1}^{n} z_i^k,$$

and update it with

$$\bar{z}^{k+1} = \bar{z}^k + (1/n)(x_{i(k)}^{k+1} - x^{k+1/2}).$$

Application to big-data problems. Consider a big-data problem setup where the data that describes Problem (1) is stored on a hard drive, but is too large to fit in a system's memory. Under this setup, an optimization algorithm that goes through the entire dataset every iteration is likely impractical.

(S-PPG) can handle this setup effectively by keeping the data and the z_i^k iterates on the hard drive as illustrated in Figure 3. At iteration k, (S-PPG) selects index $i(k)$, reads block $i(k)$ containing $f_{i(k)}$, $g_{i(k)}$, and $z_{i(k)}^k$ from the hard drive, performs computation, updates $\bar{z}^{k+1}$, and writes $z_{i(k)}^{k+1}$ back to block $i(k)$. The $\bar{z}^k$ iterate is used and updated every iteration and therefore should be stored in memory.

Interpretation as a variance reduced gradient method. Remarkably (S-PPG) converges to a solution with a fixed value of α (which is independent of n). Many traditional and modern stochastic optimization methods require their step sizes to diminish to 0, theoretically and empirically, and this limits their rates of convergence.

On the other hand, several modern "variance reduced gradient" methods take advantage of a finite sum structure similar to that of (1) and achieve a

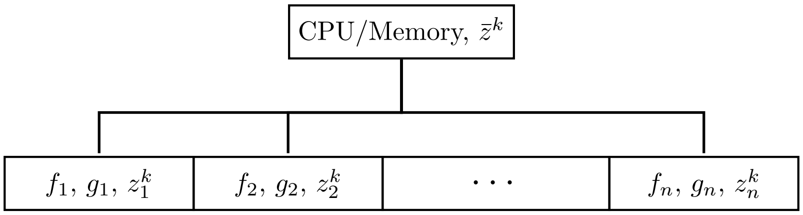

Figure 2: An illustration of (S-PPG) applied to a big-data problem. The bottom blocks represent n blocks of data stored on the hard drive. The CPU accesses only one of the n blocks per iteration.

faster rate with a constant step size. In fact, these methods achieve superior performance compared to full gradient updates. Such methods include Finito, MISO, SVRG, SAG, SAGA, and SDCA [31, 34, 26, 69, 54, 16, 17, 41, 63, 35, 29, 52, 53, 55].

In fact, (S-PPG) directly generalizes Finito and MISO. When $g_1 = \cdots = g_n = 0$ and $r = 0$ in (1), we can rewrite (S-PPG) as

$$\begin{aligned}
w^k &= \frac{1}{n}\sum_{i=1}^{n} z_i^k \\
i(k) &\sim \text{Uniform}(\{1,\ldots,n\}) \\
\phi_{i(k)}^{k+1} &= w^k \\
z_{i(k)}^{k+1} &= \phi_{i(k)}^{k+1} - \alpha \nabla f_{i(k)}(\phi_{i(k)}^{k+1}) \\
z_j^{k+1} &= z_j^k \quad \text{for } j \neq i(k),
\end{aligned}$$

which is Finito and an instance of MISO [34, 17]. Therefore it is appropriate to view (S-PPG) as a variance reduced gradient method as opposed to a stochastic gradient method. Of course, (S-PPG) is more general as it can handle a sum of non-smooth terms as well.

Comparison to SAGA. (S-PPG) is more general than SAGA [16] as it can directly handle a sum of many non-differentiable functions. However, when $g_1 = \cdots = g_n = 0$ in (1), one can use SAGA, and it is interesting to compare (S-PPG) with SAGA under this scenario.

The difference in storage requirement is small. (S-PPG) must store at least nd numbers while SAGA must store at least $(n+1)d$. This is because SAGA stores the current iterate and n gradients, while (S-PPG) only stores $z_1, \ldots, z_n$.

On the other hand, there is a difference in memory access. At each iteration (S-PPG) reads and updates $\bar{z}$ while SAGA reads and updates the current iterate and average gradient. So (S-PPG) reads n fewer numbers and writes n fewer numbers per iteration compared to SAGA. Both SAGA and (S-PPG) read and update information corresponding to a randomly chosen index, and the memory access for this is comparable.

Comparison to stochastic proximal iteration. When $f_1 = \cdots = f_n = 0$ and $r = 0$ in (1), the optimization problem is

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} g_i(x).$$

A stochastic method that can solve this problem is stochastic proximal iteration:

$$\begin{aligned} i(k) &\sim \text{Uniform}(\{1,\dots,n\}) \\ x^{k+1} &= \mathbf{prox}_{\alpha^k g_{i(k)}}(x_k), \end{aligned}$$

where α^k is a appropriately decreasing step size. Stochastic proximal iteration has been studied under many names such as stochastic proximal point, incremental stochastic gradient, and implicit stochastic gradient [30, 28, 38, 5, 60, 49, 61, 6, 51, 62, 59].

Stochastic proximal iteration requires the step size α^k to be diminishing, whereas (S-PPG) converges with a constant step size. As mentioned, optimization methods with diminishing step size tend to have slower rates, which we can observe in the numerical experiments. We experimentally compare (PPG) and (S-PPG) to stochastic proximal iteration in Section 6.

Communication efficient implementation. One way to implement (S-PPG) on a distributed computing network so that communication between nodes are minimized is to have nodes update and randomly pass around the $\overline{z}$ variable. See Figure 3. Each iteration, the current node updates $\overline{z}$ and passes it to another randomly selected node. Every neighbor and the current node is chosen with probability $1/n$.

The communication cost of this implementation of (S-PPG) is $\mathcal{O}(d)$ per iteration. When the number of iterations required for convergence is not large, this method is communication efficient. For recent work on communication efficient optimization methods, see [71, 40, 65, 24, 56, 1, 70].

Convergence. Assume Problem (1) has a solution (not necessarily unique) and meets a certain regularity condition. Furthermore, assume each f_i in has L-Lipschitz continuous gradient for $i = 1,\dots,n$, so $\|\nabla f_i(x) - \nabla f(y)\| \le L\|x-y\|$ for all $x, y \in \mathbb{R}^d$ and $i = 1,\dots,n$. Then (S-PPG) converges to a solution of Problem (1) for $0 < \alpha < 3/(2L)$. In particular, we do not assume strong convexity to establish convergence, whereas many of the mentioned variance reduced gradient methods do.

4 Convergence

In this section, we present and discuss the convergence of (PPG) and (S-PPG).

For this section, we introduce some new notation. We write $\mathbf{x} = (x_1,\dots,x_n)$, and we use other boldface letters like $\boldsymbol{\nu}$ and $\mathbf{z}$ in a similar manner. We use the

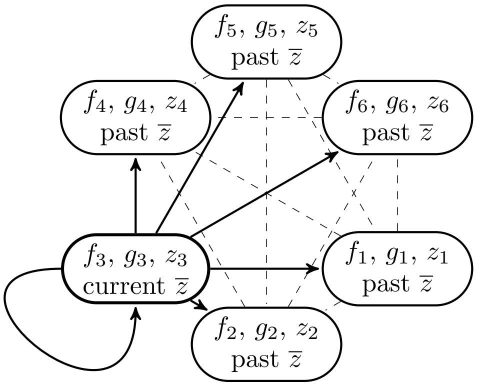

Figure 3: Distributed implementation of (S-PPG) without synchronization. Node 3 has the current copy of $\overline{z}$, and will pass it to another randomly selected node.

bar notation for $\bar{z} = (z_1 + \cdots + z_n)/n$. We write

$$\bar{f}(x) = (f_1(x) + \cdots + f_n(x))/n$$
$$\bar{g}(x) = (g_1(x) + \cdots + g_n(x))/n,$$

and, with some abuse of notation, we write

$$\bar{f}(\mathbf{x}) = (f_1(x_1) + \cdots + f_n(x_n))/n$$
$$\bar{g}(\mathbf{x}) = (g_1(x_1) + \cdots + g_n(x_n))/n.$$

Note that f_i and g_i for depend x_i instead of a common x. The main problem (1) is equivalent to

$$\begin{array}{ll} \text{minimize} & r(x) + \bar{f}(\mathbf{x}) + \bar{g}(\mathbf{x}) \\ \text{subject to} & x - x_i = 0, \quad i = 1, \ldots, n, \end{array} \quad \begin{array}{r} (2) \\ (3) \end{array}$$

where x and $\mathbf{x} = (x_1, \ldots, x_n)$ are the optimization variables. Convex duality tells us that under certain regularity conditions $x^\star$ is a solution of Problem (1) if and only if $(x^\star, \mathbf{x}^\star, \boldsymbol{\nu}^\star)$ is a saddle point of the Lagrangian

$$L(x, \mathbf{x}, \boldsymbol{\nu}) = r(x) + \bar{f}(\mathbf{x}) + \bar{g}(\mathbf{x}) + \frac{1}{n}\sum_{i=1}^{n} \nu_i(x_i - x), \tag{4}$$

where $\boldsymbol{\nu}^\star = (\nu_1^\star, \ldots, \nu_n^\star)$ is a dual solution, and $\mathbf{x}^\star = (x^\star, \ldots, x^\star)$. We simply assume the Lagrangian (4) has a saddle point. This is not a stringent requirement and is merely assumed to avoid pathologies.

Define the mapping $p(\mathbf{z})$ as

$$\begin{aligned} (p(\mathbf{z}))_i &= (1/\alpha)(x - x_i') \quad \text{for } i = 1, \dots, n, \\ \text{where} \quad x &= \mathbf{prox}_{\alpha r}(\bar{z}), \\ x_i' &= \mathbf{prox}_{\alpha g_i}\left(2x - z_i - \alpha \nabla f_i(x)\right). \end{aligned} \tag{5}$$

With this notation, we can express (PPG) as

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \alpha p(\mathbf{z}^k),$$

and we can say $\mathbf{z}$ is a fixed point of (PPG) and (S-PPG) if and only if $p(\mathbf{z}) = 0$.

Lemma 1. *$\mathbf{z}^\star = (z_1^\star, \dots, z_n^\star)$ is a fixed point of* (PPG) *and* (S-PPG) *if and only if $z_i^\star = x^\star + \alpha \nu_i^\star$ for $i = 1, \dots, n$, where $x^\star$ and $\nu_1^\star, \dots, \nu_n^\star$ are primal and dual solutions. In particular, we can recover the $x^\star$ as $x^\star = \mathbf{prox}_{\alpha r}((1/n)(z_1^\star + \cdots + z_n^\star))$.*

This lemma provides us insight as to why the method converges. Let's write $z_i^k = x_i^k + \alpha \nu_i^k$. Then the updates can be written as

$$x^{k+1/2} = \underset{x}{\operatorname{argmin}} \left\{ r(x) + (\bar{\nu}_i^k)^T(\bar{x}_i^k - x) + \frac{1}{2\alpha}\|x - \bar{x}_i^k\|_2^2 \right\}$$

and

$$\begin{aligned} x_i^{k+1} = \underset{x}{\operatorname{argmin}} \Big\{ & f(x^{k+1/2}) + (\nabla f(x^{k+1/2}))^T(x - x^{k+1/2}) \\ & + g(x) + (x - 2x^{k+1/2} + x_i^k)^T \nu_i^k + \frac{1}{2\alpha}\|x - 2x^{k+1/2} + x_i^k\|_2^2 \Big\}. \end{aligned}$$

Since $(x^\star, \mathbf{x}^\star, \boldsymbol{\nu}^\star)$ is a saddle point of the Lagrangian, we can see that $z_i^\star = x^\star + \alpha \nu_i^\star$ is a fixed point of (PPG).

4.1 Deterministic analysis.

We examine the convergence results for (PPG).

Theorem 1. *Assume $f_1, \dots, f_n$ are differentiable and have L-Lipschitz continuous gradients. Assume the Lagrangian* (4) *has a saddle point and $0 < \alpha < 3/(2L)$. Then the sequence $\|p(\mathbf{z}^k)\|_2 \to 0$ monotonically with rate*

$$\|p(\mathbf{z}^k)\|_2 \leq \mathcal{O}(1/\sqrt{k}).$$

Furthermore $\mathbf{z}^k \to \mathbf{z}^\star$, $x^{k+1/2} \to x^\star$, and $x_i^{k+1} \to x^\star$ for all $i = 1, \dots, n$, where $\mathbf{z}^\star$ is a fixed point of (PPG) *and $x^\star$ is a solution of* (1).

Theorem 1 should be understood as two related but separate results. The first result states $p(\mathbf{z}^k) \to 0$ and provides a rate. Since $p(\mathbf{z}) = 0$ implies $\mathbf{prox}_{\alpha r}(\bar{z})$ is a solution, the rate does quantify progress. The second result states that the iterates of (PPG) converge but with no guarantee of rate (just like gradient descent without strong convexity).

To obtain a more direct measure of progress, define

$$E^k = r(x^{k+1/2}) + \bar{f}(x^{k+1/2}) + \bar{g}(\mathbf{x}^{k+1}) - \big(r(x^\star) + \bar{f}(x^\star) + \bar{g}(x^\star)\big).$$

E^k is almost like the suboptimality of iterates, but not quite, as the point where $\bar{g}$ is evaluated at is different from the point where r and $\bar{f}$ is evaluated at. In fact, E^k is not necessarily positive. Nevertheless, we can show a rate on $|E^k|$.

Theorem 2. *Under the setting of Theorem 1,*

$$|E^k| \le \mathcal{O}(1/\sqrt{k}).$$

Define a similar quantity

$$e^k = \big(r(x^{k+1/2}) + \bar{f}(x^{k+1/2}) + \bar{g}(x^{k+1/2})\big) - \big(r(x^\star) + \bar{g}(x^\star) + \bar{f}(x^\star)\big).$$

While e^k truly measures suboptimality of $x^{k+1/2}$, it is possible for $e^k = \infty$ for all $k = 1, 2, \ldots$ because r and g are possibly nonsmooth and valued ∞ at some points. We need an additional assumption for e^k to be a meaningful quantity.

Corollary 1. *Assume the setting of Theorem 1. Further assume $\bar{g}(x)$ is Lipschitz continuous with parameter L_g. Then*

$$0 \le e^k \le |E^k| + L_g \|p(\mathbf{z}^k)\|$$

and

$$e^k = \mathcal{O}(1/\sqrt{k}).$$

The proof of Corollary 1 follows immediately from combining Theorems 1 and 2 with e^k's and L_g's definitions.

The $1/\sqrt{k}$ rates for Theorem 2 and Corollary 1 can be improved to the $1/k$ rates by using the *ergodic iterates*:

$$x_{\text{erg}}^{k+1/2} = \frac{1}{k}\sum_{j=1}^{k} x^{j+1/2}, \quad \mathbf{x}_{\text{erg}}^{k+1} = \frac{1}{k}\sum_{j=1}^{k} \mathbf{x}^{j+1}.$$

With these ergodic iterates, we define

$$\begin{aligned} E_{\text{erg}}^k &= \big(r(x_{\text{erg}}^{k+1/2}) + \bar{f}(x_{\text{erg}}^{k+1/2}) + \bar{g}(\mathbf{x}_{\text{erg}}^{k+1})\big) - \big(r(x^\star) + \bar{g}(\mathbf{x}^\star) + \bar{f}(x^\star)\big), \\ e_{\text{erg}}^k &= \big(r(x_{\text{erg}}^{k+1/2}) + \bar{f}(x_{\text{erg}}^{k+1/2}) + \bar{g}(x_{\text{erg}}^{k+1/2})\big) - \big(r(x^\star) + \bar{g}(\mathbf{x}^\star) + \bar{f}(x^\star)\big). \end{aligned}$$

Theorem 3. *Assume the setting of Theorem 1. Then*

$$|E_{\text{erg}}^k| \le \mathcal{O}(1/k)$$

Further assume $\bar{g}(x)$ is Lipschitz continuous with parameter L_g. Then

$$e_{\text{erg}}^k \le \mathcal{O}(1/k).$$

Finally, under rather strong conditions on the problems, linear convergence can also be shown.

Theorem 4. *Assume the setting of Theorem 1. Furthermore, assume $\bar{g}$ is differentiable with Lipschitz continuous gradient. If one (or more) of r, $\bar{g}$, or $\bar{f}$ is strongly convex, then* (PPG) *converges linearly in the sense that*

$$\|\mathbf{z}^k - \mathbf{z}^\star\|_2^2 \leq \mathcal{O}(e^{-Ck})$$

for some $C > 0$. Consequently, $|E^k|$ and e^k also converge linearly.

4.2 Stochastic analysis.

As it turns out, the condition that guarantees (S-PPG) converges is the same as that of (PPG). In particular, there is not step size reduction!

Theorem 5. *Apply the same assumptions in Theorem 1. That is, assume $f_1, \dots, f_n$ are differentiable and have L-Lipschitz continuous gradients, and assume the Lagrangian* (4) *has a saddle point and $0 < \alpha < 3/(2L)$. Then the sequence $\|p(\mathbf{z}^k)\|_2 \to 0$ with probability one at the rate*

$$\min_{i=0,\dots,k} \mathbb{E}\|p(\mathbf{z}^i)\|_2^2 \leq \mathcal{O}(1/k).$$

Furthermore $\mathbf{z}^k \to \mathbf{z}^\star$, $x^{k+1/2} \to x^\star$, and $x_i^{k+1} \to x^\star$ for all $i = 1, \dots, n$ with probability one.

The expected objective rates of (S-PPG) also match those of (PPG).

Theorem 6. *Under the same setting of Theorem 5, we have*

$$\mathbb{E}|E^k| \leq \mathcal{O}(1/\sqrt{k}) \quad \text{and} \quad |E^k_{\text{erg}}| \leq \mathcal{O}(1/k). \tag{6}$$

Further assume $\bar{g}(x)$ is Lipschitz continuous with parameter L_g. Then

$$\mathbb{E}e^k_{\text{erg}} \leq \mathcal{O}(1/k).$$

Due to space limitation, we state without proof that, under the setting of Theorem 4, (S-PPG) yields linearly convergent $\mathbb{E}\|\mathbf{z}^k - \mathbf{z}^\star\|_2$, $\mathbb{E}|E^k|$, and $\mathbb{E}|e^k|$.

5 Applications of PPG

To utilize (PPG), a given optimization problem often needs to be recast into the form of (1). In this section, we show some techniques for this while presenting some interesting applications.

All examples presented in this section are naturally posed as

$$\text{minimize} \quad r(x) + \frac{1}{n}\sum_{i=1}^n f_i(x) + \frac{1}{m}\sum_{j=1}^m g_j(x), \tag{7}$$

where $n \neq m$. There is more than one way to recast Problem (7) into the form of Problem (1).

Among these options, the most symmetric one, loosely speaking, is

$$\text{minimize} \quad r(x) + \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \big(f_i(x) + g_j(x)\big),$$

which leads to the method

$$\begin{aligned} x^{k+1/2} &= \mathbf{prox}_{\alpha r} \left(\frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij}^k \right) \\ x_{ij}^{k+1} &= \mathbf{prox}_{\alpha g_j} \left(2x^{k+1/2} - z_{ij}^k - \alpha \nabla f_i(x^{k+1/2}) \right) \\ z_{ij}^{k+1} &= z_{ij}^k + x_{ij}^{k+1} - x^{k+1/2}. \end{aligned}$$

In general, the product mn can be quite large, and if so, this approach is likely impractical. In many examples, however, mn is not large as since $n = 1$ or m is small.

Another option, feasible when neither n nor m is small, is

$$\text{minimize} \quad r(x) + \frac{1}{m+n} \left(\sum_{i=1}^{n} ((m+n)/n) f_i(x) + \sum_{j=1}^{m} ((m+n)/m) g_j(x) \right),$$

which leads to the method

$$\begin{aligned} x^{k+1/2} &= \mathbf{prox}_{\alpha r} \left(\frac{1}{n+m} \left(\sum_{i=1}^{n} y_i^k + \sum_{j=1}^{m} z_j^k \right) \right) \\ y_i^{k+1} &= x^{k+1/2} - \alpha((m+n)/n) \nabla f_i(x^{k+1/2}) \\ z_j^{k+1} &= z_j^k - x^{k+1/2} + \mathbf{prox}_{\alpha((m+n)/m) g_j} \left(2x^{k+1/2} - z_j^k \right). \end{aligned}$$

Overlapping group lasso. Let $\mathcal{G}$ be a collection of groups of indices. So $G \subseteq \{1, 2, \ldots, d\}$ for each $G \in \mathcal{G}$. The groups can be overlapping, i.e., $G_1 \cap G_2 \neq \emptyset$ is possible for $G_1, G_2 \in \mathcal{G}$ and $G_1 \neq G_2$.

We let $x_G \in \mathbb{R}^{|G|}$ denote a subvector corresponding to the indices of $G \in \mathcal{G}$, where $x \in \mathbb{R}^d$ is the whole vector. So the entries x_i for $i \in G$ form the vector x_G.

The overlapping group lasso problem is

$$\text{minimize} \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda_1 \sum_{G \in \mathcal{G}} \|x_G\|_2,$$

where $x \in \mathbb{R}^d$ is the optimization variable, $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$ are problem data, and $\lambda_1 > 0$ is a regularization parameter. As it is, the regularizer (the second term) is not proximable when the groups overlap.

Partition the collection of groups $\mathcal{G}$ into n non-overlapping collections. So $\mathcal{G}$ is a disjoint union of $\mathcal{G}_1, \dots, \mathcal{G}_n$, and if $G_1, G_2 \in \mathcal{G}_i$ and $G_1 \neq G_2$ then $G_1 \cap G_2 = \emptyset$ for $i = 1, \dots, n$. With some abuse of notation, we write

$$\mathcal{G}_i^{\mathsf{c}} = \{i \in \{1, \dots, d\} \mid i \notin G \text{ for all } G \in \mathcal{G}_i\}$$

Now we recast the problem into the form of (1)

$$\text{minimize} \quad \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{G \in \mathcal{G}_i} \lambda_2 \|x_G\|_2\right),$$

where $\lambda_2 = n\lambda_1$. The regularizer (the second term) is now a sum of n proximable terms.

For example, we can have a setup with $d = 42$, $n = 3$, $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$, and

$$\begin{aligned}
\mathcal{G}_1 &= \{\{1, \dots, 9\}, \{10, \dots, 18\}, \{19, \dots, 27\}, \{28, \dots, 36\}\} \\
\mathcal{G}_2 &= \{\{4, \dots, 12\}, \{13, \dots, 21\}, \{22, \dots, 30\}, \{31, \dots, 39\}\} \\
\mathcal{G}_3 &= \{\{7, \dots, 15\}, \{16, \dots, 24\}, \{25, \dots, 33\}, \{34, \dots, 42\}\}
\end{aligned}$$

The groups within $\mathcal{G}_i$ do not overlap for each $i = 1, 2, 3$, and $\mathcal{G}_2^{\mathsf{c}} = \{1, 2, 3, 40, 41, 42\}$.

We view the first term as the r term, the second term as the sum of $g_1, \dots, g_n$, and $f_1 = \cdots = f_n = 0$ in the notation of (1), and apply (PPG):

$$\begin{aligned}
x^{k+1/2} &= (I + \alpha A^T A)^{-1}\left(\alpha A^T b + \frac{1}{n}\sum_{i=1}^{n} z_i^k\right) \\
z_i^{k+1/2} &= 2x^{k+1/2} - z_i^k \\
(x_i^{k+1})_G &= u_{\alpha\lambda_2}\left((z_i^{k+1/2})_G\right) \quad \text{for } G \in \mathcal{G}_i \\
(x_i^{k+1})_j &= (z_i^{k+1/2})_j \quad \text{for } j \in \mathcal{G}_i^{\mathsf{c}} \\
z_i^{k+1} &= z_i^k + x_i^{k+1} - x^{k+1/2},
\end{aligned}$$

where the indices i implicitly run through $i = 1, \dots, n$, and u_α, defined in the Section 7, is the vector soft-thresholding operator.

To reduce the cost of computing $x^{k+1/2}$, we can precompute and store the Cholesky factorization of the positive definite matrix $I + \alpha A^T A$ (which costs $\mathcal{O}(m^2 d + d^3)$) and the matrix-vector product $A^T b$ (which costs $\mathcal{O}(md)$). This cost is paid upfront once, and the subsequent iterations can be done in $\mathcal{O}(d^2 + dn)$ time.

For recent work on overlapping group lasso, see [68, 72, 27, 36, 32, 67, 25, 10, 7, 64].

Low-rank and sparse matrix completion. Consider the setup where we partially observe a matrix $M \in \mathbb{R}^{d_1 \times d_2}$ on the set of indices Ω. More precisely, we observe M_{ij} for $(i, j) \in \Omega$ while M_{ij} for $(i, j) \notin \Omega$ are unknown. We assume M has a low-rank plus sparse structure, i.e., $M = L^{\text{true}} + S^{\text{true}}$ with L^{true} is low-rank and S^{true} is sparse. Here L^{true} models the true underlying structure while S^{true} models outliers. Furthermore, let's assume $0 \le M_{ij} \le 1$ for all (i,j). The goal is to estimate the unobserved entries of M, i.e., M_{ij} for $(i,j) \notin \Omega$.

To estimate M, we solve the following regularized regression

$$\begin{array}{ll} \text{minimize} & \lambda_1 \|L\|_* + \lambda_2 \|S\|_1 + \displaystyle\sum_{(i,j)\in\Omega} \ell(S_{ij} + L_{ij} - M_{ij}) \\ \text{subject to} & 0 \le S + L \le 1, \end{array}$$

where $S, L \in \mathbb{R}^{d_1 \times d_2}$ are the optimization variables, the constraint $0 \le S + L \le 1$ applies element-wise, and $\lambda_1, \lambda_2 > 0$ are regularization parameters. The constraint is proximable by Lemma 3, and we can use (PPG) either when ℓ is differentiable or when $\ell + I_{[0,1]}$ is proximable.

We view $n = 1$, the first term as the r term, the second term as the g_1 term, and the last term as the f_1 term in the notation of (1), and apply (PPG):

$$\begin{aligned}
L^{k+1/2} &= t_\alpha\left(Z^k\right) \\
S_{ij}^{k+1/2} &= s_\alpha\left(Y_{ij}^k\right) \qquad \text{for all } (i,j) \\
Z_{ij}^{k+1/2} &= \begin{cases} 2L_{ij}^{k+1/2} - Z_{ij}^k & \text{for } (i,j) \notin \Omega \\ 2L_{ij}^{k+1/2} - Z_{ij}^k - \alpha(L_{ij}^{k+1/2} + S_{ij}^{k+1/2} - M_{ij}) & \text{for } (i,j) \in \Omega \end{cases} \\
Y_{ij}^{k+1/2} &= \begin{cases} 2S_{ij}^{k+1/2} - Y_{ij}^k & \text{for } (i,j) \notin \Omega \\ 2S_{ij}^{k+1/2} - Y_{ij}^k - \alpha(L_{ij}^{k+1/2} + S_{ij}^{k+1/2} - M_{ij}) & \text{for } (i,j) \in \Omega \end{cases} \\
A^{k+1} &= Z^{k+1/2} + Y^{k+1/2} \\
B^{k+1} &= Z^{k+1/2} - Y^{k+1/2} \\
L_{ij}^{k+1} &= \frac{1}{2}\left(\Pi_{[0,1]}(A_{ij}^{k+1}) + B_{ij}^{k+1}\right) \qquad \text{for all } (i,j) \\
S_{ij}^{k+1} &= \frac{1}{2}\left(\Pi_{[0,1]}(A_{ij}^{k+1}) - B_{ij}^{k+1}\right) \qquad \text{for all } (i,j) \\
Z^{k+1} &= Z^k + L^{k+1} - L^{k+1/2} \\
Y^{k+1} &= Y^k + S^{k+1} - S^{k+1/2}.
\end{aligned}$$

t_α and s_α, defined in the Section 7, are respectively the matrix and scalar soft-thresholding operators. $\Pi_{[0,1]}$ is the projection onto the interval $[0,1]$, and the L^{k+1} and S^{k+1} updates follow from Lemma 3. The only non-trivial operation for this method is computing the SVD to evaluate $t_\alpha\left(Z^k\right)$. All other operations are elementary and embarrassingly parallel.

For a discussion on low-rank + sparse factorization, see [9].

Regression with fused lasso. Consider the problem setup where we have $Ax^{\text{true}} = b$ and we observe A and b. Furthermore, the coordinates of x^{true} are ordered in a meaningful way and we know a priori that $|x_{i+1} - x_i| \le \varepsilon$ for $i = 1, \ldots, d-1$ and some $\varepsilon > 0$. finally, we also know that x^{true} is sparse.

To estimate x, we solve the fused lasso problem

$$\begin{array}{ll} \text{minimize} & \lambda\|x\|_1 + \dfrac{1}{n}\displaystyle\sum_{i=1}^n \ell_i(x) \\ \text{subject to} & |x_{i+1} - x_i| \le \varepsilon, \quad i = 1, \dots, d-1 \end{array}$$

where

$$\ell_i(x) = (1/2)(a_i^T x - y_i)^2$$

and $x \in \mathbb{R}^d$ is the optimization variable.

We recast the problem into the form of (1)

$$\begin{array}{ll} \text{minimize} & \lambda\|x\|_1 \\ & + \dfrac{1}{2n}\left(\displaystyle\sum_{i=1}^n (\ell_i(x) + g_o(x)) + \sum_{i=1}^n (\ell_i(x) + g_e(x))\right) \end{array}$$

where

$$\begin{aligned} g_o(x) &= \sum_{i=1,3,5,\dots} I_{[-\varepsilon,\varepsilon]}(x_{i+1} - x_i) \\ g_e(x) &= \sum_{i=2,4,6,\dots} I_{[-\varepsilon,\varepsilon]}(x_{i+1} - x_i) \end{aligned}$$

and

$$I_{[-\varepsilon,\varepsilon]}(x) = \begin{cases} 0 & \text{if } |x| \le \varepsilon \\ \infty & \text{otherwise.} \end{cases}$$

Since g_o and g_e are proximable by Lemma 3, we can apply (PPG).

For recent work on fused lasso, see [57, 47, 58, 23, 33, 42, 66, 73].

Network lasso. Consider the problem setup where we have an undirected graph $G = (E, V)$. Each node $v \in V$ has a parameter to estimate $x_v \in \mathbb{R}^d$ and an associated loss function ℓ_v. Furthermore, we know that neighbors of G have similar parameters in the sense that $\|x_u - x_v\|_2$ is small if $\{u, v\} \in E$ and that x_v is sparse for each $v \in V$.

Under this model, we solve the network lasso problem

$$\text{minimize} \quad \sum_{v \in V} \lambda_1 \|x_v\|_1 + \ell_v(x_v) + \sum_{\{u,v\} \in E} \lambda_2 \|x_u - x_v\|_2,$$

where x_v for all $v \in V$ are the optimization variables, and $\ell_v(x_v)$ for all $v \in V$ a differentiable loss function, and $\lambda_1, \lambda_2 > 0$ are regularization parameters [22].

Say the G has an edge coloring $E_1, \dots, E_C$. So $E_1, \dots, E_C$ partitions E such that if $\{u, v\} \in E_c$ then $\{u, v'\} \notin E_c$ for any $v' \neq v$ and $c = 1, \dots, C$. Figure 4 illustrates this definition. (The chromatic index $\chi'(G)$ is the smallest possible

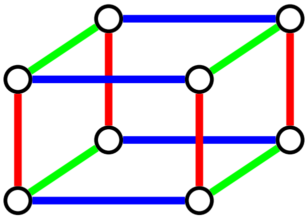

Figure 4: An edge coloring of the hypercube graph Q_3.

value of C, but C need not be $\chi'(G)$.) With the edge coloring, we recast the problem into the form of (1)

$$\begin{aligned}\text{minimize} \quad & \sum_{v\in V} \lambda_1 \|x_v\|_1 \\ & + \frac{1}{C}\sum_{c=1}^{C}\left(\sum_{v\in V}\ell_v(x_v) + \sum_{\{u,v\}\in E_c} \lambda_3 \|x_u - x_v\|_2\right),\end{aligned}$$

where $\lambda_3 = C\lambda_2$.

We view the ℓ_1 regularizer as the r term, the loss functions as the f term, and the summation over E_c as the g terms in the notation of (1), and apply (PPG):

$$\begin{aligned}
x_v^{k+1/2} &= s_{\alpha\lambda_1}\left(\frac{1}{C}\sum_{c=1}^{C} z_{cv}^k\right)\\
z_{cv}^{k+1/2} &= 2x_v^{k+1/2} - z_{cv}^k - \alpha\nabla\ell_v(x_v^{k+1/2})\\
s_c^k &= z_{cu}^{k+1/2} + z_{cv}^{k+1/2}\\
d_c^k &= z_{cu}^{k+1/2} - z_{cv}^{k+1/2}\\
x_{cu}^{k+1} &= s_c^k + u_{\alpha\lambda_3}\left(d_c^k\right) \quad \text{for } \{u,v\}\in E_c\\
x_{cv}^{k+1} &= s_c^k - u_{\alpha\lambda_3}\left(d_c^k\right) \quad \text{for } \{u,v\}\in E_c\\
x_{cv}^{k+1} &= z_{cv}^{k+1/2} \quad \text{for } \{v,u'\}\notin E_c \text{ for all } u'\in V\\
z_{cv}^{k+1} &= z_{cv}^k + x_{cv}^{k+1} - x_v^{k+1/2},
\end{aligned}$$

where the colors c implicitly run through $c = 1, \ldots, C$ unless specified otherwise and the nodes v implicitly run through all $v \in V$. Here s_α and u_α, defined in the Section 7, are respectively the scalar and vector soft-thresholding operators.

Although this algorithm, as stated, seemingly maintains C copies of x_v we can actually simplify it so that v maintains $\min\{\deg(v)+1, C\}$ copies of x_v for all $v \in V$. Since $2|E|/|V|$ is the average degree of G, storage requirement is $\mathcal{O}(|V| + |E|)$ when simplified.

Let each node have a set $N \subseteq \{1, \ldots, C\}$ such that $c \in N$ if there is a neighbor connected through an edge with color c. Write $N^{\mathsf{c}} = \{1, \ldots, C\}\backslash N$.

With this notation, we can rewrite the algorithm in a simpler, vertex-centric manner:

```
FOR EACH node
  $x^{1/2} = s_{\alpha\lambda_1}\left(\frac{1}{C}\left(|N^{\mathsf{c}}|z_{c'} + \sum_{c\in N} z_c^k\right)\right)$
  FOR EACH color $c \in S$
    $z_c^{1/2} = 2x^{1/2} - z_c - \alpha\nabla f(x^{1/2})$
    Through edge with color $c$, send $z_c^{1/2}$ and receive $z_c'^{1/2}$
    $s = z_c^{1/2} + z_c'^{1/2}$
    $d = z_c^{1/2} - z_c'^{1/2}$
    $x_c = (1/2)(s + u_{\alpha\lambda_2}(d))$
    $z_c = z_c + x_c - x^{1/2}$,
  IF $N^{\mathsf{c}} \neq \emptyset$
    $z_{c'} = x^{1/2} - \alpha\nabla f(x^{1/2})$
```

SVM. We solve the standard (primal) support vector machine setup [14]

$$\text{minimize} \quad \frac{\lambda}{2}\|x\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} g_i(x), \tag{8}$$

where $x \in \mathbb{R}^d$ is the optimization variable. The problem data is embedded in

$$g_i(x) = \max\{1 - y_i a_i^T x, 0\}$$

where $\lambda > 0$ is a regularization parameter and $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ and $y_i \in \{-1, +1\}$ are problem data for $i = 1, \dots, n$. Applying Lemma 2 and working out the details, we get a closed-form solution for the proximal operator:

$$\mathbf{prox}_{\alpha g_i}(x_0) = x_0 + \Pi_{[0,\alpha]}\left(\frac{1 - y_i a_i^T x_0}{\|a_i\|_2^2}\right) y_i a_i$$

for $i = 1, \dots, n$.

We view $r = (\lambda/2)\|x\|_2^2$ and $f = 0$ in the notation of (1), and apply (PPG):

$$\begin{aligned} x^{k+1/2} &= \frac{1}{1+\alpha\lambda}\frac{1}{n}\sum_{i=1}^{n} z_i^k \\ \beta_i^k &= y_i \Pi_{[0,\alpha]}\left(\frac{1 - y_i a_i^T(2x^{k+1/2} - z_i^k)}{\|a_i\|_2^2}\right) \\ z_i^{k+1} &= x^{k+1/2} + \beta_i a_i, \end{aligned}$$

where the indices i implicitly run through $i = 1, \dots, n$.

Generalized linear model. In the setting of generalized linear models, the maximum likelihood estimator is the solution of the optimization problem

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}\big(A(x_i^T\beta) - T_i x_i^T\beta\big),$$

where $\beta \in \mathbb{R}^d$ is the optimization variable, $x_i \in \mathbb{R}^d$ and $T_i \in \mathbb{R}$ for $i = 1, \ldots, n$ are problem data, and A is a convex function on $\mathbb{R}$.

We view $r = 0$, $f = 0$, and $g_i(\beta) = A(x_i^T\beta) - T_i x_i^T\beta$ in the notation of (1), and apply (PPG):

$$\begin{aligned}
\beta^{k+1} &= \frac{1}{n}\sum_{i=1}^{n} z_i^k \\
z_i^{k+1} &= z_i^k + \mathbf{prox}_{\alpha g_i}(2\beta^k - z_i^k) - \beta^k
\end{aligned}$$

where the indices i implicitly run from $i = 1, \ldots, n$.

For an introduction on generalized linear models, see [37].

Network Utility Maximization. In the problem of network utility maximization, one solves the optimization problem

$$\begin{array}{ll}
\text{minimize} & (1/n)\sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} & x_i \in X_i \quad i = 1, \ldots, n \\
& A_i x_i \leq y \quad i = 1, \ldots, n \\
& y \in Y,
\end{array}$$

where $f_1, \ldots, f_n$ are functions, $A_1, \ldots, A_n$ are matrices, $X_1, \ldots, X_n, Y$ are sets, and $x_1, \ldots, x_n, y$ are the optimization variables (For convenience, we convert the maximization problem into a minimization problem.) For a comprehensive discussion on network utility maximization, see [43].

This optimization problem is equivalent to the master problem

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} g_i(y) + I_Y(y)$$

where we define the subproblems

$$g_i(y) = \inf\left\{f_i(x_i)\,|\, x_i \in X_i,\, A_i x_i \leq y\right\},$$

for $i = 1, \ldots, n$. The master problem only involves the variable y, and x_i is the variable in the ith subproblem. For each $i = 1, \ldots, n$, the function g_i may not be differentiable, but it is convex if f_i and X_i are convex. If Y is convex, then the equivalent problem is convex.

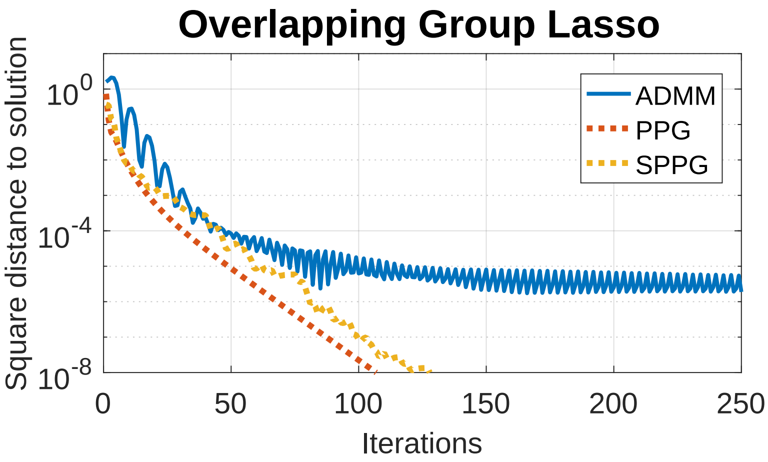

Figure 5: The error $\|x^{k+1/2} - x^\star\|^2$ vs. iteration for overlapping group lasso example.

We view $f = 0$ and $r = I_Y$ in the notation of (1), and apply (PPG):

$$\begin{aligned}
x^{k+1/2} &= \Pi_Y \left(\frac{1}{n} \sum_{i=1}^n z_i^k \right) \\
y_i^{k+1} &= \operatorname*{argmin}_{\substack{x_i \in X_i \\ A_i x_i \le y \\ y \in Y}} \left\{ \alpha f_i(x_i) + \frac{1}{2} \|y - (2x^{k+1/2} - z_i^k)\|_2^2 \right\} \\
z_i^{k+1} &= z_i^k + y_i^{k+1} - y^{k+1/2}.
\end{aligned}$$

Network utility maximization is often performed on a distributed computing network, and if so the optimization problem for evaluating the proximal operators can be solved in a distributed, parallel fashion.

6 Experiments

In this section, we present numerical experiments on two applications discussed in Section 5. The first experiment is small and is meant to serve as a proof of concept. The second experiment is more serious; the problem size is large and we compare the performance with existing methods. For the sake of scientific reproducibility, we provide the code used to generate these experiments.

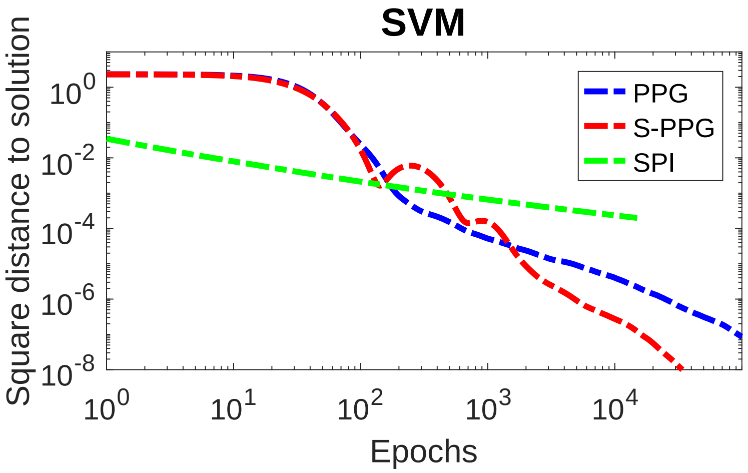

Figure 6: The error $\|x^{k+1/2} - x^\star\|^2$ vs. iteration for the SVM example.

For both experiments, we observe linear convergence. This is a pleasant surprise, as the theory presented in Section 4 and proved in Section 7 only guarantee a $\mathcal{O}(1/k)$ rate.

Overlapping group lasso. The problem size of this setup is $m = 300$, $d = 42$, $n = 3$. The groups are as described in Section 5.

The dominant cost per iteration of (PPG) is evaluating $\mathbf{prox}_{\alpha r}$ which takes $\mathcal{O}(d^2)$ time with the precomputed factorization. Since the cost per iteration of (S-PPG) is no cheaper than that of (PPG), there is no reason to use (S-PPG).

We compare the performance of (PPG) to consensus ADMM (cf. §7.1 of [8]). Both methods use the same computational subroutines and therefore have essentially the same computational cost per iteration. We show the results in Figure 5.

SVM. The problem size of this setup is $n = 2^{17} = 131,072$ and $d = 512$. The synthetic dataset A and y are randomly genearated and the regularization parameter $\lambda = 0.1$ is used. So the problem data A consists of 64×2^{20} numbers and requires 500MB storage to store in double-precision floating-point format.

First, we compare the performance of (PPG) and (S-PPG) to the stochastic proximal iteration with diminishing step size $\alpha_k = C/k$ in Figure 6. For all three methods, the parameters were roughly tuned for optimal performance.

Next, we compare the (PPG) and (S-PPG) to a state-of-the-art SVM solver

Method	Run time	Objective value
LIBLINEAR	$8.47s$	3.7699
CPU (PPG)	$33.9s$ (30 iterations)	3.7364
CPU (S-PPG)	$37.2s$ (30 epochs)	3.7364
CUDA (PPG)	$0.68s$ (30 iterations)	3.7364

Table 1: Run time as a function of grid size

LIBLINEAR [19]. Since LIBLINEAR is based on a second order method while (PPG) and (S-PPG) are first-order methods, comparing the number of iterations is not very meaningful. Rather we compare the wall-clock time these methods take to reach an equivalent level of accuracy. To compare the quality of the solutions, we use the objective value of the problem (8).

The CPU code for (PPG) and (S-PPG) are written in C++. The code is serial, not heaviliy optimized, and does not utilize BLAS (Basic Linear Algebra Subprograms) libraries or any SIMD instructions. On the other hand, LIBLINEAR is heavily optimized and does utilize BLAS.

We also implemented (PPG) on CUDA and ran it on a GPU. The algorithmic structure of (PPG) is particularly well-suited for CUDA, especially when the problem size is large. Roughly speaking, the $x^{k+1/2}$ update requires a reduce operation, which can be done effectively on CUDA. The z_i^k updates are embarassingly parallel, and can be done very effectively on CUDA. The threads must globally synchronize twice per iteration: once before computing the average of z_i^k for $i = 1, \ldots, n$ and once after $x^{k+1/2}$ has been computed. Generally speaking, global synchronization on a GPU is expensive, but we have empirically verified that the computational bottleneck is in the other operations, not the synchronization, when the problem size is reasonably large.

Table 1 shows the results. We see that CPU implementation of (PPG) and (S-PPG) are competitive with LIBLINEAR, and could even be faster than LIBLINEAR if the code is further optimized. On the other hand, the CUDA implementation of (PPG) clearly outperforms LIBLINEAR. (PPG) and (S-PPG) were run until the objective values were good as that of LIBLINEAR. These expeirments were run on an Intel Core i7-990 GPU and a GeForce GTX TITAN X GPU.

7 Convergence proofs

We say a function f is closed convex and proper if its epigraph

$$\left\{(x, \alpha) \mid x \in \mathbb{R}^d, \, |f(x)| < \infty, \, f(x) \leq \alpha\right\}$$

is a closed subset of $\mathbb{R}^{d+1}$, f is convex, and $f(x) = -\infty$ nowhere and $f(x) < \infty$ for some x.

A closed convex and proper function f has L-Lipschitz continuous gradient if f is differentiable everywhere and

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

for all $x, y \in \mathbb{R}^d$. This holds if and only if a closed convex and proper function f satisfies

$$\|\nabla f(x) - \nabla f(y)\|_2^2 \leq L(\nabla f(x) - \nabla f(y))^T(x - y)$$

for all $x, y \in \mathbb{R}^d$, which is known as the Baillon-Haddad Theorem [2]. See [3, 4, 50] for a discussion on this.

Proximal operators are firmly non-expansive. This means for any closed convex and proper function f and $x, y \in \mathbb{R}^d$,

$$\|\mathbf{prox}_f(x) - \mathbf{prox}_f(y)\|_2^2 \leq (\mathbf{prox}_f(x) - \mathbf{prox}_f(y))^T(x - y).$$

By applying Cauchy-Schwartz inequality, we can see that firmly non-expansive operators are non-expansive.

Some proximable functions. As discussed, many standard references like [11, 44] provide a list proximable functions. Here we discuss the few we use.

An optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C, \end{array}$$

where x is the optimization variable and C is a constraint set, can be transformed into the equivalent optimization problem

$$\text{minimize} \quad f(x) + I_C(x).$$

The *indicator function* I_C is defined as

$$I_C(x) = \begin{cases} 0 & \text{for } x \in C \\ \infty & \text{otherwise,} \end{cases}$$

and the proximal operator with respect to I_C is

$$\mathbf{prox}_{\alpha I_C}(x) = \Pi_C(x)$$

where Π_C is the projection onto C for any $\alpha > 0$. So I_C is proximable if the projection onto C is easy to evaluate.

The following results are well known. The proximal operator with respect to $r(x) = |x|$ is called the scalar soft-thresholding operator

$$\mathbf{prox}_{\lambda r}(x) = s_\lambda(x) = \begin{cases} x + \lambda & x < -\lambda \\ 0 & -\lambda \leq x \leq \lambda \\ x - \lambda & x > \lambda. \end{cases}$$

The proximal operator with respect to $r(x) = \|x\|_2$ is called the vector soft-thresholding operator

$$\mathbf{prox}_{\lambda r}(x) = u_\lambda(x) = \begin{cases} \max\{1 - \lambda/\|x\|_2, 0\}x & \text{for } x \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The proximal operator with respect to $r(M) = \|M\|_*$ is called the matrix soft-thresholding operator

$$\mathbf{prox}_{\lambda r}(M) = t_\lambda(M) = \{U s_\lambda(\Sigma) V^T \,|\, U\Sigma V^T = M \text{ is the SVD}\},$$

where $s_\lambda(\Sigma)$ is applied element-wise to the diagonals.

Lemma 2. *Assume $g(x) = f(a^T x)$ where $a \in \mathbb{R}^d$ and f is a closed, convex, and proper function on $\mathbb{R}$. Then the $\mathbf{prox}_g$ can be evaluated by solving a one-dimensional optimization problem.*

Proof. By examining the optimization problem that defines $\mathbf{prox}_g$

$$\mathbf{prox}_g(x_0) = \underset{x}{\text{argmin}} \left\{ f(a^T x) + \frac{1}{2}\|x - x_0\|_2^2 \right\}$$

we see that solution must be of the form $x_0 + \beta a$. So

$$\mathbf{prox}_g(x_0) = x_0 + \beta a, \qquad \beta = \underset{\beta}{\text{argmin}} \left\{ f(a^T x_0 + \beta\|a\|_2^2) + \frac{\|a\|_2^2}{2}\beta^2 \right\}.$$

□

Lemma 3. *Let*

$$g(x_1, x_2, \dots, x_n) = f(a_1 x_1 + a_2 x_2 + \cdots + a_n x_n)$$

where $x_1, \dots, x_n \in \mathbb{R}^d$, $a \in \mathbb{R}^n$, $a \neq 0$, and $f : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ is closed, convex, and proper. Then we can compute $\mathbf{prox}_g$ with

$$\begin{aligned} w &= \mathbf{prox}_{\|a\|_2^2 f}(a_1\xi_1 + a_2\xi_2 + \cdots + a_n\xi_n) \\ v &= \frac{1}{\|a\|_2^2}(a_1\xi_1 + a_2\xi_2 + \cdots + a_n\xi_n - w) \\ \mathbf{prox}_g(\xi_1, \xi_2, \dots, \xi_n) &= \begin{pmatrix} \xi_1 - a_1 v \\ \xi_2 - a_2 v \\ \vdots \\ \xi_n - a_n v \end{pmatrix}. \end{aligned}$$

Proof. The optimality conditions of $\mathbf{prox}_g(\xi_1, \xi_2, \dots, \xi_n)$ gives us

$$\begin{aligned} 0 &\in a_1^2 v + a_1(x_1 - \xi_1) \\ &\vdots \qquad\qquad \vdots \\ 0 &\in a_n^2 v + a_n(x_n - \xi_n) \end{aligned}$$

for some $v \in \partial f(a_1x_1 + \cdots + a_nx_n)$. Summing this we get

$$0 \in \|a\|_2^2 v + (a_1x_1 + \cdots + a_nx_n) - (a_1\xi_1 + \cdots + a_n\xi_n)$$

and with $w = a_1x_1 + \cdots + a_nx_n$ we have

$$w = \mathbf{prox}_{\|a\|_2^2 f}(a_1\xi_1 + a_2\xi_2 + \cdots + a_n\xi_n).$$

The expression for v and $\mathbf{prox}_g(\xi_1, \xi_2, \ldots, \xi_n)$ follows from reorganizing the equations.

□

So if $g(x, y) = f(x + y)$ then

$$\mathbf{prox}_g(x_0, y_0) = \frac{1}{2}\begin{pmatrix} x_0 - y_0 + \mathbf{prox}_{2f}(x_0 + y_0) \\ y_0 - x_0 + \mathbf{prox}_{2f}(x_0 + y_0) \end{pmatrix}.$$

If $g(x, y) = f(x - y)$ then

$$\mathbf{prox}_g(x_0, y_0) = \frac{1}{2}\begin{pmatrix} x_0 + y_0 + \mathbf{prox}_{2f}(x_0 - y_0) \\ x_0 + y_0 - \mathbf{prox}_{2f}(x_0 - y_0) \end{pmatrix}.$$

7.1 Deterministic analysis

Let h be a closed, convex, and proper function on $\mathbb{R}^d$. When

$$x = \mathbf{prox}_{\alpha h}(x_0),$$

we have

$$\alpha u + x = x_0$$

with $u \in \partial h(x_0)$. To simplify the notation, we write

$$\alpha \tilde{\nabla} h(x_0) + x = x_0$$

where $\tilde{\nabla} h(x_0) \in \partial h(x_0)$. So is $\tilde{\nabla} h(x_0)$ a subgradient of h at x_0, and which subgradient $\tilde{\nabla} h(x_0)$ is referring to depends on the context. (This notation is convenient yet potentially sloppy, but we promise to not commit any fallacy of equivocation.)

Proof of Lemma 1. Assume $\mathbf{z}^\star$ is a fixed point of (PPG) or (S-PPG). Then $p(\mathbf{z}^\star) = 0$. Then we have $x^\star = x_i'^\star$ for $i = 1, \ldots, n$ where $x^\star$ and $x_1'^\star, \ldots, x_n'^\star$ are as defined in (5). So

$$\begin{aligned} 0 &= \alpha \tilde{\nabla} r(x^\star) + x^\star - \bar{z} \\ 0 &= \alpha \tilde{\nabla} g_1(x^\star) - x^\star + z_1^\star + \alpha \nabla f_1(x^\star) \\ &\;\;\vdots \qquad\qquad\qquad \vdots \\ 0 &= \alpha \tilde{\nabla} g_n(x^\star) - x^\star + z_n^\star + \alpha \nabla f_n(x^\star). \end{aligned}$$

Adding these up and dividing by n appropriately gives us

$$0 = \tilde{\nabla} r(x^\star) + \tilde{\nabla} \bar{g}(x^\star) + \nabla \bar{f}(x^\star),$$

so $x^\star$ is a solution of Problem (1). Reorganize the definition of x in (5) to get

$$x^\star = \operatorname*{argmin}_x \left\{ r(x) - \frac{1}{\alpha}(\bar{z}^\star - x^\star)^T x + \frac{1}{2\alpha}\|x - x^\star\|_2^2 \right\}.$$

So $x^\star$ minimizes $L(\cdot, \mathbf{x}^\star, (1/\alpha)(\mathbf{z}^\star - \mathbf{x}^\star))$, where L is defined in (4). Reorganize the definition of x' in (5) to get

$$x^\star = \operatorname*{argmin}_x \left\{ f(x^\star) + (\nabla f(x^\star))^T (x - x^\star) + g(x) + \frac{1}{\alpha}(z_i^\star - x^\star)^T x + \frac{1}{2\alpha}\|x - x^\star\|_2^2 \right\}.$$

So $x^\star$ minimizes $L(x^\star, \cdot, (1/\alpha)(\mathbf{z}^\star - \mathbf{x}^\star))$. So $\boldsymbol{\nu}^\star = (1/\alpha)(\mathbf{z}^\star - \mathbf{x}^\star)$ is a dual solution of Problem (1).

The argument works in the other direction as well. If we assume $(x^\star, \boldsymbol{\nu}^\star)$ is a primal dual solution of Problem (1), we can show that $\mathbf{z}^\star = \mathbf{x}^\star + \alpha\boldsymbol{\nu}^\star$ is a fixed point of (PPG) by following a similar line of logic. □

Before we proceed to the main proofs, we introduce more notation. Define the function

$$\bar{r}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n r(x_i) + I_C(\mathbf{x}),$$

where

$$C = \{(x_1, \ldots, x_n) \,|\, x_1 = \cdots = x_n\}.$$

So

$$I_C(\mathbf{x}) = \begin{cases} 0 & \text{for } x_1 = \cdots = x_n \\ \infty & \text{otherwise.} \end{cases}$$

As before, we write

$$\bar{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(x_i), \qquad \bar{g}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n g_i(x_i).$$

With this new notation, we can recast Problem (1) into

$$\text{minimize} \quad \bar{r}(\mathbf{x}) + \bar{f}(\mathbf{x}) + \bar{g}(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^{dn}$ is the optimization variable. We can also rewrite the definition of p as the following three-step process, which starts from $\mathbf{z}$, produces intermediate points $\mathbf{x}, \mathbf{x}'$, and yields $p(\mathbf{z})$:

$$\mathbf{x} = \mathbf{prox}_{\alpha\bar{r}}(\mathbf{z}) \tag{9a}$$

$$\mathbf{x}' = \mathbf{prox}_{\alpha\bar{g}}\left(2\mathbf{x} - \mathbf{z} - \alpha\nabla\bar{f}(\mathbf{x})\right) \tag{9b}$$

$$p(\mathbf{z}) = (1/\alpha)(\mathbf{x} - \mathbf{x}'). \tag{9c}$$

Note that $x_1, \ldots, x_n$ in $\mathbf{x}$ out of (9a) are identical due to I_C. This is the same p as the p defined in (5); we're just using the new notation.

We treat the boldface variables as vectors in $\mathbb{R}^{dn}$. So the inner product between boldface variables is

$$\mathbf{x}^T\mathbf{z} = \sum_{i=1}^{n} x_i^T z_i$$

and the gradient of $\bar{f}(\mathbf{x})$ is

$$\nabla \bar{f}(\mathbf{x}) = \frac{1}{n}\begin{bmatrix} \nabla f_1(x_1) \\ \nabla f_2(x_2) \\ \vdots \\ \nabla f_n(x_n) \end{bmatrix}.$$

We use $\tilde{\nabla} g(\mathbf{x})$ and $\tilde{\nabla} r(\mathbf{x})$ in the same manner as before.

Lemma 4. *Let* $\mathbf{z}$ *and* $\tilde{\mathbf{z}}$ *be any points in* $\mathbb{R}^{dn}$. *Then*

$$\alpha\|p(\mathbf{z}) - p(\tilde{\mathbf{z}})\|^2 \leq (p(\mathbf{z}) - p(\tilde{\mathbf{z}}))^T(\mathbf{z} - \tilde{\mathbf{z}}) - (\nabla \bar{f}(\mathbf{x}) - \bar{f}(\tilde{\mathbf{x}}))^T(\mathbf{x}' - \tilde{\mathbf{x}}')$$

where $\tilde{\mathbf{x}}$ *and* $\tilde{\mathbf{x}}'$ *are obtained by applying* (9a) *and then* (9b), *respectively, to* $\tilde{\mathbf{z}}$ *instead of* $\mathbf{z}$.

Proof. This Lemma is similar to Lemma 3.3 of [15]. We reproduce the proof using this paper's notation.

$$\begin{aligned}
\|\mathbf{z} - \alpha p(\mathbf{z}) - \tilde{\mathbf{z}} + \alpha p(\tilde{\mathbf{z}})\|_2^2 &\overset{(a)}{=} \|\mathbf{z} - \mathbf{x} - \tilde{\mathbf{z}} + \tilde{\mathbf{x}}\|_2^2 + \|\mathbf{x}' - \tilde{\mathbf{x}}'\|_2^2 + 2(\mathbf{z} - \mathbf{x} - \tilde{\mathbf{z}} + \tilde{\mathbf{x}})^T(\mathbf{x}' - \tilde{\mathbf{x}}') \\
&\overset{(b)}{\leq} (\mathbf{z} - \mathbf{x} - \tilde{\mathbf{z}} + \tilde{\mathbf{x}})^T(\mathbf{z} - \tilde{\mathbf{z}}) + (\mathbf{x}' - \tilde{\mathbf{x}}')^T(2\mathbf{x} - \mathbf{z} - \alpha\nabla f(\mathbf{x}) - 2\tilde{\mathbf{x}} + \tilde{\mathbf{z}} + \alpha\nabla f(\tilde{\mathbf{x}})) \\
&\qquad + 2(\mathbf{z} - \mathbf{x} - \tilde{\mathbf{z}} + \tilde{\mathbf{x}})^T(\mathbf{x}' - \tilde{\mathbf{x}}') \\
&= (\mathbf{z} - \alpha p(\mathbf{z}) - \tilde{\mathbf{z}} + \alpha p(\tilde{\mathbf{z}}))^T(\mathbf{z} - \tilde{\mathbf{z}}) - \alpha(\nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}}))^T(\mathbf{x}' - \tilde{\mathbf{x}}'),
\end{aligned}$$

where (a) is due to $\alpha p(\mathbf{z}) = \mathbf{x} - \mathbf{x}'$ and $\alpha p(\tilde{\mathbf{z}}) = \tilde{\mathbf{x}} - \widetilde{\mathbf{x}'}$, (b) follows from the fact that the two mappings:

$$\mathbf{z} \;\mapsto\; \mathbf{z} - \mathbf{x} = \mathbf{z} - \mathbf{prox}_{\alpha\bar{r}}(\mathbf{z})$$

and

$$2\mathbf{x} - \mathbf{z} - \alpha\nabla\bar{f}(\mathbf{x}) \;\mapsto\; \mathbf{x}' = \mathbf{prox}_{\alpha\bar{g}}\left(2\mathbf{x} - \mathbf{z} - \alpha\nabla\bar{f}(\mathbf{x})\right)$$

are both firmly non-expansive. By expanding the $\|\mathbf{z} - \alpha p(\mathbf{z}) - \tilde{\mathbf{z}} + \alpha p(\tilde{\mathbf{z}})\|_2^2$ and cancellation, we obtain

$$\|\alpha p(\mathbf{z}) - \alpha p(\tilde{\mathbf{z}})\|_2^2 \leq (\alpha p(\mathbf{z}) - \alpha p(\tilde{\mathbf{z}}))^T(\mathbf{z} - \tilde{\mathbf{z}}) - \alpha(\nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}}))^T(\mathbf{x}' - \tilde{\mathbf{x}}'),$$

which proves the lemma by dividing both sides by α. □

Lemma 5. *Let $\mathbf{z}^\star$ be any fixed point of* (PPG)*, i.e., $p(\mathbf{z}^\star) = 0$. Then*

$$\|\mathbf{z}^k - \mathbf{z}^\star\|_2^2 \leq \|\mathbf{z}^0 - \mathbf{z}^\star\|_2^2 \tag{10}$$

for all $k = 0, 1, \ldots$. Moreover, we have

$$\sum_{k=0}^{\infty} \|p(\mathbf{z}^k)\|^2 < \infty \tag{11}$$

$$\sum_{k=0}^{\infty} \|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\|^2 < \infty. \tag{12}$$

Finally, $\|p(\mathbf{z}^k)\|^2$ monotonically decreases.

Proof. With the Baillon-Haddad theorem and Young's inequality on (a) we get

$$\begin{aligned}
&-(\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}}))^T(\mathbf{x}' - \tilde{\mathbf{x}}') && (13)\\
&\quad = -(\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}}))^T(\mathbf{x} - \alpha p(\mathbf{z}) - \tilde{\mathbf{x}} + \alpha p(\tilde{\mathbf{z}})) && (14)\\
&\quad = -(\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}}))^T(\mathbf{x} - \tilde{\mathbf{x}}) + \alpha(\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}}))^T(p(\mathbf{z}) - p(\tilde{\mathbf{z}})) && (15)\\
&\quad \overset{(a)}{\leq} -\tfrac{1}{L}\|\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}})\|^2 + \tfrac{3}{4L}\|\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}})\|^2 + \tfrac{L\alpha^2}{3}\|p(\mathbf{z}) - p(\tilde{\mathbf{z}})\|^2 && (16)\\
&\quad = \tfrac{L\alpha^2}{3}\|p(\mathbf{z}) - p(\tilde{\mathbf{z}})\|^2 - \tfrac{1}{4L}\|\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}})\|^2. && (17)
\end{aligned}$$

Combining Lemma 4 and equation (17), we obtain

$$-(p(\mathbf{z}) - p(\tilde{\mathbf{z}}))^T(\mathbf{z} - \tilde{\mathbf{z}}) \leq \alpha(\tfrac{\alpha L}{3} - 1)\|p(\mathbf{z}) - p(\tilde{\mathbf{z}})\|^2 - \tfrac{1}{4L}\|\nabla \bar{f}(\mathbf{x}) - \nabla \bar{f}(\tilde{\mathbf{x}})\|^2. \tag{18}$$

Applying (18) separately with $(\mathbf{z}, \tilde{\mathbf{z}}) = (\mathbf{z}^k, \mathbf{z}^\star)$ and $(\mathbf{z}^k, \mathbf{z}^{k+1})$, we get, respectively,

$$\begin{aligned}
\|\mathbf{z}^{k+1} - \mathbf{z}^\star\|^2 &= \|\mathbf{z}^k - \mathbf{z}^\star\|^2 + \alpha^2\|p(\mathbf{z}^k)\|^2 - 2\alpha(\mathbf{z}^k - \mathbf{z}^\star)^T p(\mathbf{z}^k)\\
&\leq \|\mathbf{z}^k - \mathbf{z}^\star\|^2 - \alpha^2(1 - \tfrac{2\alpha L}{3})\|p(\mathbf{z}^k)\|^2 - \tfrac{\alpha}{2L}\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\|^2 \qquad (19)
\end{aligned}$$

and

$$\begin{aligned}
\|p(\mathbf{z}^{k+1})\|^2 &= \|p(\mathbf{z}^k)\|^2 + \|p(\mathbf{z}^{k+1}) - p(\mathbf{z}^k)\|^2 - 2(p(\mathbf{z}^{k+1}) - p(\mathbf{z}^k))^T \tfrac{1}{\alpha}(\mathbf{z}^{k+1} - \mathbf{z}^k)\\
&\leq \|p(\mathbf{z}^k)\|^2 - (1 - \tfrac{2\alpha L}{3})\|p(\mathbf{z}^{k+1}) - p(\mathbf{z}^k)\|^2. \qquad (20)
\end{aligned}$$

(In (17) we can use a different parameter for Young's inequality, and improve inequalities (19) and (20) to allow $\alpha < 2/L$, which is better than $\alpha < 3/(2L)$. In fact, $\alpha < 2/L$ is sufficient for convergence in Theorem 1. However, we use the current version because we need the last term of inequality (19) for proving Theorem 3.)

Summing (19) through $k = 0, 1, \ldots$ give us the summability result. Inequality (20) states that $\|p(\mathbf{z}^k)\|^2$ monotonically decreases.

□

Proof of Theorem 1. Lemma 5 already states that $\|p(\mathbf{z}^k)\|_2^2$ decreases monotonically. Using inequality (11) of Lemma 5, we get

$$\|p(\mathbf{z}^k)\|_2^2 = \min_{i=0,1,\dots,k} \|p(\mathbf{z}^i)\|_2^2 \leq \frac{1}{k}\sum_{i=0}^{\infty} \|p(\mathbf{z}^i)\|_2^2 = C/k = \mathcal{O}(1/k)$$

for some finite constant C. (The rate $\mathcal{O}(1/k)$ can be improved to $o(1/k)$ using, say, Lemma 1.2 of [18], but we present the simpler argument.)

By (10) of Lemma 5, $\mathbf{z}^k$ is a bounded sequence and will have a limit point, which we call $\mathbf{z}^\infty$. Lemma 4 also implies p is a continuous function. Since $p(\mathbf{z}^k) \to 0$ and p is continuous, the limit point $\mathbf{z}^\infty$ must satisfy $p(\mathbf{z}^\infty) = 0$. Applying inequality (10) with $\mathbf{z}^\star = \mathbf{z}^\infty$ tells us that $\|\mathbf{z}^k - \mathbf{z}^\infty\|_2^2 \to 0$, i.e., the entire sequence converges.

Since $\mathbf{prox}_{\alpha r}$ is a continuous function

$$x^{k+1/2} = \mathbf{prox}_{\alpha r}(\bar{z}^k) \to \mathbf{prox}_{\alpha r}(\bar{z}^\star) = x^\star.$$

With this same argument, we also conclude that $x_i^k \to x^\star$ for all $i = 1, \dots, n$. □

Proof of Theorem 2. A convex function h satisfies the inequality

$$h(x) - h(\tilde{x}) \leq (\tilde{\nabla} h(x))^T (x - \tilde{x})$$

for any x and $\tilde{x}$ (so long as a subgradient $\tilde{\nabla} h(x)$ exists).

Applying this inequality we get

$$\begin{aligned}
E^k &\leq (\tilde{\nabla}\bar{r}(\mathbf{x}^{k+1/2}) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T(\mathbf{x}^{k+1/2} - \mathbf{x}^\star) + (\tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}))^T(\mathbf{x}^{k+1} - \mathbf{x}^\star) \\
&= (\tilde{\nabla}\bar{r}(\mathbf{x}^{k+1/2}) + \nabla\bar{f}(\mathbf{x}^{k+1/2}) + \tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}))^T(\mathbf{x}^{k+1/2} - \mathbf{x}^\star) - (\tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}))^T(\alpha p(\mathbf{z}^k)) \\
&= (\mathbf{x}^{k+1/2} - \alpha\tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}) - \mathbf{x}^\star)^T p(\mathbf{z}^k) \qquad (21) \\
&= ((\mathbf{z}^{k+1} - \mathbf{z}^\star) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^{k+1/2})))^T p(\mathbf{z}^k) \qquad (22) \\
&= (\mathbf{z}^{k+1} - \mathbf{z}^\star)^T p(\mathbf{z}^k) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k). \qquad (23)
\end{aligned}$$

For the second equality, we used

$$p(\mathbf{z}^k) = \tilde{\nabla}\bar{r}(\mathbf{x}^{k+1/2}) + \nabla\bar{f}(\mathbf{x}^{k+1/2}) + \tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}).$$

and combined terms. For the third equality, we used $\mathbf{x}^\star = \mathbf{z}^\star - \alpha\tilde{\nabla}\bar{r}(\mathbf{x}^\star)$ and

$$\begin{aligned}
\mathbf{x}^{k+1/2} - &\alpha\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \alpha\tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}) \\
&= \mathbf{z}^k - \alpha\tilde{\nabla}\bar{r}(\mathbf{x}^{k+1/2}) - \alpha\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \alpha\tilde{\nabla}\bar{g}(\mathbf{x}^{k+1}) \\
&= \mathbf{z}^k - \alpha p(\mathbf{z}^k) \\
&= \mathbf{z}^{k+1}.
\end{aligned}$$

Likewise we have

$$\begin{aligned}
E^k &\geq (\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star))^T(\mathbf{x}^{k+1/2} - \mathbf{x}^\star) + (\tilde{\nabla}\bar{g}(\mathbf{x}^\star))^T(\mathbf{x}^{k+1} - \mathbf{x}^\star) \\
&= (\mathbf{x}^{k+1} - \mathbf{x}^\star)^T p(\mathbf{z}^\star) + (\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star))^T \alpha p(\mathbf{z}^k) \qquad (24) \\
&= (\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star))^T \alpha p(\mathbf{z}^k). \qquad (25)
\end{aligned}$$

Here we use

$$p(\mathbf{z}^\star) = \tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star) + \tilde{\nabla}\bar{r}(\mathbf{x}^\star) = 0$$

Theorem 1 states that the sequences $\mathbf{z}^1, \mathbf{z}^2, \dots$ and $\mathbf{x}^{1+1/2}, \mathbf{x}^{2+1/2}, \dots$ both converge and therefore are bounded and that $\|p(\mathbf{z}^k)\|_2 = \mathcal{O}(1/\sqrt{k})$. Combining this with the bounds on E^k gives us $|E^k| \leq \mathcal{O}(1/\sqrt{k})$.

□

Proof of Theorem 3. By Jensen's inequality, we have

$$E^k_{\text{erg}} \leq \frac{1}{k}\sum_{i=1}^{k} E^i. \tag{26}$$

Continuing the last line of (23), we get

$$\begin{aligned} E^k &\leq (\mathbf{z}^{k+1} - \mathbf{z}^\star)^T p(\mathbf{z}^k) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k) && (27) \\ &= \tfrac{1}{2\alpha}\|\mathbf{z}^k - \mathbf{z}^\star\|^2 - \tfrac{1}{2\alpha}\|\mathbf{z}^{k+1} - \mathbf{z}^\star\|^2 - \tfrac{\alpha}{2}\|p(\mathbf{z}^k)\|^2 + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k). && (28) \end{aligned}$$

Combining (26) and (27) and after telescopic cancellation,

$$\begin{aligned} E^k_{\text{erg}} &\leq \frac{1}{2\alpha k}\|\mathbf{z}^1 - \mathbf{z}^\star\|^2 + \frac{1}{k}\sum_{i=1}^{k}\alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^{i+1/2}))^T p(\mathbf{z}^i) && (29) \\ &= \mathcal{O}(1/k) + \frac{1}{k}\alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star))^T\sum_{i=1}^{k} p(\mathbf{z}^i) + \frac{1}{k}\alpha(\nabla\bar{f}(\mathbf{x}^i) - \nabla\bar{f}(\mathbf{x}^\star))^T\sum_{i=1}^{k} p(\mathbf{z}^i) && (30) \\ &\leq \mathcal{O}(1/k) + \frac{1}{k\alpha}\|\mathbf{z}^{k+1} - \mathbf{z}^1\|\|\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star)\| + \frac{1}{2k}\sum_{i=1}^{k}\left(\alpha^2\|p(\mathbf{z}^i)\|^2 + \|\nabla\bar{f}(\mathbf{x}^i) - \nabla\bar{f}(\mathbf{x}^\star)\|^2\right) && (31) \\ &= \mathcal{O}(1/k), && (32) \end{aligned}$$

where the last line holds due to the boundedness of $\mathbf{z}^k$ and Lemma 5. With a similar argument as in (25), we get

$$\begin{aligned} E^k_{\text{erg}} &\geq (\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star))^T(\mathbf{x}^{k+1/2}_{\text{erg}} - \mathbf{x}^{k+1}_{\text{erg}}) = \frac{1}{k}\left(\sum_{i=1}^{k}\alpha p(\mathbf{z}^k)\right)^T(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star)) && (33) \\ &= \tfrac{1}{k}(\mathbf{z}^k - \mathbf{z}^0)^T(\tilde{\nabla}\bar{r}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star)) = O(1/k), && (34) \end{aligned}$$

where have once again used the boundedness of $(\mathbf{z}^k)_k$. Furthermore, since

$$|E^k_{\text{erg}} - e^k_{\text{erg}}| \leq L_g\|\mathbf{x}^{k+1}_{\text{erg}} - \mathbf{x}^{k+1/2}_{\text{erg}}\| = L_g\|\tfrac{1}{k}(\mathbf{z}^{k+1} - \mathbf{z}^1)\| = \mathcal{O}(1/k), \tag{35}$$

we have $e^k_{\text{erg}} = \mathcal{O}(1/k)$.

□

7.2 Stochastic analysis

Proof of Theorem 5. To express the updates of (S-PPG) we introduce the following notation:

$$p(\mathbf{z}^k)_{[i]} = \begin{bmatrix} 0 \\ \vdots \\ p(\mathbf{z}^k)_i \\ \vdots \\ 0 \end{bmatrix}.$$

With this notation, we can express the iterates of (S-PPG) as

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \alpha p(\mathbf{z}^k)_{[i(k)]}, \tag{36}$$

and we also have conditional expectations

$$\begin{aligned} \mathbb{E}_k p(\mathbf{z}^k)_{[i(k)]} &= \frac{1}{n} p(\mathbf{z}^k), \\ \mathbb{E}_k \|p(\mathbf{z}^k)_{[i(k)]}\|^2 &= \frac{1}{n}\|p(\mathbf{z}^k)\|^2. \end{aligned}$$

Here, we let $\mathbb{E}$ denote the expectation over all random variables $i(1), i(2), \ldots$, and $\mathbb{E}_k$ denote the expectation over $i(k)$ *conditioned on* $i(1), i(2), \ldots, i(k-1)$.

The convergence of this algorithm has been recently analyzed in [12] when $i(k)$ is chosen at random. Below, we adapt its proof to our setting with new rate results.

Note that Lemma 4 and inequality (17) remain valid, as they are not tied to a specific sequence of random samples. Hence, similar to (19), we have

$$\|\mathbf{z}^{k+1} - \mathbf{z}^*\|^2 = \|\mathbf{z}^k - \mathbf{z}^*\|^2 + \alpha^2 \|p(\mathbf{z}^k)_{[i(k)]}\|^2 - 2\alpha(\mathbf{z}^k - \mathbf{z}^*)^T p(\mathbf{z}^k)_{[i(k)]}. \tag{37}$$

We take the conditional expectation to get

$$\begin{aligned} \mathbb{E}_k\|\mathbf{z}^{k+1} - \mathbf{z}^*\|^2 &= \|\mathbf{z}^k - \mathbf{z}^*\|^2 + \alpha^2 \mathbb{E}_k \|p(\mathbf{z}^k)_{[i(k)]}\|^2 - 2\alpha(\mathbf{z}^k - \mathbf{z}^*)^T \mathbb{E}_k p(\mathbf{z}^k)_{[i(k)]} \qquad (38) \\ &= \|\mathbf{z}^k - \mathbf{z}^*\|^2 + \tfrac{\alpha^2}{n}\|p(\mathbf{z}^k)\|^2 - \tfrac{2\alpha}{n}(\mathbf{z}^k - \mathbf{z}^*)^T p(\mathbf{z}^k) \qquad (39) \\ &\leq \|\mathbf{z}^k - \mathbf{z}^*\|^2 - \tfrac{\alpha^2}{n}(1 - \tfrac{2\alpha L}{3})\|p(\mathbf{z}^k)\|^2 - \tfrac{\alpha}{2Ln}\|\nabla \bar{f}(\mathbf{x}^k) - \nabla \bar{f}(\mathbf{x}^*)\|^2. \qquad (40) \end{aligned}$$

By the same reasoning as before, we have

$$\min_{0 \leq i \leq k} \mathbb{E}\|p(\mathbf{z}^k)\|^2 \leq \mathcal{O}(1/k).$$

By (40), the sequence $\left(\|\mathbf{z}^k - \mathbf{z}^*\|^2\right)_{k \geq 0}$ is a nonnegative supermartingale. Applying Theorem 1 of [48] to (40) yields the following three properties, which hold with probability one for every fixed point $\mathbf{z}^\star$:

1. the squared fixed-point residual sequence is summable, that is,

$$\sum_{k=0}^{\infty} \|p(\mathbf{z}^k)\|^2 < \infty, \tag{41}$$

2. $\|\mathbf{z}^k - \mathbf{z}^*\|^2$ converges to a nonnegative random number, and

3. $(\mathbf{z}^k)_{k\geq 0}$ is bounded.

To proceed as before, however, we need $\|\mathbf{z}^k - \mathbf{z}^*\|^2$ to converge to a nonnegative random number (not necessarily 0) for all fixed points $\mathbf{z}^*$ with probability one. For each fixed point $\mathbf{z}^*$, the previous argument states that there is a measure one event set[1], $\Omega(\mathbf{z}^*)$, such that $\|\mathbf{z}^k - \mathbf{z}^*\|^2$ converges for all $(\mathbf{z}^k)_{k\geq 0}$ taken from $\Omega(\mathbf{z}^*)$. Note that $\Omega(\mathbf{z}^*)$ depends on $\mathbf{z}^*$ because we must select $\mathbf{z}^*$ to form (40) first. Since the number of fixed points (unless there is only one) is uncountable, $\cap_{\text{fixed point } \mathbf{z}^*}\Omega(\mathbf{z}^*)$ may not be measure one. Indeed, it is measure one as we now argue.

Let Z^* be the set of fixed points. Since $\mathbb{R}^{dn}$ is *separable* (i.e., containing a countable, dense subset), Z^* has a countable dense subset, which we write as $\{\mathbf{z}_1^*, \mathbf{z}_2^*, \dots\}$. By countability, $\Omega_c = \cap_{i=1,2,\dots}\Omega(\mathbf{z}_i^*)$ is a measure one event set, and it is defined independently of the choice of $\mathbf{z}^*$. Next we show that $\|\mathbf{z}^k - \mathbf{z}^*\|^2$ to converge to a nonnegative random number (not necessarily 0) for all fixed points $\mathbf{z}^*$ with probability one by Ω_c, that is, $\lim_k \|\mathbf{z}^k - \mathbf{z}^*\|$ exists for all $\mathbf{z}^* \in Z^*$ and all $(\mathbf{z}^k)_{k\geq 0} \in \Omega_c$.

Now consider any $\mathbf{z}^* \in Z^*$ and $(\mathbf{z}^k)_{k\geq 0} \in \Omega_c$. Then for any $\varepsilon > 0$, there is a $\mathbf{z}_i^*$ such that $\|\mathbf{z}^* - \mathbf{z}_i^*\| \leq \varepsilon$. By the triangle inequality, we can bound $\|\mathbf{z}^k - \mathbf{z}^*\|$ as

$$\begin{aligned}
\|\mathbf{z}^k - \mathbf{z}^*\| &\leq \|\mathbf{z}^k - \mathbf{z}_i^*\| + \|\mathbf{z}_i^* - \mathbf{z}^*\| \leq \|\mathbf{z}^k - \mathbf{z}_i^*\| + \varepsilon,\\
\|\mathbf{z}^k - \mathbf{z}^*\| &\geq \|\mathbf{z}^k - \mathbf{z}_i^*\| - \|\mathbf{z}_i^* - \mathbf{z}^*\| \geq \|\mathbf{z}^k - \mathbf{z}_i^*\| - \varepsilon.
\end{aligned}$$

Since $\|\mathbf{z}^k - \mathbf{z}_i^*\|$ converges, we have we have

$$\begin{aligned}
\limsup_k \|\mathbf{z}^k - \mathbf{z}^*\| &\leq \varepsilon,\\
\liminf_k \|\mathbf{z}^k - \mathbf{z}^*\| &\geq -\varepsilon.
\end{aligned}$$

As $\varepsilon > 0$ is arbitrary, $\liminf_k \|\mathbf{z}^k - \mathbf{z}^*\| = \limsup_k \|\mathbf{z}^k - \mathbf{z}^*\|$. So, $\lim_k \|\mathbf{z}^k - \mathbf{z}_i^*\|$ exists.

Finally, we can proceed with the same argument as in the proof of Theorem 1, and conclude that $\mathbf{z}^k \to \mathbf{z}^*$, $\mathbf{x}^{k+1/2} \to \mathbf{x}^*$, and $\mathbf{x}^{k+1} \to \mathbf{x}^*$ on the measure one set Ω_c.

□

[1] Each event is a randomly realized sequence of iterates $(\mathbf{z}^k)_{k\geq 0}$ in S-PPG.

Proof of Theorem 6. This proof focuses on the treatments that are different from the deterministic analysis. We only go through the steps of estimating the upper bound of $\mathbb{E}(E^k)$, skipping the similar treatment to obtain the lower bound and other rates.

We reuse a part of (23) but avoid replacing $\mathbf{z}^k - \alpha p(\mathbf{z}^k)$ by $\mathbf{z}^{k+1}$ because of (36):

$$
\begin{align}
E^k &\leq (\mathbf{x}^{k+1/2} - \alpha \tilde{\nabla} \bar{g}(\mathbf{x}^{k+1}) - \mathbf{x}^*)^T p(\mathbf{z}^k) \tag{42}\\
&= ((\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^{k+1/2})))^T p(\mathbf{z}^k) \tag{43}\\
&= (\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*)^T p(\mathbf{z}^k) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k) \tag{44}\\
&= (\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*)^T p(\mathbf{z}^k) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k) \tag{45}\\
&\quad + \alpha(\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k). \tag{46}
\end{align}
$$

By the Cauchy-Schwarz inequality,

$$
\begin{align}
\mathbb{E}(E^k) &\leq \mathbb{E}\|\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*\| \cdot \mathbb{E}\|p(\mathbf{z}^k)\| + \alpha\|\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^*)\| \cdot \mathbb{E}\|p(\mathbf{z}^k)\| \tag{47}\\
&\quad + \alpha\mathbb{E}\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\| \cdot \mathbb{E}\|p(\mathbf{z}^k)\| \tag{48}\\
&\leq \Bigg(\sqrt{\mathbb{E}\|\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*\|^2} + \alpha\|\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^*)\| \tag{49}\\
&\quad + \alpha\sqrt{\mathbb{E}\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2}\Bigg)\sqrt{\mathbb{E}\|p(\mathbf{z}^k)\|^2}. \tag{50}
\end{align}
$$

Here we have $\sqrt{\mathbb{E}\|\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*\|^2} \leq \sqrt{\|\mathbf{z}^0 - \mathbf{z}^*\|^2}$ since, similar to (19),

$$
\begin{align}
\|\mathbf{z}^k - \alpha p(\mathbf{z}^k) - \mathbf{z}^*\|^2 &= \|\mathbf{z}^k - \mathbf{z}^*\|^2 + \alpha^2\|p(\mathbf{z}^k)\|^2 - 2\alpha(\mathbf{z}^k - \mathbf{z}^*)^T p(\mathbf{z}^k) \notag\\
&\leq \|\mathbf{z}^k - \mathbf{z}^*\|^2 - \alpha^2(1 - \tfrac{2\alpha L}{3})\|p(\mathbf{z}^k)\|^2 - \tfrac{\alpha}{2L}\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2 \notag\\
&\leq \|\mathbf{z}^k - \mathbf{z}^*\|^2 \tag{51}
\end{align}
$$

and, by (40), $\mathbb{E}\|\mathbf{z}^k - \mathbf{z}^*\|^2 \leq \mathbb{E}\|\mathbf{z}^{k-1} - \mathbf{z}^*\|^2 \leq \cdots \leq \|\mathbf{z}^0 - \mathbf{z}^*\|^2$. The next term in (50), $\alpha\|\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla \bar{f}(\mathbf{x}^*)\|$, is a constant. For the third term, from

$$
\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2 \overset{(a)}{\leq} L^2\|\mathbf{x}^{k+1/2} - \mathbf{x}^*\|^2 \overset{(b)}{\leq} L^2\|\mathbf{z}^k - \mathbf{z}^*\|^2,
$$

where (a) is due to Lipschitz continuity and (b) due to nonexpansiveness of the proximal mapping, it follows that $\alpha\sqrt{\mathbb{E}\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2} \leq \alpha L\sqrt{\|\mathbf{z}^0 - \mathbf{z}^*\|^2}$. Since $\mathbb{E}\|p(\mathbf{z}^k)\|^2 = \mathcal{O}(1/k)$, we immediately have $\mathbb{E}(E^k) \leq \mathcal{O}(1/\sqrt{k})$. Similarly, we can also show $-\mathbb{E}(E^k) \leq \mathcal{O}(1/\sqrt{k})$. Therefore, $\mathbb{E}|E^k| = \mathcal{O}(1/\sqrt{k})$.

By extending the previous analysis of $|E^k_{\text{erg}}|$ and e^k_{erg} to $\mathbb{E}|E^k_{\text{erg}}|$ and $\mathbb{E}(e^k_{\text{erg}})$, respectively, along the same line of arguments, it is straightforward to show $\mathbb{E}|E^k_{\text{erg}}| = \mathcal{O}(1/k)$ and $\mathbb{E}(e^k_{\text{erg}}) = \mathcal{O}(1/k)$. □

7.3 Linear convergence analysis

We first review some definitions and inequities. Let h be a closed convex proper function. We let $\mu_h \geq 0$ be the strong-convexity constant of h, where $\mu_h > 0$ when h is strongly convex and $\mu_h = 0$ otherwise. When h is differentiable and ∇h is Lipschitz continuous, we define $(1/\beta_h)$ be the Lipschitz constant of ∇h. When h is either non-differentiable or differentiable but ∇h is not Lipschitz, we define $\beta_h = 0$. Under these definitions, we have

$$\begin{aligned} h(y)-&h(x) \\ &\geq \langle \tilde{\nabla} h(x), y-x\rangle + \underbrace{\frac{1}{2}\max\left\{\mu_h\|x-y\|^2, \beta_h\|\tilde{\nabla} h(x)-\tilde{\nabla} f(y)\|^2\right\}}_{S_h(x,y)}, \end{aligned} \tag{52}$$

for any points x, y where the subgradients $\tilde{\nabla} h(x), \tilde{\nabla} h(y)$ exist. Note that $S_h(x,y) = S_h(y,x)$.

For the three convex functions $\bar{r}, \bar{f}, \bar{g}$, we introduce their parameters $\mu_{\bar{r}}, \beta_{\bar{r}}, \mu_{\bar{f}}, \beta_{\bar{f}}, \mu_{\bar{g}}, \beta_{\bar{g}}$, as well as the combination

$$S(\mathbf{z}, \mathbf{z}^*) = S_{\bar{r}}(\mathbf{x}, \mathbf{x}^*) + S_{\bar{f}}(\mathbf{x}, \mathbf{x}^*) + S_{\bar{g}}(\mathbf{x}', \mathbf{x}^*). \tag{53}$$

As we assume each f_i has L-Lipschitz gradient, we set $\beta_{\bar{f}} = 1/L$. Since $\bar{r}$ includes the indicator function I_C, which is non-differentiable, we set $\beta_{\bar{r}} = 0$. The values of remaining parameters $\mu_{\bar{r}}, \mu_{\bar{f}}, \mu_{\bar{g}}, \beta_{\bar{g}} \geq 0$ are kept unspecified. Applying (52) to each pair of the three functions in

$$E = \left(\bar{r}(\mathbf{x}) + \bar{f}(\mathbf{x}) + \bar{g}(\mathbf{x}')\right) - \left(\bar{r}(\mathbf{x}^*) + \bar{f}(\mathbf{x}^*) + \bar{g}(\mathbf{x}^*)\right)$$

yields

$$E \leq (\tilde{\nabla}\bar{r}(\mathbf{x}) + \nabla\bar{f}(\mathbf{x}))^T(\mathbf{x}-\mathbf{x}^*) + (\tilde{\nabla}\bar{g}(\mathbf{x}'))^T(\mathbf{x}'-\mathbf{x}^*) - S(\mathbf{z},\mathbf{z}^*), \tag{54}$$

$$E \geq (\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla\bar{f}(\mathbf{x}^*))^T(\mathbf{x}-\mathbf{x}^*) + (\tilde{\nabla}\bar{g}(\mathbf{x}^*))^T(\mathbf{x}'-\mathbf{x}^*) + S(\mathbf{z},\mathbf{z}^*). \tag{55}$$

In this fashion, both the upper and lower bounds on E^k, which we previously derive, are tightened by $S(\mathbf{z}^k, \mathbf{z}^*)$. In particular, we can tightened (28) and (25) as

$$E^k \leq \tfrac{1}{2\alpha}\|\mathbf{z}^k-\mathbf{z}^*\|^2 - \tfrac{1}{2\alpha}\|\mathbf{z}^{k+1}-\mathbf{z}^*\|^2 - \tfrac{\alpha}{2}\|p(\mathbf{z}^k)\|^2 + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k) - S(\mathbf{z}^k,\mathbf{z}^*), \tag{56}$$

$$E^k \geq (\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla\bar{f}(\mathbf{x}^*))^T \alpha p(\mathbf{z}^k) + S(\mathbf{z}^k,\mathbf{z}^*), \tag{57}$$

where the two terms involving $S(\mathbf{z}^k, \mathbf{z}^*)$ are newly added. Combining the upper and lower bounds of E^k yields

$$\tfrac{1}{2\alpha}\|\mathbf{z}^{k+1}-\mathbf{z}^*\|^2 \leq \tfrac{1}{2\alpha}\|\mathbf{z}^k-\mathbf{z}^*\|^2 - Q, \tag{58}$$

where

$$
\begin{aligned}
Q &= -\alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla\bar{f}(\mathbf{x}^{k+1/2}))^T p(\mathbf{z}^k) + \alpha(\tilde{\nabla}\bar{r}(\mathbf{x}^*) + \nabla\bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k) + \tfrac{\alpha}{2}\|p(\mathbf{z}^k)\|^2 + 2S(\mathbf{z}^k,\mathbf{z}^*) \quad (59)\\
&= -\alpha(\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k) + \tfrac{\alpha}{2}\|p(\mathbf{z}^k)\|^2 + 2S(\mathbf{z}^k,\mathbf{z}^*) \quad (60)\\
&= \Big(-\alpha(\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k) + \tfrac{\alpha}{2}\|p(\mathbf{z}^k)\|^2 + 2S_{\bar{f}}(\mathbf{z}^k,\mathbf{z}^*)\Big) + 2S_{\bar{r}}(\mathbf{z}^k,\mathbf{z}^*) + 2S_{\bar{g}}(\mathbf{z}^k,\mathbf{z}^*) \quad (61)\\
&\geq c_1\Big(\|p(\mathbf{z}^k)\|^2 + \|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*)\|^2\Big) + \frac{\mu_{\bar{f}}}{2}\|\mathbf{x}^{k+1/2} - \mathbf{x}^\star\|^2 + 2S_{\bar{r}}(\mathbf{z}^k,\mathbf{z}^*) + 2S_{\bar{g}}(\mathbf{z}^k,\mathbf{z}^*), \quad (62)
\end{aligned}
$$

where $c_1 > 0$ is a constant and the inequality follows from the Young's inequality:

$$
\alpha(\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*))^T p(\mathbf{z}^k) \leq \tfrac{\alpha}{4}\|p(\mathbf{z}^k)\|^2 + \alpha\|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*)\|^2
$$

and $\alpha < \beta_{\bar{f}}$. Later on, in three different cases, we will show

$$
Q \geq C\|\mathbf{z}^k - \mathbf{z}^*\|^2. \tag{63}
$$

Hence, by substituting (63) into (58), we obtain the Q-linear (or quotient-linear) convergence relation

$$
\|\mathbf{z}^{k+1} - \mathbf{z}^*\| \leq \sqrt{1 - 2\alpha C}\|\mathbf{z}^k - \mathbf{z}^*\|, \tag{64}
$$

from which it is easy to further derive the Q-linear convergence results for $|E^k|$ and e^k.

Case 1. Assume $\bar{g}$ is both strongly convex and has Lipschitz gradient, i.e., $\mu_{\bar{g}}, \beta_{\bar{g}} > 0$, (and $\bar{f}$ still has Lipschitz gradient). In this case,

$$
\begin{aligned}
Q &= c_1\Big(\|p(\mathbf{z}^k)\|^2 + \|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*)\|^2\Big) + 2S_{\bar{g}}(\mathbf{z}^k,\mathbf{z}^*) \quad (65)\\
&\geq c_1\Big(\|p(\mathbf{z}^k)\|^2 + \|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^*)\|^2\Big) + \mu_{\bar{g}}\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|^2. \quad (66)
\end{aligned}
$$

(Here we use $\nabla\bar{g}$ instead of $\tilde{\nabla}\bar{g}$ since $\bar{g}$ is differentiable.) By the identities

$$
\begin{aligned}
\mathbf{z}^k &= \mathbf{x}^{k+1} - \alpha\big(\nabla\bar{g}(\mathbf{x}^{k+1}) + \nabla\bar{f}(\mathbf{x}^{k+1/2})\big) + 2\alpha p(\mathbf{z}^k), \quad (67)\\
\mathbf{z}^* &= \mathbf{x}^\star \quad\;\; - \alpha\big(\nabla\bar{g}(\mathbf{x}^\star) + \nabla\bar{f}(\mathbf{x}^\star)\big), \quad (68)
\end{aligned}
$$

the triangle inequality, and $\|\nabla\bar{g}(\mathbf{x}^{k+1}) - \nabla\bar{g}(\mathbf{x}^\star)\| \leq (1/\beta_{\bar{g}})\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|$, we get

$$
\begin{aligned}
\|\mathbf{z}^k - \mathbf{z}^*\|^2 &= \big\|(\mathbf{x}^{k+1} - \mathbf{x}^\star) - \alpha\big(\nabla\bar{g}(\mathbf{x}^{k+1}) - \nabla\bar{g}(\mathbf{x}^\star)\big) - \alpha\big(\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^\star)\big) + 2\alpha p(\mathbf{z}^k)\big\|^2 \quad (69)\\
&\leq 3\big\|(\mathbf{x}^{k+1} - \mathbf{x}^\star) - \alpha\big(\nabla\bar{g}(\mathbf{x}^{k+1}) - \nabla\bar{g}(\mathbf{x}^\star)\big)\big\|^2 + 3\alpha^2\|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^\star)\|^2 + 12\alpha^2\|p(\mathbf{z}^k)\|^2 \quad (70)\\
&\leq 3(1 + \tfrac{\alpha}{\beta_{\bar{g}}})^2\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|^2 + 3\alpha^2\|\nabla\bar{f}(\mathbf{x}^{k+1/2}) - \nabla\bar{f}(\mathbf{x}^\star)\|^2 + 12\alpha^2\|p(\mathbf{z}^k)\|^2. \quad (71)
\end{aligned}
$$

Since (71) is bounded by (66) up to a constant factor, we have established (63) for this case.

Case 2. $\bar{f}$ is strongly convex and $\bar{g}$ has Lipschitz gradient, i.e., $\mu_{\bar{f}}, \beta_{\bar{g}} > 0$ (and $\bar{f}$ still has Lipschitz gradient). In this case,

$$Q = c_1\Big(\|p(\mathbf{z}^k)\|^2 + \|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2\Big) + \mu_{\bar{f}}\|\mathbf{x}^{k+1/2} - \mathbf{x}^*\|^2. \tag{72}$$

By the identities

$$\mathbf{z}^k = \mathbf{x}^{k+1/2} - \alpha\big(\nabla \bar{g}(\mathbf{x}^{k+1}) + \nabla \bar{f}(\mathbf{x}^{k+1/2})\big) + \alpha p(\mathbf{z}^k), \tag{73}$$

$$\mathbf{z}^* = \mathbf{x}^\star \qquad - \alpha\big(\nabla \bar{g}(\mathbf{x}^\star) + \nabla \bar{f}(\mathbf{x}^\star)\big), \tag{74}$$

the triangle inequality, and $\|\nabla \bar{g}(\mathbf{x}^{k+1}) - \nabla \bar{g}(\mathbf{x}^\star)\| \leq (1/\beta_{\bar{g}})\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|$, we get

$$\|\mathbf{z}^k - \mathbf{z}^*\|^2 = \Big\|(\mathbf{x}^{k+1/2} - \mathbf{x}^\star) - \alpha\big(\nabla \bar{g}(\mathbf{x}^{k+1}) - \nabla \bar{g}(\mathbf{x}^\star)\big) - \alpha\big(\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\big) + \alpha p(\mathbf{z}^k)\Big\|^2 \tag{75}$$

$$\leq 4\|\mathbf{x}^{k+1/2} - \mathbf{x}^\star\|^2 + 4\alpha^2\|\nabla \bar{g}(\mathbf{x}^{k+1}) - \nabla \bar{g}(\mathbf{x}^\star)\|^2 + 4\alpha^2\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\|^2 + 4\alpha^2\|p(\mathbf{z}^k)\|^2 \tag{76}$$

$$\leq 4\|\mathbf{x}^{k+1/2} - \mathbf{x}^\star\|^2 + 4\tfrac{\alpha^2}{\beta_{\bar{g}}^2}\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|^2 + 4\alpha^2\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\|^2 + 4\alpha^2\|p(\mathbf{z}^k)\|^2 \tag{77}$$

$$\leq 4(1 + \tfrac{2\alpha^2}{\beta_{\bar{g}}^2})\|\mathbf{x}^{k+1/2} - \mathbf{x}^\star\|^2 + 4\alpha^2\|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^\star)\|^2 + 4\alpha^2(1 + \tfrac{2\alpha^2}{\beta_{\bar{g}}^2})\|p(\mathbf{z}^k)\|^2, \tag{78}$$

where the last line follows from

$$\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}^{k+1/2} - \mathbf{x}^\star - \alpha p(\mathbf{z}^k)\|^2 \leq 2\|\mathbf{x}^{k+1/2} - \mathbf{x}^\star\|^2 + 2\alpha^2\|p(\mathbf{z}^k)\|^2.$$

Since (78) is bounded by (72) up to a constant factor, we have established (63) for this case.

Case 3. $\bar{r}$ is strongly convex and $\bar{g}$ has Lipschitz gradient, in short, $\mu_{\bar{r}}\beta_{\bar{g}} > 0$, (and $\bar{f}$ still has Lipschitz gradient). In this case,

$$Q = c_1\Big(\|p(\mathbf{z}^k)\|^2 + \|\nabla \bar{f}(\mathbf{x}^{k+1/2}) - \nabla \bar{f}(\mathbf{x}^*)\|^2\Big) + \mu_{\bar{r}}\|\mathbf{x}^{k+1/2} - \mathbf{x}^*\|^2. \tag{79}$$

We still use (78), which is bounded by (79) up to a constant factor, we have established (63) for this case.

8 Conclusion and future work

In this paper we presented (PPG) and a variant (S-PPG). By discussing possible applications, we demonstrated how (PPG) expands the class of optimization problems that can be solved with a simple and scalable method. We proved convergence and demonstrated the effectiveness, especially in parallel computing, of the methods through computational experiments.

An interesting future direction is to consider cyclic and asynchronous variations of (PPG). Generally speaking, random coordinate updates can be computationally inefficient, and cyclic coordinate updates access the data more efficiently. (PPG) is a synchronous algorithm; at each iteration the $z_1, \dots, z_n$ are updated synchronously and the synchronization can cause inefficiency. Asynchronous updates avoid this problem.

References

[1] T. Arjevani and O. Shamir. Communication complexity of distributed convex learning and optimization. In *NIPS*, pages 1756–1764. 2015.

[2] J.-B. Baillon and G. Haddad. Quelques propriétés des opérateurs angle-bornés et n-cycliquement monotones. *Israel Journal of Mathematics*, 26(2):137–150, 1977.

[3] H. H. Bauschke and P. L. Combettes. *The Baillon-Haddad Theorem Revisited*, 17(3–4):781–787, 2010.

[4] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* 2011.

[5] D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2):163–195, 2011.

[6] P. Bianchi. Ergodic convergence of a stochastic proximal point algorithm. *SIAM Journal on Optimization*, 26(4):2235–2260, 2016.

[7] J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1111–1141, 2013.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[9] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[10] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.

[11] P. L. Combettes and J.-C. Pesquet. *Proximal Splitting Methods in Signal Processing*, pages 185–212. 2011.

[12] P. L. Combettes and J.-C. Pesquet. Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping. *SIAM Journal on Optimization*, 25(2):1221–1248, 2015.

[13] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[14] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[15] Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, Jun 2017.

[16] A. Defazio, F. Bach, and A. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654. 2014.

[17] A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *ICML*, volume 32, pages 1125–1133, 2014.

[18] W. Deng, M.-J. Lai, Z. Peng, and W. Yin. Parallel multi-block ADMM with $o(1/k)$ convergence. *Journal of Scientific Computing*, 2016.

[19] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[20] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40, 1976.

[21] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problmes de Dirichlet non linéaires. *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, 9(2):41–76, 1975.

[22] D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *KDD*, pages 387–396, 2015.

[23] H. Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[24] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. In *NIPS*. 2014.

[25] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

[26] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323. 2013.

[27] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, pages 543–550, 2010.

[28] B. Kulis and P. L. Bartlett. Implicit online learning. In *ICML*, pages 575–582, 2010.

[29] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. 2015.

[30] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

[31] N. Le Roux, M. Schmidt, and F. Bach. Stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, 2012.

[32] J. Liu and J. Ye. Moreau-Yosida regularization for grouped tree structure learning. In *NIPS*, pages 1459–1467. 2010.

[33] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *KDD*, pages 323–332. 2010.

[34] J. Mairal. Optimization with first-order surrogate functions. In *ICML*, pages 783–791, 2013.

[35] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[36] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski. Network flow algorithms for structured sparsity. In *NIPS*, pages 1558–1566. 2010.

[37] P. McCullagh and J. A. Nelder. *Generalized linear models.* 2nd edition, 1989.

[38] H. B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *AISTATS*, 2011.

[39] G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal*, 29(3):341–346, 1962.

[40] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

[41] A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS*, pages 1574–1582. 2014.

[42] G. Nowak, T. Hastie, J. R. Pollack, and R. Tibshirani. A fused lasso latent feature model for analyzing multi-sample aCGH data. *Biostatistics*, 12(4):776–791, 2011.

[43] D. P. Palomar and M. Chiang. Alternative distributed algorithms for network utility maximization: Framework and applications. *IEEE Transactions on Automatic Control*, 52(12):2254–2269, 2007.

[44] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[45] G. B. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *Journal of Mathematical Analysis and Applications*, 72(2):383–390, 1979.

[46] H. Raguet, J. Fadili, and G. Peyré. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226, 2013.

[47] F. Rapaport, E. Barillot, and J.-P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.

[48] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Herbert Robbins Selected Papers*, pages 111–135. 1985.

[49] E. K. Ryu and S. Boyd. Stochastic proximal iteration: A non-asymptotic improvement upon stochastic gradient descent. 2014.

[50] E. K. Ryu and S. Boyd. Primer on monotone operator methods. *Applied and Computational Mathematics*, 15(1):3–43, 2016.

[51] A. Salim, P. Bianchi, W. Hachem, and J. Jakubowicz. A stochastic proximal point algorithm for total variation regularization over large scale graphs. In *IEEE CDC*, pages 4490–4495, 2016.

[52] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, pages 1–30, 2016.

[53] S. Shalev-Shwartz. SDCA without duality, regularization and individual convexity. In *ICML*, pages 747–754, 2016.

[54] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14:567–599, 2013.

[55] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2016.

[56] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *ICML*, 2014.

[57] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

[58] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18, 2008.

[59] P. Toulis and E. M. Airoldi. Asymptotic and finite-sample properties of estimators based on stochastic gradients. *Annals of Statistics (To appear)*, 2017.

[60] P. Toulis, J. Rennie, and E. M. Airoldi. Statistical analysis of stochastic gradient methods for generalized linear models. In *ICML*, pages 667–675, 2014.

[61] M. Wang and D. P. Bertsekas. Incremental constraint projection methods for variational inequalities. *Mathematical Programming*, 150(2):321–363, 2015.

[62] M. Wang and D. P. Bertsekas. Stochastic first-order methods with random constraint projection. *SIAM Journal on Optimization*, 26(1):681–717, 2016.

[63] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[64] E. Yang, A. Lozano, and P. Ravikumar. Elementary estimators for high-dimensional linear regression. In *ICML*, pages 388–396, 2014.

[65] T. Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *NIPS*. 2013.

[66] G.-B. Ye and X. Xie. Split Bregman method for large scale fused lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569, 2011.

[67] L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *NIPS*, pages 352–360. 2011.

[68] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

[69] L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *NIPS*, pages 980–988. 2013.

[70] Y. Zhang and L. Lin. Disco: Distributed optimization for self-concordant empirical loss. In *ICML*, 2015.

[71] Y. Zhang, M. J. Wainwright, and J. C. Duchi. Communication-efficient algorithms for statistical optimization. In *NIPS*, pages 1502–1510. 2012.

[72] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, (6A):3468–3497, 2009.

[73] J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *KDD*, pages 1095–1103. 2012.