A Stochastic Decoupling Method for Minimizing the Sum of Smooth and Non-Smooth Functions

Konstantin Mishchenko[1] **Peter Richtárik**[1,2]
[1] King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[2] Moscow Institute of Physics and Technology, Dolgoprudny, Russia

Abstract

We consider the problem of minimizing the sum of three convex functions: i) a smooth function f in the form of an expectation or a finite average, ii) a non-smooth function g in the form of a finite average of proximable functions g_j, and iii) a proximable regularizer R. We design a variance reduced method which is able progressively learn the proximal operator of g via the computation of the proximal operator of a single randomly selected function g_j in each iteration only. Our method can provably and efficiently accommodate many strategies for the estimation of the gradient of f, including via standard and variance-reduced stochastic estimation, effectively decoupling the smooth part of the problem from the non-smooth part. We prove a number of iteration complexity results, including a general $\mathcal{O}(1/t)$ rate, $\mathcal{O}(1/t^2)$ rate in the case of strongly convex f, and several linear rates in special cases, including accelerated linear rate. For example, our method achieves a linear rate for the problem of minimizing a strongly convex function f under linear constraints under no assumption on the constraints beyond consistency. When combined with SGD or SAGA estimators for the gradient of f, this leads to a very efficient method for empirical risk minimization with large linear constraints. Our method generalizes several existing algorithms, including forward-backward splitting, Douglas-Rachford splitting, proximal SGD, proximal SAGA, SDCA, randomized Kaczmarz and Point-SAGA. However, our method leads to many new specific methods in special cases; for instance, we obtain the first randomized variant of the Dykstra's method for projection onto the intersection of closed convex sets.

1 Introduction

In this paper we address optimization problems of the form

$$\min_{x\in\mathbb{R}^d} F(x) := f(x) + \frac{1}{m}\sum_{j=1}^m g_j(x) + R(x), \tag{1}$$

where $f:\mathbb{R}^d\to\mathbb{R}$ is a smooth convex function, and $R, g_1, \dots, g_m\colon \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ are proper closed convex functions, admitting efficiently computable proximal operators[1]. We also assume throughout that $\operatorname{dom} F := \{x : F(x) < +\infty\} \neq \emptyset$ and, moreover, that the set of minimizers of (1), $\mathcal{X}^*$, is non-empty.

The main focus of this work is on how the difficult non-smooth term

$$g(x) := \frac{1}{m}\sum_{j=1}^m g_j(x) \tag{2}$$

[1] The proximal operator of function R is defined as $\operatorname{prox}_{\eta R}(x) := \operatorname{argmin}_{u\in\mathbb{R}^d}\left\{R(u) + \frac{1}{2\eta}\|u-x\|^2\right\}$.

Preprint. Under review.

should be treated in order to construct an efficient algorithm for solving the problem. We are specifically interested in the case when m is very large, and when the proximal operators of g and $g + R$ are impossible or prohibitively difficult to evaluate. We thus need to rely on splitting approaches which make calls to proximal operators of functions $\{g_j\}$ and R separately.

Existing methods for solving problem (1) can efficiently handle the case $m = 1$ only [1]. There were a few attempts to design methods capable of handling the general m case, such as [2, 44, 49] and [15]. None of the existing methods offer a linear rate for non-smooth problem except for random projection. In cases when sublinear rates are established, the assumptions on the functions g_j are very restrictive. For instance, the results in [2] are limited to Lipschitz continuous g_j only, and [15] assumes g_j to be strongly convex. This is very unfortunate because the majority of problems appearing in popular data science and machine learning applications lack these properties. For instance, if we want to find a minimum of a smooth function over the intersection of m convex sets, g_j will be characteristic functions of sets, which are neither Lipschitz nor strongly convex.

Applications. There is a long list of applications of the non-smooth finite-sum problem (1), including convex feasibility [3], constrained optimization [41], decentralized optimization [38], support vector machine [14], Dantzig selector [8], overlapping group LASSO [60], and fused LASSO. In Appendix A we elaborate in detail how these problems can be mapped to the general problem (1) (in particular, see Table 3).

Variance reduction. Stochastic variance reduction methods are a major breakthrough of the last decade, whose success started with the Stochastic Dual Coordinate Ascent (SDCA) method [52] and the invention of the Stochastic Average Gradient (SAG) method [48]. Variance reduction has attracted enormous attention and now its reach covers strongly convex, convex and non-convex [32] stochastic problems. Despite being originally developed for finite-sum problems, variance reduction was shown to be applicable even to problems with f expressed as a general expectation [31, 40]. Further generalizations and extensions include variance reduction for minimax problems [42], coordinate descent in the general R case [25], and minimization with arbitrary sampling [24]. However, very little is known about variance reduction for non-smooth finite sum problems.

2 Summary of Contributions

The departure point of our work is the observation that there is a class of non-smooth problems for which variance reduction is *not* required; these are the linear feasibility problems: given $\mathbf{A} \in \mathbb{R}^{m\times d}$ and $b \in \mathbb{R}^m$, find $x \in \mathbb{R}^d$ such that $\mathbf{A}x = b$. Assuming the system is consistent, this problem can be cast as an instance of (1), with $R \equiv 0$, $f(x) = \frac{1}{2}\|x\|^2$ and g_j corresponding to the characteristic function of the j-th equation in the system. Efficient SGD methods (or equivalently, randomized projection methods) with linear convergence rates were recently developed for this problem [23, 46, 56], as well as accelerated variants [56, 46, 22] whose linear rate yields a quadratic improvement in the iteration complexity. However, it is *not* known whether these or similar linear rates could be obtained when one considers f to be an arbitrary smooth and strongly convex function. While our work was originally motivated by the quest to answer this question, and we answer in the affirmative, we were able to build a much more general theory, as we explain below.

We now summarize some of the most important contributions of our work:

First variance reduction for g. We propose a *variance reduction* strategy for progressively approximating the proximal operator of the average of a large number of *non-smooth* functions g_j via only evaluating the proximal operator of a single function g_j in each iteration. That is, unlike existing approaches, we are able to treat the difficult term (2) for any m. Combined with a gradient-type step in f (we allow for multiple ways in which the gradient estimator is built; more on that below), and a proximal step for R, this leads to a *new and remarkably efficient method* (Algorithm 1) for solving problem (1).

Compatibility with any gradient estimator for f. Our variance reduction scheme for the non-smooth term g is *decoupled* from the way we choose to construct gradient estimators for f. This allows us to use the most efficient and suitable estimators depending on the structure of f. In this regard, two cases are of particular importance: i) $f = \mathbb{E}_\xi f_\xi$, where $f_\xi \colon \mathbb{R}^d \to \mathbb{R}$ is almost surely convex and smooth, and ii) $f = \frac{1}{n}\sum_i f_i$, where $\{f_i\}$ are convex and smooth. In case i) one may consider the standard stochastic gradient estimator $\nabla f_{\xi^k}(x^k)$, or a mini-batch variant thereof, and

f	g_j	R	η	Method	Comment
$f_1 = f, n = 1$	0	R	$< {}^2/_L$	Forward-Backward	[39, 12]
0	$g_1 = g, m = 1$	R	any	Douglas-Rachford	[33]
$\mathbb{E}_\xi f_\xi$	0	R	$\le {}^1/_{4L}$	Proximal SGD	[19]
${}^1/_n \sum_i f_i$	0	R	$\le {}^1/_{5L}$	Proximal SAGA	[16]
${}^1/_2 \|x - x^0\|^2$	g_j	0	$\eta = {}^1/_m$	SDCA	[52]
${}^1/_2 \|x - x^0\|^2$	$\chi_{\mathcal{C}_j}$	0	$\eta = {}^1/_m$	Randomized Dykstra's algorithm	NEW
${}^1/_2 \|x - x^0\|^2$	$\chi_{\{x: a_j^\top x = b_j\}}$	0	$\eta = {}^1/_m$	Randomized Kaczmarz method	[28, 53]
0	g_j	0	any	Point-SAGA	[15]
$f_1 = f, n = 1$	$g_1 = g, m = 1$	R	$< {}^2/_L$	Condat-Vũ algorithm	[58, 13]

Table 1: Selected special cases of our method. For Dykstra's algorithm, $\mathcal{C}_1, \dots, \mathcal{C}_m$ are closed convex sets; and we wish to find projection onto their intersection. Randomized Kaczmarz is a special case for linear constraints (i.e. $\mathcal{C}_j = \{x : a_j^\top x = b_j\}$). We do not prove convergence under the same assumptions as Point-SAGA as they require strong convexity and smoothness of each g_j, but it is still a special case.

in case ii) one may consider the batch gradient $\nabla f(x^k)$ if n is small, or a variance-reduced gradient estimator, such as SVRG [27, 30] or SAGA [16, 45], if n is large. *Our general analysis allows for any estimator to be used as long as it satisfies a certain technical assumption (Assumption 2).* In particular, to illustrate the versatility of our approach, we show that this assumption holds for estimators used by Gradient Descent, SVRG, SAGA and over-parameterized SGD. We also claim without a proof that a variant of coordinate descent [25] satisfies our assumption, but leave it for future work.

Future-proof design. Our analysis is compatible with a wide array of other estimators of the gradient of f beyond the specific ones listed above. Therefore, new specific variants of our generic method for solving problem (1) can be obtained in the future by marrying any such new estimators with our variance reduction strategy for the non-smooth finite sum term g.

Special cases. Special cases of our method include randomized Kaczmarz method [28, 53], Douglas-Rachford splitting [33], forward-backward splitting [39, 12], a variant of SDCA [52], and Point-SAGA [15]. Also, we obtain the first *randomized* variant of the famous Dykstra's algorithm [20] for projection onto the intersection of convex sets. These special cases are summarized in Table 1.

Sublinear rates. We first prove convergence of the iterates to the solution set in a Bregman sense, without quantifying the rate (see Appendix F.3). Next, we establish $\mathcal{O}\left({}^1/_t\right)$ rate with constant stepsizes under no assumption on problem (1) beyond the existence of a solution and a few technical assumptions (see Thm 1). The rate improves to $\mathcal{O}\left({}^1/_{t^2}\right)$ once we assume strong convexity of f, and allow for carefully designed decreasing stepsizes (see Thm 2).

Linear rate in the non-smooth case with favourable data. Consider the special case of (1) with f being strongly convex, $R \equiv 0$ and $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$, where $\phi_j : \mathbb{R}^{d_j} \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions, and $\mathbf{A}_j \in \mathbb{R}^{d \times d_j}$ are given (data) matrices:

$$\min_{x \in \mathbb{R}^d} f(x) + \frac{1}{m} \sum_{j=1}^m \phi_j(\mathbf{A}_j^\top x). \tag{3}$$

If the smallest eigenvalue of $\mathbf{A}^\top \mathbf{A}$ is positive, i.e. $\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) > 0$, where $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_m] \in \mathbb{R}^{d \times \sum_j d_j}$, then our method converges linearly (see Thm 4; and note that this can only happen if $\sum_j d_j \le d$). Moreover, picking j with probability proportional to $\|\mathbf{A}_j\|$ is optimal (Cor 2). In the special case when $\phi_j(y) = \chi_{\{x \,:\, \mathbf{A}_j^\top x = b_j\}}(x)$ for some vectors $b_1 \in \mathbb{R}^{d_1}, \dots, b_m \in \mathbb{R}^{d_1}$ i.e. if we are minimizing a strongly convex function under a linear constraint,

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) \ : \ \mathbf{A}^\top x = b \right\},$$

Problem	f scvx	g_j smooth	Method for f	Rate	Theorem
$\mathbb{E}f_\xi(x) + \frac{1}{m}\sum_{j=1}^m g_j(x) + R(x)$	✗	✗	SGD	$\mathcal{O}\left(1/\sqrt{t}\right)$	Cor. 1
	✓	✗		$\mathcal{O}\left(1/t\right)$	3
$\frac{1}{n}\sum_{i=1}^n f_i(x) + \frac{1}{m}\sum_{j=1}^m g_j(x) + R(x)$	✗	✗	GD, SVRG and SAGA	$\mathcal{O}\left(1/t\right)$	1
	✓	✗		$\mathcal{O}\left(1/t^2\right)$	2
	✓	✓		Linear	6
$\frac{1}{n}\sum_{i=1}^n f_i(x) + \frac{1}{m}\sum_{j=1}^m \phi_j(\mathbf{A}_j^\top x)$	✓	✗		Linear	4, 5

Table 2: Summary of iteration complexity results. We assume by default that all functions are convex, but provide different rates based on whether f is strongly convex (scvx) and whether $g_1, \dots, g_m$ are smooth functions, which is represented by the check marks.

then the rate is linear even if $\mathbf{A}^\top\mathbf{A}$ is not positive definite[2]. The rate will depend on $\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})$, i.e. the smallest positive eigenvalue (see Thm 5).

Linear and accelerated rate in the smooth case. If $g_1, \dots, g_m$ are smooth functions, the rate is linear (see Thm 6). If m is big enough, then it is also *accelerated* (Cor 3). A summary of our iteration complexity results is provided in Table 2.

Related work. The problems that we consider recently received a lot of attention. However, we are the first to show linear convergence on non-smooth problems. $\mathcal{O}\left(1/t\right)$ convergence with stochastic variance reduction was obtained in [49] and [44], although both works do not have $\mathcal{O}\left(1/t^2\right)$ rate as we do. On the other hand, works such as [61, 10] managed to prove $\mathcal{O}\left(1/t^2\right)$ convergence, but only with all functions from f and g used at every iteration. Stochastic $\mathcal{O}\left(1/t^2\right)$ for constrained minimization can be found in [36]. There is also a number of works that consider parallel [17] ($\mathcal{O}\left(1/t\right)$ rate) and stochastic [63, 35] variants of ADMM, which work with one non-smooth term composed with a linear transformation. To show linear convergence they require matrix in the transformation to be positive-definite. Variance reduced ADMM for compositions, which is an orthogonal direction to ours, was considered in [59]. There is a method for non-smooth problems with $f \equiv 0$ and proximal operator preconditioning that was analyzed in detail in [11], we discuss the relation to it in Appendix **??**. Many methods were designed to work with non-smooth functions in parallel only, and one can obtain more of them from three-operator splitting methods such as the Condat-Vũ algorithm [58, 13]. Several works obtained linear convergence for smooth g [18, 42]. Coordinate descent methods for two non-smooth functions were considered in [1]. Work [62] designed a method for (1) assuming that the variance of $\nabla f_\xi(x)$ is uniformly bounded over all possible x, which we do not require, and it has to evaluate proximal operators of all terms in g at each iteration, making it the bottleneck of the algorithm.

3 Preliminaries

Convexity and smoothness. A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is called *μ-strongly convex* if $f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x-y\|^2$ for all $x, y \in \mathbb{R}^d$. It is called *convex* if this holds with $\mu = 0$. A convex function $f : \mathbb{R}^d \to \mathbb{R}$ is called *L-smooth* if it is differentiable and satisfies $f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x-y\|^2$ for all $x, y \in \mathbb{R}^d$.

Bregman divergence. To simplify the notation and proofs, it is convenient to work with Bregman divergences. The Bregman divergence associated with a differentiable convex function f is the function $D_f(x, y) := f(x) - f(y) - \langle \nabla f(x), x - y\rangle$. It is important to note that the *Bregman divergence* of a convex function is always non-negative and is a (non-symmetric) notion of "distance" between x and y. For $x^* \in \mathcal{X}^*$, the quantity $D_f(x, x^*)$ serves as a generalization of the functional gap $f(x) - f(x^*)$ in cases when $\nabla f(x^*) \neq 0$.

Useful inequalities related to convexity, strong convexity and smoothness are summarized in Appendix D. We will make the following assumption related to optimality conditions.

[2] By $\chi_{\mathcal{C}}(x)$ we denote the characteristic function of the set $\mathcal{C}$, defined as follows: $\chi_{\mathcal{C}}(x) = 0$ if $x \in \mathcal{C}$ and $\chi_{\mathcal{C}}(x) = +\infty$ if $x \notin \mathcal{C}$

Algorithm 1 Stochastic Decoupling Method (SDM).

```
Input: Stepsize $\eta$, initial vectors $x^0$, $y_1^0, \dots, y_m^0$, oracle that gives gradient estimates
for $t = 0, 1, \dots$ do
    Produce a gradient estimate $v^t$, and set $y^t = \frac{1}{m}\sum_{k=1}^m y_k^t$
    $z^t = \mathrm{prox}_{\eta R}(x^t - \eta v^t - \eta y^t)$
    Sample $j$ from $\{1, \dots, m\}$ with probabilities $\{p_1, \dots, p_m\}$ and set $\eta_j = \frac{\eta}{m p_j}$
    $x^{t+1} = \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t)$
    $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1})$
end for
```

Assumption 1. *There exists $x^* \in \mathcal{X}^*$ and vectors $y_1^* \in \partial g_1(x^*), \dots, y_m^* \in \partial g_m(x^*)$ and $r^* \in \partial R(x^*)$ such that $\nabla f(x^*) + \frac{1}{m}\sum_{j=1}^m y_j^* + r^* = 0$.*

Throughout the paper, we will assume that some x^* and $y_1^*, \dots, y_m^*$ satisfying Assumption 1 are fixed and all statements relate to these objects. We will denote $y^* := \frac{1}{m}\sum_{j=1}^m y_j^*$. A commentary and further details related to this assumption can be found in Appendix E.

4 The Algorithm

Our method is very general and can work with different types of gradient update. One only needs to have for each x^t an estimate of the gradient v^t such that $\mathbb{E}v^t = \nabla f(x^t)$ plus an additional assumption about its variance. We also maintain an estimate y^t of full proximal step with respect to g, which allows us to make an intermediate step $z^t = \mathrm{prox}_{\eta R}(x^t - \eta v^t - \eta y^t)$. The key idea of this work is then to combine it with variance reduction in the non-smooth part. In fact, it mimics variance reduction step from [15], which was motivated by the SAGA algorithm [16]. Essentially, the expression above for z^t does not allow for update of y^t, so we do one more step,

$$x^{t+1} = \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t).$$

This can additionally be rewritten using the identity $\mathrm{prox}_{\eta g}(x) \in x - \eta \partial g(\mathrm{prox}_{\eta g}(x))$ as

$$x^{t+1} \in x^t - \eta(v^t + \partial R(z^t) + y^t) - \eta_j(\partial g_j(x^{t+1}) - y_j^t) \approx \mathrm{prox}_{\eta(R+g)}(x^t - \eta \nabla f(x^t)).$$

To make sure that the approximation works, we want to make y_j^t be close to $\partial g_j(x^{t+1})$, which we do not know in advance. However, we do it in hindsight by updating y_j^{t+1} with a particular subgradient from $\partial g_j(x^{t+1})$, namely $y_j^{t+1} = \frac{1}{\eta_j}(z^t + \eta_j y_j^t - \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t)) \in \partial g_j(x^{t+1})$.

We also need to accurately estimate $\nabla f(x^t)$, and there several options for this. The simplest choice is setting $v^t = \nabla f(x^t)$. Often this is too expensive and one can instead construct v^t using a variance reduction technique, such as SAGA [16] (see Algorithm 2). To a reader familiar with Fenchel duality, it might be of some interest that there is an explanation of our ideas using the dual.[3]

5 Gradient Estimators

Since we want to have analysis that puts many different methods under the same umbrella, we need an assumption that is easy to satisfy. In particular, the following will fit our needs.

Assumption 2. *Let $w^t := x^t - \eta v^t$ and $w^* := x^* - \eta \nabla f(x^*)$. We assume that the oracle produces v^t and (potentially) updates some other variables in such a way that for some constants $\eta_0 > 0$, $\omega > 0$ and non-negative sequence $\{\mathcal{M}^t\}_{t=0}^{+\infty}$, such that the following holds for any $\eta \le \eta_0$:*

[3] Indeed, problem (1) can be recast into $\min_x \max_{y_1,\dots,y_m} f(x) + R(x) + \frac{1}{m}\sum_{j=1}^m x^\top y_j - \frac{1}{m} g_j^*(y_j)$, where g_j^* is the Fenchel conjugate of g_j. Then, the proximal gradient step in x would be $z = \mathrm{prox}_{\eta R}\left(x - \eta \nabla f(x) - \eta \frac{1}{m} y_j\right)$. In contrast, our update in y_j is a proximal block coordinate ascent step, so the overall process is akin to proximal alternating gradient descent-ascent. However, this is not how we developed nor analyze the method, ans so this should not be seen as a formal explanation.

Algorithm 2 SAGA Oracle.

```
Input: $x^t$, table of past gradients $\nabla f_1(u_1^t), \dots, \nabla f_n(u_n^t)$ and their average $\alpha^t$
  Sample subset $S$ from $\{1, \dots, n\}$ of size $\tau$
  $v^t = \frac{1}{\tau}\sum_{i \in S}\left(\nabla f_i(x^t) - \nabla f_i(u_i^t)\right) + \alpha^t$
  For all $i \in S$ update $\nabla f_i(u_i^{t+1})$ with $u_i^{t+1} = x^t$
  return $v^t$
```

(a) If f is convex, then $\mathbb{E}\|w^t - w^*\|^2 + \mathcal{M}^{t+1} \le \|x^t - x^*\|^2 - \omega\eta D_f(x^t, x^*) + \mathcal{M}^t$.

(b) If f is μ-strongly convex, then either $\mathcal{M}^t = 0$ for all t or there exists $\rho > 0$ such that

$$\mathbb{E}\|w^t - w^*\|^2 + \mathcal{M}^{t+1} \le (1 - \omega\eta\mu)\|x^t - x^*\|^2 + (1 - \rho)\mathcal{M}^t.$$

We note that we could easily make a slightly different assumption to allow for a strongly convex R, but this would be at the cost of analysis clarity. Since the assumption above is already quite general, we choose to stick to it and claim without a proof that in the analysis it is possible to transfer strong convexity from f to R.

Another observation is that part (a) of Assumption 2 implies its part (b) with $\omega/2$. However, to achieve tight bounds for gradient descent we need to consider them separately.

Lemma 1 (Proof in Appendix F.1). *If f is convex, Gradient Descent satisfies Assumption 2(a) with any $\eta_0 < {}^2/L$, $\omega = 2 - \eta_0 L$ and $\mathcal{M}^t = 0$. If f is μ-strongly convex, Gradient Descent satisfies Assumption 2(b) with $\eta_0 = \frac{2}{L+\mu}$, $\omega = 1$ and $\mathcal{M}^t = 0$.*

Since $\mathcal{M}^t = 0$ for Gradient Descent, one can ignore ρ in the convergence results or treat it as $+\infty$.

Lemma 2 (Proof in Appendix F.11). *In SVRG and SAGA, if f_i is L-smooth and convex for all i, Assumption 2(a) is satisfied with $\eta_0 = {}^1/6L$, $\omega = {}^1/3$ and $\mathcal{M}^t = \frac{3\eta^2}{n}\sum_i \mathbb{E}\|\nabla f_i(u_i^t) - \nabla f_i(x^*)\|^2$, where in SVRG $u_i^t = u^t$ is the reference point of the current loop, and in SAGA u_i^t is the point whose gradient is stored in memory for function f_i. If f is also strongly convex, then Assumption 2 holds with $\eta_0 = {}^1/5L$, $\omega = 1$, $\rho = {}^1/3n$ and the same $\mathcal{M}^t$.*

Lemma 3 (Proof in Appendix F.12). *Assume that at an optimum x^* the variance of stochastic gradients is finite, i.e. $\sigma_*^2 := \mathbb{E}_\xi\|\nabla f_\xi(x^*) - \nabla f(x^*)\|^2 < +\infty$. Then, SGD that terminates after at most t_0 iterations satisfies Assumption 2(a) with $\eta_0 = \frac{1}{4L}$, $\omega = 1$ and $\rho = 0$. In this case, sequence $\{\mathcal{M}^t\}_{t=0}^{t_0}$ is given by $\mathcal{M}^t = 2\eta^2(t_0 - t)\sigma_*^2$. If f is strongly convex and $\sigma_* = 0$, it satisfies Assumption 2(b) with $\eta_0 = \frac{1}{2L}$, $\omega = 1$ and $\mathcal{M}^t = 0$.*

There are two important cases for SGD. If the model is overparameterized, i.e. $\sigma_* \approx 0$, we get almost the same guarantees for SGD as for GD. If, $\sigma_* \gg 0$, then one needs to choose $\eta = \mathcal{O}\left({}^1/(\sqrt{t_0}L)\right)$ in order to keep $\mathcal{M}^0$ away from $+\infty$. This effectively changes the $\mathcal{O}\left({}^1/t\right)$ rate to $\mathcal{O}\left({}^1/\sqrt{t}\right)$, see Cor 1. Moreover, obtaining a $\mathcal{O}\left({}^1/t\right)$ rate for strongly convex case requires a separate proof.

6 Convergence

Let $\gamma := \min_{j=1,\dots,m}\frac{1}{\eta_j L_j}$, where $L_j \in \mathbb{R} \cup \{+\infty\}$ is the smoothness constant of g_j, in most cases giving $L_j = +\infty$ and $\gamma = 0$. Tho goal of our analysis is to show that with introducing new term in the Lyapunov function, $\mathcal{Y}^t := (1 + \gamma)\sum_{k=1}^m \eta_k^2 \mathbb{E}\|y_k^t - y_k^*\|^2$, the convergence is not significantly hurt. This term will be always incorporated in the full Lyapunov function defined as

$$\mathcal{L}^t := \mathbb{E}\|x^t - x^*\|^2 + \mathcal{M}^t + \mathcal{Y}^t,$$

where $\mathcal{M}^t$ is from Assumption 2. In the proof of $\mathcal{O}\left({}^1/t^2\right)$ rate we will use decreasing stepsizes and $\mathcal{Y}^t$ will be defined slightly differently, but except for this, it is going to be the same Lyapunov function everywhere.

6.1 $\mathcal{O}({}^1/t)$ convergence for general convex problem

Theorem 1 (Proof in Appendix F.4). *Assume f is L-smooth and μ-strongly convex, $g_1, \dots, g_m, R$ are proper, closed and convex. If we use a method for generating v^t which satisfies Assumption 2* *and $\eta \le \eta_0$, then*

$$\mathbb{E} D_f(\overline{x}^t, x^*) \le \frac{1}{\omega\eta t}\mathcal{L}^0,$$

where $\mathcal{L}^0 := \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m \eta_k^2 \|y_k^0 - y_k^*\|^2$ and $\overline{x}^t := \frac{1}{t}\sum_{k=0}^{t-1} x^k$.

If $R \equiv 0$ and $g_j \equiv 0$ for all j, then this transforms into $\mathcal{O}(1/t)$ convergence of $f(x^t) - \min f(x)$, which is the correct rate.

The next result takes care of the case when SGD is used, which requires special consideration.

Corollary 1. *If we use SGD for t iterations with constant stepsize, the method converges to a neighborhood of radius $\mathcal{M}^0/\eta t = 2\eta\sigma_*^2$. If we choose the stepsize $\eta = \Theta\left(1/(L\sqrt{t})\right)$, then $2\eta\sigma_*^2 = \mathcal{O}(1/\sqrt{t})$, and we recover $\mathcal{O}\left(1/\sqrt{t}\right)$ rate.*

6.2 $\mathcal{O}(1/t^2)$ convergence for strongly convex f

In this section, we consider a variant of Algorithm 1 with time-varying stepsizes,

$$z^t = \mathrm{prox}_{\eta^{t+1}R}(x^t - \eta^t v^t - \eta^t y^t), \qquad x^{t+1} = \mathrm{prox}_{\eta_j^t g_j}(z^t + \eta_j^t y_j^t).$$

Theorem 2 (Proof in Appendix F.5)**.** *Consider updates with time-varying stepsizes, $\eta^t = \frac{2}{\mu\omega(a+t)}$ and $\eta_j^t = \frac{\eta^t}{mp_j}$ for $j = 1, \ldots, m$, where $a \ge 2\max\left\{\frac{1}{\omega\mu\eta_0}, \frac{1}{\rho}\right\}$. Then*

$$\mathbb{E}\|x^t - x^*\|^2 \le \frac{a^2}{(t+a-1)^2}\mathcal{L}^0,$$

where $\mathcal{L}^0 = \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m (\eta_k^0)^2\|y_k^0 - y_k^*\|^2$.

This improves upon $\mathcal{O}(1/t)$ convergence proved in [15] under similar assumptions and matches the bound in [11].

In Cor 1 we obtained $\mathcal{O}(1/\sqrt{t})$ rate for SGD with $\sigma_* \neq 0$. It is not surprising that the rate is worse as it is so even with $g \equiv 0$. For standard SGD we are able to improve the guarantee above to $\mathcal{O}(1/t)$ when the objective is strongly convex.

Theorem 3 (Proof in Appendix F.6)**.** *Assume f is μ-strongly convex, f_ξ is almost surely convex and L-smooth. Let the update be produced by SGD, i.e. $v^t = \nabla f_{\xi^t}(x^t)$, and let us use time-varying stepsizes $\eta^{t-1} = \frac{2}{a+\mu t}$ with $a \ge 4L$. Then*

$$\mathbb{E}\|x^t - x^*\|^2 \le \frac{8\sigma_*^2}{\mu(a+\mu t)} + \frac{a^2}{(a+\mu t)^2}\mathcal{L}^0.$$

6.3 Linear convergence for linear non-smoothness

We now provide two linear convergence rates in the case when $R \equiv 0$ and $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$.

Theorem 4 (Proof in Appendix F.7)**.** *Assume that f is μ-strongly convex, $R \equiv 0$, $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$ for $j = 1, \ldots, m$ and take a method satisfying Assumption 2 with $\rho > 0$. Then, if $\eta \le \eta_0$,*

$$\mathbb{E}\|x^t - x^*\|^2 \le \left(1 - \min\{\rho, \omega\eta\mu, \rho_A\}\right)^t \mathcal{L}^0,$$

where $\rho_A := \lambda_{\min}(\mathbf{A}^\top\mathbf{A})\min_j \left(p_j/\|\mathbf{A}_j\|\right)^2$, and $\mathcal{L}^0 := \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m \eta_k^2\|y_k^0 - y_k^*\|^2$.

Corollary 2. *If oracle from Algorithm 2 (SAGA) is used with probabilities $p_j \propto \|\mathbf{A}_j\|$, then to get $\mathbb{E}\|x^t - x^*\|^2 \le \varepsilon$, it is enough to run it for*

$$\mathcal{O}\left(\left(n + \frac{L}{\mu} + \frac{\|\mathbf{A}\|_{2,1}^2}{\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}\right)\log\frac{1}{\varepsilon}\right)$$

iterations.

Now let us show that this can be improved to depend only on positive eigenvalues if the problem is linearly constrained.

Theorem 5 (Proof in Appendix F.8)**.** *Under the same assumptions as in Thm 4 and assuming, in addition, that* $g_j = \chi_{\{x:\mathbf{A}_j^\top x=b_j\}}$ *it holds* $\mathbb{E}\|x^t - x^*\|^2 \le (1 - \min\{\rho, \omega\eta\mu, \rho_A\})\mathcal{L}^0$ *with* $\rho_A = \lambda_{\min}^+(\mathbf{A}^\top \mathbf{A}) \min_j \left({}^{p_j}/_{\|\mathbf{A}_j\|}\right)^2$*, i.e.* ρ_A *depends only on the smallest positive eigenvalue of* $\mathbf{A}^\top\mathbf{A}$.

One implication of Thm 5 is that just by taking a solver such as SVRG we immediately obtain a method for decentralized optimization that will converge linearly. Furthermore, if the problem is ill-conditioned or the communication graph is well conditioned, the leading term is still ${}^{L}/_{\mu}$, meaning that the rate for decentralized method is the same as for centralized up to constant factors. In Appendix F.8, we also give a version of our method specialized to the linearly constrained problem that requires only one extra vector, y^t.

6.4 Linear convergence if all g_j are smooth

Theorem 6 (Proof in Appendix F.9)**.** *Assume that* f *is* L*-smooth and* μ*-strongly convex,* g_j *is* L_j*-smooth for all* j*, Assumption 2(b) is satisfied and* $\eta \le \eta_0$*. Then, Algorithm 1 converges as*

$$\mathbb{E}\|x^t - x^*\|^2 \le \left(1 - \min\left\{\omega\eta\mu, \rho, \frac{\gamma}{m(1+\gamma)}\right\}\right)^t \mathcal{L}^0,$$

where $\gamma \coloneqq \min_{j=1,\ldots,m}(\eta_j L_j)^{-1}$.

Based on the theorem above, we suggest to choose probabilities p_j to maximize γ, which can be done by using $p_j \propto L_j$. If $p_j = \frac{L_j}{\sum_{k=1}^m L_k}$, then $\gamma = \min_{j=1,\ldots,m} \frac{mp_j}{\eta L_j} = \frac{1}{\eta \overline{L}}$ with $\overline{L} \coloneqq \frac{1}{m}\sum_{j=1}^m L_j$.

Corollary 3 (Proof in Appendix F.10)**.** *Choose as solver for* f *SVRG or SAGA without minibatching, which satisfy Assumption 2 with* $\eta_0 = {}^1/_{5L}$ *and* $\rho = {}^1/_n$*, and consider for simplicity situation where* $L_1 = \cdots = L_m \coloneqq L_g$ *and* $p_1 = \cdots = p_m$*. Define* $\eta_{best} \coloneqq (\omega\mu m L_g)^{-1/2}$*, and set the stepsize to* $\eta = \min\{\eta_0, \eta_{best}\}$*. Then the complexity to get* $\mathbb{E}\|x^t - x^*\|^2 \le \varepsilon$ *is*

$$\mathcal{O}\left(\left(n + m + \frac{L}{\mu} + \sqrt{\frac{mL_g}{\mu}}\right)\log\frac{1}{\varepsilon}\right).$$

Notably, the rate in Cor 3 is accelerated in g, suggesting that the proposed update is in some cases optimal. Moreover, if m becomes large, the last term is dominating everything else meaning that acceleration in f might not be needed at all.

7 Implementation Details and Experiments

Randomly generated linear system. In this experiment, we first generate a matrix with independent Gaussian entries of zero mean and scale ${}^1/_{\sqrt{d}}$, where $d = 100$, and after that we set $\mathbf{W} \in \mathbb{R}^{d\times d}$ to be the product of the generated matrix with itself plus identity matrix with coefficient 10^{-2} to make sure $\mathbf{W}$ is positive definite. We also generated a random vector $x^* \in \mathbb{R}^d$ and took $b = \mathbf{W}x^*$. The problem is to solve $\mathbf{W}x = b$, or, equivalently, to minimize $\|\mathbf{W}x - b\|^2$. We made this choice because it makes estimation of the parameters of accelerated Sketch-and-Project easier.

To run our method, we choose

$$f(x) = \frac{1}{2}\|x\|^2$$

and

$$g_j(x) = \chi_{\{x:w_j^\top x=b_j\}}(x), \quad j = 1, \ldots, d,$$

where $\chi_{\{x\,:\,w_j^\top x=b_j\}}(x)$ is the characteristic function, whose value is 0 if $w_j^\top x = b_j$ and $+\infty$ otherwise. Then, the proximal operator of g_j is the projection operator onto the corresponding constraint. We found that the choice of stepsize is important for fast convergence and that the value approximately equal $1.3 \cdot 10^{-4} \ll 1 = {}^2/_{(L+\mu)}$ led to the best performance for this matrix.

We compare our method to the accelerated Sketch-and-Project method of [22] using optimal parameters. The other method that we consider is classic Kaczmarz method that projects onto randomly chosen constraint. We run all methods with uniform sampling.

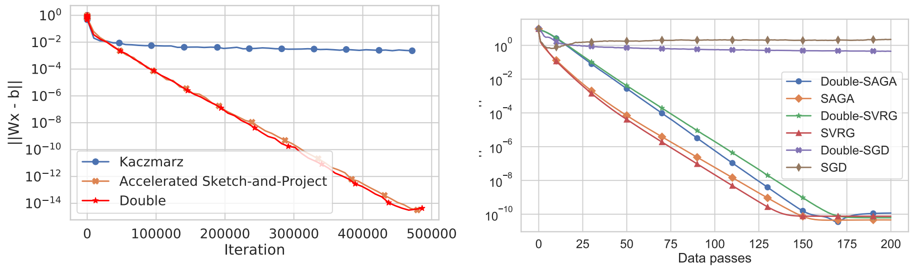


Figure 1: Left: convergence of the Stochastic Decoupling method, Kaczmarz and accelerated Kaczmarz of [22] when solving $\mathbf{W}x = b$ with random positive-definite $\mathbf{W} \in \mathbb{R}^{d\times d}$, where $d = 100$. It is immediate to observe that the method we propose performs on a par with the accelerated Sketch-and-Project. Right: linear regression with A9a dataset from LIBSVM [**?**] with first 50 observation used as linear constraints. We compare convergence of SVRG with full projections (labeled as 'SVRG') to the same method combined with Algorithm 1 (labeled as 'Double-').

Linear regression with linear constraints. We took A9a dataset from LIBSVM and ran ℓ_2-regularized linear regression, using first 50 observations of the dataset as tough constraints. We compare iteration complexity to precise projection onto all constraints and observe that it takes almost the same number of iterations, although stochastic iterations are significantly cheaper. For each method we chose minibatch of size 20 and stepsizes of order $^1/_L$ for all methods.

More experiments are provided in Appendix G.

References

[1] Ahmet Alacaoglu, Quoc Tran Dinh, Olivier Fercoq, and Volkan Cevher. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Advances in Neural Information Processing Systems*, pages 5852–5861, 2017.

[2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

[3] HH Bauschke and JM Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, 1996.

[4] Amir Beck. *First order methods in optimization*. MOS-SIAM Series on Optimization, 2017.

[5] Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

[6] Dimitri P Bertsekas and Athena Scientific. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

[7] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[8] Emmanuel Candès and Terence Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[9] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[10] Volkan Cevher, Băng Công Vũ, and Alp Yurtsever. Stochastic forward-Douglas-Rachford splitting for monotone inclusions. Technical report, Springer International Publishing, 2018.

[11] Antonin Chambolle, Matthias J Ehrhardt, Peter Richtárik, and Carola-Bibiane Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

[12] Patrick Louis Combettes and jean-Christophe Pesquet. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, chapter Proximal splitting methods in signal processing, pages 185–212. Springer Optimization and Its Applications. Springer, 2011.

[13] Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

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

[15] Aaron Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems*, pages 676–684, 2016.

[16] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[17] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multi-block admm with o (1/k) convergence. *Journal of Scientific Computing*, 71(2):712–736, 2017.

[18] Simon S Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. *arXiv preprint arXiv:1802.01504*, 2018.

[19] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934, 2009.

[20] Richard L Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[21] Matthias J Ehrhardt, Pawel Markiewicz, Antonin Chambolle, Peter Richtárik, Jonathan Schott, and Carola-Bibiane Schönlieb. Faster pet reconstruction with a stochastic primal-dual hybrid gradient method. In *Wavelets and Sparsity XVII*, volume 10394, page 103941O. International Society for Optics and Photonics, 2017.

[22] Robert M Gower, Filip Hanzely, Peter Richtárik, and Sebastian U Stich. Accelerated stochastic matrix inversion: general theory and speeding up bfgs rules for faster second-order optimization. In *Advances in Neural Information Processing Systems*, pages 1619–1629, 2018.

[23] Robert M Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[24] Robert M Gower, Peter Richtárik, and Francis Bach. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *arXiv preprint arXiv:1805.02632*, 2018.

[25] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: Variance reduction via gradient sketching. In *Advances in Neural Information Processing Systems*, pages 2082–2093, 2018.

[26] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313, 2015.

[27] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

[28] S Kaczmarz. Angenaherte auflosung von systemen linearer glei-chungen. *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, pages 355–357, 1937.

[29] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ell_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[30] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. *arXiv preprint arXiv:1901.08689*, 2019.

[31] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156, 2017.

[32] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2348–2358, 2017.

[33] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[34] Ji Liu and Stephen Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.

[35] Yuanyuan Liu, Fanhua Shang, and James Cheng. Accelerated variance reduced stochastic admm. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[36] Konstantin Mishchenko and Peter Richtárik. A stochastic penalty model for convex and nonconvex optimization with big constraints. *arXiv preprint arXiv:1810.13387*, 2018.

[37] Ion Necoara, Peter Richtárik, and Andrei Patrascu. Randomized projection methods for convex feasibility problems: conditioning and convergence rates. *arXiv:1801.04873*, 2018.

[38] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[39] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[40] Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *arXiv preprint arXiv:1811.10105*, 2018.

[41] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag New York, 2006.

[42] Balamurugan Palaniappan and Francis Bach. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, pages 1416–1424, 2016.

[43] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[44] Fabian Pedregosa, Kilian Fatras, and Mattia Casotto. Proximal splitting meets variance reduction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1–10, 2019.

[45] Xun Qian, Zheng Qu, and Peter Richtárik. SAGA with arbitrary sampling. In *The 36th International Conference on Machine Learning*, 2019.

[46] Peter Richtárik and Martin Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv preprint arXiv:1706.01108*, 2017.

[47] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[48] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

[49] Ernest K Ryu and Wotao Yin. Proximal-proximal-gradient method. *arXiv preprint arXiv:1708.06908*, 2017.

[50] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated subgradient solver for SVM. In *24th International Conference on Machine Learning*, pages 807–814, 2007.

[51] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30, 2011.

[52] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

[53] Thomas Strohmer and Roman Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[54] Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, pages 537–552, 2013.

[55] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[56] Stephen Tu, Shivaram Venkataraman, Ashia C Wilson, Alex Gittens, Michael I Jordan, and Benjamin Recht. Breaking locality accelerates block Gauss-Seidel. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3482–3491. JMLR. org, 2017.

[57] Huy V Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Perez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. *arXiv preprint arXiv:1904.03148*, 2019.

[58] Băng Công Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

[59] Yue Yu and Longbo Huang. Fast stochastic variance reduced admm for stochastic composition optimization. *arXiv preprint arXiv:1705.04138*, 2017.

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

[61] Alp Yurtsever, Băng Công Vũ, and Volkan Cevher. Stochastic three-composite convex minimization. In *Advances in Neural Information Processing Systems*, pages 4329–4337, 2016.

[62] Renbo Zhao and Volkan Cevher. Stochastic three-composite convex minimization with a linear operator. In *International Conference on Artificial Intelligence and Statistics*, pages 765–774, 2018.

[63] Shuai Zheng and James T Kwok. Fast-and-light stochastic admm. In *International Joint Conferences on Artificial Intelligence Organization*, pages 2407–2613, 2016.

Supplementary Material

A Stochastic Decoupling Method for Minimizing the Sum of Smooth and Non-Smooth Functions

Contents

A Applications

In this section we list a number of selected applications for our method:

- Compressed sensing [9].
- Total Generalized Variance (TGV) image denoising [7].
- Decentralized optimization over networks [38].
- Support-vector machine [14].
- Dantzig selector [8].
- Group-Lasso [60].
- Network utility maximization.
- Square-root lasso [5].
- ℓ_1 trend filtering [29].
- Convex relaxation of unsupervised image matching and object discovery [57].

In the rest of this section we formulate some of them explicitly. A summary of the mapping of these problems to the structure of problem (1) is provided in Table 3.

Special case of problem (1)	$f(x)$	$g_j(x)$	$R(x)$
Constrained optimization (4)	$f(x)$	$\chi_{\mathcal{C}_j}(x)$	$R(x)$
Convex projection	$\frac{1}{2}\|x - x^0\|^2$	$\chi_{\mathcal{C}_j}(x)$	0
Convex feasibility	0	$\chi_{\mathcal{C}_j}(x)$	0
Dantzig selector (5)	0	$\chi_{\mathcal{B}_\lambda^j}(x)$	$\|x\|_1$
Decentralized optimization (6)	$f_i(x_i)$	$\chi_{\{x \,:\, w_j^\top x = 0\}}(x)$	0
Support vector machine (7)	$f(x) = \frac{\lambda}{2}\|x\|^2$, $n = 1$	$\max\{0, 1 - b_j a_j^\top x\}$	0
Overlapping group lasso (8)	$f_i(x) = \frac{1}{2}(a_i^\top x - b_i)^2$	$\|x\|_{G_j}$	0
Fused lasso (9)	$\frac{1}{2}(a_i^\top x - b_i)^2$	$\chi_{\mathcal{C}_j^\varepsilon}(x)$	$\lambda\|x\|_1$
Fused lasso (10)	$\frac{1}{2}(a_i^\top x - b_i)^2$	$\lambda_2\|\mathbf{D}_{j:}x\|$	$\lambda_1\|x\|_1$

Table 3: Selected applications of Algorithm 1 for solving problem (1).

A.1 Constrained Optimization

Let $\mathcal{C}_j \subseteq \mathbb{R}^d$ be closed convex sets with a non-empty intersection and consider the constrained composite optimization problem

$$\begin{aligned} \min \quad & f(x) + R(x) \\ \text{subject to} \quad & x \in \bigcap_{j=1}^m \mathcal{C}_j . \end{aligned}$$

If we let $g_j \equiv \chi_{\mathcal{C}_j}$ be the characteristic function of $\mathcal{C}_j$, defined as follows: $\chi_{\mathcal{C}_j}(x) = 0$ for $x \in \mathcal{C}_j$ and $\chi_{\mathcal{C}_j}(x) = +\infty$ for $x \notin \mathcal{C}_j$, this problem can be written in the form

$$\min_{x \in \mathbb{R}^d} f(x) + R(x) + \frac{1}{m} \sum_{j=1}^m \underbrace{\chi_{\mathcal{C}_j}(x)}_{g_j(x)} . \tag{4}$$

We remark that [1] considered the case $m = 1$.

For $f(x) = \frac{1}{2}\|x - x^0\|^2$ and $R \equiv 0$, this specialized to the *best approximation* problem. For $f \equiv 0$ and $R \equiv 0$, this problem specializes to the *convex feasibility* problem [37].

A.2 Dantzig Selector

Dantzig selector [8] solves the problem of estimating sparse parameter x from a linear model. Given an input matrix $\mathbf{A} \in \mathbb{R}^{m\times d}$, output vector $b \in \mathbb{R}^m$ and threshold parameter $\lambda \geq 0$, define

$$\mathcal{B}_\lambda \coloneqq \{x \;:\; \|\mathbf{A}^\top(b - \mathbf{A}x)\|_\infty \leq \lambda\} = \bigcap_{j=1}^m \mathcal{B}_\lambda^j,$$

where $\mathcal{B}_\lambda^j \coloneqq \left\{x \;:\; \left|\left(\mathbf{A}^\top(b-\mathbf{A}x)\right)_j\right| \leq \lambda\right\}$. The goal of the Dantzig selector problem is to find the solution to

$$\min_{x\in\mathbb{R}^d} \|x\|_1 + \chi_{\mathcal{B}_\lambda}(x),$$

which can equivalently be written in the finite-sum form

$$\min_{x\in\mathbb{R}^d} \underbrace{\|x\|_1}_{R(x)} + \frac{1}{m}\sum_{j=1}^m \underbrace{\chi_{\mathcal{B}_\lambda^j}(x)}_{g_j(x)}. \tag{5}$$

A.3 Decentralized Optimization

The problem of minimizing the sum of functions over a network [38] can be reformulated as

$$\min_{x=(x_1,\dots,x_n)} \frac{1}{n}\sum_{i=1}^n f_i(x_i) + \chi_{\{x\,:\,\mathbf{W}x=0\}}(x),$$

where $\mathbf{W}$ is a matrix such that $\mathbf{W}x = 0$ if and only if $x_1 = \cdots = x_n$. Functions $f_1, \dots, f_n$ are stored on different nodes and each node has access only to its own function. Matrix $\mathbf{W}$ is often derived from a communication graph, which defines how the nodes can communicate with each other. Then, one can solve the problem above by sampling constraints and projecting onto them, which corresponds to averaging of the iterates among a subset of nodes. Formally, if $\mathbf{W} = (w_1^\top, \dots, w_m^\top)^\top$, we rewrite the problem above as

$$\min_{x=(x_1,\dots,x_n)} \frac{1}{n}\sum_{i=1}^n \underbrace{f_i(x_i)}_{f_i(x)} + \frac{1}{m}\sum_{j=1}^m \underbrace{\chi_{\{x\,:\,w_j^\top x=0\}}(x)}_{g_j(x)}. \tag{6}$$

A.4 Support-Vector Machine (SVM)

Support-vector machine [14] is a very popular method for supervised classification. The primal formulation of SVM is given by

$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{\lambda}{2}\|x\|^2}_{f(x)} + \frac{1}{m}\sum_{j=1}^m \underbrace{\max\{0, 1 - b_j a_j^\top x\}}_{g_j(x)}, \tag{7}$$

where $a_1, \dots, a_m \in \mathbb{R}^d$ and $b_1, \dots, b_m$ are the features and the outputs. It is easy to verify that for $g_j(x) \coloneqq \max\{0, 1 - b_j a_j^\top x\}$ the proximal operator is given by

$$\mathrm{prox}_{\eta_j g_j}(x) = x + \Pi_{[0,\eta_j]}\left(\frac{1 - b_j a_j^\top x}{\|a_j\|^2}\right) b_j a_j.$$

The celebrated stochastic subgradient descent method Pegasos [50, 51, 54] for SVMs achieves $\mathcal{O}(1/t)$ rate.

A.5 Overlapping Group Lasso

This is a generalization of LASSO proposed in [60] to efficiently select groups of features that are most valuable for the given objective. Let us assume that we are given sets of indices $G_1, \dots, G_m \subseteq \{1, \dots, d\}$ and let $\|x\|_{G_j} \coloneqq \sqrt{\sum_{i \in G}[x]_i^2}$, where $[x]_i$ is the i-th coordinate of vector x. Then, assuming that we are given vectors $a_1, \dots, a_n \in \mathbb{R}^d$ and scalars $b_1, \dots, b_n$, the objective we want to minimize is

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\frac{1}{2}(a_i^\top x - b_i)^2}_{f_j(x)} + \frac{1}{m} \sum_{j=1}^{m} \underbrace{\|x\|_{G_j}}_{g_j(x)}. \tag{8}$$

It is easy to verify that if $g_j(x) = \|x\|_{G_j}$, then

$$[\mathrm{prox}_{\eta_j g_j}(x)]_i = \begin{cases} [x]_i, & \text{if } i \notin G_j, \\ \max\left\{0, \left(1 - \frac{\eta_j}{\|x\|_{G_j}}\right)\right\} [x]_i, & \text{if } i \in G_j. \end{cases}$$

Vector y_j^t will always have at most $|G_j|$ nonzeros, so one can store in memory only the coordinates of y_j^t from G_j.

A.6 Fused Lasso

The Fused Lasso problem [55] is defined as

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\frac{1}{2}(a_i^\top x - b_i)^2}_{f_i(x)} + \underbrace{\lambda \|x\|_1}_{R(x)} + \frac{1}{d-1} \sum_{j=1}^{d-1} \underbrace{\chi_{\mathcal{C}_j^\varepsilon}(x)}_{g_j(x)}, \tag{9}$$

where

$$\mathcal{C}_j^\varepsilon \coloneqq \{x \ : \ |[x]_j - [x]_{j+1}| \le \varepsilon\},$$

$[x]_j$ is the j-th entry of vector x, $a_1, \dots, a_n \in \mathbb{R}^d$ and $b_1, \dots, b_n \in \mathbb{R}$ are given vectors and scalars, ε is given thresholding parameter.

Another formulation of the Fused Lasso is done by using penalty functions. Define $\mathbf{D}$ to be zero everywhere except for $\mathbf{D}_{i,i} = 1$ and $\mathbf{D}_{i,i+1} = -1$ with $i = 1, \dots, d-1$. Note that $\|\mathbf{D}x\|_1 = \sum_{j=1}^{m} |\mathbf{D}_{j:}x|$, where m is the number of rows of $\mathbf{D}$. Then the reformulated objective is

$$\min_{x} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\frac{1}{2}(a_i^\top x - b_i)^2}_{f_i(x)} + \underbrace{\lambda_1 \|x\|_1}_{R(x)} + \frac{1}{m} \sum_{j=1}^{m} \underbrace{\lambda_2 |\mathbf{D}_{j:}x|}_{g_j(x)}. \tag{10}$$

In our notation, this means $\mathbf{A} = \mathbf{D}^\top$ and $\mathbf{A}^\top \mathbf{A}$ is a tridiagonal matrix given by

$$\mathbf{A}^\top \mathbf{A} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

It can be shown that the eigenvalue of a tridiagonal matrix $\mathbf{W}$ of size $(d-1) \times (d-1)$ with a on its main diagonal and b on the other two diagonals are given by $\lambda_k(\mathbf{W}) = a + 2|b| \cos\left(\frac{k\pi}{d}\right)$, $k = 1, \dots, d-1$. Thus, $\lambda_{\min}(\mathbf{A}^\top \mathbf{A}) = 2 + 2\cos\left(\left(1 - \frac{1}{d}\right)\pi\right) = 2 - 2\cos\left(\frac{\pi}{d}\right) \approx \frac{1}{2d^2}$ and $\min_j \frac{1}{\|\mathbf{A}_j\|^2} = \frac{1}{6}$. Therefore, if in (9) or (10) $\lambda_1 = 0$, we guarantee linear convergence with the aforementioned constants.

A.7 Square-root Lasso

The approach gets its name from minimizing the square root of the regular least squares, i.e. $\|\mathbf{D}w - b\|$ instead of $\|\mathbf{D}w - b\|^2$. This is then combined with ℓ_1-penalty for feature selection, which gives the objective

$$\min_{w\in\mathbb{R}^d} \|\mathbf{D}w - b\| + \lambda\|w\|_1.$$

Equivalently, by introducing a new variable z we can put constraints $\mathbf{D}_{j:}x - [z]_j = 0$ for $j = 1, \dots, m$, which can be written as $a_j^\top (w^\top, z^\top)^\top = 0$ with $a_j = (\mathbf{D}_{j:}, e_j^\top)^\top$ and $e_j := (0, 0, \dots, \underbrace{1}_{j}, \dots, 0)$. Then, the reformulation is

$$\min_{x=(w,z)\in\mathbb{R}^{d+m}} \frac{1}{m}\sum_{j=1}^m \underbrace{\chi_{\{x: a_j^\top x = 0\}}}_{g_j(x)=g_j(w,z)} + \underbrace{\|z - b\| + \lambda\|w\|_1}_{R(x)=R(w,z)}.$$

The proximal operator of R is that of a block-separable function, which is easy to evaluate:

$$\mathrm{prox}_{\eta R}(x) = \begin{pmatrix} \mathrm{prox}_{\eta\lambda\|\cdot\|_1}(w) \\ \mathrm{prox}_{\eta\|\cdot - b\|}(z) \end{pmatrix}.$$

B Relation to Existing Methods

B.1 SDCA, Dykstra's algorithm and the Kaczmarz method

Here we formulate SDCA [52], Dykstra's algorithm and Kaczmarz method. SDCA is a method for solving

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{j=1}^m g_j(x) + \frac{1}{2}\|x-x^0\|^2.$$

Its iterates can be defined by the following recursion:

$$\begin{aligned} x^{t+1} &= \mathrm{prox}_{\eta g_j}(x^t + \overline{y}_j^t), \\ \overline{y}_j^{t+1} &= \overline{y}_j^t + x^t - x^{t+1}. \end{aligned}$$

If we restrict our attention to characteristic functions, i.e.

$$g_j(x) = \chi_{\mathcal{C}_j}(x) = \begin{cases} 0, & \text{if } x\in\mathcal{C}_j \\ +\infty, & \text{otherwise} \end{cases},$$

then the proximal operator step is replaced with projection:

$$x^{t+1} = \Pi_{\mathcal{C}_j}(x^t + \overline{y}_j^t).$$

This is known as Dykstra's algorithm. Finally, if $\mathcal{C}_j = \{x : a_j^\top x = b_j\}$, then it boils down to random projections, i.e.

$$x^{t+1} = \Pi_{\{a_j^\top x = b_j\}}(x^t),$$

which is the method of Kaczmarz.

Originally SDCA is formulated differently and

Theorem 7. *Consider the regularized minimization problem of SDCA, which is*

$$\min_{x} \frac{1}{m}\sum_{j=1}^m g_j(x) + \frac{1}{2}\|x-x^0\|^2$$

with convex $g_1,\dots,g_m$. Then, SDCA is a special cases of Algorithm 1 obtained by applying it with $f(x) = \frac{1}{2}\|x-x^0\|^2$, $R(x)\equiv 0$, stepsize $\eta = \frac{1}{m}$ and initialization $y_1^0 = \dots = y_m^0 = 0$. Furthermore, if we consider special case $g_j = \chi_{\mathcal{C}_j}$, where $\mathcal{C}_j \neq \emptyset$ is a closed convex set, then we also obtain Dykstra's algorithm, and if every $\mathcal{C}_j$ is a linear subspace, then we recover the Kaczmarz method.

Proof. We will show by induction that $y^t = x^0 - x^t$ and $x^{t+1} = \mathrm{prox}_{\eta_j g_j}(x^t + \eta_j y_j^t)$.

Indeed, it holds for y^0 by initialization and then by induction assumption we have

$$z^t = x^t - \eta(x^t - x^0) - \eta y^t = x^t - \eta(x^t - x^0) - \eta(x^0 - x^t) = x^t.$$

Therefore, if we denote $\overline{y}_j^t \coloneqq \eta_j y_j^t$, then

$$x^{t+1} = \mathrm{prox}_{\eta_j g_j}(x^t + \overline{y}_j^t),$$

which is the update rule of x^{t+1} in SDCA. Moreover, we have

$$\overline{y}_j^{t+1} = \eta y_j^{t+1} = \eta y_j^t + x^t - x^{t+1} = \overline{y}_j^t + x^t - x^{t+1}.$$

Finally,

$$y^{t+1} = y^t + z^t - x^{t+1} = x^0 - x^t + x^t - x^{t+1} = x^0 - x^{t+1},$$

which yields our induction step and the proof itself. □

B.2 Accelerated Kaczmarz

Accelerated Kaczmarz [34] performs the following updates:

$$\begin{aligned} z^t &= (1-\alpha_t)x^t - \alpha_t y^t, \\ x^{t+1} &= \Pi_{\{x: a_i^\top x = b_i\}}(z^t), \\ y^{t+1} &= y^t + \gamma_t(z^t - x^{t+1}) + (1-\beta_t)(z^t - y^t) \end{aligned}$$

with some parameters $\alpha_t, \gamma_t, \beta_t$. While the original analysis [34] suggests $\beta_t < 1$, our method gives the same update when $f(x) = \frac{1}{2}\|x\|^2$, $R \equiv 0$, $\alpha_t = \eta$, $\beta_t = 1$, $\gamma_t = \frac{1}{\eta n}$.

B.3 ADMM and Douglas-Rachford splitting

ADMM, also known as Douglas-Rachford splitting, in its simplest form as presented in [43] is a special case of Algorithm 1 when $f \equiv 0$ and $m = 1$.

B.4 Point-SAGA, SAGA, SVRG and proximal GD

In the trivial case $f \equiv 0$ and $R \equiv 0$, we recover Point-SAGA. Methods such as SAGA, SVRG and Proximal Gradient Descent are obtained, in contrast, by setting $g \equiv 0$. We would like to mention that introducing g does not change the stepsizes for which those methods work, e.g. Gradient Descent works with arbitrary $\eta < {}^{2}/_{L}$, which is tight. The similarity suggests that small η should be used when solving this problem and this observation is validated by our experiments.

B.5 Stochastic Primal-Dual Hybrid Gradient

The relation to the Stochastic Primal-Dual Hybrid Gradient (SPDHG) is complicated. On the one hand, SPDHG is a general method with three parameters and it preconditions proximal operators with matrices, so our method can not be its strict generalization. On the other hand, SPDHG does not allow for f. Moreover, when $f \equiv 0$ and some parameters are set to specific values in SPDHG, the methods coincide, but the guarantees are not the same. In particular, we show below that one of the parameters in SPDHG, θ, should be set to 1, in which case linear convergence for smooth $g_1, \dots, g_m$ was not known for SPDHG. Therefore, potentially the tools developed in this work can lead to new discoveries about full version of SPDHG as well.

Let us now formulate the method explicitly. After a simple rescaling of the functions, SPDHG from [21] can be formulated as a method to solve the problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{j=1}^m \phi_j(\mathbf{A}_j^\top x) + R(x). \tag{11}$$

Renaming the variables for our convenience and choosing for simplicity uniform probabilities of sampling j from $\{1, \dots, m\}$, the update rules of SPDHG can be written as

$$\begin{aligned} w^t &= \mathrm{prox}_{\eta R}(w^{t-1} - \eta \overline{y}^t), \\ y_j^{t+1} &= \mathrm{prox}_{\sigma \phi_j^*}(\sigma \mathbf{A}_j^\top w^t + y_j^t), \\ y^{t+1} &= y^t + \frac{1}{m}\mathbf{A}_j(y_j^{t+1} - y_j^t), \\ \overline{y}^{t+1} &= \overline{y}^t + \theta(y_j^{t+1} - y_j^t), \end{aligned}$$

where η, σ and θ are the method's parameters and ϕ_j^* is the Fenchel conjugate of ϕ_j. The initialization that we are interested in is with $y^0 = \frac{1}{m}\sum_{j=1}^m y_j^0$, $\overline{y}^0 = y^0$, $w^0 = x^0$.

One can immediately see that one big difference with our approach is that the method puts $\mathbf{A}_j$ outside of the proximal operator, which also leads to different iteration complexity. In particular, when $\phi_1, \dots, \phi_m$ are smooth, the complexity proved in [11] is

$$\mathcal{O}\left(\left(m + \sum_{j=1}^m \|\mathbf{A}_j\| \sqrt{\frac{L_\phi}{\mu_R}}\right) \log \frac{1}{\varepsilon}\right),$$

where μ_R is the strong convexity constant of R and L_ϕ is the smoothness constant of $\phi_1, \dots, \phi_m$. Since function $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$ is at most $L_\phi \|\mathbf{A}_j\|^2$ smooth, our rate from Corollary 3 with μ-strongly convex and L-smooth f is

$$\mathcal{O}\left(\left(n + m + \frac{L}{\mu} + \sqrt{m\frac{L_\phi}{\mu}} \max_j \|\mathbf{A}_j\|\right) \log\frac{1}{\varepsilon}\right).$$

If, in addition, we use sampling with probabilities proportional to $\|\mathbf{A}_j\|$, then we can achieve

$$\mathcal{O}\left(\left(n + m + \frac{L}{\mu} + \frac{1}{\sqrt{m}}\sum_{j=1}^m \|\mathbf{A}_j\| \sqrt{\frac{L_\phi}{\mu_R}}\right) \log\frac{1}{\varepsilon}\right).$$

We do not prove this, but the complexity for our method will be similar if we use strongly convex R rather than f, so our rates should match or be even be superior to that of SPDHG, at the cost of evaluating potentially harder proximal operators.

Now, let us prove that our method is indeed connected to SPDHG via choice of $\theta = 1$ and $\eta\sigma = 1$.

Theorem 8. *If we apply SPDHG with identity matrices $\mathbf{A}_j = \mathbf{I}$, i.e. $\phi_j(x) = g_j(x)$, and choose parameters $\theta = 1$ and $\eta\sigma = 1$, then it is algorithmically equivalent to Algorithm 1 with $f \equiv 0$.*

Proof. Since ϕ_j and g_j are the same, we will use in the proof g_j only.

First, mention that it is straightforward to show by induction that $y^t = \frac{1}{m}\sum_{j=1}^m y_j^t$, which coincides with our update. Our goal is to show by induction that in SPDHG it holds

$$w^{t-1} - \eta\overline{y}^t = x^t - \eta y^t,$$

where we define sequence x^t as

$$x^{t+1} := \mathrm{prox}_{\eta g_j}(w^t + \eta y_j^t) = \mathrm{prox}_{\frac{1}{\sigma}g_j}(w^t + \eta y_j^t).$$

We will see that implicitly x^{t+1} is present in every update of SPDHG. To this end, let us first rewrite the update for y_j^{t+1}. We have by Moreau's identity

$$y_j^{t+1} = \mathrm{prox}_{\sigma g_j^*}(\sigma w^t + y_j^t) = \sigma w^t + y_j^t - \sigma\,\mathrm{prox}_{\frac{1}{\sigma}g_j}\left(\frac{\sigma w^t + y_j^t}{\sigma}\right).$$

Since we consider $\sigma = \frac{1}{\eta}$, it transforms into

$$y_j^{t+1} = y_j^t + \frac{1}{\eta}\left(w^t - \mathrm{prox}_{\eta g_j}(w^t + \eta y_j^t)\right) = y_j^t + \frac{1}{\eta}\left(w^t - x^{t+1}\right)$$

The only missing thing is rewriting update for w^t in terms of x^t and y^t. From the update rule for y_j^{t+1} we derive

$$\overline{y}^{t+1} = y^t + \theta(y_j^{t+1} - y_j^t) = y^t + \frac{\theta}{\eta}(w^t - x^{t+1}).$$

Hence,

$$\begin{aligned} w^{t+1} &= \mathrm{prox}_{\eta R}(w^t - \eta\overline{y}^{t+1}) \\ &= \mathrm{prox}_{\eta R}(w^t - \eta y^{t+1} - \theta(w^t - x^{t+1})) \\ &\overset{\theta=1}{=} \mathrm{prox}_{\eta R}(x^{t+1} - \eta y^{t+1}). \end{aligned}$$

Thus, update for w^t, y_j^t and y^t completely coincide under this choice of parameters. □

Since our method under $f \equiv 0$ reduced to Point-SAGA [15], we obtain the following results that was unknown.

Corollary 4. *Point-SAGA [15] is a special case of Stochastic Primal-Dual Hybrid Gradient [11].*

C Evaluating Proximal Operators

For some functions, the proximal operator admits a closed form solution, for instance if $g_j(x) = \chi_{\{x\,:\,a_j^\top x = b_j\}}(x)$, then

$$\mathrm{prox}_{\eta_j g_j}(x) = x - \tfrac{a_j^\top x - b_j}{\|a_j\|^2} a_j.$$

If, however, the proximal operator is not given in a closed form, then it is still possible to efficiently evaluate it. If $g_j = \phi_j(\mathbf{A}_j^\top x)$, $\mathbf{A}_j \in \mathbb{R}^{d \times d_j}$ and $\mathbf{A}_j$ is of full column rank, then the proximal operator is the solution of a d_j-dimensional strongly convex problem.

Lemma 4. *Let $\phi_j : \mathbb{R}^{d_j} \to \mathbb{R}$ be a convex lower semi-continuous function such that Range $\left(\mathbf{A}_j^\top\right)$ has a point of* $\mathrm{dom}(\phi)$*. If $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$, then*

$$x - \mathrm{prox}_{\eta_j g_j}(x) \in \textit{Range}\left(\mathbf{A}_j\right).$$

Proof. Let us fix x. Any vector $z \in \mathbb{R}^d$ can be decomposed as $z = x + \mathbf{A}_j\beta + w$, where $\beta \in \mathbb{R}^{d_j}$ and $\mathbf{A}_j^\top w = 0$, from which it also follows $g_j(z) = \phi_j(\mathbf{A}_j^\top x + \mathbf{A}^\top \mathbf{A}_j \beta)$. Then

$$\begin{aligned}
\mathrm{prox}_{\eta_j g_j}(x) &:= \operatorname*{argmin}_{z \in \mathbb{R}^d} \left\{ \eta_j \phi_j(\mathbf{A}_j^\top z) + \frac{1}{2}\|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{z = x + \mathbf{A}_j\beta + w} \left\{ \eta_j \phi_j(\mathbf{A}_j^\top x + \mathbf{A}_j^\top \mathbf{A}_j \beta) + \frac{1}{2}\|\mathbf{A}_j\beta + w\|^2 \right\} \\
&= \operatorname*{argmin}_{z = x + \mathbf{A}_j\beta + w} \left\{ \eta_j \phi_j(\mathbf{A}_j^\top x + \mathbf{A}_j^\top \mathbf{A}_j \beta) + \frac{1}{2}\|\mathbf{A}_j\beta\|^2 + \frac{1}{2}\|w\|^2 \right\}.
\end{aligned}$$

Clearly, the last expression achieves its minimum only when $w = 0$. □

We can simply the expression for the proximal operator even more. It is straightforward to verify that for any matrix $\mathbf{B} \in \mathbb{R}^{d_1 \times d_2}$ and constant vector $c \in \mathbb{R}^{d_2}$ we have and function Φ with $\mathrm{dom}(\Phi(\mathbf{B}\beta + c) \neq \emptyset$ it holds

$$\begin{aligned}
\operatorname*{argmin}_{\beta = \mathbf{B}(\alpha + c), \beta \in \mathbb{R}^{d_2}} \Phi(\beta) &= \operatorname*{argmin}_{\beta = \mathbf{B}(\alpha + c), \beta \in \mathbb{R}^{d_2}} \Phi(\mathbf{B}(\alpha + c)) \\
&= \mathbf{B} \operatorname*{argmin}_{u = \alpha + c, \alpha \in \mathbb{R}^{d_1}} \Phi(\alpha + c) \\
&= \mathbf{B}\left(\operatorname*{argmin}_{u \in \mathbb{R}^{d_1}} \Phi(u) - c \right). \qquad (12)
\end{aligned}$$

Since we know by chain rule that $u := \mathrm{prox}_{\eta_j g_j}(x) = x + \mathbf{A}_j \beta_j$ for some $\beta_j \in \mathbb{R}^{d_j}$, we can write the necessary and sufficient optimality condition for u:

$$\begin{aligned}
\mathrm{prox}_{\eta_j g_j}(x) &= \operatorname*{argmin}_{u = x + \mathbf{A}_j\beta,\ \beta \in \mathbb{R}^{d_j}} \left\{ \phi_j(\mathbf{A}_j^\top u) + \frac{1}{2\eta_j}\|x - u\|^2 \right\} \\
&\overset{(12)}{=} x + \mathbf{A}_j \operatorname*{argmin}_{\beta \in \mathbb{R}^{d_j}} \left\{ \phi_j\left(\mathbf{A}_j^\top (x + \mathbf{A}_j\beta)\right) + \tfrac{1}{2\eta_j}\|\mathbf{A}_j\beta\|^2 \right\} \\
&= x + \mathbf{A}_j \operatorname*{argmin}_{\beta \in \mathbb{R}^{d_j}} \left\{ \phi_j\left(\mathbf{A}_j^\top x + \mathbf{A}_j^\top \mathbf{A}_j \beta\right) + \tfrac{1}{2\eta_j}\|\mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}\mathbf{A}_j^\top \mathbf{A}_j\beta\|^2 \right\} \\
&\overset{(12)}{=} x + \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1} \operatorname*{argmin}_{\alpha = \mathbf{A}_j^\top \mathbf{A}_j \beta} \left\{ \phi_j\left(\mathbf{A}_j^\top x + \alpha\right) + \tfrac{1}{2\eta_j}\|\mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}\alpha\|^2 \right\} \\
&\overset{(12)}{=} x + \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1} \Big(\operatorname*{argmin}_{\theta = \alpha + \mathbf{A}_j^\top x} \left\{ \phi_j\left(\theta\right) + \tfrac{1}{2\eta_j}\|\mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}(\theta - \mathbf{A}_j^\top x)\|^2 \right\} - \mathbf{A}^\top x \Big).
\end{aligned}$$

Note that

$$\begin{aligned}
\|\mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}(\theta - \mathbf{A}_j^\top x)\|^2 &= (\theta - \mathbf{A}_j^\top x)^\top (\mathbf{A}_j^\top \mathbf{A}_j)^{-1} \mathbf{A}_j^\top \mathbf{A}_j (\mathbf{A}_j^\top \mathbf{A}_j)^{-1} (\theta - \mathbf{A}_j^\top x) \\
&= \|\theta - \mathbf{A}_j^\top x\|^2_{(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}},
\end{aligned}$$

where for any positive semi-definite matrix $\mathbf{W}$ we denote $\|x\|_{\mathbf{W}}^2 := x^\top \mathbf{W} x$. Denoting similarly $\mathrm{prox}_{\eta_j \phi_j}^{\mathbf{W}}(x) := \mathrm{argmin}_\theta \{\phi_j(\theta) + \frac{1}{2\eta_j}\|\theta - x\|_{\mathbf{W}}^2\}$, we obtain

$$\begin{aligned} \mathrm{prox}_{\eta_j g_j}(x) &= x + \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1} \left(\underset{\theta \in \mathbb{R}^{d_j}}{\mathrm{argmin}} \left\{ \phi_j\left(\beta\right) + \tfrac{1}{2\eta_j}\|\theta - \mathbf{A}_j^\top x\|_{(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}} \right\} - \mathbf{A}^\top x \right) \\ &= x + \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1} \left(\mathrm{prox}_{\eta_j \phi_j}^{(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}} \left(\mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}\mathbf{A}_j^\top x\right) - \mathbf{A}^\top x \right). \end{aligned}$$

Thus, we only need to know how to efficiently evaluate $\mathrm{prox}_{\lambda \phi_j}^{(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}}(z)$ for arbitrary $\lambda > 0$ and $z \in \mathbb{R}^{d_j}$, assuming that matrix $(\mathbf{A}_j^\top \mathbf{A}_j)^{-1}$ can be precomputed. For example, if $\mathbf{A}_j = a_j \in \mathbb{R}^d$, then

$$\mathrm{prox}_{\eta_j \phi_j}^{(a_j^\top a_j)^{-1}}(x) = \mathrm{prox}_{\eta_j \|a_j\|^2 \phi_j}(x).$$

If, in addition, $\phi_j \colon \mathbb{R} \to \mathbb{R}$ is given by

$$\phi_j(z) = \begin{cases} b_j z, & \text{if } z \le 0, \\ c_j z, & \text{otherwise} \end{cases}$$

with $b_j < c_j$, then $\mathrm{prox}_{\lambda \phi_j}(z) = z - \lambda b_j$ for $z \le \lambda b_j$, $\mathrm{prox}_{\lambda \phi_j}(z) = 0$ for $z \in (\lambda b_j, \lambda c_j]$ and $\mathrm{prox}_{\lambda \phi_j}(z) = z - \lambda c_j$ for $z > \lambda c_j$. Therefore,

$$\mathrm{prox}_{\eta_j g_j}(x) = \begin{cases} x - \eta_j a_j b_j, & \text{if } a_j^\top x \le \|a_j\|^2 b_j, \\ x - \frac{a_j^\top x}{\|a_j\|^2} a_j, & \text{if } \|a_j\|^2 b_j \le a_j^\top x \le \|a_j\|^2 c_j, \\ x - \eta_j a_j c_j, & \text{otherwise} \end{cases}.$$

Note that if $\|a_j\|^2 b_j \le a_j^\top x \le \|a_j\|^2 c_j$, then $a_j^\top \mathrm{prox}_{\eta_j g_j}(x) = 0$.

D Inequalities Related to Smoothness, Convexity and Proximal Operators

Since many of our proofs are easier to write when one uses Bregman divergences, we will formulate most of the required properties in terms of $D_f(\cdot, \cdot)$.

Proposition 1. *Let f be convex and L-smooth, then we have for any y*

$$\|\nabla f(x) - \nabla f(y)\|^2 \le 2LD_f(x, y), \tag{13}$$

$$\|\nabla f(x) - \nabla f(y)\|^2 \le L \langle \nabla f(x) - \nabla f(y), x - y \rangle . \tag{14}$$

Proposition 2. *Let f be μ-strongly convex, including the case $\mu = 0$, which holds when f is simply convex. Then, for arbitrary x and y*

$$\frac{\mu}{2}\|x - y\|^2 + D_f(x, y) \le \langle \nabla f(x) - \nabla f(y), x - y \rangle . \tag{15}$$

The proposition above is convenient for proofs of SVRG and SAGA, but it is not tight if we want to show that Gradient Descent converges for any $\eta \le \frac{2}{L+\mu}$ when the objective is μ-strongly convex. To make the analysis tighter, we require the following statement.

Proposition 3. *Let f be differentiable and μ-strongly convex. Then we have for any x and y*

$$\mu\|x - y\|^2 \le \langle \nabla f(x) - \nabla f(y), x - y \rangle . \tag{16}$$

Moreover, if f is also L-smooth, then

$$\frac{\mu L}{L + \mu}\|x - y\|^2 + \frac{1}{L + \mu}\|\nabla f(x) - \nabla f(y)\|^2 \le \langle \nabla f(x) - \nabla f(y), x - y \rangle . \tag{17}$$

This is the tightest inequality one can get and, in particular, (17) implies (14) when $\mu = 0$.

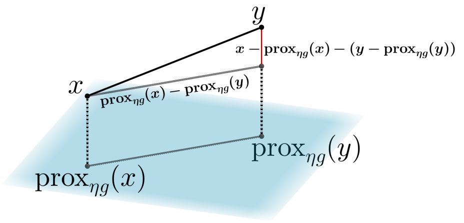

y
$x - \mathrm{prox}_{\eta g}(x) - (y - \mathrm{prox}_{\eta g}(y))$
x
$\mathrm{prox}_{\eta g}(x) - \mathrm{prox}_{\eta g}(y)$
$\mathrm{prox}_{\eta g}(y)$
$\mathrm{prox}_{\eta g}(x)$

Figure 2: Illustration of property (18) with characteristic function of a linear subspace, $g(x) = \chi_{\{x \,:\, a^\top x = b\}}$. In this case the proximal operator returns the projection of a point onto the subspace, and Inequality (18) becomes identity and follows from Pythagorean theorem.

An important property of the proximal operator is firm non-expansiveness:

Proposition 4. *Let $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a proper closed convex function. Then its proximal operator is firmly non-expansive. That is, for all $\eta \in \mathbb{R}$,*

$$\|\mathrm{prox}_{\eta g}(x) - \mathrm{prox}_{\eta g}(z)\|^2 \le \|x - z\|^2 - \left(1 + \tfrac{1}{\eta L_g}\right)\|x - \mathrm{prox}_{\eta g}(x) - (z - \mathrm{prox}_{\eta g}(z))\|^2, \tag{18}$$

where $L_g \in \mathbb{R} \cup \{+\infty\}$ is the smoothness constant of function g (for non-smooth functions, $L_g = +\infty$).

Inequality (18) was also the main inspiration for one of the authors, who believed that if a method for non-smooth variance reduction exists, then it is possible to show convergence using (18). We derived the method by playing with this inequality and trying to see how it can be combined with a full gradient step ∇f, and later extended it to stochastic gradients and introduced penalty term R. Moreover, we would like to note that Equation 18 is tight if $g(x) = \chi_{\{x \,:\, a^\top x = b\}}$ for some vector a and scalar b, as is shown in Figure 2.

E Optimality Conditions

We now comment on the nature of Assumption 1. In view of the first-order necessary and sufficient condition for the solution of (1), we have

$$x^* \in \mathcal{X}^* \quad \Leftrightarrow \quad 0 \in \partial F(x^*) = \nabla f(x^*) + \partial (g + R)(x^*).$$

By the weak sum rule [4, Cor 3.38], we have

$$\partial F(x) \supseteq \nabla f(x) + \frac{1}{m} \sum_{j=1}^{m} \partial g_j(x) + \partial R(x)$$

for all $x \in \operatorname{dom} F \supseteq \mathcal{X}^*$. Under the regularity condition $\cap_{j=1}^{k} (\operatorname{dom} g_j) \cap_{j=k+1}^{m} \operatorname{ri}(\operatorname{dom} g_j) \cap \operatorname{ri}(\operatorname{dom} R) \neq \emptyset$, where $g_1, \dots, g_k$ are polyhedral functions, the inclusions becomes an identity [47, Thm 23.8], which means that Assumption 1 is satisfied.

For functions g_j of the form $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$, where $\phi_j : \mathbb{R}^{d_j} \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions and $\mathbf{A}_j \in \mathbb{R}^{d \times d_j}$, we shall instead consider the following (slightly stronger) assumption:

Assumption 3. *There exists $x^* \in \mathcal{X}^*$ and vectors $y_1^* \in \mathbf{A}_1 \partial \phi_1(\mathbf{A}_1^\top x^*), \dots, y_m^* \in \mathbf{A}_m \partial \phi_m(\mathbf{A}_m^\top x^*)$ and $r^* \in \partial R(x^*)$ such that $\nabla f(x^*) + \frac{1}{m} \sum_{j=1}^{m} y_j^* + r^* = 0$.*

Since $\mathbf{A}_j \partial \phi_j(\mathbf{A}_j^\top x) \subseteq \partial g_j(x)$ for all $x \in \operatorname{dom} g_j$ [4, Thm 3.43], Assumption 3 is indeed stronger than Assumption 1. If Range $\left(\mathbf{A}_j^\top\right)$ contains a point from $\operatorname{ri}(\operatorname{dom} g_j)$, or g_j is polyhedral and Range $\left(\mathbf{A}_j^\top\right)$ contains a point from mere $\operatorname{dom}(g_j)$, then $\partial g_j(x) = \mathbf{A}_j \partial \phi_j(\mathbf{A}_j^\top x)$ for any x [47, Thm 23.9], and these two assumptions are the same.

Below we provide another stationarity condition that shows why x^* is a fixed-point of our method.

Lemma 5 (Optimality conditions)**.** *Let x^* be a solution of* (1) *and let Assumption 1 be satisfied. Then for any $\eta, \eta_j \in \mathbb{R}$,*

$$x^* = \operatorname{prox}_{\eta R}(x^* - \eta \nabla f(x^*) - \eta y^*), \qquad x^* = \operatorname{prox}_{\eta_j g_j}(x^* + \eta_j y_j^*).$$

Proof. Let $z = \operatorname{prox}_{\eta R}(x^* - \eta \nabla f(x^*) - \eta y^*) = \operatorname{argmin}_u \{\eta R(u) + \frac{1}{2}\|u - (x^* - \eta \nabla f(x^*) - \eta y^*)\|^2\}$. R is convex, so the problem inside argmin is strongly convex, and the necessary and sufficient condition for z to be its solution is

$$0 \in z - x^* + \eta \nabla f(x^*) + \eta y^* + \eta \partial R(z).$$

By Assumption 1 it holds for $z = x^*$, implying the first equation that we want to prove. The second one follows by exactly the same argument applied to $\operatorname{argmin}_u \{\eta_j g_j(u) + \frac{1}{2}\|u - (x^* + \eta_j y_j^*)\|^2\}$. □

F Convergence Proofs

In this section, we provide the proofs of our convergence results. Each lemma, theorem and corollary is first restated and only then proved to simplify the reading.

F.1 Proof of Lemma 1 (Gradient Descent)

Algorithm 3 Stochastic Decoupling Method with Gradient Descent.

```
Input: Stepsize $\eta$, initial vectors $x^0$, $y_1^0, \dots, y_m^0$, $y^0 = \frac{1}{m}\sum_{j=1}^m y_j^0$, oracle that gives gradient estimates
for $t = 0, 1, \dots$ do
    $z^{t+1} = \mathrm{prox}_{\eta R}(x^t - \eta \nabla f(x^t) - \eta y^t)$
    Sample $j$ from $\{1, \dots, m\}$ with probabilities $\{p_1, \dots, p_m\}$ and set $\eta_j = \frac{\eta}{m p_j}$
    $x^{t+1} = \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t)$
    $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1})$
    $y^{t+1} = y^t + \frac{1}{m}(y_j^{t+1} - y_j^t)$
end for
```

Here we prove that Gradient Descent update on f satisfies our assumption on the method with the best possible stepsizes.

Lemma 1. *If f is convex, Gradient Descent satisfies Assumption 2(a) with any $\eta_0 < \frac{2}{L}$, $\omega = 2 - \eta_0 L$ and $\mathcal{M}^t = 0$. If f is μ-strongly convex, Gradient Descent satisfies Assumption 2(b) with $\eta_0 = \frac{2}{L+\mu}$, $\omega = 1$ and $\mathcal{M}^t = 0$.*

Proof. Since we consider Gradient Descent, we have

$$w^t = x^t - \eta \nabla f(x^t).$$

First, if f is convex and smooth, then for any $\eta \le \eta_0 < \frac{2}{L}$

$$\begin{aligned}
\|w^t - w^*\|^2 &= \|x^t - x^*\|^2 - 2\eta \left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^* \right\rangle + \eta^2 \|\nabla f(x^t) - \nabla f(x^*)\|^2 \\
&\le \|x^t - x^*\|^2 - \eta(2 - \eta_0 L) \left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^* \right\rangle \\
&\quad - \eta\eta_0 L \left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^* \right\rangle + \eta\eta_0 \|\nabla f(x^t) - \nabla f(x^*)\|^2 \\
&\overset{(14)}{\le} \|x^t - x^*\|^2 - \eta(2 - \eta_0 L) \left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^* \right\rangle \\
&\overset{(15)}{\le} \|x^t - x^*\|^2 - \eta\left(2 - \eta_0 L\right) D_f(x^t, x^*).
\end{aligned}$$

Now let us consider μ-strongly convex f. We have

$$\begin{aligned}
\|w^t - w^*\|^2 &= \|x^t - x^*\|^2 - 2\eta \left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^* \right\rangle + \eta^2 \|\nabla f(x^t) - \nabla f(x^*)\|^2 \\
&\overset{(17)}{\le} \left(1 - \frac{2\eta\mu L}{L+\mu}\right) \|x - y\|^2 - \eta \left(\frac{2}{L+\mu} - \eta\right) \|\nabla f(x^t) - \nabla f(x^*)\|^2 \\
&\overset{(16)}{\le} \left(1 - \frac{2\eta\mu L}{L+\mu}\right) \|x - y\|^2 - \eta \left(\frac{2}{L+\mu} - \eta\right) \mu^2 \|x^t - x^*\|^2 \\
&= (1 - \eta\mu)^2 \|x^t - x^*\|^2 \\
&\le (1 - \eta\mu) \|x^t - x^*\|^2.
\end{aligned}$$

The last step simply uses $1 - \eta\mu \le 1$, which, of course, makes our guarantees slightly weaker, but, on the other hand, puts Gradient Descent under the umbrella of Assumption 2. □

F.2 Key lemma

The result below is the most important lemma of our paper as it lies at the core of our analysis. It provides a very generic statement about the step with stochastic proximal operators. At the same time, it is a mere corollary of firm non-expansiveness of the proximal operator.

Lemma 6. *Let $z^t = \operatorname{prox}_{\eta R}(w^t - \eta y^t)$ and $x^{t+1} = \operatorname{prox}_{\eta_j g_j}(z^t + y_j^t)$, where j is sampled from $\{1, \dots, m\}$ with probabilities $\{p_1, \dots, p_m\}$, $\eta_j = \frac{\eta}{p_j}$ and η is a positive number. If $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1})$ and $y_k^{t+1} = y_k^t$ for all $k \neq j$, it holds*

$$\mathbb{E}\left[\|x^{t+1} - x^*\|^2 + \mathcal{Y}^{t+1}\right] \le \mathbb{E}\left[\|w^t - w^*\|^2 + \left(1 - \frac{\gamma}{m(1+\gamma)}\right)\mathcal{Y}^t - \|z^t - w^t - (x^* - w^*)\|^2\right],$$

where $\gamma := \min_{j=1,\dots,m} \frac{1}{\eta_j L_j}$ and $L_j \in \mathbb{R} \cup \{+\infty\}$ is the smoothness constant of g_j.

Proof. Mention that $x^* = \operatorname{prox}_{\eta_j g_j}(x^* + \eta_j y_j^*)$ by optimality condition. In addition, it holds by definition $y_j^{t+1} = \frac{1}{\eta_j}(z^t + \eta_j y_j^t - x^{t+1})$, so property (18) yields

$$\|x^{t+1} - x^*\|^2 + \left(1 + \frac{1}{\eta_j L_g}\right)\eta_j^2 \|y_j^{t+1} - y_j^*\|^2 \le \|z^t + \eta_j y_j^t - (x^* + \eta_j y_j^*)\|^2$$

and we can replace $1 + \frac{1}{\eta_j L_g}$ with $1 + \gamma$ since $\gamma \le \frac{1}{\eta_j L_g}$.

Let $\mathbb{E}_j$ the expectation with respect to sampling of j. Then, we observe

$$\begin{aligned}
&\mathbb{E}_j \|z^t + \eta_j y_j^t - (x^* + \eta_j y_j^*)\|^2 \\
&= \|z^t - x^*\|^2 + \mathbb{E}_j\left[\frac{\eta^2}{m^2 p_j^2}\|y_j^t - y_j^*\|^2\right] + 2\left\langle z^t - x^*, \eta \mathbb{E}_j\left[\frac{1}{m p_j}(y_j^t - y_j^*)\right]\right\rangle \\
&= \|z^t - x^*\|^2 + \frac{\eta^2}{m^2}\sum_{k=1}^m \frac{1}{p_k}\|y_k^t - y_k^*\|^2 + 2\eta\left\langle z^t - x^*, y^t - y^*\right\rangle. \qquad (19)
\end{aligned}$$

Denote $w^* := x^* - \eta \nabla f(x^*)$. Another optimality condition from Lemma 5 is $x^* = \operatorname{prox}_{\eta R}(w^* - \eta y^*)$, so let us use (18) one more time to obtain

$$\begin{aligned}
\|z^t - x^*\|^2 &\le \|w^t - \eta y^t - (w^* - \eta y^*)\|^2 - \|w^t - \eta y^t - z^t - (w^* - \eta y^* - x^*)\|^2 \\
&= \eta^2\|y^t - y^*\|^2 - 2\eta\left\langle w^t - w^*, y^t - y^*\right\rangle + \|w^t - w^*\|^2 - \|w^t - \eta y^t - z^t - (w^* - \eta y^* - x^*)\|^2.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\|w^t - \eta y^t - z^t - (w^* - \eta y^* - x^*)\|^2 &= \|w^t - z^t - (w^* - x^*)\|^2 \\
&\quad - 2\eta\left\langle w^t - z^t - (w^* - x^*), y^t - y^*\right\rangle + \eta^2\|y^t - y^*\|^2,
\end{aligned}$$

so

$$\begin{aligned}
\|z^t - x^*\|^2 &\le -2\eta\left\langle w^t - w^*, y^t - y^*\right\rangle + \|w^t - w^*\|^2 + 2\eta\left\langle w^t - z^t - (w^* - x^*), y^t - y^*\right\rangle \\
&\quad - \|w^t - z^t - (w^* - x^*)\|^2 \\
&= \|w^t - w^*\|^2 - 2\eta\left\langle z^t - x^*, y^t - y^*\right\rangle - \|w^t - z^t - (w^* - x^*)\|^2.
\end{aligned}$$

Together with the previously obtained bounds it adds up to

$$\mathbb{E}_j\|z^t + \eta_j y_j^t - (x^* + \eta_j y_j^*)\|^2 \le \|w^t - w^*\|^2 + \frac{\eta^2}{m^2}\sum_{k=1}^m \frac{1}{p_k}\|y_k^t - y_k^*\|^2 - \|z^t - w^t - (x^* - w^*)\|^2.$$

To get the expression in the left-hand side of this lemma's statement, let us add the missing sum and evaluate its expectation:

$$\mathbb{E}\left[\sum_{k=1}^m \eta_k^2 \|y_k^{t+1} - y_k^t\|^2\right] = \mathbb{E}\|y_j^{t+1} - y_j^*\|^2 + \mathbb{E}\left[\sum_{k\neq j}\eta_k^2\|y_k^{t+1} - y_k^t\|^2\right].$$

Clearly, all summands in the last sum were not changed at iteration t, so

$$\begin{aligned}\mathbb{E}_j\left[\sum_{k\neq j}\eta_k^2\|y_k^{t+1}-y_k^t\|^2\right] &= \mathbb{E}_j\left[\sum_{k\neq j}\eta_k^2\|y_k^t-y_k^t\|^2\right]\\ &= \sum_{k=1}^m(1-p_k)\eta_k^2\|y_k^t-y_k^t\|^2\\ &= \sum_{k=1}^m\eta_k^2\|y_k^t-y_k^t\|^2-\frac{\eta^2}{m^2}\sum_{k=1}^m\frac{1}{p_k}\|y_k^t-y_k^t\|^2.\end{aligned}$$

The negative sum will cancel out with the same in equation (19) and we conclude the proof. □

F.3 Convergence of Bregman divergence to 0 almost surely

Here we formulate a result that we only briefly mentioned in the main text. It states that for convex problems, Bregman divergence $D_f(x^t, x^*)$ almost surely converges to 0. To show it, let us borrow the classical result on supermartingale convergence.

Proposition 5 ([6], Proposition A.4.5)**.** *Let $\{X^t\}_t$, $\{Y^t\}_t$, $\{Z^t\}_t$ be three sequences of non-negative random variables and let $\{\mathcal{F}^t\}_t$ be a sequence of σ-algebras such that $\mathcal{F}^t\subset\mathcal{F}^{t+1}$ for all t. Assume that:*

- *The random variables X^t, Y^t, Z^t are non-negative and $\mathcal{F}^t$-measurable.*
- *For each t, we have $\mathbb{E}[X^{t+1}\mid\mathcal{F}^t]\le X^t-Y^t+Z^t$.*
- *There holds, with probability 1,*

$$\sum_{t=0}^\infty Z^t<\infty.$$

Then X^t converges to a non-negative random variable X and we have $\sum_{t=0}^\infty Y^t<\infty$ with probability 1.

Theorem 9. *Take a method that satisfies Assumption 2(a), a stepsize $\eta\le\eta_0$ and an optimum x^* satisfying Assumption 1. Then, with probability 1 it holds $D_f(x^t,x^*)\to 0$.*

Proof. Fix any solution x^*, $y_1^*,\dots,y_m^*$. Let $\mathcal{F}^t=\sigma(x^0,y_1^0,\dots,y_m^0,\dots,x^t,y_1^t,\dots,y_m^t)$ be the σ-algebra generated by all random variables prior to moment t, and let $\overline{\mathcal{M}}^t$ be $\mathcal{M}^t$ conditioned on $\mathcal{F}^t$, i.e. $\overline{\mathcal{M}}^t=\mathcal{M}^t|\mathcal{F}^t$, from which it follows $\mathcal{M}^t=\mathbb{E}\overline{\mathcal{M}}^t$. Then, the assumptions of Proposition 5 are satisfied for sequences

$$\begin{aligned}X^t&=\|x^t-x^*\|^2+\overline{\mathcal{M}}^t+(1+\gamma)\sum_{k=1}^m\eta_k^2\|y_k^t-y_k^*\|^2,\\ Y^t&=\omega\eta D_f(x^t,x^*),\\ Z^t&=0.\end{aligned}$$

Therefore, we have that $\sum_{t=0}^\infty Y^t<\infty$ and $Y^t\to 0$ almost surely, from which it follows $D_f(x^t,x^*)\to 0$. □

The almost sure guarantee is not applicable to SGD which has $\mathcal{M}^0$ proportional to the number of iterations. We leave its analysis as well as analysis of convergence of x^t to an optimum for future work.

F.4 Proof of Theorem 1 ($\mathcal{O}\left(1/t\right)$ rate)

Below we provide the proof of $\mathcal{O}\left(1/t\right)$ rate for general convex functions.

Theorem 1. *Assume f is L-smooth and μ-strongly convex, $g_1, \dots, g_m, R$ are convex, closed and lower semi-continuous. Take a method satisfying Assumption 2 and $\eta \le \eta_0$, then*

$$\mathbb{E}D_f(\overline{x}^t, x^*) \le \frac{1}{\omega\eta t}\mathcal{L}^0,$$

where $\mathcal{L}^0 := \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m \eta_k^2\|y_k^0 - y_k^*\|^2$ and $\overline{x}^t := \frac{1}{t}\sum_{k=0}^{t-1} x^k$.

Proof. Recall that

$$\mathcal{L}^t := \mathbb{E}\|x^t - x^*\|^2 + \mathcal{M}^t + \mathcal{Y}^t,$$

and by Assumption 2 combined with Lemma 6

$$\mathcal{L}^{t+1} \le \mathcal{L}^t - \omega\eta\mathbb{E}D_f(x^t, x^*).$$

Telescoping this inequality from 0 to t, we obtain

$$\mathbb{E}\left[\sum_{k=0}^t D_f(x^t, x^*)\right] \le \frac{1}{\omega\eta}(\mathcal{L}^0 - \mathcal{L}^{t+1}) \le \frac{1}{\omega\eta}\mathcal{L}^0.$$

By convexity of f, the left-hand side is lower bounded by $t\mathbb{E}D_f(\overline{x}^t, x^*)$, so dividing both sides by t finishes the proof. □

F.5 Proof of Theorem 2 ($\mathcal{O}(1/t^2)$ rate)

In this subsection, we show the $\mathcal{O}\left(1/t^2\right)$ rate.

Theorem 2. *Consider updates with time-varying stepsizes, $\eta^t = \frac{2}{\omega\mu(a+t)}$ and $\eta_j^t = \frac{\eta^t}{mp_j}$ for $j = 1, \dots, m$, where $a \ge 2\max\left\{\frac{1}{\omega\mu\eta_0}, \frac{1}{\rho}\right\}$. Then, it holds*

$$\mathbb{E}\|x^t - x^*\|^2 \le \frac{a^2}{(t+a-1)^2}\mathcal{L}^0,$$

where $\mathcal{L}^0 = \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m (\eta_k^0)^2\|y_k^0 - y_k^*\|^2$.

Proof. For this proof, we redefine the sequence $\mathcal{Y}^t$ to have time-varying stepsizes:

$$\mathcal{Y}^t := \sum_{k=1}^m (\eta_k^t)^2\mathbb{E}\|y_k^t - y_k^*\|^2.$$

Before writing a new recurrence, let us note that

$$(1 - \omega\eta^t\mu)\left(\frac{\eta^{t-1}}{\eta^t}\right)^2 = \frac{\left(1 - \frac{2}{a+t}\right)(a+t)^2}{(a+t-1)^2} = \frac{(a+t-2)(a+t)}{(a+t-1)^2} < 1,$$

so $1 - \omega\eta^t\mu \le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2$. Then, Lemma 6 gives a similar recurrence to what we have seen in other proofs, but the stepsizes in the right-hand side are now time-dependent:

$$\begin{aligned}
\mathcal{L}^{t+1} &= \mathbb{E}\|x^{t+1} - x^*\|^2 + \mathcal{M}^{t+1} + \mathcal{Y}^{t+1} \\
&\le (1 - \omega\eta^t\mu)\mathbb{E}\|x^t - x^*\|^2 + (1-\rho)\mathcal{M}^t + \sum_{k=1}^m (\eta_k^t)^2\mathbb{E}\|y_k^t - y_k^*\|^2 \\
&\le (1 - \omega\eta^t\mu)\mathbb{E}\left[\|x^t - x^*\|^2 + \mathcal{M}^t\right] + \left(\frac{\eta^t}{\eta^{t-1}}\right)\sum_{k=1}^m (\eta_k^{t-1})^2\mathbb{E}\|y_k^t - y_k^*\|^2 \\
&\le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2\mathbb{E}\left[\|x^t - x^*\|^2 + \mathcal{M}^t\right] + \left(\frac{\eta^t}{\eta^{t-1}}\right)^2\sum_{k=1}^m (\eta_k^{t-1})^2\mathbb{E}\|y_k^t - y_k^*\|^2 \\
&= \left(\frac{\eta^t}{\eta^{t-1}}\right)^2\mathcal{L}^t.
\end{aligned}$$

Thus,

$$\mathcal{L}^{t+1} \le \mathcal{L}^0 \prod_{k=1}^{t} \left(\frac{\eta^k}{\eta^{k-1}}\right)^2 = \left(\frac{\eta^t}{\eta^0}\right)^2 \mathcal{L}^0 = \left(\frac{a}{a+t}\right)^2 \mathcal{L}^0.$$

□

F.6 Proof of Theorem 3 ($\mathcal{O}(1/t)$ rate of SGD)

Here we consider the case where $f(x)$ is given as expectation parameterized by a random variable ξ,

$$f(x) = \mathbb{E}_\xi f_\xi(x).$$

While it is often assumed in the literature that $\mathbb{E}\|\nabla f_\xi(x) - \nabla f(x)\|^2 \le \sigma^2$ uniformly over x, we do not need this assumption and bound the variance using the following lemma.

Lemma 7. *Let $w^t = x^t - \eta \nabla f_{\xi^t}(x^t)$, where random function $f_\xi(x)$ is almost surely convex and L-smooth. Then,*

$$\mathbb{E}\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*)\|^2 \le 4L\mathbb{E}D_f(x^t, x^*) + 2\sigma_*^2, \tag{20}$$

where $\sigma_*^2 := \mathbb{E}\|\nabla f_\xi(x^*) - \nabla f(x^*)\|^2$, i.e. σ_*^2 is the variance at an optimum. If more than one x^* exists, take the one that minimizes σ_*^2.

Proof. This proof is based on existing results for SGD and goes in a very standard way. By Young's inequality

$$\begin{aligned}\mathbb{E}\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*)\|^2 &\le 2\mathbb{E}\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*;\xi^t)\|^2 + 2\mathbb{E}\|\nabla f_{\xi^t}(x^*) - \nabla f(x^*)\|^2 \\ &\overset{(13)}{\le} 4L\mathbb{E}D_{f_{\xi^t}}(x^t, x^*) + 2\sigma_*^2 \\ &= 4L\mathbb{E}D_f(x^t, x^*) + 2\sigma_*^2.\end{aligned}$$

□

In the proof of Theorem 3 we will again need time-varying stepsize and $\mathcal{Y}^t$ should be defined as

$$\mathcal{Y}^t := \sum_{k=1}^{m} (\eta_k^t)^2 \mathbb{E}\|y_k^t - y_k^*\|^2.$$

Lemma 8. *But before let us prove a simple statement about sequences with contraction and additive error. Assume that sequence $\{\mathcal{L}^t\}_t$ satisfies inequality $\mathcal{L}^{t+1} \le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \mathcal{L}^t + 2(\eta^t)^2\sigma_*^2$ with some constant $\sigma_* \ge 0$. Then, it holds*

$$\mathcal{L}^t \le \left(\frac{\eta^{t-1}}{\eta^0}\right)^2 \mathcal{L}^0 + 2t(\eta^{t-1})^2\sigma_*^2.$$

Proof. We will prove the inequality by induction. For $t = 0$ it is straightforward. The induction step follows from

$$\begin{aligned}\mathcal{L}^{t+1} &\le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \mathcal{L}^t + 2(\eta^t)^2\sigma_*^2 \\ &\le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \left(\frac{\eta^{t-1}}{\eta^0}\right)^2 \mathcal{L}^0 + 2\left(\frac{\eta^t}{\eta^{t-1}}\right)^2 (\eta^{t-1})^2 t\sigma_*^2 + 2t(\eta^{t-1})^2\sigma_*^2 \\ &= \left(\frac{\eta^t}{\eta^0}\right)^2 \mathcal{L}^0 + 2(t+1)(\eta^{t-1})^2\sigma_*^2.\end{aligned}$$

□

Now we are ready to prove the theorem.

Theorem 3. *Assume f is μ-strongly convex, $f(\cdot;\xi)$ is almost surely convex and L-smooth. Let the update be produced by SGD, i.e. $v^t = \nabla f(x^t;\xi^t)$, and let us use time-varying stepsizes $\eta^{t-1} = \frac{2}{a+\mu t}$ with $a \ge 4L$. Then, it holds*

$$\mathbb{E}\|x^t - x^*\|^2 \le \frac{8\sigma_*^2}{\mu(a+\mu t)} + \frac{a^2}{(a+\mu t)^2}\mathcal{L}^0.$$

Proof. It holds by Lemma 7

$$\mathbb{E}\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*)\|^2 \le 4L\mathbb{E}D_f(x^t, x^*) + 2\sigma_*^2.$$

Therefore, for $w^t := x^t - \eta^t v^t = x^t - \eta^t \nabla f_{\xi^t}(x^t)$ and $w^* := x^* - \eta^t \nabla f(x^*)$ we have

$$\begin{aligned}\mathbb{E}\|w^t - w^*\|^2 &= \mathbb{E}\left[\|x^t - x^*\|^2 - 2\eta^t\left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^*\right\rangle + (\eta^t)^2\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*)\|^2\right] \\ &\overset{(20)}{\le} \mathbb{E}\left[\|x^t - x^*\|^2 - 2\eta^t\left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^*\right\rangle + 4(\eta^t)^2 L D_f(x^t, x^*) + 2(\eta^t)^2\sigma_*^2\right] \\ &\overset{(15)}{\le} \mathbb{E}\Big[(1-\eta^t\mu)\|x^t - x^*\|^2 - 2\eta^t(\underbrace{1 - 2\eta^t L}_{\ge 0})D_f(x^t, x^*) + 2(\eta^t)^2\sigma_*^2\Big] \\ &\le (1-\eta^t\mu)\mathbb{E}\|x^t - x^*\|^2 + 2(\eta^t)^2\sigma_*^2.\end{aligned}$$

Using the same argument as in the proof of Theorem 2, we can show that $1 - \eta^t\mu \le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2$. Combining these results with Lemma 6, we obtain for $\mathcal{L}^{t+1} := \mathbb{E}\|x^{t+1} - x^*\|^2 + \mathcal{Y}^{t+1}$

$$\begin{aligned}\mathcal{L}^{t+1} &\le (1-\eta^t\mu)\mathbb{E}\|x^t - x^*\|^2 + \sum_{k=1}^m (\eta_k^t)^2\mathbb{E}\|y_k^t - y_k^*\|^2 + 2(\eta^t)^2\sigma_*^2 \\ &\le \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \mathbb{E}\|x^t - x^*\|^2 + \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \mathcal{Y}^t + 2(\eta^t)^2\sigma_*^2 \\ &= \left(\frac{\eta^t}{\eta^{t-1}}\right)^2 \mathcal{L}^t + 2(\eta^t)^2\sigma_*^2.\end{aligned}$$

By Lemma 8

$$\mathbb{E}\|x^t - x^*\|^2 \le \mathcal{L}^t \le \left(\frac{\eta^{t-1}}{\eta^0}\right)^2 \mathcal{L}^0 + 2t(\eta^{t-1})^2\sigma_*^2 \le \frac{a^2}{(a+\mu t)^2}\mathcal{L}^0 + \frac{8t}{(a+\mu t)\mu t}\sigma_*^2.$$

□

F.7 Proof of Theorem 4 (linear rate for $g_j = \phi_j(\mathbf{A}_j^\top x)$)

Let us now show linear convergence of our method when the consider problem has linear structure, i.e. $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$.

We first need a lemma on the nature of $y_1^t, \dots, y_m^t$ in the considered case.

Lemma 9. *Let the proximal sum be of the form $\frac{1}{m}\sum_{j=1}^m \phi_j(\mathbf{A}_j^\top x)$ with some matrices $\mathbf{A}_j \in \mathbb{R}^{d\times d_j}$, and $y_j^0 = \mathbf{A}_j\beta_j^0$ for $j = 1, \dots, m$. Then, if Assumption 3 is satisfied, for any t and j we have*

$$y_j^t = \mathbf{A}_j\beta_j^t, \quad y^t = \tfrac{1}{m}\sum_{j=1}^m y_j^t = \tfrac{1}{m}\mathbf{A}\beta^t, \quad y_j^* = \mathbf{A}_j\beta_j^*, \quad y^* = \tfrac{1}{m}\sum_{j=1}^m y_j^* = \tfrac{1}{m}\mathbf{A}\beta^*$$

with some vectors $\beta_i^t, \beta_i^* \in \mathbb{R}^{d_i}$ with $i = 1, \dots, m$, $\beta^t := ((\beta_1^t)^\top, \dots, (\beta_m^t)^\top)^\top$, $\beta^* := ((\beta_1^*)^\top, \dots, (\beta_m^*)^\top)^\top$ and $\mathbf{A} := [\mathbf{A}_1, \dots, \mathbf{A}_m]$.

Proof. By definition $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1}) = \frac{1}{\eta_j}(z^t + \eta_j y_j^t - x^{t+1})$. In addition, by Lemma 4 there exists $\beta_j^{t+1} \in \partial\phi_j(\mathbf{A}_j^\top x^{t+1})$ such that $x^{t+1} = \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t) \in z^t + \eta_j y_j^t - \eta_j\mathbf{A}_j\beta_j^{t+1}$ and, thus, $y_j^{t+1} = \mathbf{A}_j\beta_j^{t+1}$. Therefore, we also have $y^t = \frac{1}{m}\mathbf{A}\beta^t$.

The claims about $y_1^*, \dots, y_m^*$ and y^* follow from Assumption 3. □

Now it is time to prove Theorem 4.

Theorem 4. *Assume that f is μ-strongly convex, $R \equiv 0$, $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$ for $j = 1, \dots, m$ and take a method satisfying Assumption 2 with $\rho > 0$. Then, if $\eta \le \eta_0$,*

$$\mathbb{E}\|x^t - x^*\|^2 \le (1 - \min\{\rho, \omega\eta\mu, \rho_A\})^t \, \mathcal{L}^0,$$

where $\rho_A \coloneqq \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \min_j \left(\frac{p_j}{\|\mathbf{A}_j\|}\right)^2$, and $\mathcal{L}^0 \coloneqq \|x^0 - x^*\|^2 + \mathcal{M}^0 + \sum_{k=1}^m \eta_k^2 \|y_k^0 - y_k^*\|^2$.

Proof. Lemma 6 with Assumption 2 yields

$$\mathcal{L}^{t+1} \le (1 - \min\{\rho, \omega\eta\mu\}) \left(\mathbb{E}\|x^t - x^*\|^2 + \mathcal{M}^t\right) + \mathcal{Y}^t - \mathbb{E}\|z^t - w^t - (x^* - w^*)\|^2.$$

By Lemma 9

$$\mathcal{Y}^t = \sum_{k=1}^m \eta_k^2 \mathbb{E}\|y_k^t - y_k^*\|^2 = \sum_{k=1}^m \eta_k^2 \|\mathbf{A}_k\|^2 \mathbb{E}\|\beta_k^t - \beta_k^*\|^2.$$

Since we assume $R \equiv \text{const}$, we have $z^t - w^t = x^t - \eta v^t - \eta y^t - (x^t - \eta v^t) = -\eta y^t$ and

$$\begin{aligned}
\|z^t - w^t - (x^* - w^*)\|^2 &= \eta^2 \|y^t - y^*\|^2 \\
&= \frac{\eta^2}{m^2} \|\mathbf{A}(\beta^t - \beta^*)\|^2 \\
&\ge \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \frac{\eta^2}{m^2} \|\beta^t - \beta^*\|^2 \\
&= \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \sum_{k=1}^m \frac{p_k^2}{\|\mathbf{A}_k\|^2} \eta_k^2 \|\mathbf{A}_k\|^2 \|\beta_k^t - \beta_k^*\|^2 \\
&\ge \lambda_{\min}(\mathbf{A}^\top \mathbf{A}) \min_k \frac{p_k^2}{\|\mathbf{A}_k\|^2} \sum_{k=1}^m \eta_k^2 \|\mathbf{A}_k\|^2 \|\beta_k^t - \beta_k^*\|^2 \\
&= \rho_A \mathcal{Y}^t.
\end{aligned}$$

Therefore,

$$\mathcal{Y}^t - \mathbb{E}\|z^t - w^t - (x^* - w^*)\|^2 \le (1 - \rho_A)\mathcal{Y}^t.$$

Putting the pieces together, we obtain

$$\mathcal{L}^{t+1} \le (1 - \min\{\rho, \omega\eta\mu, \rho_A\})\mathcal{L}^t,$$

from which it follows that $\mathcal{L}^t$ converges to 0 linearly. Finally, note that $\mathbb{E}\|x^t - x^*\|^2 \le \mathcal{L}^t \le (1 - \min\{\rho, \omega\eta\mu, \rho_A\})^t \mathcal{L}^0$. □

F.8 Proof of Theorem 5 (linear constraints)

Here we discuss the problem of linearly constrained minimization

$$\min_x \{f(x) : \mathbf{A}^\top x = b\}.$$

We split matrix $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_m]$ and vector $b = (b_1^\top, \dots, b_m^\top)^\top$ and define projection operator $\Pi_j(\cdot) \coloneqq \Pi_{\{x : \mathbf{A}_j^\top x = b_j\}}(\cdot)$. Since $y_j^t \in \text{Range}\,(\mathbf{A}_j)$, it is orthogonal to the hyperplane $\{x : \mathbf{A}_j^\top x = b_j\}$ for any x it holds

$$\Pi_j(x + y_j^t) = \Pi_j(x).$$

This allows us to write a memory-efficient version of Algorithm 1 as given in Algorithm 4. If only a subset of functions $g_1, \dots, g_m$ is linear equality constraints, then similarly the corresponding vectors y_j^t are not need in the update, although they are still useful for the analysis.

Here we show that if f is strongly convex and the non-smooth part is constructed of linear constraints, then we can guarantee linear rate of convergence. Moreover, the rate will depend only on the smallest nonzero eigenvalue of $\mathbf{A}^\top \mathbf{A}$, implying that even if $\mathbf{A}^\top \mathbf{A}$ is degenerate, convergence will be linear.

Algorithm 4 Double method for linearly constrained problem.

```
Input: Stepsize $\eta$, initial vectors $x^0$, $y^0$, oracle that gives gradient estimates
for $t = 0, 1, \ldots$ do
    Produce a gradient estimate $v^t$
    $z^t = \mathrm{prox}_{\eta R}(x^t - \eta v^t - \eta y^t)$
    Sample $j$ from $\{1, \ldots, m\}$ with probabilities $\{p_1, \ldots, p_m\}$
    $x^{t+1} = \Pi_j(z^t)$
    $y^{t+1} = y^t + \frac{p_j}{\eta}(z^t - x^{t+1})$
end for
```

Theorem 5. *Under the same assumptions as in Theorem 4 and assuming, in addition, that* $g_j(x) = \chi_{\{x \,:\, \mathbf{A}_j^\top x = b_j\}}$ *it holds* $\mathbb{E}\|x^t - x^*\|^2 \le (1 - \min\{\rho, \omega\eta\mu, \rho_A\})\mathcal{L}^0$ *with* $\rho_A = \lambda_{\min}^+(\mathbf{A}^\top \mathbf{A}) \min_j \left(\frac{p_j}{\|\mathbf{A}_j\|}\right)^2$*, i.e.* ρ_A *depends only on the smallest* ***positive*** *eigenvalue of* $\mathbf{A}^\top\mathbf{A}$*.*

Proof. The main reason we get an improved guarantee for linear constraints is that one can write a closed form expression for the proximal operators:

$$\mathrm{prox}_{\eta_j g_j}(x) = x - \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^\dagger(\mathbf{A}_j^\top x - b_j).$$

Assume that j was sampled at iteration t, then

$$\begin{aligned} y_j^{t+1} &= \frac{1}{\eta_j}\left(z^t + \eta_j y_j^t - \mathrm{prox}_{\eta_j g_j}(z^t + \eta_j y_j^t)\right) \\ &= \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^\dagger(\mathbf{A}_j^\top(z^t + \eta_i y_j^t) - b_j). \end{aligned}$$

Therefore, for any j and t there exists a vector x_j^t such that

$$\begin{aligned} y_j^{t+1} &= \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^\dagger(\mathbf{A}_j^\top x_j^t - b_j) \\ &= \mathbf{A}_j(\mathbf{A}_j^\top \mathbf{A}_j)^\dagger(\mathbf{A}_j^\top x_j^t - \mathbf{A}_j^\top x^*), \end{aligned}$$

where the second step is by the fact that x^* is from the set $\{x : \mathbf{A}_j^\top x = b\}$. Then, $y_j^t - y_j^* = \frac{1}{m}\mathbf{A}_j(\beta_j^t - \beta_j^*)$ with $\beta_j^t - \beta_j^* \in \mathrm{Range}\left(\mathbf{A}_j^\top\right)$. This, in turn, implies $\beta^t - \beta^* \in \mathrm{Range}\left(\mathbf{A}^\top\right)$, so

$$\begin{aligned} \|z^t - w^t - (x^* - w^*)\|^2 &= \eta^2\|y^t - y^*\|^2 \\ &= \frac{\eta^2}{m^2}\|\mathbf{A}(\beta^t - \beta^*)\|^2 \\ &\ge \lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})\frac{\eta^2}{m^2}\|\beta^t - \beta^*\|^2. \end{aligned}$$

The rest of the proof goes the same way as that of Theorem 4 in Appendix F.7. □

F.9 Proof of Theorem 6 (smooth g_j)

This is the only proof where Lemma 6 is used with finite smoothness constants. However, we will not use the negative square term from Lemma 6, which is rather needed in the case $g_j(x) = \phi_j(\mathbf{A}_j^\top x)$.

Theorem 6. *Assume that f is L-smooth and μ-strongly convex, g_j is L_j-smooth for $j = 1, \ldots, m$ and Assumption 2(b) is satisfied. Then, Algorithm 1 converges as*

$$\mathbb{E}\|x^t - x^*\|^2 \le \left(1 - \min\left\{\omega\eta\mu, \rho, \frac{\gamma}{m(1+\gamma)}\right\}\right)\mathcal{L}^0,$$

where $\gamma \coloneqq \min_{j=1,\ldots,m}\frac{1}{\eta_j L_j}$*.*

Proof. Following the same lines as in the proof of Theorem 4, we get a contraction in $\mathcal{Y}^t$. Now we obtain it from the fact that functions $g_1, \ldots, g_m$ are smooth, so the recursion is

$$\begin{aligned} \mathcal{L}^{t+1} &\le (1 - \omega\eta\mu)\|x^t - x^*\|^2 + (1-\rho)\mathcal{M}^t + \left(1 - \frac{\gamma}{m(1+\gamma)}\right)\mathcal{Y}^t \\ &\le \left(1 - \min\left\{\omega\eta\mu, \rho, \frac{\gamma}{m(1+\gamma)}\right\}\right)\mathcal{L}^t. \end{aligned}$$

This is sufficient to show the claimed result. □

F.10 Proof of Corollary 3 (optimal stepsize)

Corollary 3 is a very interesting statement about the optimal stepsizes for the case where $g_1, \dots, g_m$ are smooth functions. Its proof is a mere check that the choice of stepsizes gives the claimed complexity.

Corollary 3. *Choose as solver for f SVRG or SAGA without minibatching, which satisfy Assumption 2 with $\eta_0 = \frac{1}{5L}$ and $\rho = \frac{1}{n}$, and consider for simplicity situation where $L_1 = \dots = L_m := L_g$ and $p_1 = \dots = p_m$. Define $\eta_{best} := \frac{1}{\sqrt{\omega\mu m L_g}}$, and set the stepsize to be $\eta = \min\{\eta_0, \eta_{best}\}$. Then the complexity to get $\mathbb{E}\|x^t - x^*\|^2 \le \varepsilon$ is $\mathcal{O}\left(\left(n + m + \frac{L}{\mu} + \sqrt{m\frac{L_g}{\mu}}\right)\log\frac{1}{\varepsilon}\right)$.*

Proof. According to Theorem 6, in general, for any $\eta \le \eta_0$ the complexity to get $\mathbb{E}\|x^t - x^*\|^2$ is

$$\mathcal{O}\left(\left(\frac{1}{\rho} + m + \frac{1}{\omega\eta\mu} + \frac{1}{\gamma}m\right)\log\frac{1}{\varepsilon}\right),$$

where $\frac{1}{\gamma}$ simplifies to ηL_g when $L_1 = \dots = L_m = L_g$ and $p_1 = \dots = p_m = \frac{1}{m}$. In addition, for SVRG and SAGA, ω is a constant close to 1, so we can ignore it. Since m and $\frac{1}{\rho} = 3n$ do not depend on η, we only need to simplify the other two terms. One of them decreases with η and the other increases, so the best complexity is achieved when the two quantities are equal to each other. The corresponding equation is

$$\omega\eta^2\mu m L_g = 1,$$

whose solution is

$$\eta = \eta_{best} = \frac{1}{\sqrt{\omega\mu m L_g}}.$$

Thus, we see that η_{best} is optimal. Moreover, if $\eta_{best} \le \eta_0$ and $\eta = \eta_{best}$, the two terms in the complexity become equal

$$\frac{1}{\omega\eta_{best}\mu} = m\eta_{best}L_g = \sqrt{\frac{mL_g}{\omega\mu}}.$$

However, if $\eta_{best} > \eta_0$, then $\eta = \min\{\eta_0, \eta_{best}\} = \eta_0$ is relatively small and the dominating term in the complexity is $\frac{1}{\omega\eta\mu}$ rather than $\eta L_g m$. Therefore, the complexity is

$$\mathcal{O}\left(n + m + \frac{1}{\eta_0\mu}\right) = \mathcal{O}\left(n + m + \frac{L}{\mu}\right).$$

Combining the two complexities into one, we get the result. □

F.11 Proof of Lemma 2 (SVRG and SAGA)

Here we consider the update rule by SVRG and SAGA with minibatch of size τ. Following [26], we analyze SVRG and SAGA together by treating them both as memorization methods. In this sense, SAGA simply stores each gradient estimate, $\nabla f_i(u_i^t)$ individually, and SVRG stores only the reference point, u^t, itself and every iteration reevaluate $\nabla f_i(u^t)$ for all sampled i. To avoid any confusion, we provide the explicit formulation of our method with SVRG solver in Algorithm 5.

First of all, let us show that the estimate that we use, v^t, is unbiased.

Lemma 10. *Let us sample a set of indices S of size τ from $\{1, \dots, n\}$. Then, it holds for*

$$v^t := \frac{1}{\tau}\sum_{i \in S}\left(\nabla f_i(x^t) - \nabla f_i(u_i^t) + \alpha^t\right)$$

that it is unbiased

$$\mathbb{E}v^t = \mathbb{E}\nabla f(x^t). \tag{21}$$

Algorithm 5 Stochastic Decoupling Method with SVRG.

```
Input: Stepsize $\eta$, initial vectors $x^0$, $u^0$, $\nabla f(u^0)$, $y_1^0, \dots, y_m^0$, $y^0 = \frac{1}{m}\sum_{j=1}^m y_j^0$, minibatch size $\tau$
for $t = 0, 1, \dots$ do
    Sample subset $S$ from $\{1, \dots, n\}$ of size $\tau$ with probabilities $q_1, \dots, q_n$
    $v^t = \sum_{i \in S} \frac{1}{q_i n}\left(\nabla f_i(x^t) - \nabla f_i(u^t) + \nabla f(u^t)\right)$
    $z^t = \mathrm{prox}_{\eta R}(x^t - \eta v^t - \eta y^t)$
    $u^{t+1} = \begin{cases} x^t, & \text{with probability } \frac{\tau}{n}, \\ u^t, & \text{with probability } 1 - \frac{\tau}{n} \end{cases}$
    Sample $j$ from $\{1, \dots, m\}$ with probabilities $\{p_1, \dots, p_m\}$ and set $\eta_j = \frac{\eta}{m p_j}$
    $x^{t+1} = \mathrm{prox}_{\eta_j g_j}\left(z^t + \eta_j y_j^t\right)$
    $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1})$
    $y^{t+1} = y^t + \frac{1}{m}(y_j^{t+1} - y_j^t)$
end for
```

Proof. Clearly, since i is sampled with probability $\frac{\tau}{n}$, it holds

$$\frac{1}{\tau}\mathbb{E}\left[\nabla f_i(x^t) - \nabla f_i(u_i^t)\right] = \mathbb{E}\left[\frac{1}{n}\sum_{k=1}^n (\nabla f_k(x^t) - \nabla f_k(u_k^t))\right] = \mathbb{E}\left[\nabla f(x^t) - \frac{1}{n}\sum_{k=1}^n \nabla f_k(u_k^t)\right].$$

Therefore, $\mathbb{E}v^t = \mathbb{E}\nabla f(x^t)$. □

We continue our analysis with the following lemma.

Lemma 11. *Consider SVRG and SAGA solver for f. Assume that every f_i is convex and L-smooth and define*

$$\mathcal{M}^t := \frac{3\eta^2}{\tau}\sum_{i=1}^n \mathbb{E}\|\nabla f_i(u_i^t) - \nabla f_i(x^*)\|^2, \tag{22}$$

where for SVRG $u_1^t = \dots = u_n^t$ is the reference point at moment t and for SAGA it is the point whose gradient is saved in memory for function f_i. Then,

$$\mathcal{M}^{t+1} \le \left(1 - \frac{\tau}{n}\right)\mathcal{M}^t + 6\eta^2 L D_f(x^t, x^*).$$

Proof. We have for SVRG that $\mathcal{M}^{t+1}$ changes with probability $\frac{\tau}{n}$ and with probability $1 - \frac{\tau}{n}$ it remains the same. Then,

$$\mathbb{E}\sum_{k=1}^n \|\nabla f_k(u_k^{t+1}) - \nabla f_k(x^*)\|^2 = \frac{\tau}{n}\sum_{k=1}^n \mathbb{E}\|\nabla f_k(x^t) - \nabla f_k(x^*)\|^2 + \left(1 - \frac{\tau}{n}\right)\mathcal{M}^t.$$

Similarly, for SAGA we update exactly τ out of n gradient in the memory, which leads to the following identity:

$$\begin{aligned}
&\mathbb{E}\sum_{k=1}^n \|\nabla f_k(u_k^{t+1}) - \nabla f_k(x^*)\|^2 \\
&\quad = \mathbb{E}\sum_{i \in S} \|\nabla f_i(u_i^{t+1}) - \nabla f_i(x^*)\|^2 + \mathbb{E}\sum_{i \notin S} \|\nabla f_i(u_i^{t+1}) - \nabla f_i(x^*)\|^2 \\
&\quad = \frac{\tau}{n}\sum_{k=1}^n \mathbb{E}\|\nabla f_k(x^t) - \nabla f_k(x^*)\|^2 + \left(1 - \frac{\tau}{n}\right)\mathcal{M}^t.
\end{aligned}$$

In both cases, we obtained the same recursion. Now let us bound the gradient difference in the identity above:

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n}\|\nabla f_k(x^t)-\nabla f_k(x^*)\|^2 &\overset{(13)}{\le} \frac{1}{n}\sum_{k=1}^{n} 2LD_{f_k}(x^t,x^*)\\ &= 2LD_f(x^t,x^*).\end{aligned}$$

This gives us the claimed inequality. □

Now let us show how the recursion looks like when $\mathcal{M}^{t+1}$ is combined with $\|w^{t+1}-w^*\|^2$.

Lemma 12. *Consider the iterates of Algorithm 5 (SVRG) or 6 (SAGA). Let $f_1,\dots,f_n$ be convex and L-smooth, f be μ-strongly convex, S be a subset of $\{1,\dots,n\}$ of size τ sampled with probabilities $q_1,\dots,q_n$. If $\eta\le\frac{1}{5L}$, $\alpha^t=\frac{1}{n}\sum_{i=1}^n\nabla f_i(u_i^t)$ and $w^t=x^t-\eta v^t$ with*

$$v^t=\frac{1}{\tau}\sum_{i\in S}\left(\nabla f_i(x^t)-\nabla f_i(u_i^t)+\alpha^t\right)$$

then we have

$$\mathbb{E}\left[\|w^t-w^*\|^2+\mathcal{M}^{t+1}\right]\le(1-\rho)\left[\|x^t-x^*\|^2+\mathcal{M}^t\right],$$

where $w^*:=x^*-\eta\nabla f(x^*)$ and $\rho:=\min\left\{\eta\mu,\frac{1}{3n}\right\}$.

Proof. It holds

$$\begin{aligned}\mathbb{E}\|w^t-w^*\|^2 &= \mathbb{E}\|x^t-x^*-\eta(v^t-\nabla f(x^*))\|^2\\ &= \mathbb{E}\left[\|x^t-x^*\|^2-2\eta\left\langle x^t-x^*,\mathbb{E}[v^t\mid x^t]-\nabla f(x^*)\right\rangle+\eta^2\|v^t-\nabla f(x^*)\|^2\right]\\ &\overset{(21)}{=} \mathbb{E}\left[\|x^t-x^*\|^2-2\eta\left\langle x^t-x^*,\nabla f(x^t)-\nabla f(x^*)\right\rangle+\eta^2\|v^t-\nabla f(x^*)\|^2\right].\\ &\overset{(15)}{\le} \mathbb{E}\left[(1-\eta\mu)\|x^t-x^*\|^2-2\eta D_f(x^t,x^*)+\eta^2\|v^t-\nabla f(x^*)\|^2\right].\end{aligned}$$

On the other hand, by Jensen's and Young's inequalities

$$\begin{aligned}\mathbb{E}\|v^t-\nabla f(x^*)\|^2 &= \mathbb{E}\left\|\frac{1}{\tau}\sum_{i\in S}(\nabla f_i(x^t)-\nabla f_i(u_i^t))+\alpha^t-\nabla f(x^*))\right\|^2\\ &\le \frac{1}{\tau}\mathbb{E}\sum_{i\in S}\left\|\nabla f_i(x^t)-\nabla f_i(x^*)+\nabla f_i(x^*)-\nabla f_i(u_i^t)+\alpha^t-\nabla f(x^*)\right\|^2\\ &\le \frac{2}{\tau}\mathbb{E}\sum_{i\in S}\left\|\nabla f_i(x^t)-\nabla f_i(x^*)\right\|^2+\frac{2}{\tau}\mathbb{E}\sum_{i\in S}\|\nabla f_i(x^*)-\alpha_i^t+\alpha^t-\nabla f(x^*)\|^2\\ &\overset{(13)}{\le} 4L\mathbb{E}D_f(x^t,x^*)+\frac{2}{n}\sum_{k=1}^{n}\mathbb{E}\left\|\nabla f_k(x^*)-\alpha_k^t+\alpha^t-\nabla f(x^*)\right\|^2.\end{aligned}$$

Using inequality $\mathbb{E}\|X-\mathbb{E}X\|^2\le\mathbb{E}\|X\|^2$ that holds for any random variable X, the second term can be simplified to

$$\begin{aligned}&\frac{2}{n}\sum_{k=1}^{n}\mathbb{E}\left\|\nabla f_k(x^*)-\alpha_k^t+\alpha^t-\nabla f(x^*)\right\|^2\\ &\le \frac{2}{n}\sum_{k=1}^{n}\mathbb{E}\|\nabla f_k(u_k^t)-\nabla f_k(x^*)\|^2\\ &= \frac{2\tau}{3n}\mathcal{M}^t.\end{aligned}$$

Thus,

$$\begin{aligned}\mathbb{E}\left[\|w^t-w^*\|^2+\mathcal{M}^{t+1}\right] &\le (1-\eta\mu)\mathbb{E}\|x^t-x^*\|^2-2\eta\left(1-2\eta L-3\eta L\right)\mathbb{E}D_f(x^t,x^*)\\ &\quad+\left(\left(1-\frac{\tau}{n}\right)+\frac{2\tau}{3n}\right)\mathcal{M}^t. \qquad (23)\end{aligned}$$

The second term in the right-hand side can be dropped as $1 - 2\eta L - \frac{cL}{n\eta} = 1 - 2\eta L - 3\eta L \le 0$. In addition, $\rho \le \eta\mu$ and $\rho \le \frac{1}{3n} = \frac{1}{n} - \frac{2\eta^2}{c}$, so the claim follows. □

Now we are ready to prove Lemma 2.

Lemma 2. *In SVRG and SAGA, if f_i is L-smooth and convex for all i, Assumption 2 is satisfied with $\eta_0 = \frac{1}{6L}$, $\omega = \frac{1}{3}$, $\rho = \frac{1}{3n}$ and*

$$\mathcal{M}^t = \frac{3\eta^2}{n}\sum_{i=1}^n \mathbb{E}\|\nabla f_i(u_i^t) - \nabla f_i(x^*)\|^2,$$

where in SVRG $u_i^t = u^t$ is the reference point of the current loop, and in SAGA u_i^t is the point whose gradient is stored in memory for function f_i. If f is also strongly convex, then Assumption 2 holds with $\eta_0 = \frac{1}{5L}$, $\omega = 1$, $\rho = \frac{1}{3n}$ and the same $\mathcal{M}^t$.

Proof. Equation (23) gives immediately the second part of the claim.

Similarly, if $\eta \le \frac{1}{6L}$, from Equation 23 we obtain by mentioning $1 - 2\eta L - \frac{cL}{n\eta} = 1 - 5\eta L \le \frac{1}{6}$ that

$$\mathbb{E}\left[\|w^t - w^*\|^2 + \frac{c}{n}\sum_{k=1}^n \|\alpha_k^{t+1} - \alpha_k^*\|^2\right] \le \|x^t - x^*\|^2 - \frac{\eta}{3}D_f(x^t, x^*) + \frac{c}{n}\sum_{k=1}^n \|\alpha_i^t - \alpha_i^*\|^2.$$

□

F.12 Proof of Lemma 3 (SGD)

Lemma 3. *Assume that at an optimum x^* the variance of stochastic gradients is finite, i.e. $\sigma_*^2 := \mathbb{E}\|\nabla f_{\xi^t}(x^*) - \nabla f(x^*)\|^2 < +\infty$. Then, SGD that terminates after at most t_0 iterations satisfies Assumption 2(a) with $\eta_0 = \frac{1}{4L}$, $\omega = 1$ and $\rho = 0$. If f is strongly convex, it satisfies Assumption 2(b) with $\eta_0 = \frac{1}{2L}$, $\omega = 1$ and $\rho = 0$. In both cases, sequence $\{\mathcal{M}^t\}_{t=0}^{t_0}$ is given by*

$$\mathcal{M}^t = 2\eta^2(t_0 - t)\sigma_*^2.$$

Proof. Clearly, we have

$$\begin{aligned}\mathbb{E}\|w^t - w^*\|^2 &= \|x^t - x^*\|^2 - 2\eta\left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^*\right\rangle + \eta^2\mathbb{E}\|\nabla f_{\xi^t}(x^t) - \nabla f(x^*)\|^2 \\ &\overset{(20)}{\le} \|x^t - x^*\|^2 - 2\eta\left\langle \nabla f(x^t) - \nabla f(x^*), x^t - x^*\right\rangle + \eta^2\left(4LD_f(x^t, x^*) + 2\sigma_*^2\right) \\ &\overset{(15)}{\le} (1 - \eta\mu)\|x^t - x^*\|^2 - 2\eta(1 - 2\eta L)D_f(x^t, x^*) + 2\eta^2\sigma_*^2.\end{aligned}$$

If f is not strongly convex, then $\mu = 0$ and by assuming $\eta \le \eta_0 = \frac{1}{4L}$ we get $1 - 2\eta L \ge \frac{1}{2}$ and

$$\mathbb{E}\|w^t - w^*\|^2 \le \|x^t - x^*\|^2 - \eta D_f(x^t, x^*) + 2\eta^2\sigma_*^2.$$

In case $\mu = 0$, by defining $\{\mathcal{M}^t\}_{t=0}^{t_0}$ with recursion

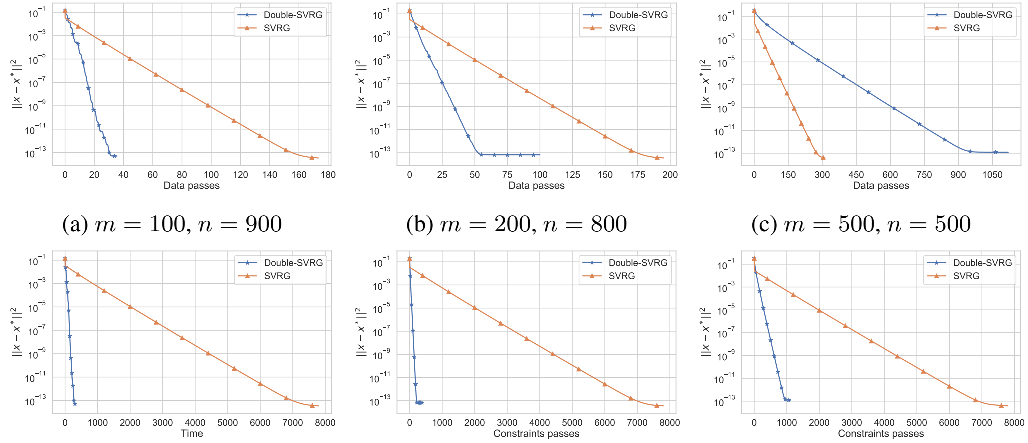
$$\mathcal{M}^{t+1} = \mathcal{M}^t - 2\eta^2\sigma_*^2,$$

we can verify Assumption 2(a) as long as $\mathcal{M}^{t_0} = \mathcal{M}^0 - 2t_0\eta^2\sigma_*^2 \ge 0$. This is the reason we choose $\mathcal{M}^0 = 2t_0\eta^2\sigma_*^2$.

On the other hand, when $\mu > 0$ and $\sigma_* = 0$, it follows from $\eta \le \eta_0 = \frac{1}{2L}$ that

$$\mathbb{E}\|w^t - w^*\|^2 \le (1 - \eta\mu)\|x^t - x^*\|^2.$$

□



(a) $m = 100$, $n = 900$ (b) $m = 200$, $n = 800$ (c) $m = 500$, $n = 500$

(d) $m = 100$, $n = 900$ (e) $m = 200$, $n = 800$ (f) $m = 500$, $n = 500$

Figure 3: Comparison of SVRG with precise projection onto all constraints (labeled as 'SVRG') to our stochastic version of SVRG (labeled as 'Double-SVRG').

G Additional Experiments

Here we want to see how changing m and n affects the comparison between SVRG with exact projection and decoupled SVRG with one stochastic projection. The problem that we consider is again ℓ_2-regularized constrained linear regression. We took Gisette dataset from LIBSVM, whose dimension is $d = 5000$, and used its first 1000 observations to construct f and g. In particular, we split these observations into soft loss $f_i(x) = \frac{1}{2}\|a_i^\top x - b_i\|^2$ and hard constraints $g_j(x) = \chi_{\{x: a_j^\top x = b_j\}}$ with $n + m = 1000$ and we considered three choices of n: 250, 500 and 750. To make sure that the constraints can be satisfied, we generated a random vector x_0 from normal distribution $\mathcal{N}(0, 1/\sqrt{d})$ and set $b = \mathbf{A}x_0$. In all cases, first part of data was used in f and the rest in g. To better see the effect of changing n, we used fixed ℓ_2 penalty of order $1/(n+m)$ for all choices of n.

Computing the projection of a point onto the intersection of all constraints as at least as expensive as m individual projections and we count it as such for SVRG. In practice it might be by orders of magnitude slower than this estimate for big matrices, but the advantage of our method can be seen even without taking it into account. On the other hand, to make the comparison fair in terms of computation trade-off, we use SVRG with minibatch 20 and our method with minibatch 1. The stepsize for both methods is $1/(2L)$.

As we can from Figure 3, the trade-off between projections and gradients changes dramatically when m increases. When $m = 100$, which implies that the term corresponding to $\mathbf{A}$ in the complexity is small, the difference is tremendous, partially because minibatching for SVRG improves only part of its complexity [24]. In the setting $m = n = 500$, we see that the number of data passes taken by our method to solve the problem is a few times bigger than than that taken by SVRG. Clearly, this happens because the term related to $\mathbf{A}$ becomes dominating in the complexity and SVRG uses $m = 500$ times more constraints at each iteration than our method.

H SAGA: Solver for f

To provide a detailed example of how everything should look together, we give here a pseudocode of our method with SAGA.

Algorithm 6 Stochastic Decoupling Method with SAGA.

```
Input: Stepsize $\eta$, initial vectors $x^0$, $\nabla f_1(u_1^0), \ldots, \nabla f_n(u_n^0)$, $\alpha^0 = \frac{1}{n}\sum_{i=1}^n \nabla f_i(u_i^0)$, $y_1^0, \ldots, y_m^0$,
    $y^0 = \frac{1}{m}\sum_{j=1}^m y_j^0$, minibatch size $\tau$
for $t = 0, 1, \ldots$ do
    Sample subset $S$ from $\{1, \ldots, n\}$ of size $\tau$ with probabilities $q_1, \ldots, q_n$
    $v^t = \sum_{i \in S} \frac{1}{q_i n} (\nabla f_i(x^t) - \nabla f_i(u_i^t)) + \alpha^t$
    $z^t = \mathrm{prox}_{\eta R}(x^t - \eta v^t - \eta y^t)$
    for $i \in S$ do
        Update $\nabla f_i(u_i^{t+1})$ with $u_i^{t+1} = x^t$
    end for
    $\alpha^{t+1} = \alpha^t + \frac{1}{n}\sum_{i \in S} \nabla f_i(u_i^{t+1}) - \nabla f_i(u_i^t)$
    Sample $j$ from $\{1, \ldots, m\}$ with probabilities $\{p_1, \ldots, p_m\}$ and set $\eta_j = \frac{\eta}{m p_j}$
    $x^{t+1} = \mathrm{prox}_{\eta_j g_j}\left(z^t + \eta_j y_j^t\right)$
    $y_j^{t+1} = y_j^t + \frac{1}{\eta_j}(z^t - x^{t+1})$
    $y^{t+1} = y^t + \frac{1}{m}(y_j^{t+1} - y_j^t)$
end for
```

While we consider in our theory only uniform probabilities, i.e. $q_1 = \ldots = q_n$, we still borrow the arbitrary sampling oracle from [45] to provide a more general method.

I Table of Key Notation

Objective function F	$F := f + g + R$
Domain of F	$\operatorname{dom} F := \{x \,:\, F(x) < +\infty\} \neq \emptyset$
Primal variable	$x \in \mathbb{R}^d$
Set of optimal solutions	$\mathcal{X}^* := \{x^* \in \mathbb{R}^d \,:\, F(x) \geq F(x^*) \;\forall x \in \mathbb{R}^d\}$ (non-empty)
f in finite-sum form	$f(x) = \frac{1}{n} \sum\limits_{i=1}^{n} f_i(x)$ (f_i are differentiable and convex)
f in expectation form	$f(x) = \mathbb{E}_\xi f_\xi(x)$ (f_ξ are differentiable and convex)
Standard Euclidean norm	$\lVert x \rVert := (\sum_{l=1}^{d} x_l^2)^{1/2}$
Gradient noise at optimum	$\sigma_*^2 := \mathbb{E}_\xi \lVert \nabla f_\xi(x^*) - \nabla f(x^*) \rVert^2$
Smoothness constant of f	L
Strong convexity constant of f	μ
Function g_j	$g_j : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ (proper, closed, convex, proximable)
Function g_j in a structured form	$g_j(x) = \phi_j(\mathbf{A}_j^\top x)$, $\mathbf{A}_j \in \mathbb{R}^{d \times d_j}$
Function ϕ_j	$\phi_j : \mathbb{R}^{d_j} \to \mathbb{R} \cup \{+\infty\}$ (proper, closed, convex, proximable)
Function g	$g(x) := \frac{1}{m} \sum\limits_{j=1}^{m} g_j(x)$ (proper, closed, convex)
Function R	$R(x) : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ (proper, closed, convex, proximable)
Primal iterates	$x^t \in \mathbb{R}^d$
Estimator of $\nabla f(x^t)$	v^t
Dual variables	$y_1^t, \dots, y_j^t \in \mathbb{R}^d$
Probability of selecting index j	p_j
Stepsize associated with f and R	η
Stepsize associated with g_j	$\eta_j = \frac{\eta}{m p_j}$
Lyapunov function	$\mathcal{L}^t := \mathbb{E} \lVert x^t - x^* \rVert^2 + \mathcal{M}^t + \mathcal{Y}^t$
$\mathcal{Y}^t$	$\mathcal{Y}^t := (1 + \gamma) \sum_{k=1}^{m} \eta_j^2 \mathbb{E} \lVert y_j^t - y_j^* \rVert^2$
Smoothness constant of g_j	$L_j \in \mathbb{R} \cup \{+\infty\}$ ($L_j = +\infty$ if g_j is non-smooth)
Parameter γ	$\gamma := \min_{j=1,\dots,m} \frac{1}{\eta_j L_j}$ ($\gamma = 0$ if any g_j is non-smooth)
Subdifferential of R	$\partial R(x) := \{s \,:\, R(y) \geq R(x) + \langle s, y - x \rangle\}$, $x \in \operatorname{dom} R$
Proximal operator of function R	$\operatorname{prox}_{\eta R}(x) := \operatorname{argmin}_u \left\{ R(u) + \frac{1}{2\eta} \lVert u - x \rVert^2 \right\}$
Proximal operator of function g_j	$\operatorname{prox}_{\eta_j g_j}(x) := \operatorname{argmin}_u \left\{ g_j(u) + \frac{1}{2\eta_j} \lVert u - x \rVert^2 \right\}$
Characteristic function of a set $\mathcal{C}$	$\chi_{\mathcal{C}}(x) := \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & x \notin \mathcal{C} \end{cases}$
Projection onto set $\mathcal{C}$	$\Pi_{\mathcal{C}}(x) := \operatorname{prox}_{\chi_{\mathcal{C}}}(x) = \operatorname{argmin}_{u \in \mathcal{C}} \lVert u - x \rVert$
Bregman divergence of f	$D_f(x, y) := f(x) - f(y) - \langle \nabla f(x), x - y \rangle$

Table 4: Key notation used in this paper.