Algorithms for Large-scale Regularized Optimization

by

LEE Ching-pei

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 05/22/2019

The dissertation is approved by the following members of the Final Oral Committee:

Stephen J. Wright, Professor, Computer Sciences
Michael C. Ferris, Professor, Computer Sciences
Po-Ling Loh, Assistant Professor, Statistics
Jerry Zhu, Professor, Computer Sciences
Dimitris Papailiopoulos, Assistant Professor, Electrical and Computer Engineering

To my family.

ACKNOWLEDGMENTS

I would like to give my sincerest gratitude to my advisor, Stephen Wright. Four years ago, he kindly accepted me, who had only limited knowledge in optimization back then, as his student when I just decided to come to Madison. Steve has shaped my penchant and viewpoint for optimization research, equipped me with the ability to conduct research, and showed me as a role model what a great researcher is like, and how to balance research and life. Without his guidance, I could not get to where I stand now, with confidence and knowledge in optimization, and devoted to be a researcher for this area. Becoming a researcher like him is a goal that I have been, and will continue, pursuing.

I would also like to thank all my committee members, Michael, Po-Ling, Jerry, and Dimitris, for their advices and suggestions. The suggestions from Michael and Po-Ling in my preliminary exam has especially helped improve the quality of the final dissertation.

My master's advisor, Chih-Jen Lin, opened my door to academic research. Although that was a different research area, I thank him for forming my attitude toward solid research. Popo has been my mentor during my master's years and without him I would have not joined Prof. Lin's group and not to mention all the consequential steps that led to my Ph.D study.

Po-Wei is the person who made me decide to pursue research in optimization. I have to admit that I was not able to understand many research-related things he described to me while I was still a master's student, but even just those parts that I had a grasp of, I sensed how beautiful and astonishing optimization is as a research area. Discussions with him have also established my research standard and taste to a great deal. For now, with much more knowledge I gained, discussing with him has become inspiring and motivating for me.

I would also like to thank the office mates and group mates for the time we spent together in WID, especially to Cong, Mike, Clement, Silvia, Kwang, and Amanda. Cong was super friendly to me since I arrived in Madison, and he has provided lots of essential information throughout the process. I also enjoyed all the random discussions in politics, films, and life in Madison.

Badminton club is the major part of my off-research life in Madison. I have made great friends that we not only play badminton together but also share our lives and support each other. Yutong, Weiyue, Andy, and Donny have been there since I joined the group, and I enjoyed the time spent with them. Sandro, Adrien, Dan, Haowei, and Hongchi spent much time playing singles with me; Steph, It, and Hide are great presidents and I enjoyed playing with them a lot; Xinwu regarded me as a coach for her, though I was just providing advices for her, but I also appreciate the time we trained together. Martin has been a very good friend throughout all the years, and I appreciate all the moments we spent on court and off court. I can still remember those profound conversations we had on various issues and I also thank him for all the helps throughout the years.

My friends in Taiwan have also been an important part for me to go through the process of PhD. My thank and love to Xian, Iya, KSM, and Dai. Finally, I thank to dearest family for their unconditioned support for my capricious decisions and the changes in my life plans. I love them all and I would like to dedicate the dissertation to them.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Regularized optimization that minimizes the sum of a smooth and a nonsmooth function is widely seen in applications including signal processing, engineering, and data analysis, where the smooth term is for data fitting and the nonsmooth term is for promoting desirable properties in the solution. Optimization methods that utilize smoothness of the data-related term are usually preferred for this type of problems as they have close relation with the unregularized, smooth counterparts. In this dissertation, we study efficient methods for large-scale regularized optimization problems, covering second- and first-order methods as well as incremental methods that update a block of variables at a time. We provide theory for convergence speed, novel algorithms with efficient practical performance, and analysis for existing methods sharper than existing theory. We also discuss its application in distributed implementations, in which multiple machines are used to cope with extremely large-scale data sets.

1 INTRODUCTION

Nonsmooth optimization arises naturally in many important applications, especially as a means of inducing desired properties in the solution of the optimization problem. For general nonsmooth optimization problems, usually we can only rely on subgradient-type methods that converge slowly. Fortunately, by utilizing problem structures in specific nonsmooth optimization settings, various algorithms with much better convergence guarantees can be designed. Among them, regularized optimization is a natural extension from smooth optimization, which covers many widely-used optimization problems in machine learning, signal processing, statistics, just to name a few.

In regularized optimization, a common setting is that there is one smooth term and one nonsmooth but convex term, and the appearance of the latter usually generates a regularized solution with desired property such as sparsity or feasibility. One popular special case is constrained optimization of the form

$$\begin{aligned} &\min_{x\mathbb{R}^n} f_0(x) \\ &\text{subject to } f_i(x) \leq 0, i = 1, \ldots, m, \end{aligned} \tag{1.1}$$

where f_0 is differentiable and $f_i, i = 1, \ldots, m$ are convex. We can consider the constraints as a form of regularization, and reformulate the problem as an equivalent regularized optimization problem using an indicator function.

$$\min_x f_0(x) + \mathbb{1}_{\{x|f_i(x)\leq 0, i=1,\ldots,m\}}(x).$$

Another widely seen case is that the regularization term is a certain norm on the variables, either smooth or nonsmooth, to control certain properties of the solution. This type of minimization problem can be

traced back to the 40s by Tikhonov to cope with ill-conditioned problems, where the regularization is the squared-Euclidean norm (Tikhonov, 1943) and the data-related term is linear least square regression, resulting in the following problem.

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 + \lambda\|x\|_2^2, \tag{1.2}$$

where A is a given data matrix, b is the target to approximate, and $\lambda \geq 0$ is a user-specified parameter to balance the two terms. In statistics, (1.2) is also known as ridge regression. The main purpose of (1.2) is to generate a solution of bounded norm, therefore avoiding overfitting the given data and expecting for better generalization ability on new data points. See, for example, Shalev-Shwartz and Ben-David (2014, Chapter 13). From their respective optimality conditions, (1.2) generates the same solution as (1.1) with $f_0(x) = \|Ax-b\|_2^2$ and $m = 1, f_1(x) = \|x\|_2^2 - \rho$ for some $\rho \geq 0$. The regularization term in (1.2) also serves as a way to improve the problem condition and make it easier to solve than the nonregularized counterpart. In other cases, however, regularized or constrained versions are harder to solve.

The squared-Euclidean norm is a very special case, and most other widely considered regularized problems are nonsmooth. One of the most considered problem is the LASSO (Tibshirani, 1996).

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 + \lambda\|x\|_1. \tag{1.3}$$

The ℓ_1 norm in (1.3) induces sparse solutions (that is, x contains few nonzero elements) and has important applications in high-dimensional settings and feature selection. In signal processing, this formulation is used for compressive sensing (Donoho et al., 2006; Baraniuk, 2007) for recovering signals and can be applied in image denoising. The problem (1.3) can be extended to accommodate problems beyond the least square

regression, including classification and ordinal regression. Another important extension of (1.3) that induces sparsity in the individual variables is to use $\ell_{2,1}$ norms to induce group sparsity in the blocks of the variables.

From the Bayesian view, these norms as the regularization can also be interpreted as in the maximum likelihood estimation a certain prior of the model, or the solution to the optimization problem (Rasmussen, 2003), where we assume in addition that the model follows a given probability distribution. For example, the ℓ_1 norm is equivalent to assuming a Laplace distribution centered at 0:

$$p(x) = \frac{\lambda}{2} \exp\left(-\lambda x\right).$$

Similarly, the squared-Euclidean norm assumes a Gaussian distribution of the solution centered at 0 with standard deviation $1/\lambda$.

One can also extend the discussion to matrix variables, where the counterparts to the ℓ_1 norm and the ℓ_2 norm (squared-Euclidean norm) are the nuclear norm and the Frobenius norm, respectively. The former is the sum of the singular values of the matrix variable, while the latter is the sum of entry-wise square. The nuclear norm induces low-rank solutions, while the Frobenius norm controls the overall size of the matrix.

These applications illustrate the popularity and importance of regularized optimization. Unfortunately, except for some special cases, they are difficult to minimize due to the presence of nonsmoothness. General-purpose methods for nonsmooth optimization tend to converge slowly and are not suitable for these problems that have a special structure such that there is one smooth term. In the special case of (1.1), under the additional assumption that projection to the feasible set can be performed efficiently, the problem structure has long been utilized to extend smooth optimization algorithms to obtain more efficient solvers for this setting, including the famous gradient projection method (Rosen, 1960) and the projected Newton method (Bertsekas, 1982).

By generalizing the projection operation, these projection methods are extended to the type of proximal methods that can deal with nonsmooth regularizations beyond indicator to the feasible set. Given a regularization term ψ and a norm $\|\cdot\|$, the proximal operator that being used is defined as

$$\text{prox}_{\psi}^{\|\cdot\|}(v) := \arg\min_{x} \ \psi(x) + \frac{1}{2}\|x - v\|^2, \tag{1.4}$$

which covers the projection operation as a special case in which ψ is the indicator function of the feasible set. Proximal methods are also called operator splitting methods (Lions and Mercier, 1979), and include the proximal gradient method (by taking the Euclidean norm) and the proximal Newton method (by taking the norm induced by the Hessian of f, assuming positive-definiteness) as prominent examples. Similar to the projection approach, the proximal approach assumes that the operation (1.4) can be conducted efficiently. A more detailed survey on proximal algorithms can be found in the monograph Parikh and Boyd (2014).

An advantage of regularized optimization is that usually one can extend existing algorithms for smooth optimization to it with little effort theoretically, to obtain convergence guarantees similar to the smooth case. Among them, the most well-studied is proximal gradient and its accelerated variant, but studies for other algorithms are rather limited, mostly because of the difficulty of finding the exact solution to the proximal operator (1.4). This comes from two practicality issues in the proximal operator. The first one is the assumption of a simple structure in the function ψ, which may not be the case beyond simple cases such as when ψ is the ℓ_1-norm or an indicator function for simple sets. Even when the feasible set is simply a polyhedron, the projection can be time consuming. Secondly, when the norm used in the proximal operator is not the Euclidean norm, the computation can also be difficult even if ψ has a simple structure.

In this dissertation, algorithms for regularized optimization problems are studied. We generalize the proximal algorithms to consider the cases

that the subproblem (1.4) cannot be solved to optimality but just approximately, possibly from another iterative process. This flexibility extends the possibility of operator splitting to make second-order methods practically feasible. Under a mild block-separability assumption of the regularization term, we further develop second-order block-coordinate approaches that can converge much faster than the first-order ones. We also study its realization in applications with extremely large problem size where multiple processors are required to process the problem-defining data.

1.1 Overview

Chapters 2-5 are derived from our papers Lee and Wright (2019b, 2018a, 2019a); Lee et al. (2018), respectively.

We first consider a general framework that includes proximal Newton and proximal quasi-Newton as special classes in Chapter 2. The central idea in this chapter is to consider approximate solutions of the proximal operator (1.4) with arbitrary precision defined in terms of the objective value of the proximal operation. This relaxation therefore removes the necessity of assuming simple structure in the regularizer and allows broader choices of the norm used to define the proximal operator. Under only the assumption that the smooth term is Lipschitz-continuously differentiable, We show that this framework can deal with both convex and nonconvex problems, and we give detailed convergence rates. By using inexact proximal operations, our framework makes extensions of high-order methods for smooth optimization to the regularized setting empirically feasible. We discuss in detail how the inexactness in the proximal operation affects the convergence rates, and in particular show that the impact is mild both theoretically and empirically. In the detailed convergence analysis, we show that global linear convergence of our framework holds on a wide class of problems beyond the strongly convex ones, and local linear con-

vergence can be obtained when a sharpness condition holds. For convex problems, the widely observed phenomenon that many algorithms can obtain an approximate solution with medium precision is justified in theory, and this behavior is used to sharpen existing iteration complexities. We further show that when the problem is nonconvex, our framework generates iterates that converge to the set of stationary points, on which the point zero is included in the set of generalized gradient.

In Chapter 3, we extend the framework to a stochastic second-order block-coordinate descent approach for problems with the additional assumption that the regularization term is block-separable. This framework can utilize the problem structure to have much lower cost per iteration by updating just one block of the variables. In particular, the proximal operator (1.4) is applied on just one block of the variables while other parts remain intact. Within each block, our framework allows flexible use of higher-order information of the smooth term with line search to accelerate the convergence. By picking the blocks using a certain probability distribution, we show that faster convergence rates in the expected objective value can be obtained. When applied to the special case of stochastic proximal coordinate descent, our result shows that sampling according to the block-wise Lipschitz constant can improve the convergence rate greatly both in the convex and the nonconvex setting. These results generalize the sampling strategy of coordinate descent for smooth optimization proposed by Nesterov (Nesterov, 2012) to regularized optimization. We also show how to make use of the second-order derivative and conduct line search with low implementation cost.

After discussion of the general frameworks in Chapters 2 and 3, we turn to specific algorithms in Chapter 4. Our discussion focus on special properties of first-order methods, in particular proximal gradient and proximal coordinate descent. We show that when applied on convex problems, these algorithms provides an implicit regularization such that the

iterates generated lie within a bounded region even without such a constraint given in the optimization problem. This implicit regularization is then used to improve the $O(1/k)$ convergence we obtained in the previous chapters to $o(1/k)$, where k is the iteration counter. As gradient descent and coordinate descent are special cases of proximal gradient and proximal coordinate descent, the well-established convergence speed of these fundamental and extensively studied algorithms are also improved by our analysis.

In Chapter 5, we tackle a real-world application of regularized optimization in distributed optimization, where multiple machines are connected through a local network to store and process data with extremely large volume. In particular, we consider the problem setting in which the smooth term $f(x)$ is of the form $\tilde{f}(A^\top x)$, where $\tilde{f}$ is Lipschitz-continuously differentiable and A is the data matrix whose columns are disjointly stored on multiple machines. This setting of distributed optimization arises naturally when A has size beyond the capacity of a single machine and when A is by nature collected in a distributed manner. The major bottleneck of this setting is that synchronization of the variables and the computation of the gradient requires expensive inner-machine communication, which can be magnitudes slower than accessing data locally. By utilizing the framework proposed in Chapter 2, we propose a communication- and computation-efficient proximal quasi-Newton-type algorithm for this application. Prior to our approach, only first-order methods can be used in this setting because of the distributed data storage, but in practice first-order methods can be unsatisfactorily slow for any real applications. By properly utilizing the gradient of the smooth term from previous iterations, our framework provides much faster empirical performance than state of the art for distributed optimization.

2 INEXACT SUCCESSIVE QUADRATIC APPROXIMATION FOR REGULARIZED OPTIMIZATION

2.1 Introduction

In this chapter, we consider the following regularized optimization problem:

$$\min_x \; F(x) \coloneqq f(x) + \psi(x), \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz-continuously differentiable, and $\psi : \mathbb{R}^n \to \mathbb{R}$ is convex, extended-valued, proper, and closed, but might be nondifferentiable. Moreover, we assume that F is lower-bounded and the solution set Ω of (2.1) is non-empty. Unlike the many other works on this topic, we focus on the case in which ψ does *not* necessarily have a simple structure, such as (block) separability, which allows a prox-operator to be calculated economically, often in closed form. Rather, we assume that subproblems that involve ψ explicitly are solved inexactly, by an iterative process.

Problems of the form (2.1) arise in many contexts. The function ψ could be an indicator function for a trust region or a convex feasible set. It could be a multiple of an ℓ_1 norm or a sum-of-ℓ_2 norms. It could be the nuclear norm for a matrix variable, or the sum of absolute values of the elements of a matrix. It could be a smooth convex function, such as $\|\cdot\|_2^2$ or the squared Frobenius norm of a matrix. Finally, it could be a combination of several of these elements, as happens when different types of structure are present in the solution. In some of these situations, the prox-operator involving ψ is expensive to calculate exactly.

We consider algorithms that generate a sequence $\{x^k\}_{k=0,1,\ldots}$ from some starting point x^0, and solve the following subproblem inexactly at each

iteration, for some symmetric matrix H_k:

$$\arg\min_{d\in\mathbb{R}^n} Q_{H_k}^{x^k}(d) := \nabla f\left(x^k\right)^T d + \frac{1}{2} d^T H_k d + \psi\left(x^k + d\right) - \psi\left(x^k\right). \quad (2.2)$$

We abbreviate the objective in (2.2) as $Q_k(\cdot)$ (or as $Q(\cdot)$ when we focus on the inner workings of iteration k). In some results, we allow H_k to have zero or negative eigenvalues, provided that Q_k itself is strongly convex. (Strong convexity in ψ may overcome any lack of strong convexity in the quadratic part of (2.2).)

In the special case of the proximal-gradient algorithm (Combettes and Wajs, 2005; Wright et al., 2009), where H_k is a positive multiple of the identity, the subproblem (2.2) can often be solved cheaply, particularly when ψ is (block) separable, by means of a prox-operator involving ψ. For more general choices of H_k, or for more complicated regularization functions ψ, it may make sense to solve (2.2) by an iterative process, such as accelerated proximal gradient or coordinate descent. Since it may be too expensive to run this iterative process to obtain a high-accuracy solution of (2.2), we consider the possibility of an inexact solution. In this chapter, we assume that the inexact solution satisfies the following condition, for some constant $\eta \in [0, 1)$:

$$Q(d) - Q^* \leq \eta\left(Q(0) - Q^*\right) \quad \Leftrightarrow \quad Q(d) \leq (1-\eta)Q^*, \quad (2.3)$$

where $Q^* := \inf_d Q(d)$ and $Q(0) = 0$. The value $\eta = 0$ corresponds to exact solution of (2.2). Other values $\eta \in (0, 1)$ indicate solutions that are inexact to within a *multiplicative* constant.

The condition (2.3) is studied in (Bonettini et al., 2016, Section 4.1), which applies a primal-dual approach to (2.2) to satisfy it. In this connection, note that if we have access to a lower bound $Q_{LB} \leq Q^*$ (obtained by finding a feasible point for the dual of (2.2), or other means), then any d satisfying $Q(d) \leq (1-\eta)Q_{LB}$ also satisfies (2.3).

In practical situations, we need not enforce (2.3) explicitly for some chosen value of η. In fact, we do not necessarily require η to be known, or (2.3) to be checked at all. Rather, we can take advantage of the convergence rates of whatever solver is applied to (2.2) to ensure that (2.3) holds for *some* value of $\eta \in (0, 1)$, possibly unknown. For instance, if we apply an iterative solver to the strongly convex function Q in (2.2) that converges at a global linear rate $(1 - \tau)$, then the "inner" iteration sequence $\{d^{(t)}\}_{t=0,1,\ldots}$ (starting from some $d^{(0)}$ with $Q(d^{(0)}) \leq 0$) satisfies

$$Q(d^{(t)}) - Q^* \leq (1 - \tau)^t \left(Q\left(0\right) - Q^*\right), \quad t = 0, 1, 2, \ldots . \tag{2.4}$$

If we fix the number of inner iterations at T (say), then $d^{(T)}$ satisfies (2.3) with $\eta = (1-\tau)^T$. Although τ might be unknown as well, we can implicitly tune the accuracy of the solution by adjusting T. On the other hand, if we wish to attain a certain target accuracy η and have an estimate of rate τ, we can choose the number of iterations T large enough that $(1 - \tau)^T \leq \eta$. Note that τ depends on the extreme eigenvalues of H_k in some algorithms; we can therefore choose H_k to ensure that τ is restricted to a certain range for all k.

Empirically, we observe that Q-linear methods for solving (2.2) often have rapid convergence in their early stages, with slower convergence later. Thus, a moderate value of η may be preferable to a smaller value, because moderate accuracy is attainable in disproportionately fewer iterations than high accuracy.

A practical stopping condition for the subproblem solver in our framework is just to set a fixed number of iterations, provided that a linearly convergent method is used to solve (2.2). This guideline can be combined with other more sophisticated approaches, possibly adjusting the number of inner iterations (and hence implicitly η) according to some heuristics. For simplicity, our analysis assumes a fixed choice of $\eta \in (0, 1)$. We examine in particular the number of outer iterations required to solve (2.1) to a

given accuracy ϵ. We show that the dependence of the iteration complexity on the inexactness measure η is benign, increasing only modestly with η over approaches that require exact solution of (2.2) for each k.

Quadratic Approximation Algorithms

To build complete algorithms around the subproblem (2.2), we either do a step size line search along the inexact solution d^k, or adjust H_k and recompute d^k, seeking in both cases to satisfy a familiar "sufficient decrease" criterion. We present two algorithms that reflect each of these approaches. The first uses a line search approach on the step size with a modified Armijo rule, as presented in Tseng and Yun (2009). We consider a backtracking line-search procedure for simplicity; the analysis could be adapted for more sophisticated procedures. Given the current point x^k, the update direction d^k and parameters $\beta, \gamma \in (0, 1)$, backtracking finds the smallest nonnegative integer i such that the step size $\alpha_k = \beta^i$ satisfies

$$F\left(x^k + \alpha_k d^k\right) \leq F\left(x^k\right) + \alpha_k \gamma \Delta_k, \tag{2.5}$$

where

$$\Delta_k \coloneqq \nabla f\left(x^k\right)^T d^k + \psi\left(x^k + d^k\right) - \psi\left(x^k\right). \tag{2.6}$$

This version appears as Algorithm 1. The exact version of this algorithm can be considered as a special case of the block-coordinate descent algorithm of Tseng and Yun (2009).[1] In Bonettini et al. (2016), Algorithm 1 (with possibly a different criterion on d^k) is called the "variable metric inexact line-search-based method". (We avoid the term "metric" because we consider the possibility of indefinite H_k in some of our results.) More complicated metrics, not representable by a matrix norm, were also con-

[1] The definition of Δ in Tseng and Yun (2009) contains another term $\omega d^T H d/2$, where $\omega \in [0, 1)$ is a parameter. We take $\omega = 0$ for simplicity, but our analysis can be extended in a straightforward way to the case of $\omega \in (0, 1)$.

sidered in Bonettini et al. (2016). Since our analysis makes use only of the smallest and largest eigenvalues of H_k (which correspond to the strong convexity and Lipschitz continuity parameters of the quadratic approximation term), we could also generalize our approach to this setting. We present only the matrix-representable case, however, as it allows a more direct comparison with the second algorithm presented next.

Algorithm 1 Inexact Successive Quadratic Approximation with Backtracking Line Search

Given $\beta, \gamma \in (0,1)$, $x^0 \in \mathbb{R}^n$;
for $k = 0, 1, 2, \ldots$ **do**
  Choose a symmetric H_k that makes Q_k strongly convex;
  Obtain from (2.2) a vector d^k satisfying (2.3), for some fixed $\eta \in [0, 1)$;
  Compute Δ_k by (2.6);
  $\alpha_k \leftarrow 1$;
  while (2.5) is not satisfied **do**
    $\alpha_k \leftarrow \beta \alpha_k$;
  $x^{k+1} \leftarrow x^k + \alpha_k d^k$;

The second algorithm uses the following sufficient decrease criterion from Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018):

$$F(x) - F(x+d) \geq -\gamma Q_H^x(d) \geq 0, \tag{2.7}$$

for a given parameter $\gamma \in (0, 1]$. If this criterion is not satisfied, the algorithm modifies H and recomputes d^k. The criterion (2.7) is identical to that used by trust-region methods (see, for example, (Nocedal and Wright, 2006, Chapter 4)), in that the ratio between the actual objective decrease and the decrease predicted by Q is bounded below by γ; that is,

$$\frac{F(x) - F(x+d)}{Q_H^x(0) - Q_H^x(d)} \geq \gamma.$$

We consider two variants of modifying H such that (2.7) is satisfied.

The first successively increases H by a factor β^{-1} (for some parameter $\beta \in (0,1)$) until (2.7) holds. We require in this variant that the initial choice of H is positive definite, so that all eigenvalues grow by a factor of β^{-1} at each multiplication. The second variant uses a similar strategy, except that H is modified by adding a successively larger multiple of the identity, until (2.7) holds. (This algorithm allows negative eigenvalues in the initial estimate of H.) These two approaches are defined as the first and the second variants of Algorithm 2, respectively.

Algorithm 2 Inexact Successive Quadratic Approximation with Modification of the Quadratic Term

Given $\beta, \gamma \in (0,1]$, $x^0 \in \mathbb{R}^n$;
for $k = 0, 1, 2, \ldots$ **do**
  if Variant 1 **then** Choose $H_k^0 \succ 0$;
  if Variant 2 **then** Choose a suitable H_k^0;
  $\alpha_k \leftarrow 1$, $H_k \leftarrow H_k^0$;
  Obtain from (2.2) a vector d^k satisfying (2.3), for some fixed $\eta \in [0,1)$;
  while (2.7) is not satisfied **do**
    if Variant 1 **then** $\alpha_k \leftarrow \beta\alpha_k$, $H_k \leftarrow H_k^0/\alpha_k$;
    if Variant 2 **then** $H_k \leftarrow H_k^0 + \alpha_k^{-1} I$, $\alpha_k \leftarrow \beta\alpha_k$;
    Obtain from (2.2) a vector d^k satisfying (2.3);
  $x^{k+1} \leftarrow x^k + d^k$;

Algorithm 1 and Variant 1 of Algorithm 2 are direct extensions of backtracking line search in the smooth case, in the sense that when ψ is not present, both approaches are identical to shrinking the step size. However, aside from the sufficient decrease criteria, the two differ when the regularization term is present.

The second variant of Algorithm 2 is similar to the method proposed in Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018), with the only difference being the inexactness criterion of the subproblem solution. This variant of modifying H can be seen as interpolating between the step from

the original H and the proximal gradient step. It is also a generalization of the trust-region technique for smooth optimization. When ψ is not present, adding a multiple of the identity to H in (2.2) is equivalent to shrinking the trust region (Moré and Sorensen, 1983). We can therefore think of Algorithm 2, Variant 2 as a generalized trust-region approach for regularized problems.

Rather than our multiplicative criterion (2.3), the works Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018) use an *additive* criterion to measure inexactness of the solution. In the analysis of Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018), this tolerance must then be reduced to zero at a certain rate as the algorithm progresses, resulting in growth of the number of inner iterations per outer iteration as the algorithms progress. By contrast, we attain satisfactory performance (both in theory and practice) for a fixed value $\eta \in (0, 1)$ in (2.3).

Which of the algorithms described above is "best" depends on the circumstances. When (2.2) is expensive to solve, Algorithm 1 may be preferred, as it requires approximate solution of this subproblem just once on each outer iteration. On the other hand, when ψ has special properties, such as inducing sparsity or low rank in x, Algorithm 2 might benefit from working with sparse iterates and solving the subproblem in spaces of reduced dimension.

Variants and special cases of the algorithms above have been discussed extensively in the literature. Proximal gradient algorithms have $H = \xi I$ for some $\xi > 0$ (Combettes and Wajs, 2005; Wright et al., 2009); proximal-Newton uses $H = \nabla^2 f$ (Lee et al., 2014; Rodomanov and Kropotov, 2016; Li et al., 2017a); proximal-quasi-Newton and variable metric use quasi-Newton approximations for H_k (Scheinberg and Tang, 2016; Ghanbari and Scheinberg, 2018). The term "successive quadratic approximation" is also used by Byrd et al. (2016). Our methods can even be viewed as a special case of block-coordinate descent (Tseng and Yun, 2009) with a

single block. The key difference in this work is the use of the inexactness criterion (2.3), while existing works either assume exact solution of (2.2), or use a different criterion that requires increasing accuracy as the number of outer iterations grows. Some of these works provide only an asymptotic convergence guarantee and a local convergence rate, with a lack of clarity about when the fast local convergence rate will take effect. An exception is Bonettini et al. (2016), which also makes use of the condition (2.3). However, Bonettini et al. (2016) gives convergence rate only for convex f and requires existence of a scalar $\mu \geq 1$ and a sequence $\{\zeta_k\}$ such that

$$\sum_{k=0}^{\infty} \zeta_k < \infty, \quad \zeta_k \geq 0, \quad H_{k+1} \preceq (1+\zeta_k) H_k, \quad \mu I \succeq H_k \succeq \frac{1}{\mu} I, \quad \forall k, \tag{2.8}$$

where $A \succeq B$ means that $A - B$ is positive semidefinite. This condition may preclude such useful and practical choices of H_k as the Hessian and quasi-Newton approximations. We believe that our setting may be more general, practical, and straightforward in some situations.

Contribution

This chapter shows that, when the initial value of H_k at all outer iterations k is chosen appropriately, and that (2.3) is satisfied for all iterations, then the objectives of the two algorithms converge at a global Q-linear rate under an "optimal set strong convexity" condition defined in (2.10), and at a sublinear rate for general convex functions. When F is nonconvex, we show sublinear convergence of the first-order optimality condition. Moreover, to discuss the relation between the subproblem solution precision and the convergence rate, we show that the iteration complexity is proportional to either $1/(1-\eta)$ or $1/(2(1-\sqrt{\eta}))$, depending on the properties of f and ψ, and the algorithm parameter choices.[2]

[2]Note that for $\eta \in [0, 1)$, $1/(1-\eta) > 1/(2(1-\sqrt{\eta}))$.

In comparison to existing works, our major contributions in this chapter are as follows.

- We quantify how the inexactness criterion (2.3) affects the step size of Algorithm 1, the norm of the final H in Algorithm 2, and the iteration complexity of these algorithms. We discuss why the process for finding a suitable value of α_k in each algorithm can potentially improve the convergence speed when the quadratic approximations incorporate curvature information, leading to acceptance of step sizes whose values are close to one.
- We provide a global convergence rate result on the first-order optimality condition for the case of nonconvex f in (2.1) for general choices of H_k, without assumptions beyond the Lipschitzness of ∇f.
- The global R-linear convergence case of a similar algorithm in Ghanbari and Scheinberg (2018) when F is strongly convex is improved to a global Q-linear convergence result for a broader class of problems.
- For general convex problems, in addition to the known sublinear $(1/k)$ convergence rate, we show linear convergence with a rate independent of the conditioning of the problem in the early stages of the algorithm.
- Faster linear convergence in the early iterations also applies to problems with global Q-linear convergence, explaining in part the empirical observation that many methods converge rapidly in their early stages before settling down to a slower rate. This observation also allows improvement of iteration complexities.

Related Work

Our general framework and approach, and special cases thereof, have been widely studied in the literature. Some related work has already been

discussed above. We give a broader discussion in this section.

When ψ is the indicator function of a convex constraint set, our approach includes an inexact variant of a constrained Newton or quasi-Newton method. There are a number of papers on this approach, but their convergence results generally have a different flavor from ours. They typically show only asymptotic convergence rates, together with global convergence results without rates, under weaker smoothness and convexity assumptions on f than we make here. For example, when ψ is the indicator function of a "box" defined by bound constraints, Conn et al. (1988) applies a trust-region framework to solve (2.2) approximately, and shows asymptotic convergence. The paper Byrd et al. (1995) uses a line-search approach, with H_k defined by an L-BFGS update, and omits convergence results. For constraint sets defined by linear inequalities, or general convex constraints, Burke et al. (1990) shows global convergence of a trust region method using the Cauchy point. A similar approach using the exact Hessian as H_k is considered in Lin and Moré (1999), proving local superlinear or quadratic convergence in the case of linear constraints.

Turning to our formulation (2.1) in its full generality, Algorithm 1 is analyzed in Bonettini et al. (2016), which refers to the condition (2.3) as "η-approximation." (Their η is equivalent to $1 - \eta$ in our notation.) This paper shows asymptotic convergence of $Q_k(d)$ to zero without requiring convexity of F, Lipschitz continuity of ∇f, or a fixed value of η. The only assumptions are that $Q_k(d^k) < 0$ for all k and the sequence of objective function values converges (which always happens when F is bounded below). Under the additional assumptions that ∇f is Lipschitz continuous, F is convex, (2.8), and (2.3), they showed convergence of the objective value at a $1/k$ rate. The same authors considered convergence for nonconvex functions satisfying a Kurdyka-Łojasiewicz condition in Bonettini et al. (2017), but the exact rates are not given. Our results differ in not requiring the assumption (2.8), and we are more explicit about the dependence of

the rates on η. Moreover, we show detailed convergence rates for several additional classes of problems.

A version of Algorithm 2 without line search but requiring H_k to *overestimate* the Hessian, as follows:

$$f(x^k + d) \leq f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d$$

is considered in Chouzenoux et al. (2014). Asymptotic convergence is proved, but no rates are given.

Convergence of an inexact proximal-gradient method (for which $H_k = LI$ for all k) is discussed in Schmidt et al. (2011). With this choice of H_k, (2.7) always holds with $\gamma = 1$. They also discuss its accelerated version for convex and strongly convex problems. Instead of our multiplicative inexactness criterion, they assume an additive inexactness criterion in the subproblem, of the form

$$Q_k\left(d^k\right) \leq Q_k^* + \epsilon_k. \tag{2.9}$$

Their analysis also allows for an error e^k in the gradient term in (2.2). The paper shows that for general convex problems, the objective value converges at a $1/k$ rate provided that $\sum_k \sqrt{\epsilon_k}$ and $\sum_k \|e^k\|$ converge. For strongly convex problems, they proved R-linear convergence of $\|x^k - x^*\|$, provided that the sequence $\{\|e^k\|\}$ and $\{\sqrt{\epsilon_k}\}$ both decrease linearly to zero. When our approaches are specialized to proximal gradient ($H_k = LI$), our analysis shows a Q-linear rate (rather than R-linear) for the strongly convex case, and applies to the convergence of the objective value rather than the iterates. Additionally, our results shows convergence for nonconvex problems.

Variant 2 of Algorithm 2 is proposed in Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018) for convex and strongly convex objectives, with inexactness defined additively as in (2.9). For convex f, Scheinberg

and Tang (2016) showed that if $\sum_{k=0}^{\infty} \epsilon_k/\|H_k\|$ and $\sum_{k=0}^{\infty} \sqrt{\epsilon_k/\|H_k\|}$ converge then a $1/k$ convergence rate is achievable. The same rate can be achieved if $\epsilon_k \leq (a/k)^2$ for any $a \in [0, 1]$. When F is μ-strongly convex, Ghanbari and Scheinberg (2018) showed that if $\sum \epsilon_k/\rho^k$ is finite (where $\rho = 1 - (\gamma\mu)/(\mu + M)$, M is the upper bound for $\|H_k\|$, and γ is as defined in (2.7)), then a global R-linear convergence rate is attained. In both cases, the conditions require a certain rate of decrease for ϵ_k, a condition that can be achieved by performing more and more inner iterations as k increases. By contrast, our multiplicative inexactness criterion (2.3) can be attained with a fixed number of inner iterations. Moreover, we attain a Q-linear rather than an R-linear result.

Algorithm 1 is also considered in Lee et al. (2014), with H_k set either to $\nabla^2 f(x^k)$ or a BFGS approximation. Asymptotic convergence and a local rate are shown for the exact case. For inexact subproblem solutions, local results are proved under the assumption that the unit step size is always taken (which may not happen for inexact steps). A variant of Algorithm 1 with a different step size criterion is discussed in Byrd et al. (2016), for the special case of $\psi(x) = \|x\|_1$. Inexactness of the subproblem solution is measured by the norm of a proximal-gradient step for Q. By utilizing specific properties of the ℓ_1 norm, this paper showed a global convergence rate on the norm of the proximal gradient step on F to zero, without requiring convexity of f — a result similar to our nonconvex result. However, the extension of their result to general ψ is not obvious and, moreover, our inexactness condition avoids the cost of computing the proximal gradient step on Q. When H_k is $\nabla^2 f(x^k)$ or a BFGS approximation, they obtain for the inexact version local convergence results similar to the exact case proved in Lee et al. (2014).

For the case in which f is convex, thrice continuously differentiable, and self-concordant, and ψ is the indicator function of a closed convex set, Tran-Dinh et al. (2014) analyzed global and local convergence rates

of inexact damped proximal Newton with a fixed step size. The paper Li et al. (2017a) extends this convergence analysis to general convex ψ. However, generalization of these results beyond the case of $H_k = \nabla^2 f(x^k)$ and self-concordant f is not obvious.

Accelerated inexact proximal gradient is discussed in Schmidt et al. (2011); Villa et al. (2013) for convex f to obtain an improved $O(1/k^2)$ convergence rate. The work Jiang et al. (2012) considers acceleration with more general choices of H under the requirement $H_k \succeq H_{k+1}$ for all k, which precludes many interesting choices of H_k. This requirement is relaxed by Ghanbari and Scheinberg (2018) to $\theta_k H_k \succeq \theta_{k+1} H_{k+1}$ for scalars θ_k that are used to decide the extrapolation step size. However, as shown in the experiment in Ghanbari and Scheinberg (2018), extrapolation may not accelerate the algorithm. Our analysis does not include acceleration using extrapolation steps, but by combining with the Catalyst framework (Lin et al., 2018), similar improved rates could be attained.

Outline: Remainder of the Chapter

The remainder of this chapter is organized as follows. In Section 2.2, we introduce notation and prove some preliminary results. Convergence analysis appears in Section 2.3 for Algorithms 1 and 2, covering both convex and nonconvex problems. Some interesting and practical choices of H_k are discussed in Section 2.4 to show that our framework includes many existing algorithms. We provide some preliminary numerical results in Section 2.5, and make some final comments in Section 2.6.

2.2 Notations and Preliminaries

The norm $\|\cdot\|$, when applied on vectors, denotes the Euclidean norm. When applied to a symmetric matrix A, it denotes the corresponding induced norm, which is equivalent to the spectral radius of A. For any

symmetric matrix A, $\lambda_{\min}(A)$ denotes its smallest eigenvalue. For any two symmetric matrices A and B, $A \succeq B$ (respectively $A \succ B$) denotes that $A - B$ is positive semidefinite (respectively positive definite). For our nonsmooth function F, ∂F denotes the set of generalized gradient defined as

$$\partial F(x) := \nabla f(x) + \partial \psi(x),$$

where $\partial \psi$ denotes the subdifferential (as ψ is convex). When the minimum F^* of $F(x)$ is attainable, we denote the solution set by $\Omega := \{x \mid F(x) = F^*\}$, and define $P_\Omega(x)$ as the (Euclidean-norm) projection of x onto Ω.

In some results, we use a particular strong convexity assumption to obtain a faster rate. We say that F satisfies the *optimal set strong convexity* condition with modulus $\mu \geq 0$ if for any x and any $\lambda \in [0, 1]$, we have

$$F(\lambda x + (1-\lambda) P_\Omega(x)) \leq \lambda F(x) + (1-\lambda) F^* - \frac{\mu \lambda (1-\lambda)}{2} \|x - P_\Omega(x)\|^2. \tag{2.10}$$

This condition does not require the strong convexity to hold globally, but only between the current point and its projection onto the solution set. Examples of functions that are not strongly convex but satisfy (2.10) include:

- $F(x) = h(Ax)$ where h is strongly convex, and A is any matrix;
- $F(x) = h(Ax) + \mathbf{1}_X(x)$, where X is a polyhedron;
- Squared-hinge loss: $F(x) = \sum \max(0, a_i^T x - b_i)^2$.

A similar condition is the "quasi-strong convexity" condition proposed by Necoara et al. (2018), which always implies (2.10), and can be implied by optimal set strong convexity if F is differentiable. However, since we allow ψ (and therefore F) to be nonsmooth, we need a different definition here.

Turning to the subproblem (2.2) and the definition of Δ_k in (2.6), we find a condition for d to be a descent direction.

Lemma 2.1. *If ψ is convex and f is differentiable, then d is a descent direction for F at x if $\Delta < 0$.*

Proof. We know that d is a descent direction for F at x if the directional derivative

$$F'(x;d) := \lim_{\alpha \to 0} \quad \frac{F(x+\alpha d) - F(x)}{\alpha}$$

is negative. Note that since f is differentiable and ψ is convex,

$$F'(x;d) = \max_{s \in \partial F(x)} s^T d = \nabla f(x)^T d + \max_{\hat{s} \in \partial \psi(x)} \hat{s}^T d$$

is well-defined. Now from the convexity of ψ,

$$\psi(x+d) \geq \psi(x) + \hat{s}^T d, \quad \forall \hat{s} \in \partial \psi(x),$$

so

$$\max_{\hat{s} \in \partial \psi(x)} \hat{s}^T d + \nabla f(x)^T d \leq \psi(x+d) - \psi(x) + \nabla f(x)^T d = \Delta.$$

Therefore, when $\Delta < 0$, the directional derivative is negative and d is a descent direction. □

The following lemma motivates our algorithms.

Lemma 2.2. *If Q and ψ are convex and f is differentiable, then $Q(d) < 0$ implies that d is a descent direction for F at x.*

Proof. Note that $Q(0) = 0$. Therefore, if Q is convex, we have

$$\lambda \nabla f(x)^T d + \frac{\lambda^2}{2} d^T H d + \psi(x + \lambda d) - \psi(x) = Q_H^x(\lambda d) \leq \lambda Q_H^x(d) < 0,$$

for all $\lambda \in (0, 1]$. It follows that $\nabla f(x)^T(\lambda d) + \psi(x + \lambda d) - \psi(x) < 0$ for all sufficiently small λ. Therefore, from Lemma 2.1, λd is a descent direction, and since d and λd only differ in their lengths, so is d. □

Positive semidefiniteness of H suffices to ensure convexity of Q. However, Lemma 2.2 may be used even when H has negative eigenvalues, as ψ may have a strong convexity property that ensures convexity of Q. Lemma 2.2 then suggests that no matter how coarse the approximate solution of (2.2) is, as long as it is better than $d = 0$ for a convex Q, it results in a descent direction. This fact implies finite termination of the backtracking line search procedure in Algorithm 1.

2.3 Convergence Analysis

We start our analysis for both algorithms by showing finite termination of the line search procedures. We then discuss separately three classes of problems involving different assumptions on F, namely, that F is convex, that F satisfies optimal set strong convexity (2.10), and that F is nonconvex. Different iteration complexities are proved in each case. The following condition is assumed throughout our analysis in this section.

Assumption 1. *In (2.1), f is L-Lipschitz-continuously differentiable for some $L > 0$; ψ is convex, extended-valued, proper, and closed; F is lower-bounded; and the solution set Ω of (2.1) is nonempty.*

Line Search Iteration Bound

We show that the line search procedures have finite termination. The following lemma for the backtracking line search in Algorithm 1 does not require H to be positive definite, though it does require strong convexity of Q (2.2).

Lemma 2.3. *If Assumption 1 holds, Q is σ-strongly convex for some $\sigma > 0$, and the approximate solution d to (2.2) satisfies (2.3) for some $\eta < 1$, then for Δ defined in (2.6), we have*

$$\begin{aligned}\Delta &\leq -\frac{1}{2}\left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\sigma \|d\|^2 + d^T H d\right) \\ &\leq -\frac{1}{2}\left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\sigma + \lambda_{\min}(H)\right)\|d\|^2 . \end{aligned} \tag{2.11}$$

Moreover, if

$$(1-\sqrt{\eta})\sigma + (1+\sqrt{\eta})\lambda_{\min}(H) > 0,$$

then the backtracking line search procedure in Algorithm 1 terminates in finite steps and produces a step size α that satisfies the following lower bound:

$$\alpha \geq \min\left\{1, \beta(1-\gamma)\frac{\left(1-\sqrt{\eta}\right)\sigma + \left(1+\sqrt{\eta}\right)\lambda_{\min}(H)}{L\left(1+\sqrt{\eta}\right)}\right\}. \tag{2.12}$$

Proof. From (2.3) and strong convexity of Q, we have that for any $\lambda \in [0,1]$,

$$\begin{aligned}\frac{1}{1-\eta}(Q(0) - Q(d)) &\geq Q(0) - Q^* \\ &\geq Q(0) - Q(\lambda d) \\ &\geq Q(0) - \left(\lambda Q(d) + (1-\lambda)Q(0) - \frac{\sigma\lambda(1-\lambda)}{2}\|d\|^2\right).\end{aligned} \tag{2.13}$$

Since $Q(0) = 0$, we obtain by substituting from the definition of Q that

$$\begin{aligned}&\frac{1}{1-\eta}\left(\nabla f(x)^T d + \frac{1}{2}d^T H d + \psi(x+d) - \psi(x)\right) \\ &\leq \lambda\left(\nabla f(x)^T d + \frac{1}{2}d^T H d + \psi(x+d) - \psi(x)\right) - \frac{\sigma\lambda(1-\lambda)}{2}\|d\|^2 .\end{aligned}$$

Since $1/(1-\eta) \geq 1 \geq \lambda$, we have

$$\begin{aligned}\left(\frac{1}{1-\eta} - \lambda\right)\Delta &\leq -\frac{\sigma\lambda\left(1-\lambda\right)}{2}\left\|d\right\|^2 + \frac{1}{2}\left(\lambda - \frac{1}{1-\eta}\right)d^T H d \\ &\leq -\left(\frac{\sigma\lambda\left(1-\lambda\right)}{2} + \frac{1}{2}\left(\frac{1}{1-\eta} - \lambda\right)\lambda_{\min}\left(H\right)\right)\left\|d\right\|^2.\end{aligned} \tag{2.14}$$

It follows immediately that the following bound holds for any $\lambda \in [0,1]$:

$$\Delta \leq -\frac{1}{2}\left(\frac{\sigma\lambda\left(1-\lambda\right)}{\left(\frac{1}{1-\eta} - \lambda\right)} + \lambda_{\min}\left(H\right)\right)\left\|d\right\|^2.$$

We make the following specific choice of λ:

$$\lambda = \frac{1-\sqrt{\eta}}{1-\eta} \in (0,1]. \tag{2.15}$$

for which

$$1 - \lambda = \sqrt{\eta}\lambda, \quad \frac{1}{1-\eta} - \lambda = \frac{\sqrt{\eta}}{1-\eta}.$$

The result (2.11) follows by substituting these identities into (2.14).

If the right-hand side of (2.11) is negative, then we have from the Lipschitz continuity of ∇f, the convexity of ψ, and the mean value theorem that the following relationships are true for all $\alpha \in [0,1]$:

$$\begin{aligned}&F\left(x+\alpha d\right) - F\left(x\right) \\ &= f\left(x+\alpha d\right) - f\left(x\right) + \psi\left(x+\alpha d\right) - \psi\left(x\right) \\ &\leq \alpha\nabla f\left(x\right)^T d - \alpha\left(\psi\left(x\right) - \psi\left(x+d\right)\right) + \alpha\int_0^1 \left(\nabla f\left(x+t\alpha d\right) - \nabla f\left(x\right)\right)^T d\, dt \\ &\leq \alpha\Delta + \frac{L\alpha^2}{2}\left\|d\right\|^2 \\ &\leq \alpha\Delta - \frac{L\alpha^2(1+\sqrt{\eta})}{\left(1-\sqrt{\eta}\right)\sigma + \left(1+\sqrt{\eta}\right)\lambda_{\min}\left(H\right)}\Delta.\end{aligned}$$

Therefore, (2.5) is satisfied if

$$\alpha\Delta - \frac{L\alpha^2(1+\sqrt{\eta})}{\left(1-\sqrt{\eta}\right)\sigma + \left(1+\sqrt{\eta}\right)\lambda_{\min}(H)}\Delta \leq \alpha\gamma\Delta.$$

We thus get that (2.5) holds whenever

$$\alpha \leq (1-\gamma)\frac{\left(1-\sqrt{\eta}\right)\sigma + \left(1+\sqrt{\eta}\right)\lambda_{\min}(H)}{L\left(1+\sqrt{\eta}\right)}.$$

This leads to (2.12), when we introduce a factor β to account for possible undershoot of the backtracking procedure. □

Note that Lemma 2.3 allows indefinite H, and suggests that we can still obtain a certain amount of objective decrease as long as $\lambda_{\min}(H)$ is not too negative in comparison to the strong convexity parameter of Q. When the strong convexity of Q is accounted for completely by the quadratic part (that is, $\lambda_{\min}(H) = \sigma > 0$) we have the following simplification of Lemma 2.3.

Corollary 2.4. *If Assumption 1 holds, $\lambda_{\min}(H) = \sigma > 0$, and the approximate solution d to (2.2) satisfies (2.3) for some $\eta < 1$, we have*

$$\Delta \leq -\frac{1}{1+\sqrt{\eta}}d^T H d \leq -\frac{\sigma}{1+\sqrt{\eta}}\|d\|^2. \tag{2.16}$$

Moreover, the backtracking line search procedure in Algorithm 1 terminates in finite steps and produces a step size that satisfies the following lower bound:

$$\alpha \geq \bar{\alpha} := \min\left\{1, \frac{2\beta(1-\gamma)\sigma}{L\left(1+\sqrt{\eta}\right)}\right\}. \tag{2.17}$$

Proof. Following (2.13), we have from convexity of ψ for any $\lambda \in [0,1]$ that

$$\begin{aligned}&\frac{1}{1-\eta}\left(\nabla f\left(x\right)^{T} d+\frac{1}{2} d^{T} H d+\psi\left(x+d\right)-\psi\left(x\right)\right)\\ \leq\ &\lambda\left(\nabla f\left(x\right)^{T} d+\frac{\lambda}{2} d^{T} H d+\psi\left(x+d\right)-\psi\left(x\right)\right).\end{aligned}$$

Therefore,

$$\left(\frac{1}{1-\eta}-\lambda\right)\Delta \leq \left(\lambda^{2}-\frac{1}{1-\eta}\right)\frac{1}{2} d^{T} H d. \tag{2.18}$$

Using (2.15) in (2.18), we obtain (2.16). The bound (2.17) follows by substituting $\sigma = \lambda_{\min}(H)$ into (2.12). □

Note that the first inequality in (2.11) and the second inequality in (2.16) make use of the pessimistic lower bound $d^T H d \geq \lambda_{\min}(H)\|d\|^2$, in practice, we observe (see Section 2.5) that the unit step $\alpha_k = 1$ is often accepted in practice (significantly larger than the lower bounds (2.12) and (2.17)) when H_k is the actual Hessian $\nabla^2 f(x^k)$ or its quasi-Newton approximation.

Next we consider Algorithm 2.

Lemma 2.5. *If Assumption 1 holds, Q is σ-strongly convex for some $\sigma > 0$, and d is an approximate solution to (2.2) satisfying (2.3) for some $\eta \in [0,1)$, then (2.7) is satisfied if*

$$(1-\gamma)\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\sigma + \lambda_{\min}(H) \geq L. \tag{2.19}$$

Therefore, in Algorithm 2, if the initial H_k^0 satisfies

$$m_0 I \preceq H_k^0 \preceq M_0 I \tag{2.20}$$

for some $M_0 > 0$, $m_0 \leq M_0$, then for Variant 2, the final H_k satisfies

$$\|H_k\| \leq \tilde{M}_2(\eta) \coloneqq M_0 + \max\left\{1, \frac{1}{\beta}\left(\frac{L\left(1+\sqrt{\eta}\right)}{2-\gamma\left(1-\sqrt{\eta}\right)} - m_0\right)\right\}. \tag{2.21}$$

For Variant 1, if we assume in addition that $m_0 > 0$, we have

$$\|H_k\| \leq \tilde{M}_1(\eta) \coloneqq M_0 \max\left\{1, \frac{L\left(1+\sqrt{\eta}\right)}{\beta\left(2-\gamma\left(1-\sqrt{\eta}\right)\right)m_0}\right\}. \tag{2.22}$$

Proof. From Lipschitz continuity of ∇f, we have that

$$\begin{aligned}
&F(x) - F(x+d) + \gamma Q_H^x(d) \\
&= f(x) - f(x+d) + \gamma \nabla f(x)^T d + \frac{\gamma}{2} d^T H d + (1-\gamma)(\psi(x) - \psi(x+d)) \\
&\geq (\gamma - 1)\nabla f(x)^T d - \frac{L}{2}\|d\|^2 + \frac{\gamma}{2} d^T H d + (1-\gamma)(\psi(x) - \psi(x+d)) \\
&= (\gamma-1)\Delta - \frac{L}{2}\|d\|^2 + \frac{\gamma}{2} d^T H d \qquad (2.23) \\
&\geq \frac{1-\gamma}{2}\left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\sigma\|d\|^2 + d^T H d\right) - \frac{L}{2}\|d\|^2 + \frac{\gamma}{2} d^T H d, \qquad (2.24)
\end{aligned}$$

where in (2.23) we used the definition (2.6), and in (2.24) we used Lemma 2.3. By noting $d^T H d \geq \lambda_{\min}(H)\|d\|^2$, (2.24) shows that (2.19) implies (2.7).

Since ψ is convex, we have that $\sigma \geq \lambda_{\min}(H)$, so that a sufficient condition for (2.19) is that

$$\left((1-\gamma)\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}} + 1\right)\lambda_{\min}(H) \geq L,$$

which is equivalent to

$$\frac{2-\gamma(1-\sqrt{\eta})}{1+\sqrt{\eta}}\lambda_{\min}(H) \geq L.$$

Let the coefficient of $\lambda_{\min}(H)$ in the above inequality be denoted by C_1, this observation suggests that for Variant 1 the smallest eigenvalue of the final H is no larger than $L/(C_1\beta)$, and since the proportion between the largest and the smallest eigenvalues of H_k remains unchanged after scaling the whole matrix, we obtain (2.22).

For Variant 2, to satisfy $C_1 H \succeq LI$, the coefficient for I must be at least $L/C_1 - m_0$. Considering the overshoot, and that the difference between the largest and the smallest eigenvalues is fixed after adding a multiple of identity, we obtain the condition (2.21).

□

By noting the simplification from $d^T H d \geq \lambda_{\min}(H)\|d\|^2$, we rarely observe the worst-case bounds (2.22) or (2.21) in practice, unless H^0 is a multiple of the identity.

Iteration Complexity

Now we turn to the iteration complexity of our algorithms, considering three different assumptions on F: convexity, optimal set strong convexity, and the general (possibly nonconvex) case.

The following lemma is modified from some intermediate results in Ghanbari and Scheinberg (2018), which shows R-linear convergence of Variant 2 of Algorithm 2 for a strongly convex objective when the inexactness is measured by an additive criterion. A proof can be found in Appendix 2.A.

Lemma 2.6. *Let F^* be the optimum of F. If Assumption 1 holds, f is convex and F is μ-optimal-set-strongly convex as defined in (2.10) for some $\mu \geq 0$, then*

for any given x and H, and for all $\lambda \in [0,1]$, we have

$$\begin{aligned} Q^* &\leq \lambda\left(F^* - F\left(x\right)\right) - \frac{\mu\lambda\left(1-\lambda\right)}{2}\left\|x - P_\Omega\left(x\right)\right\|^2 \\ &\qquad + \frac{\lambda^2}{2}\left(x - P_\Omega\left(x\right)\right)^T H\left(x - P_\Omega\left(x\right)\right) \\ &\leq \lambda\left(F^* - F\left(x\right)\right) + \frac{1}{2}\left\|x - P_\Omega\left(x\right)\right\|^2\left(\left\|H\right\|\lambda^2 - \mu\lambda\left(1-\lambda\right)\right), \end{aligned} \tag{2.25}$$

where Q^* is the optimal objective value of (2.2). In particular, by setting $\lambda = \mu/(\mu + \|H\|)$ (as in Ghanbari and Scheinberg (2018)), we have

$$Q^* \leq \frac{\mu}{\mu + \|H\|}(F^* - F(x)). \tag{2.26}$$

Note that we allow $\mu = 0$ in Lemma 2.6.

Sublinear Convergence for General Convex Problems

We start with case of F convex, that is, $\mu = 0$ in the definition (2.10). In this case, the first inequality in (2.25) reduces to

$$Q_k^* \leq \lambda\left(F^* - F\left(x^k\right)\right) + \lambda^2\frac{\left(x^k - P_\Omega\left(x^k\right)\right)^T H_k\left(x^k - P_\Omega\left(x^k\right)\right)}{2}, \tag{2.27}$$

for all $\lambda \in [0,1]$. We assume the following in this subsection.

Assumption 2. *There exists finite $R_0, M > 0$ such that*

$$\sup_{x:F(x)\leq F(x_0)} \|x - P_\Omega(x)\| = R_0 < \infty \;\; \textit{and} \;\; \|H_k\| \leq M, \;\; k = 0,1,2,\ldots. \tag{2.28}$$

Using this assumption, we can bound the second term in (2.27) by

$$\hat{A} := \sup_k \left(x^k - P_\Omega\left(x^k\right)\right)^T H_k\left(x^k - P_\Omega\left(x^k\right)\right) \leq MR_0^2. \tag{2.29}$$

The bound $\hat{A} \leq MR_0^2$ is quite pessimistic, but we use it for purposes of comparing with existing works.

The following lemma is inspired by (Bach, 2015, Lemma 4.4) but contains many nontrivial modifications, and will be needed in proving the convergence rate for general convex problems. Its proof can be found in Appendix 2.B.

Lemma 2.7. *Assume we have three nonnegative sequences* $\{\delta_k\}_{k\geq 0}$, $\{c_k\}_{k\geq 0}$, *and* $\{A_k\}_{k\geq 0}$, *and a constant* $A > 0$ *such that for all* $k = 0, 1, 2, \ldots$, *and for all* $\lambda_k \in [0, 1]$, *we have*

$$0 < A_k \leq A, \quad \delta_{k+1} \leq \delta_k + c_k \left(-\lambda_k \delta_k + \frac{A_k}{2}\lambda_k^2\right). \tag{2.30}$$

Then for $\delta_k \geq A_k$, *we have*

$$\delta_{k+1} \leq \left(1 - \frac{c_k}{2}\right)\delta_k. \tag{2.31}$$

In addition, if we define $k_0 := \arg\min\{k : \delta_k < A\}$, *then*

$$\delta_k \leq \frac{2A}{\sum_{t=k_0}^{k-1} c_t + 2}, \quad \textit{for all } k \geq k_0. \tag{2.32}$$

By Lemma 2.7 together with Assumption 2, we can show that the algorithms converge at a global sublinear rate (with a linear rate in the early stages) for the case of convex F, provided that the final value of H_k for each iteration k of Algorithms 1 and 2 is positive semidefinite.

Theorem 2.8. *Assume that* f *is convex, Assumptions 1 and 2 hold,* $H_k \succeq 0$ *for all* k, *and there is some* $\eta \in [0, 1)$ *such that the approximate solution* d^k *of* (2.2) *satisfies* (2.3) *for all* k. *Then the following claims for Algorithm 1 are true.*

1. *When* $F(x^k) - F^* \geq (x^k - P_\Omega(x^k))^T H_k (x^k - P_\Omega(x^k))$, *we have a linear*

improvement of the objective error at iteration k, that is,

$$F\left(x^{k+1}\right) - F^* \leq \left(1 - \frac{(1-\eta)\,\gamma\alpha_k}{2}\right)\left(F\left(x^k\right) - F^*\right). \tag{2.33}$$

2. *For any $k \geq k_0$, where $k_0 := \arg\min\{k : F(x^k) - F^* < MR_0^2\}$, we have*

$$F\left(x^k\right) - F^* \leq \frac{2MR_0^2}{\gamma(1-\eta)\sum_{t=k_0}^{k-1}\alpha_t + 2}, \tag{2.34}$$

suggesting sublinear convergence of the objective error. If there exists $\bar{\alpha} > 0$ such that $\alpha_k \geq \bar{\alpha}$ for all k, we have

$$k_0 \leq \max\left\{0, 1 + \frac{2}{\gamma\,(1-\eta)\,\bar{\alpha}}\log\frac{F\left(x^0\right) - F^*}{MR_0^2}\right\}. \tag{2.35}$$

For Algorithm 2 under the condition (2.20), the above results still hold, with $\bar{\alpha} = 1$, $\alpha_k \equiv 1$ for all k, and M replaced by $\tilde{M}_1(\eta)$ defined in (2.22) for Variant 1, and $\tilde{M}_2(\eta)$ defined in (2.21) for Variant 2.

Proof. Denoting $\delta_k := F(x^k) - F^*$, we have for Algorithm 1 that the sufficient decrease condition (2.5) together with $H_k \succeq 0$ imply that

$$\delta_{k+1} - \delta_k \leq \alpha_k\gamma\Delta_k = \alpha_k\gamma\left(Q_k\left(d^k\right) - \frac{1}{2}\left(d^k\right)^T H_k d^k\right) \leq \alpha_k\gamma Q_k\left(d^k\right). \tag{2.36}$$

By defining

$$A_k := \left(x^k - P_\Omega\left(x^k\right)\right)^T H_k\left(x^k - P_\Omega\left(x^k\right)\right), \quad A := MR_0^2,$$

(note that $A_k \leq A$ follows from (2.29)) and using (2.3), (2.36), and (2.27), we obtain

$$\delta_{k+1} - \delta_k \leq \alpha_k\gamma\,(1-\eta)\left(-\lambda_k\delta_k + \frac{A_k\lambda_k^2}{2}\right), \quad \forall\lambda_k \in [0,1]. \tag{2.37}$$

We note that (2.37) satisfies (2.30) with

$$c_k = \alpha_k \gamma \left(1 - \eta\right).$$

The results now follow directly from Lemma 2.7.

For Algorithm 2, from (2.7) and (2.3), we get that for any $k \geq 0$,

$$\delta_{k+1} - \delta_k \leq \gamma \left(1 - \eta\right) Q_k^*, \tag{2.38}$$

and the remainder of the proof follows the above procedure starting from the right-hand side of (2.36) with $\alpha_k \equiv 1$.

□

The conditions of Parts 1 and 2 of Theorem 2.8 bear further consideration. When the regularization term ψ is not present in F, and M is a global bound on the norm of the true Hessian $\nabla^2 f(x)$, the condition in Part 2 of Theorem 2.8 is satisfied for $k_0 = 0$, since $f(x^0) - f^* \leq \frac{1}{2} M \|x^0 - P_\Omega(x^0)\|^2 \leq \frac{1}{2} M R_0^2$. Under these circumstances, the linear convergence result of Part 1 may appear not to be interesting. We note, however, that the contribution from ψ may make a significant difference in the general case (in particular, it may result in $F(x^0) - F^* > M R_0^2$) and, moreover, a choice of H_k with $\|H_k\|$ significantly less than M may result in the condition of Part 1 being satisfied intermittently during the computation. In particular, Part 1 lends some support to the empirical observation of rapid convergence on the early stages of the algorithms, as we discuss further below. Note that (Nesterov, 2013, Theorem 4) suggests that when the algorithm is exact proximal gradient, we get $F(x^k) - F^* \leq M R_0^2$ for all $k \geq 1$, but this is not always the case when a different H is picked or when (2.2) is solved only approximately.

By combining Theorem 2.8 with Lemma 2.3 and Corollary 2.4 (which yield lower bounds on α_k), we obtain the following results for Algorithm 1.

Corollary 2.9. *Assume the conditions of Theorem 2.8 are all satisfied. Then we have the following.*

1. *If there exists $\sigma > 0$ such that $\lambda_{\min}(H_k) \geq \sigma$ for all k, then (2.33) becomes*

$$\frac{F\left(x^{k+1}\right) - F^*}{F\left(x^k\right) - F^*} \leq 1 - \frac{\gamma}{2} \min\left\{(1-\eta), \frac{2\left(1-\sqrt{\eta}\right)\beta(1-\gamma)\sigma}{L}\right\}, \tag{2.39}$$

   (2.34) becomes

$$F\left(x^k\right) - F^* \leq \frac{2MR_0^2}{\gamma(k-k_0)\min\left\{1-\eta, \frac{2\left(1-\sqrt{\eta}\right)\beta(1-\gamma)\sigma}{L}\right\} + 2},$$

   and (2.35) becomes

$$k_0 < 1 + \frac{2}{\gamma}\max\left\{0, \log\frac{F\left(x^0\right) - F^*}{MR_0^2}\right\} \cdot \max\left\{\frac{1}{(1-\eta)}, \frac{L}{2(1-\sqrt{\eta})\beta\left(1-\gamma\right)\sigma}\right\}.$$

2. *If Q_k is σ-strongly convex and $H_k \succeq 0$ for all k, then (2.33) becomes*

$$\frac{F\left(x^{k+1}\right) - F^*}{F\left(x^k\right) - F^*} \leq 1 - \frac{\gamma}{2} \min\left\{1-\eta, \frac{\left(1-\sqrt{\eta}\right)^2\beta(1-\gamma)\sigma}{L}\right\},$$

   (2.34) becomes

$$F\left(x^k\right) - F^* \leq \frac{2MR_0^2}{\gamma(k-k_0)\min\left\{1-\eta, \frac{(1-\sqrt{\eta})^2\beta(1-\gamma)\sigma}{L}\right\} + 2},$$

   and (2.35) becomes

$$k_0 < 1 + \frac{2}{\gamma}\max\left\{0, \log\frac{F\left(x^0\right) - F^*}{MR_0^2}\right\} \max\left\{\frac{1}{(1-\eta)}, \frac{L}{(1-\sqrt{\eta})^2\beta\left(1-\gamma\right)\sigma}\right\}.$$

We make some remarks on the results above.

Remark 2.10. *For any $\eta \in [0, 1)$, we have*

$$\frac{1}{2(1-\sqrt{\eta})} < \frac{1}{1-\eta} < \frac{1}{(1-\sqrt{\eta})^2}.$$

Therefore, Algorithm 1 with positive definite H_k has better dependency on η than the case in which we set $\lambda_{\min}(H_k) = 0$ and rely on ψ to make Q_k strongly convex. If ψ is strongly convex, we can move some of its curvature to H_k without changing the subproblems (2.2). This strategy may require us to increase M, but this has only a slight effect on the bounds in Corollary 2.9. These bounds give good reasons to capture the curvature of Q_k in the Hessian H_k alone, so henceforth we focus our discussion on this case.

Remark 2.11. *For Algorithm 2, when we use the bounds (2.22) and (2.21) for M in (2.28), the dependency of the global complexity on η becomes*

$$\max\left\{\frac{1}{1-\eta}, \frac{1}{\left(2-\gamma\left(1-\sqrt{\eta}\right)\right)(1-\sqrt{\eta})}\right\} \leq \max\left\{\frac{1}{1-\eta}, \frac{1}{(2-\gamma)(1-\sqrt{\eta})}\right\},$$

This result is slightly worse than that of using positive definite H in Algorithm 1 if we compare the second part in the max operation.

Remark 2.12. *The bound in (2.29) is not tight for general H, unless $H_k \equiv MI$, as in standard prox-gradient methods. This observation gives further intuition for why second-order methods tend to perform well even though their iteration complexities (which are based on the bound (2.29)) tend to be worse than first-order methods. Moreover, when H_k incorporates curvature information for f, step sizes α_k are often much larger than the worst-case bounds that are used in Corollary 2.9. Theorem 2.8, which shows how the convergence rates are related directly to the α_k, would give tighter bounds in such cases. Line search on H_k in Algorithm 2 does not improve the rate directly, but we note that using H_k with*

smaller norm whenever possible gives more chances of switching to the intermittent linear rate (2.33).

Part 1 of Theorem 2.8 also explains why solving the *subproblem* (2.2) approximately can save the running time significantly, since because of fast early convergence rate, a solution of moderate accuracy can be attained relatively quickly.

Linear Convergence for Optimal Set Strongly Convex Functions

We now consider problems that satisfy the μ-optimal-set-strong-convexity condition (2.10) for some $\mu > 0$, and show that our algorithms have a global linear convergence property.

Theorem 2.13. *If Assumption 1 holds, f is convex, F is μ-optimal-set-strongly convex for some $\mu > 0$, there is some $\eta \in [0, 1)$ such that at every iteration of Algorithm 1, the approximate solution d of (2.2) satisfies (2.3), and*

$$\sigma I \preceq H_k \preceq MI, \quad \text{for some } M \geq \sigma > 0, \quad \forall k. \tag{2.40}$$

Then for $k = 0, 1, 2, \ldots$, we have

$$\frac{F\left(x^{k+1}\right) - F^*}{F\left(x^k\right) - F^*} \leq 1 - \frac{\alpha_k \gamma \left(1 - \eta\right) \mu}{\mu + \|H_k\|} \tag{2.41a}$$

$$\leq 1 - \frac{\gamma\mu}{\mu + M} \min\left\{ \left(1 - \eta\right), \frac{2\left(1 - \sqrt{\eta}\right) \beta \left(1 - \gamma\right) \sigma}{L} \right\}. \tag{2.41b}$$

Moreover, on iterates k for which $F(x^k) - F^* \geq (x^k - P_\Omega(x^k))^T H_k (x^k - P_\Omega(x^k))$, these per-iteration contraction rates can be replaced by the faster rates (2.33) *and* (2.39).

Proof. By rearranging (2.36), we have

$$\begin{aligned} F\left(x^{k+1}\right) - F^* &\leq F\left(x^k\right) - F^* + \alpha_k \gamma Q_k\left(d^k\right) \\ &\leq F\left(x^k\right) - F^* + \alpha_k \gamma \left(1-\eta\right) Q_k^* \qquad (2.42a) \\ &\leq F\left(x^k\right) - F^* - \alpha_k \gamma \left(1-\eta\right) \frac{\mu}{\mu + \|H_k\|} \left(F\left(x^k\right) - F^*\right) \\ &\qquad (2.42b) \\ &= \left(1 - \alpha_k \gamma \left(1-\eta\right) \frac{\mu}{\mu + \|H_k\|}\right) \left(F\left(x^k\right) - F^*\right), \end{aligned}$$

where in (2.42a) we used the inexactness condition (2.3) and in (2.42b) we used (2.26). Using the result in Corollary 2.4 to lower-bound α_k, we obtain (2.41b).

To show that the part for the early fast rate in (2.33) and (2.39) can be applied, we show that Assumption 2 holds. Then because f is assumed to be convex as well here, Theorem 2.8 and Corollary 2.9 apply as well. Consider (2.10), by rearranging the terms, we get

$$\begin{aligned} \lambda\left(F(x) - F^*\right) &\geq \frac{\mu\lambda(1-\lambda)}{2} \left\|x - P_\Omega\left(x\right)\right\|^2 + F\left(\lambda x + \left(1-\lambda\right) P_\Omega\left(x\right)\right) - F^* \\ &\geq \frac{\mu\lambda(1-\lambda)}{2} \left\|x - P_\Omega\left(x\right)\right\|^2, \quad \forall \lambda \in [0,1], \end{aligned} \tag{2.43}$$

as $F\left(\lambda x + \left(1-\lambda\right) P_\Omega\left(x\right)\right) \geq F^*$ from optimality. By dividing both sides of (2.43) by λ and letting $\lambda \to 0$, we get the bound

$$F(x^0) - F^* \geq F(x) - F^* \geq \frac{\sigma}{2} \|x - P_\Omega(x)\|, \forall x : F(x) \leq F(x^0), \tag{2.44}$$

validating Assumption 2.

□

Note that the parameter μ in the theorem above is decided by the problem and cannot be changed, while σ can be altered according to the

algorithm choice. We have a similar result for Algorithm 2.

Theorem 2.14. *If Assumption 1 holds, f is convex, F is μ-optimal-set-strongly convex for some $\mu > 0$, there exists some $\eta \in [0, 1)$ such that at every iteration of Algorithm 2, the approximate solution d of (2.2) satisfies (2.3), and the conditions for H_k^0 in Lemma 2.5 are satisfied for all k. Then we have*

$$\frac{F\left(x^{k+1}\right) - F^*}{F\left(x^k\right) - F^*} \leq 1 - \gamma \frac{\mu\left(1-\eta\right)}{\mu + \|H_k\|}, \quad k = 0, 1, 2, \ldots, \tag{2.45}$$

and the right-hand side of (2.45) can be further bounded by

$$1 - \gamma \frac{\mu\left(1-\eta\right)}{\mu + \tilde{M}_1(\eta)} \quad \text{and} \quad 1 - \gamma \frac{\mu\left(1-\eta\right)}{\mu + \tilde{M}_2(\eta)} \tag{2.46}$$

for Variant 1 and Variant 2, respectively, where $\tilde{M}_1(\eta)$ and $\tilde{M}_2(\eta)$ are defined in Lemma 2.5. Moreover, when $F(x^k) - F^* \geq (x^k - P_\Omega(x^k))^T H_k (x^k - P_\Omega(x^k))$, the faster rate (2.33) (with $\alpha_k \equiv 1$ and the modification for Algorithm 2 mentioned in Theorem 2.8) can be used to replace (2.45).

Proof. From (2.26) and (2.38), we have

$$\begin{aligned} F\left(x^{k+1}\right) - F^* &\leq F\left(x^k\right) - F^* + \gamma Q_k\left(d^k\right) \\ &\leq F\left(x^k\right) - F^* + \gamma\left(1-\eta\right) Q_k^* \\ &\leq \left(1 - \gamma \frac{\mu}{\mu + \|H_k\|}\left(1-\eta\right)\right)\left(F\left(x^k\right) - F^*\right), \end{aligned}$$

proving (2.45). From Lemma 2.5, we ensure that $\|H_k\|$ is upper-bounded by $\tilde{M}_1(\eta)$ and $\tilde{M}_2(\eta)$ for the two variants respectively, leading to (2.46). The statement concerning (2.33) follows from the same reasoning as in the proof for Theorem 2.13.

□

By reasoning with the extreme eigenvalues of H_k, we can see that

the convergence rates still depend on the conditioning of f. For Algorithm 1, if we select $M \leq L$, then backtracking may be necessary, and the bound (2.41b) (in which a factor μ/L appears) is germane. This same factor appears in both (2.41a) and (2.41b) when $M > L$. Often, however, the backtracking line search chooses a value of α_k that is not much less than 1, which is why we believe that the bounds (2.33), (2.34), and (2.41a) (which depend explicitly on α_k) have some value in revealing the actual performance of the algorithm. Similar comments apply to Algorithm 2, because (2.7) may be satisfied with $\|H_k\|$ much smaller than the bounds for properly chosen H_k^0.

In the interesting case in which we choose $H_k \equiv LI$ and $\eta = 0$, we have $m_0 = \|H_k\| = L$ in Algorithm 2, and modification of H_k is not needed, since (2.7) always holds for $\gamma = 1$. The bound (2.34) becomes $(F(x^k) - F^*) \leq 2LR_0^2/(k+2)$, which matches the known convergence rates of proximal gradient (Nesterov, 2013) and gradient descent (Nesterov, 2004). The global linear rate in Theorem 2.14 also matches that of existing proximal gradient analysis for strongly convex problems, but the intermittent linear rate (2.33) that applies to both cases is new. For the case of accelerated proximal gradient covered in Nesterov (2013), although not covered directly by our framework studied in this work, one can combine our algorithm and analysis with the Catalyst framework (Lin et al., 2018) to obtain similar accelerated rates for both the strongly convex and the general convex cases.

Convergence Rates for a Sharpness Condition

We then consider the case that in addition to convexity, the function satisfies a sharpness condition

$$\frac{S}{r} \|x - P_\Omega(x)\|^r \leq F(x) - F^*, \forall x \tag{2.47}$$

for some $S > 0$ and some $r \geq 1$. Two noteworthy special cases are the weak sharp minima (Burke and Ferris, 1993) where $r = 1$ and the quadratic growth condition where $r = 2$. The case of $r = 2$ is studied in Peng et al. (2018) and they show linear global convergence is achieved in this case. Here we extend their result to develop more general results. The key idea is to notice that in (2.25), we can remove the term related to μ and use the inequality in (2.47) to develop convergeence rates related to the objective value. For this class of problems, we obtain the following results.

Theorem 2.15. *Assume that Assumption 1 holds, f is convex, there is some $\eta \in [0, 1)$ such that at every iteration of Algorithm 1, the approximate solution d of (2.2) satisfies (2.3), there is some $\eta \in [0, 1)$ such that at every iteration of Algorithm 1, the approximate solution d of (2.2) satisfies (2.3), and (2.40) holds for all k. If F satisfies (2.47) for some $S > 0$ and some $r \geq 1$, then we have the following for Algorithm 1.*

1. *When $r \in [1, 2)$, the objective value decrease per iteration is at least*

$$F\left(x^k\right) - F\left(x^{k+1}\right) \geq \frac{(1-\eta)\,\alpha_k \gamma}{2}\left(M\left(\frac{r}{S}\right)^{\frac{2}{r}}\right)^{\frac{3r}{r-2}} \tag{2.48}$$

   as long as

$$F\left(x^k\right) - F^* \geq \left(M\left(\frac{r}{S}\right)^{\frac{2}{r}}\right)^{\frac{r}{r-2}}. \tag{2.49}$$

   Let k_0 be the first index such that (2.49) fails, then we further have that the objective value converges Q-linearly to the optimum for all $k \geq k_0$.

2. *When $r = 2$, we have global Q-linear convergence.*

3. *When $r > 2$, we have Q-linear convergence as long as (2.49) holds.*

Proof. We will use (2.42a). To begin, we consider Lemma 2.6 and notice

that since (2.10) does not hold, we can use (2.27) and (2.47). This leads to

$$Q_k^* \leq \lambda \left(F^* - F\left(x^k\right)\right) + \frac{M\lambda^2}{2}\left(\frac{r}{S}\left(F\left(x^k\right) - F^*\right)\right)^{\frac{2}{r}}, \forall \lambda \in [0,1]. \quad (2.50)$$

As the right-hand side of (2.50) is convex with respect to λ, the minimum happens at either where the derivative is zero or the boundary. Namely,

$$\lambda = \min\left\{1, \frac{\left(F\left(x^k\right) - F^*\right)^{1-\frac{2}{r}}}{M\left(\frac{r}{S}\right)^{\frac{2}{r}}}\right\}.$$

Thus we need to consider when

$$\left(F\left(x^k\right) - F^*\right)^{1-\frac{2}{r}} \leq M\left(\frac{r}{S}\right)^{\frac{2}{r}} \quad (2.51)$$

and when

$$\left(F\left(x^k\right) - F^*\right)^{1-\frac{2}{r}} > M\left(\frac{r}{S}\right)^{\frac{2}{r}} \quad (2.52)$$

Therefore, we discuss the three cases $r < 2$, $r = 2$, and $r > 2$ separately.

Case I: $r \in [1, 2)$**.**

As $1 - \frac{2}{r} < 0$, (2.51) holds if and only if (2.49) holds. In this situation, by using

$$\lambda = \frac{\left(F\left(x^k\right) - F^*\right)^{1-\frac{2}{r}}}{M\left(\frac{r}{S}\right)^{\frac{2}{r}}}, \quad (2.53)$$

(2.50) becomes

$$Q_k^* \leq -\frac{\left(F\left(x^k\right) - F^*\right)^{2-\frac{2}{r}}}{2M\left(\frac{r}{S}\right)^{\frac{2}{r}}}. \quad (2.54)$$

Now as $2 - 2/r \geq 0$, by using (2.49) again, we get that

$$Q_k^* \leq -\frac{\left(M\left(\frac{r}{S}\right)^{\frac{2}{r}}\right)^{\frac{r}{r-2}(2-\frac{2}{r})}}{2M\left(\frac{r}{S}\right)^{\frac{2}{r}}} \leq -\frac{\left(M\left(\frac{r}{S}\right)^{\frac{2}{r}}\right)^{\frac{r}{r-2}(2-\frac{2}{r})-1}}{2},$$

which together with (2.42a) leads to (2.48).

Next, when (2.52) holds, we let $\lambda = 1$ in (2.50) and get

$$Q_k^* \leq \left(F\left(x^k\right) - F^*\right)\left(-1 + \frac{M}{2}\left(\frac{r}{S}\right)^{\frac{2}{r}}\left(F\left(x^k\right) - F^*\right)^{\frac{2}{r}-1}\right) \leq \frac{1}{2}\left(F\left(x^k\right) - F^*\right), \tag{2.55}$$

where the second inequality is from (2.52). This result together with (2.42a) then implies the desired local Q-linear convergence.

Case II: $r = 2$**.**

In this case, $1 - 2/r = 0$, so

$$Q_k^* \leq \left(F\left(x^k\right) - F^*\right)\left(-\lambda + \frac{\lambda^2 M}{2}\left(\frac{r}{S}\right)^{\frac{2}{r}}\right), \forall \lambda \in [0, 1].$$

Notice that the coefficient for $F(x^k) - F^*$ is independent of the function value, so global linear convergence follows directly from (2.42a).

Case III: $r > 2$**.**

In this case, we have $1 - 2/r \geq 0$ so (2.52) holds if and only if (2.49) holds. When (2.49) holds, we can use $\lambda = 1$ to get (2.55), and the rest then follows directly from the same argument. □

We note that when $r > 2$, when (2.49) stops to hold, we can use the result for the convex case to obtain global convergence results. The result for Algorithm 2 is very similar and thus omitted.

Sublinear Convergence of the First-order Optimality Condition for Nonconvex Problems

We consider now the case of nonconvex F. In this situation, Lemma 2.6 cannot be used, so we consider other properties of Q. We can no longer guarantee the convergence of the objective value to the global minimum. Instead, we consider the norm of the exact solution of the subproblem as the indicator of closeness to the first-order optimality condition $0 \in \partial F(x)$ for (2.1) (see, for example, (Fletcher, 1987, (14.2.16))). In particular, it is known that $0 \in \partial F(x)$ if and only if

$$0 = \arg\min_d \; Q_I^x(d) = \arg\min_d \; \nabla f\,(x)^T\, d + \frac{1}{2} d^T d + \psi\,(x+d) - \psi\,(x)\,. \quad (2.56)$$

This is a consequence of the following lemma.

Lemma 2.16. *Given any $H \succ 0$, and Q_H^x defined as in (2.2), the following are true.*

1. *A point x satisfies the first-order optimality condition $0 \in \partial F(x)$ if and only if*
$$0 = \arg\min_d \; Q_H^x(d).$$
2. *For any x, defining d^* to be the minimizer of $Q_H^x(\cdot)$, we have*
$$Q_H^x(d^*) \leq -\frac{1}{2}\lambda_{\min}\,(H)\|d^*\|^2. \quad (2.57)$$

Proof. Part 1 is well known. For Part 2, we have from the optimality conditions for d^* that $-\nabla f(x) - Hd^* \in \partial\psi(x+d^*)$. By convexity of ψ, we thus have

$$\psi(x) \geq \psi(x+d^*) + (d^*)^T(\nabla f(x) + Hd^*) \quad \Rightarrow \quad 0 \geq Q_H^x(d^*) + \frac{1}{2}(d^*)^T H d^*,$$

from which the result follows.

□

As in (2.56), we consider the following measure of closeness to a stationary point:

$$G_k \coloneqq \arg\min_d \; Q_I^{x^k}(d). \tag{2.58}$$

We show that the minimum value of the norm of this measure over the first k iterations converges to zero at a sublinear rate of $O(1/\sqrt{k})$. The first step is to show that the minimum of $|Q_k|$ converges at a $O(1/k)$ rate.

Lemma 2.17. *Assume that there is an $\eta \in [0, 1)$ such that (2.3) is satisfied at all iterations. For Algorithm 1, if Assumption 1 holds and $H_k \succeq \sigma I$ for some $\sigma > 0$ and all k, we have*

$$\min_{0\le t\le k} \left|Q_t\left(d^t\right)\right| \le \frac{F\left(x^0\right) - F^*}{\gamma\left(k+1\right)\min_{0\le t\le k}\alpha_t} \le \frac{F\left(x^0\right) - F^*}{\gamma\left(k+1\right)} \max\left\{1, \frac{(1+\sqrt{\eta})L}{2\beta\left(1-\gamma\right)\sigma}\right\}. \tag{2.59}$$

For Algorithm 2 (requires $H_k^0 \succ 0$ for the first variant), we have

$$\min_{0\le t\le k} \left|Q_t\left(d^t\right)\right| \le \frac{F\left(x^0\right) - F^*}{\gamma\left(k+1\right)}.$$

Proof. From (2.36), we have that for any $k \ge 0$,

$$F^* - F\left(x^0\right) \le F\left(x^{k+1}\right) - F\left(x^0\right) \le \gamma \sum_{t=0}^{k} \alpha_t Q_t\left(d^t\right) \le \gamma \min_{0\le t\le k} \alpha_t \sum_{t=0}^{k} Q_t\left(d^t\right). \tag{2.60}$$

From Corollary 2.4, we have that α_t for all t is lower bounded by a positive value. Therefore, using $|Q_t\left(d^t\right)| = -Q_t\left(d^t\right)$ for all t, we obtain

$$\min_{0\le t\le k} \left|Q_t\left(d^t\right)\right| \le -\frac{1}{k+1}\sum_{t=0}^{k} Q_t\left(d^t\right) \le \frac{F\left(x^0\right) - F^*}{\gamma\left(k+1\right)\min_{0\le t\le k}\alpha_t}.$$

Substituting the lower bound for α from Corollary 2.4 gives the desired

result (2.59). The result for Algorithms 2 follows from the same reasoning applied to (2.7).

□

The following lemma is from Tseng and Yun (2009). (Its proof is omitted.)

Lemma 2.18 ((Tseng and Yun, 2009, Lemma 3)). *Given H_k satisfying (2.40) for all k, we have*

$$\|G_k\| \leq \frac{1 + \frac{1}{\sigma} + \sqrt{1 - 2\frac{1}{M} + \frac{1}{\sigma^2}}}{2} M \left\|d^{k*}\right\|,$$

where

$$d^{k*} := \arg\min Q_k.$$

We are now ready to show the convergence of $\|G_k\|$.

Corollary 2.19. *Assume that (2.3) holds at all iterations for some $\eta \in [0, 1)$ and that Assumption 1 holds. Let $\tilde{M}_1(\eta)$ and $\tilde{M}_2(\eta)$ be as defined in Lemma 2.5. For Algorithm 1, suppose that H_k satisfies (2.40) for all $k \geq 0$. We then have for all $k = 0, 1, , 2, \ldots$ that*

$$\begin{aligned}
\min_{0 \leq t \leq k} \|G_t\|^2 \leq & \frac{F(x^0) - F^*}{\gamma(k+1)} \frac{M^2 \left(1 + \frac{1}{\sigma} + \sqrt{1 - \frac{2}{M} + \frac{1}{\sigma^2}}\right)^2}{2(1-\eta)\sigma \min_{0 \leq t \leq k} \alpha_t} \\
\leq & \frac{F(x^0) - F^*}{\gamma(k+1)} \frac{M^2 \left(1 + \frac{1}{\sigma} + \sqrt{1 - \frac{2}{M} + \frac{1}{\sigma^2}}\right)^2}{2\sigma} \\
& \max\left\{\frac{1}{1-\eta}, \frac{L}{2\left(1 - \sqrt{\eta}\right)(1-\gamma)\sigma\beta}\right\}.
\end{aligned}$$

For Algorithm 2, if the initial H_k^0 satisfies $M_0 I \succeq H_k^0 \succeq m_0 I$ with $M_0 \geq m_0 > 0$

then for Variant 1 we have:

$$\min_{0\leq t\leq k}\|G_t\|^2 \leq \frac{F\left(x^0\right))-F^*}{\gamma\left((k+1)\right)}\frac{\tilde{M}_1(\eta)^2\left(1+\frac{1}{m_0}+\sqrt{1-\frac{2}{\tilde{M}_1(\eta)}+\frac{1}{m_0^2}}\right)^2}{2\left(1-\eta\right)m_0}.$$

For Variant 2, we have under the same assumptions on H_k^0 that the same bound is satisfied,[3] *with $\tilde{M}_1(\eta)$ replaced by $\tilde{M}_2(\eta)$.*

Proof. Let $\bar{k} := \arg\min_{0\leq t\leq k}|Q_t(d^t)|$, the condition (2.3) and Lemmas 2.16 and 2.18 imply

$$\begin{aligned}
-Q_{\bar{k}}\left(d^{\bar{k}}\right) &\geq -\left(1-\eta\right)Q_{\bar{k}}^* \\
&\geq \frac{\sigma\left(1-\eta\right)}{2}\left\|d^{\bar{k}*}\right\|^2 \\
&\geq \frac{2\sigma\left(1-\eta\right)}{M^2\left(1+\frac{1}{\sigma}+\sqrt{1-\frac{2}{M}+\frac{1}{\sigma^2}}\right)^2}\left\|G_{\bar{k}}\right\|^2. \qquad (2.61)
\end{aligned}$$

Finally, we note that $\|G_{\bar{k}}\| \geq \min_{0\leq t\leq k}\|G_t\|$. The proof is finished by combining (2.61) with Lemma 2.17.

□

If we replace the definition of G_k in (2.58) by the solution of (2.2), the inequality in Lemma 2.18 is not needed. In particular, when we use the proximal gradient algorithm with $H_k = LI$ and $\eta = 0$ (so that (2.7) holds with $\gamma = 1$, and $M = L$) we obtain a bound of $2(F(x^0) - F^*)/(L(k+1))$ on $\|d^k\|^2$, matching the result shown in Nesterov (2013); Drusvyatskiy and Lewis (2018).

Comparison Among Different Approaches

Algorithms 1 and 2 both require evaluation of the function F for each choice of the parameter α_k, to check whether the decrease conditions (2.5)

[3] We could instead require only $H_k^0 \succeq 0$ and start with $H_k + I$ instead.

and (2.7) (respectively) are satisfied. The difference is that Algorithm 2 may also require solution of the subproblem (2.2) for each α_k. This additional computation comes with two potential benefits. First, the second variant of Algorithm 2 allows the initial choice of approximate Hessian H_k^0 to be indefinite, although the final value H_k at each iteration needs to be positive semidefinite for our analysis to hold. (There is a close analogy here to trust-region methods for nonconvex smooth optimization, where an indefinite Hessian is adjusted to be positive semidefinite in the process of solving the trust-region subproblem.) Second, because full steps are always taken in Algorithm 2, any structure induced in the iterates x^k by the regularizer ψ (such as sparsity) will be preserved. This fact in turn may lead to faster convergence, as the algorithm will effectively be working in a low-dimensional subspace.

2.4 Choosing H_k

Here we discuss some ways to choose H_k so that the algorithms are well defined and practical, and our convergence theory can be applied.

When H_k are chosen to be positive multiples of identity ($H_k = \zeta_k I$, say), our algorithms reduce to variants of proximal gradient. If we set $\zeta_k \geq L$, then the unit step size is always accepted even if the problem is not solved exactly, because $Q_k(d^k)$ is an upper bound of $F(x^k) - F(x^k + d^k)$. When L is not known in advance, adaptive strategies can be used to find it. For Algorithm 2, we could define ζ_k^0 (such that $H_k^0 = \zeta_k^0 I$) to be the final value ζ_{k-1} from the previous iteration, possibly choosing a smaller value at some iterations to avoid being too conservative. For Algorithm 1, we could increase ζ_k^0 over ζ_{k-1} if backtracking was necessary at iteration $k-1$, and shrink it when a unit stepsize sufficed for several successive iterations.

The proximal Newton approach of setting $H_k = \nabla^2 f(x^k)$ is a common choice in the convex case (Lee et al., 2014), where we can guarantee that H_k

is at least positive semidefinite. In Lee et al. (2014), it is shown that in some neighborhood of the optimum, when d^k is the exact solution of (2.2), then unit step size is always taken, and superlinear or quadratic convergence to the optimum ensues. (A global complexity condition is not required for this result.) Generally, however, indefiniteness in $\nabla^2 f(x^k)$ may lead to the search direction d^k not being a descent direction, and the backtracking line search will not terminate in this situation. (Our convergence results for Algorithm 1 do not apply in the case of H_k indefinite.) A common fix is to use damping, setting $H_k = \nabla^2 f(x^k) + \zeta_k I$, for some $\zeta_k \geq 0$ that at least ensures positive definiteness of H_k. Strategies for choosing ζ_k adaptively have been the subject of much research in the context of smooth minimization, for example, in trust-region methods and the Levenberg-Marquardt method for nonlinear least squares (see Nocedal and Wright (2006)). Variant 2 of our Algorithm 2 uses this strategy. It is desirable to ensure that $\zeta_k \to 0$ as the iterates approach a solution at which local convexity holds, to ensure rapid local convergence.

An L-BFGS approximation of $\nabla^2 f(x^k)$ could also be used for H_k. When ψ is not present in (2.1) and f is strongly convex, it is shown in Liu and Nocedal (1989) that this approach has global linear convergence because the eigenvalues of H_k are restricted to a bounded positive interval. This proof can be extended to our algorithms, when a convex ψ is present in (2.1). When f is not strongly convex, one can apply safeguards to the L-BFGS update procedure (as described in Li and Fukushima (2001)) to ensure that the upper and lower eigenvalues of H_k are bounded uniformly away from zero.

Another interesting choice of H_k is a block-diagonal approximation of the Hessian, which (when ψ can be partitioned accordingly) allows the subproblem (2.2) to be solved in parallel while still retaining some curvature information. Strategies like this one are often used in distributed optimization for machine learning problems (see, for example, Yang (2013);

Lee and Chang (2017); Zheng et al. (2017)).

2.5 Numerical Results

We sketch some numerical simulations that support our theoretical results. We conduct experiments on two different problems: ℓ_1-regularized logistic regression, and the Lagrange dual problem of ℓ_2-regularized squared-hinge loss minimization. The algorithms are implemented in C/C++.

ℓ_1-regularized Logistic Regression

Given training data points $(a_i, b_i) \in \mathbb{R}^n \times \{-1, 1\}$, $i = 1, \ldots, l$, and a specified parameter $C > 0$, we solve the following convex problem

$$\min_{x \in \mathbb{R}^n} C \sum_{i=1}^{l} \psi\left(1 + \exp\left(-b_i a_i^T x\right)\right) + \|x\|_1. \tag{2.62}$$

We define H_k to be the limited-memory BFGS approximation (Liu and Nocedal, 1989) based on the past 10 steps, with a safeguard mechanism proposed in Li and Fukushima (2001) to ensure uniform boundedness of H_k. The subproblems (2.2) are solved with SpaRSA (Wright et al., 2009), a proximal-gradient method which, for bounded H_k, converges globally at a linear rate. We consider the publicly available data sets listed in Table 2.1,[4] and present empirical convergence results by showing the relative objective error, defined as

$$\frac{F(x) - F^*}{F^*}, \tag{2.63}$$

where F^* is the optimum, obtained approximately through running our algorithm with long enough time. For all variants of our framework, we used

[4]Downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

parameters $\beta = 0.5$, and $\gamma = 10^{-4}$. Further details of our implementation are described in Chapter 5.

We use the two smaller data sets a9a and rcv1 to quantify the relationship between accuracy of the subproblem solution and the number of outer iterations. We compare running SpaRSA with a fixed number of iterations $T \in \{5, 10, 15, 20, 25, 30\}$. Figure 2.1 shows that, in all cases, the number of outer iterations decreases monotonically as the (fixed) number of inner iterations is increased. For $T \geq 15$, the degradation in number of outer iterations resulting from less accurate solution of the subproblems is modest, as our theory suggests. We also observe the initial fast linear rates in the early stages of the method that are predicted by our theory, settling down to a slower linear rate on later iterations, but with sudden drops of the objective, possibly as a consequence of intermittent satisfaction of the condition in Part 1 of Theorem 2.8.

Next, we examine empirically the step size distribution for Algorithm 1 and how often in Algorithm 2 the matrix H_k needs to be modified. On both a9a and rcv1, the initial step estimate $\alpha = 1$ is accepted on over 99.5% of iterations in Algorithm 1, while in both variants of Algorithm 2, the initial choice of H_k is used without modification on over 99% of iterations. These statistics hold regardless of the value of T (the number of inner iterations), though in the case of Algorithm 2, we see a faint trend toward more adjustments for larger values of T. When adjustments are needed, they never number more than 4 at any one iteration, except for a single case (a9a for Variant 1 of Algorithm 2 with $T = 5$) for which up to 8 adjustments are needed.

We next compare our inexact method with an exact version, in which the subproblems (2.2) are solved to near-optimality at each iteration. Since the three algorithms behave similarly, we use Algorithm 1 as the representative for this investigation. We use a local cluster with 16 nodes for the two larger data sets rcv1 and epsilon, while for the small data set a9a,

Data set	l	n	#nonzeros
a9a	$32,561$	123	$451,592$
rcv1_test.binary	$677,399$	$47,236$	$49,556,258$
epsilon	$400,000$	$2,000$	$800,000,000$

Table 2.1: Properties of the Data Sets

only one node is used. Iteration counts and running time comparisons are shown in Figure 2.2. The exact version requires fewer iterations, as expected, but the inexact version requires only modestly more iterations. In terms of runtime, the inexact versions with moderate amount of inner iterations (at least 30) has the advantage, due to the savings obtained by solving the subproblem inexactly.

We note that the approach of gradually increasing the number of inner iterations, suggested in Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018), produces good results for this application, the number of iterations required being comparable to those for the exact solver while the running time is slightly faster than that of $T = 30$ for epsilon and competitive with it for the rest two data sets.

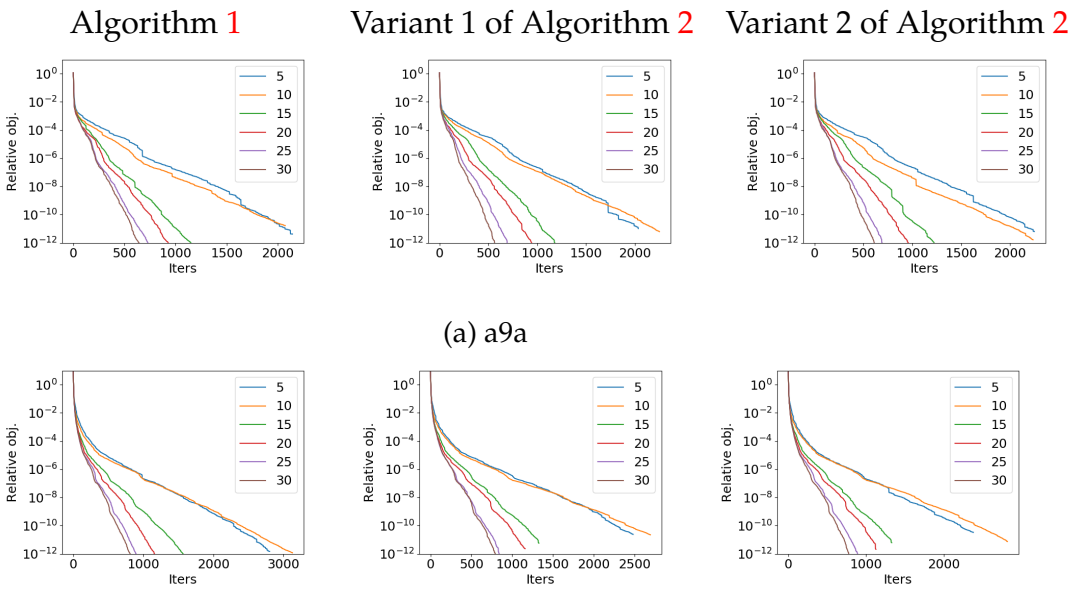
Dual of ℓ_2-regularized Squared-Hinge Loss Minimization

Given the same binary-labelled data points as in the previous experiment and a parameter $C > 0$, the ℓ_2-regularized squared-hinge loss minimization problem is

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x\|_2^2 + C\sum_{i=1}^{l} \max(1 - b_i a_i^T x, 0)^2.$$

With the notation $A \coloneqq (b_1 a_1, b_2 a_2, \ldots, b_l a_l)$, the dual of this problem is

$$\min_{\alpha\geq 0} \frac{1}{2}\alpha^T A^T A\alpha - \mathbf{1}^T\alpha + \frac{1}{4C}\|\alpha\|_2^2, \tag{2.64}$$



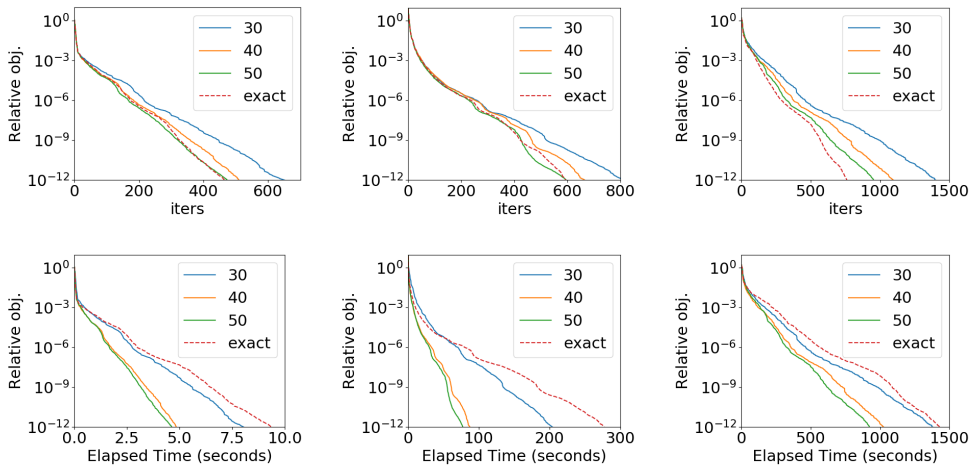
(a) a9a

(b) rcv1t

Figure 2.1: Comparison of different subproblem solution exactness in solving (2.62). The y-axis is the relative objective error (2.63), and the x-axis is the iteration count.

which is $(1/2C)$-strongly convex. We consider the distributed setting such that the columns of A are stored across multiple processors. In this setup, only the block-diagonal parts (up to a permutation) of $A^T A$ can be easily formed locally on each processor. We take H_k to be the matrix formed by these diagonal blocks, so that the subproblem (2.2) can be decomposed into independent parts. We use cyclic coordinate descent with random permutation (RPCD) as the solver for each subproblem. (Note that this algorithm partitions trivially across processors, because of the block-diagonal structure of H_k.)

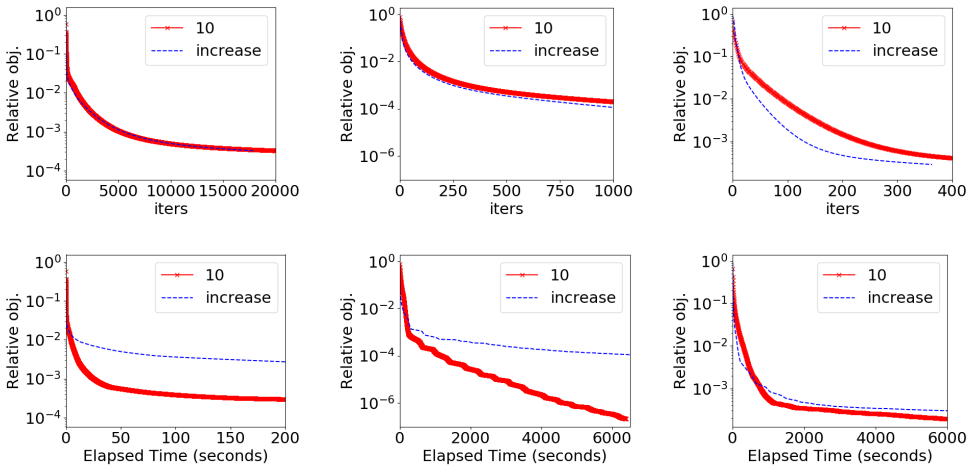
Our experiment compares the strategy of performing a fixed number of RPCD iterations for each subproblem with one of increasing the number of inner iterations as the algorithm proceeds, as in Scheinberg and Tang



(a) a9a (b) rcv1 (c) epsilon

Figure 2.2: Comparison between the exact version and the inexact version of Algorithm 1 for solving (2.62). Top: outer iterations; bottom: running time. The y-axis is the relative objective error (2.63).

(2016); Ghanbari and Scheinberg (2018). We use the data sets in Table 2.1, and compare the two strategies on Algorithm 1, but use an exact line search to choose α_k rather than the backtracking approach. (An exact line search is made easy by the quadratic objective.) For the first strategy, we use ten iterations of RPCD on each subproblem, while for the second strategy, we perform $1+\lfloor k/10 \rfloor$ iterations of RPCD at the kth outer iteration as suggested by Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018). The implementation is a modification of the experimental code of Lee and Roth (2015). We run the algorithms on a local cluster with 16 machines, so that H_k contains 16 diagonal blocks. Results are shown in Figure 2.3. Since the choice of H_k in this case does not capture global curvature information adequately, the strategy of increasing the accuracy of subproblem solution on later iterations does not reduce the number of iterations as significantly as in the previous experiment. The runtime results show a significant advantage for the first strategy of a fixed number



(a) a9a (b) rcv1 (c) epsilon

Figure 2.3: Comparison of two strategies for inner iteration count in Algorithm 1 applied to (2.2): Increasing accuracy on later iterations (blue) and a fixed number of inner iterations (red). Top: outer iterations; bottom: running time. Vertical axis shows relative objective error (2.63).

of inner iterations, particularly on the a9a and rcv1 data sets. Judging from the trend in the approach of increasing inner iterations, we can expect that the exact version will show huger disadvantage for running time in this case. We also observe the faster linear rate on early iterations, matching our theory.

2.6 Conclusions

We have analyzed global convergence rates of three practical inexact successive quadratic approximation algorithms under different assumptions on the objective function, including the nonconvex case. Our analysis shows that inexact solution of the subproblems affects the rates of convergence in fairly benign ways, with a modest factor appearing in the bounds. When linearly convergent methods are used to solve the subproblems, the

inexactness condition holds when a fixed number of inner iterations is applied at each outer iteration k.

Appendix

2.A Proof of Lemma 2.6

Proof. We have

$$
\begin{aligned}
Q^* &= \min_d \ \nabla f(x)^T d + \frac{1}{2} d^T H d + \psi(x+d) - \psi(x) \\
&\leq \min_d \ f(x+d) + \psi(x+d) + \frac{1}{2} d^T H d - F(x) && \text{(2.65a)} \\
&\leq F(x + \lambda(P_\Omega(x) - x)) + \frac{\lambda^2}{2} (P_\Omega(x) - x)^T H (P_\Omega(x) - x) - F(x) \quad \forall \lambda \in [0,1] && \text{(2.65b)} \\
&\leq (1-\lambda) F(x) + \lambda F^* - \frac{\mu \lambda (1-\lambda)}{2} \|x - P_\Omega(x)\|^2 && \text{(2.65c)} \\
&\quad + \frac{\lambda^2}{2} (x - P_\Omega(x))^T H (x - P_\Omega(x)) - F(x) \quad \forall \lambda \in [0,1] \\
&\leq \lambda(F^* - F(x)) - \frac{\mu \lambda (1-\lambda)}{2} \|x - P_\Omega(x)\|^2 + \frac{\lambda^2}{2} \|H\| \|x - P_\Omega(x)\|^2 \quad \forall \lambda \in [0,1],
\end{aligned}
$$

where in (2.65a) we used the convexity of f, in (2.65b) we set $d = \lambda(P_\Omega(x) - x)$, and in (2.65c) we used the optimal set strong convexity (2.10) of F. Thus we obtain (2.25).

□

2.B Proof of Lemma 2.7

Proof. Consider

$$
\lambda_k = \arg \min_{\lambda \in [0,1]} \ -\lambda \delta_k + \frac{\lambda^2}{2} A_k, \tag{2.66}
$$

then by setting the derivative to zero in (2.66), we have

$$\lambda_k = \min\left\{1, \frac{\delta_k}{A_k}\right\}. \tag{2.67}$$

When $\delta_k \geq A_k$, we have from (2.67) that $\lambda_k = 1$. Therefore, from (2.30) we get

$$\delta_{k+1} \leq \delta_k + c_k\left(-\delta_k + \frac{A_k}{2}\right) \leq \delta_k + c_k\left(-\delta_k + \frac{\delta_k}{2}\right) = \left(1 - \frac{c_k}{2}\right)\delta_k,$$

proving (2.31).

On the other hand, since $A \geq A_k > 0, c_k \geq 0$ for all k, (2.30) can be further upper-bounded by

$$\delta_{k+1} \leq \delta_k + c_k\left(-\lambda_k\delta_k + \frac{A_k}{2}\lambda_k^2\right) \leq \delta_k + c_k\left(-\lambda_k\delta_k + \frac{A}{2}\lambda_k^2\right), \quad \forall \lambda_k \in [0,1].$$

Now take

$$\lambda_k = \min\left\{1, \frac{\delta_k}{A}\right\}. \tag{2.68}$$

For $\delta_k \geq A \geq A_k$, (2.31) still applies. If $A > \delta_k$, we have from (2.68) that $\lambda_k = \delta_k/A$, hence

$$\delta_{k+1} \leq \delta_k - \frac{c_k}{2A}\delta_k^2. \tag{2.69}$$

This together with (2.31) imply that $\{\delta_k\}$ is a monotonically decreasing sequence. Dividing both sides of (2.69) by $\delta_{k+1}\delta_k$, and from the fact that δ_k is decreasing and nonnegative, we conclude

$$\delta_k^{-1} \leq \delta_{k+1}^{-1} - \frac{c_k\delta_k}{2\delta_{k+1}A} \leq \delta_{k+1}^{-1} - \frac{c_k}{2A}$$

Summing this inequality from k_0, and using $\delta_{k_0} < A$, we obtain

$$\delta_k^{-1} \geq \delta_{k_0}^{-1} + \frac{\sum_{t=k_0}^{k-1} c_t}{2A} \geq \frac{\sum_{t=k_0}^{k-1} c_t + 2}{2A} \Rightarrow \delta_k \leq \frac{2A}{\sum_{t=k_0}^{k-1} c_t + 2},$$

proving (2.32). □

3 INEXACT VARIABLE METRIC STOCHASTIC BLOCK-COORDINATE DESCENT FOR REGULARIZED OPTIMIZATION

3.1 Introduction

In this chapter, we consider the following regularized convex minimization problem:

$$\min_x F(x) := f(x) + \psi(x), \tag{3.1}$$

where f is block-wise Lipschitz-continuously differentiable but not necessarily convex, and ψ is convex, extended-valued, proper, closed, and block-separable, but possibly nondifferentiable. We assume F is lower-bounded and the solution set Ω is non-empty. For simplicity, we assume $x \in \mathbb{R}^n$, but our methods can be applied to matrix variables too. We decompose $x \in \mathbb{R}^n$ into N blocks such that

$$x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^n, \quad x_i \in \mathbb{R}^{n_i}, \quad n_i \in \mathbb{N}, \quad \sum_{i=1}^{N} n_i = n,$$

and assume throughout that the function ψ can be decomposed into $\psi(x) = \sum_{i=1}^{N} \psi_i(x_i)$, with all ψ_i convex. Many regularized empirical risk minimization (ERM) problems in machine learning have this structure with $n_i > 1$ for all i, see, for example, Yuan and Lin (2006); Meier et al. (2008); Crammer and Singer (2002); Lebanon and Lafferty (2002); Tsochantaridis et al. (2005); Lee and Lin (2013). For the block-separability of x, we use the column submatrices of the identity denoted by $U_1, \ldots, U_N$, where U_i corresponds to the indices in the ith block of x. Thus we have

$$x_i = U_i^\top x, \quad x = \sum_i^N U_i x_i, \quad \text{and } \nabla_i f = U_i^\top \nabla f.$$

By f being block-wise Lipschitz-continuously differentiable, we mean that there exist constants $L_i > 0$ for all i such that

$$\|\nabla_i f(x + U_i h) - \nabla_i f(x)\| \leq L_i \|h\|, \quad \forall h \in \mathbb{R}^{n_i}, \quad \forall x \in \mathbb{R}^n. \tag{3.2}$$

We consider randomized block-coordinate-descent (BCD) type methods to optimize (3.1) where only one block of variables is updated each time. Moreover, we define subproblems with varying quadratic terms, and use possibly non-uniform sampling to select which blocks to update at each step. Significantly, in order to accommodate general quadratic terms and complicated regularizers ψ_i, we also allow inexactness in computation of the update step.

The kth iteration of the "exact" version of our approach proceeds as follows. Given the current iterate x^k, we pick a block i, according to some discrete probability distribution over $\{1, \ldots, N\}$, and minimize a quadratic approximation of f plus the original ψ, restricted to block i, to obtain the update direction d_i^k. That is,

$$d_i^k \coloneqq \arg\min_{d \in \mathbb{R}^{n_i}} Q_{H_i^k}^{x^k}(d), \tag{3.3}$$

where

$$Q_{H_i^k}^{x^k}(d) \coloneqq \nabla_i f\left(x^k\right)^\top d + \frac{d^\top H_i^k d}{2} + \psi_i\left(x_i^k + d\right) - \psi_i\left(x_i^k\right), \tag{3.4}$$

H^k is some positive-definite matrix that can change over iterations, and $H_i^k \in \mathbb{R}^{n_i \times n_i}$ is the ith diagonal block of H^k. (The subscript i refers to block index rather than coordinate index.) A backtracking line search along d_i^k is then performed to determine the step. We denote the objective in (3.4) by $Q_i^k(\cdot)$, or by Q_i when discussing the internal workings of some iteration k.

Stochastic methods of this type have been discussed in existing works

(Nesterov, 2012; Tappenden et al., 2016; Fountoulakis and Tappenden, 2018), but under various assumptions that may be impractical for some problems. Nesterov (2012) require the prior knowledge of the component-wise Lipschitz constants, and assume that (3.3) is solved to optimality, which is usually possible only when ψ_i possesses some simple structure and H^k is diagonal. Tappenden et al. (2016) restricts H^k to be fixed over iterations. Its extension Fountoulakis and Tappenden (2018) is close to our framework, but their subproblem termination condition may be expensive to check except for specific choices of ψ (as they point out). By contrast, we aim for more general applicability by requiring only that (3.3) is solved arbitrarily inexactly, in a sense defined below in (3.5). Moreover, these works consider only uniform sampling for the regularized problem in which $\psi \not\equiv 0$.[1] Since Nesterov (2012) showed possible advantages of non-uniform sampling in the non-regularized (smooth) case, we wish to consider non-uniform sampling in the regularized setting too. Others studied the cyclic version under different assumptions (Chouzenoux et al., 2016; Sun and Hong, 2015; Tseng and Yun, 2009; Yun, 2014), but it is known that this variant is significantly slower than the randomized one in the worst case (Sun and Ye, 2016).

This chapter contributes both to theory and practice. From the practical angle, we extend randomized BCD for regularized functions to a more flexible framework, involving variable quadratic terms and line searches, recovering existing BCD algorithms as special cases. Knowledge of block-wise Lipschitz constants is not assumed. We thus develop more practical algorithms, applicable to wider problem classes, including convex and nonconvex ones, without prior knowledge of parameters. Our framework leads to algorithms that are significantly faster than existing ones when applied to real-world problems. The theoretical contributions are as follows.

[1] For the special case $\psi \equiv 0$, works including Tappenden et al. (2016) considered arbitrary samplings.

1. For convex problems, our analysis reflects a phenomenon that is widely observed in practice for BCD on convex problems: fast Q-linear convergence in the early stages of the algorithm, until a modest degree of suboptimality is attained. This result can be used to strongly weaken the dependency of the iteration complexity on the initial objective value.

2. We show that global linear convergence holds under an assumption weaker than strong convexity. By combining this fact with the fast rate above, even for strongly convex problems, we can get iteration complexities sharper than existing analyses.

3. Our convergence analysis allows arbitrary sampling probabilities for the blocks; we show that non-uniform distributions can reduce the iteration complexity significantly in some cases including both convex and nonconvex problems.

4. We show that the inexactness of subproblem solution affects the convergence rates in a benign way. It follows that if approximate solutions can be obtained cheaply for the subproblems, overall running time of the algorithm can be reduced significantly.

Special cases of our algorithm of diagonal H extends existing analysis for smaller classes of problems, showing that for (3.1), with the additional information of blockwise Lipschitz constants, sampling with probability proportional to the value of these constants L_i enjoys the same improvement of the convergence by a factor of $L_{\max}/L_{\text{avg}}$, where

$$L_{\max} := \max_{1\leq i\leq N} L_i, \quad L_{\text{avg}} := \frac{\sum_{i=1}^N L_i}{N}, \quad \text{and } L_{\min} := \min_{1\leq i\leq N} L_i,$$

over uniform sampling, a novel result in the regularized setting (3.1), to our knowledge. We also show the advantage of the same sampling strategy for nonconvex problems, which is novel even for the non-regularized case.

We introduce our assumptions and the proposed algorithm in Section 3.2. Section 3.3 provides detailed convergence analysis for various classes of problems, including the convex and the nonconvex ones as well as the cases where our algorithm enjoys global linear convergence. The special case of traditional BCD with the extension of non-uniform sampling is studied in Section 3.4. We then discuss related works in Section 3.5 and efficient implementation of our algorithm for ERM problems in Section 3.6. Computational results are shown in Section 3.7 with some concluding remarks in Section 3.8.

3.2 Proposed Algorithm

We consider the case in which (3.3) is too difficult to be solved in closed form, so it must be solved inexactly by an iterative method, such as coordinate descent, proximal gradient, or their accelerated variants. We assume that d_i^k is an η-approximate solution to (3.3) for some $\eta \in [0, 1)$ fixed over all k and all i, satisfying the following condition similar to (2.3):

$$\eta \left(Q_i^k (0) - \left(Q_i^k \right)^* \right) \geq Q_i^k \left(d_i^k \right) - \left(Q_i^k \right)^*, \tag{3.5}$$

where $(Q_i^k)^* := \inf_d Q_i^k(d)$. We do not necessarily need to know η or to verify this condition explicitly as discussed in Chapter 2. Our analysis can be extended easily to variable, adaptive choices of η, which might lead to better iteration complexity, but for better interpretability and simplicity, we fix η for all k in the discussion below.

Our Method

In each iteration of our algorithm, with the current iterate x, a block i is chosen according to some discrete probability distribution over $\{1, 2, \ldots, N\}$, with probabilities $p_1, p_2, \ldots, p_N > 0$. For the selected block i, we com-

Algorithm 3 Inexact variable-metric block-coordinate descent for (3.1)

Given $\beta, \gamma \in (0,1)$, $\eta \in [0,1)$, and $x = x^0 \in \mathbb{R}^n$;
for $k = 0, 1, 2, \ldots$ **do**
    Pick a probability distribution $p_1, \ldots, p_N > 0$, $\sum_i p_i = 1$, and sample i accordingly;
    Compute $\nabla_i f(x)$ and choose a positive-definite H_i;
    Approximately solve (3.3) to obtain a solution d_i satisfying (3.5);
    Compute Δ_i by (3.7), $\alpha_i \leftarrow 1$;
    while (3.6) is not satisfied **do**
        $\alpha_i \leftarrow \beta \alpha_i$;
    $x \leftarrow x + \alpha_i U_i d_i$;

pute $\nabla_i f$, and choose a positive-definite H_i, thus defining the subproblem (3.4). The selection of H_i is application-dependent, but in most cases, the (generalized) Hessian, [2] its quasi-Newton approximation, and its diagonal entries, possibly plus a diagonal damping term, are obvious choices. We then find an approximate solution to (3.3) that satisfies (3.5) for some $\eta \in [0,1)$.

After obtaining d_i, we conduct a backtracking line search, as in Tseng and Yun (2009), and require a sufficient decrease condition to hold. Given $\beta, \gamma \in (0,1)$, we let α_i be the largest value in $\{1, \beta^1, \beta^2, \ldots\}$ such that

$$F\left(x + \alpha_i U_i d_i\right) \leq F\left(x\right) + \alpha_i \gamma \Delta_i \tag{3.6}$$

holds, where

$$\Delta_i \coloneqq \nabla_i f\left(x\right)^\top d_i + \psi_i\left(x_i + d_i\right) - \psi_i\left(x_i\right). \tag{3.7}$$

Then the iterate is updated to $x + \alpha_i U_i d_i$. Our algorithm is summarized in Algorithm 3.

[2] Since $\nabla_i f$ is Lipschitz continuous, it is differentiable almost everywhere. Therefore, we can at least define a generalized Hessian as suggested by Hiriart-Urruty et al. (1984).

3.3 Convergence Analysis

The convergence analysis extends that in Chapter 2, which can be considered as a special case of the framework in this chapter with $N = 1$ block. Nontrivial modifications are needed to allow for multiple blocks and non-uniform sampling. The following result about lower bound on the step size tracks Corollary 2.3 and its proof is therefore removed.

Lemma 3.1. *If the ith block is selected, $H_i \succeq m_i$ for some $m_i > 0$, and (3.5) holds, then we have*

$$\Delta_i \leq -\frac{1}{1+\sqrt{\eta}} d_i^\top H_i d_i \leq -\frac{m_i}{1+\sqrt{\eta}} \|d_i\|^2. \tag{3.8}$$

Moreover, the backtracking line search procedure in Algorithm 3 terminates finitely with a step size lower bounded by

$$\alpha_i \geq \bar{\alpha}_i := \min\left\{1, \frac{2\beta(1-\gamma)m_i}{L_i\left(1+\sqrt{\eta}\right)}\right\}. \tag{3.9}$$

The bound $\bar{\alpha}_i$ in (3.9) is a worst-case guarantee. For properly selected H_i (for example, when H_i includes true second-order information about f confined to the ith block), the steps will usually be closer to 1 because the last inequality in (3.8) can be loose in this case.

We proceed on to deal with the case that F is convex and that F is not necessarily convex, respectively.

Convex Case

The following technical lemma is crucial for both the convergence rate proofs and for motivating the choice of p_i, $i = 1, 2, \ldots, N$.

Lemma 3.2. *If f and ψ are convex and F satisfies (2.10) for $\mu \geq 0$, then for any point x, matrices $H_i \succeq 0$ with $H_i \in \mathbb{R}^{n_i \times n_i}$, probability distribution $\{p_i\} > 0$,*

and step sizes $\{\alpha_i\} > 0$*, by defining*

$$\mathcal{P} \coloneqq \mathit{diag}(p_1 I_{n_1}, \ldots, p_N I_{n_N}), \quad \mathcal{A} \coloneqq \mathit{diag}(\alpha_1 I_{n_1}, \ldots, \alpha_N I_{n_N}),$$
$$H \coloneqq \mathit{diag}(H_1, \ldots, H_N),$$

we have that for Q_i *defined by* (3.4) *with the given* H_i *and* x*, the following holds for all* $\lambda \in [0, 1]$ *and all* θ *such that* $0 \leq \theta \leq \alpha_i p_i$*,* $i = 1, \ldots, N$*:*

$$\mathbb{E}_i \left[\alpha_i Q_i^* \middle| x \right] \leq \theta\lambda \left(F^* - F\left(x\right) \right) - \frac{\mu\theta\lambda \left(1 - \lambda\right) \left\| x - P_\Omega\left(x\right) \right\|^2}{2} + \frac{\theta^2\lambda^2}{2} \left(x - P_\Omega\left(x\right) \right)^\top \mathcal{P}^{-1}\mathcal{A}^{-1} H \left(x - P_\Omega\left(x\right) \right). \tag{3.10}$$

Proof. Given any $d \in \mathbb{R}^n$, let $\tilde{d} \coloneqq \mathcal{A}\mathcal{P}d$. We can then obtain by change of variables that

$$\begin{aligned}
&\mathbb{E}_i[\alpha_i Q_i^* | \, x] \\
&= \min_d \quad \nabla f\left(x\right)^\top \mathcal{A}\mathcal{P}d + \frac{1}{2} d^\top H \mathcal{A}\mathcal{P}d + \sum_{i=1}^N \alpha_i p_i \left(\psi_i\left(x_i + d_i\right) - \psi_i\left(x_i\right) \right) \\
&= \min_{\tilde{d}} \quad \nabla f\left(x\right)^\top \tilde{d} + \frac{1}{2} \tilde{d}^\top \mathcal{P}^{-1}\mathcal{A}^{-1} H \tilde{d} + \sum_{i=1}^N \alpha_i p_i \left(\psi_i\left(x_i + \frac{\tilde{d}_i}{\alpha_i p_i} \right) - \psi_i\left(x_i\right) \right) \\
&\leq \min_{\tilde{d}} \min_{\theta : \theta \in [0,1], \frac{\theta}{\alpha_i p_i} \leq 1, \forall i} \nabla f\left(x\right)^\top \left(\theta\tilde{d}\right) + \frac{1}{2} \left(\theta\tilde{d}\right)^\top \mathcal{P}^{-1}\mathcal{A}^{-1} H \left(\theta\tilde{d}\right) + \\
&\qquad \sum_{i=1}^N \alpha_i p_i \left(\psi_i\left(x_i + \frac{\theta\tilde{d}_i}{\alpha_i p_i} \right) - \psi_i\left(x_i\right) \right).
\end{aligned} \tag{3.11}$$

Next, from the convexity of f, we have

$$\nabla f\left(x\right)^\top \theta\tilde{d} = \theta \left(\nabla f\left(x\right)^\top \tilde{d} \right) \leq \theta \left(f\left(x + \tilde{d} \right) - f\left(x\right) \right),$$

and from that $\theta/(\alpha_i p_i) \leq 1$ for all i and the convexity of ψ, we get

$$\begin{aligned}\psi_i\left(x_i + \frac{\theta\tilde{d}_i}{\alpha_i p_i}\right) &\leq \left(1 - \frac{\theta}{\alpha_i p_i}\right)\psi_i\left(x_i\right) + \frac{\theta}{\alpha_i p_i}\psi_i\left(x_i + \tilde{d}_i\right) \\ &= \frac{\theta}{\alpha_i p_i}\left(\psi_i\left(x_i + \tilde{d}_i\right) - \psi_i\left(x_i\right)\right) + \psi_i\left(x_i\right).\end{aligned}$$

Therefore,

$$\begin{aligned}&\min_{\tilde{d}} \min_{\theta:\theta\in[0,1],\frac{\theta}{\alpha_i p_i}\leq 1,\forall i} \nabla f\left(x\right)^\top\left(\theta\tilde{d}\right) + \frac{1}{2}\left(\theta\tilde{d}\right)^\top \mathcal{P}^{-1}\mathcal{A}^{-1}H\left(\theta\tilde{d}\right) + \\ &\qquad \sum_{i=1}^{N}\alpha_i p_i\left(\psi_i\left(x_i + \frac{\theta\tilde{d}_i}{\alpha_i p_i}\right) - \psi_i\left(x_i\right)\right) \\ \leq\ &\min_{\tilde{d}} \min_{\theta:\theta\in[0,1],\frac{\theta}{\alpha_i p_i}\leq 1,\forall i} \theta\left(F(x+\tilde{d}) - F(x)\right) + \frac{\theta^2}{2}\tilde{d}^\top\mathcal{P}^{-1}\mathcal{A}^{-1}H\tilde{d} \\ \leq\ &\min_{\lambda\in[0,1]} \min_{\theta:\theta\in[0,1],\frac{\theta}{\alpha_i p_i}\leq 1,\forall i} \theta\left(F\left(x + \lambda\left(P_\Omega\left(x\right) - x\right)\right) - F\left(x\right)\right) + \\ &\qquad \frac{\theta^2\lambda^2}{2}\left(P_\Omega\left(x\right) - x\right)^\top P^{-1}\mathcal{A}^{-1}H\left(P_\Omega\left(x\right) - x\right). \qquad (3.12)\end{aligned}$$

The desired result (3.10) then follows from combining (3.11)-(3.12) and (2.10).

□

By the positive-definiteness of H, (3.6) implies that

$$\begin{aligned}F\left(x + \alpha_i U_i d_i\right) - F\left(x\right) &\leq \gamma\alpha_i\left(\Delta_i + \frac{1}{2}d_i^\top H_i d_i\right) = \gamma\alpha_i Q_i(d_i) \\ &\leq (1-\eta)\gamma\alpha_i Q_i^*. \qquad (3.13)\end{aligned}$$

Thus Lemma 3.2 can be applied to the right-hand side of this bound to obtain an estimate of the decrease in F at the current step.

Now we are ready to state the convergence speed results. Given any

x^0, we define

$$R_0 \coloneqq \sup_{x:f(x)\leq f(x^0)} \|x - P_\Omega(x)\|. \tag{3.14}$$

For the case of general convex problems, we make the additional assumption that for any x^0, R_0 defined in (3.14) is finite and obtain the following convergence rate results.

Theorem 3.3. *Assume that f and ψ are convex and (3.2) holds. If at all iteration of Algorithm 3 (3.5) is satisfied with a fixed $\eta \in [0, 1)$, and H_i^k are chosen such that*

$$H_i^k \succeq m_i, \quad k = 0, 1, \ldots, \tag{3.15}$$

for some $m_i > 0$ for all i. We can then guarantee that the step sizes α_i^k are lower-bounded away from zero for all i and all k by Lemma 3.1 and get the following.

1. *At the kth iteration, given any probability distribution $\{p_i^k\}_{i=1}^N > 0$ for picking the block i_k to update, and given the final step sizes $\{\alpha_i^k\}_{i=1}^N > 0$ generated by the backtracking line search for different blocks. Let*

$$\mathcal{P}_k \coloneqq diag(p_1^k I_{n_1}, \ldots, p_N^k I_{n_N}), \quad \mathcal{A}_k \coloneqq diag(\alpha_1^k I_{n_1}, \ldots, \alpha_N^k I_{n_N}),$$
$$H^k \coloneqq diag(H_1^k I_{n_1}, \ldots, H_N^k I_{n_N}).$$

If

$$F\left(x^k\right) - F^* \geq \left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1} \mathcal{A}_k^{-1} H^k \left(x^k - P_\Omega\left(x^k\right)\right) \min_{1\leq i\leq N} \alpha_i^k p_i^k,$$

the convergence rate of the expected objective value is Q-linear:

$$\frac{\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \middle| x^k\right]}{\left(F\left(x^k\right) - F^*\right)} \leq \left(1 - \frac{(1-\eta)\,\gamma \min_{1\leq i\leq N} \alpha_i^k p_i^k}{2}\right). \tag{3.16}$$

2. *Assume in addition that there exist* $M_i \geq m_i, i = 1, \ldots, N$, *and we define*

$$M \coloneqq diag(M_1 I_{n_1}, \ldots, M_N I_{n_N}), \quad \bar{\mathcal{A}} \coloneqq diag(\bar{\alpha}_1 I_{n_1}, \ldots, \bar{\alpha}_N I_{n_N}),$$

where $\bar{\alpha}_i$ *are defined in Lemma 3.1. Given a probability distribution* $\{p_i\} > 0$ *and define*

$$k_0 \coloneqq \arg\min \left\{ k : F\left(x^k\right) - F^* < \|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\| \min_{1\leq i\leq N} \bar{\alpha}_i p_i R_0^2 \right\}, \tag{3.17}$$

if for all $k \geq k_0$,

$$M_i \succeq H_i^k \succeq m_i, \quad i = 1, \ldots, N \tag{3.18}$$

and the sampling of i_k *follows the distribution* $\{p_i\}$ *for all* $k \geq k_0$, *then the expected objective follows a sublinear convergence rate*

$$\mathbb{E}_{i_{k_0}, i_{k_0+1}, \ldots, i_{k-1}} \left[F\left(x^k\right) \middle| x^{k_0} \right] - F^* \leq \frac{2\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\| R_0^2}{2N + (1-\eta)\gamma(k-k_0)}. \tag{3.19}$$

3. *If a fixed probability distribution* $\{p_i\}_{i=1}^N > 0$ *is used throughout to sample the blocks and* (3.18) *holds at all iterations, then we have that for all* $k < \bar{k}_0$, *where*

$$\bar{k}_0 \coloneqq \left\lceil \max \left\{ 0, \frac{\log \frac{F\left(x^0\right) - F^*}{\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| \min_i \bar{\alpha}_i p_i R_0^2}}{\log\left(\frac{2}{2-(1-\eta)\gamma \min_i \bar{\alpha}_i p_i}\right)} \right\} \right\rceil, \tag{3.20}$$

the expected objective is upper bounded by

$$\mathbb{E}_{i_0, \ldots, i_{k-1}} \left[F\left(x^k\right) - F^* \middle| x^0 \right] \leq \left(1 - \frac{(1-\eta)\gamma \min_i \bar{\alpha}_i p_i}{2}\right)^k \left(F\left(x^0\right) - F^*\right), \tag{3.21}$$

and for all $k \geq \bar{k}_0$, the expected objective is upper bounded by

$$\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[F\left(x^k\right) - F^* \middle| x^0\right] \leq \frac{2\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2}{2N + (1-\eta)\gamma(k-\bar{k}_0)}. \tag{3.22}$$

Proof. Consider Lemma 3.2. For the general convex case, we have $\mu = 0$ and thus only two terms are left in (3.10):

$$\begin{aligned} &\mathbb{E}_{i_k}\left[\alpha_{i_k}^k \left(Q_{i_k}^k\right)^* \middle| x^k\right] \\ &\leq \theta\lambda\left(F^* - F\left(x^k\right)\right) + \frac{\theta^2\lambda^2}{2}\left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\left(x^k - P_\Omega\left(x^k\right)\right) \end{aligned} \tag{3.23}$$

for all $\lambda \in [0,1]$ and all $\theta \in [0, \min_i \alpha_i^k p_i^k]$. We then set $\theta = \min_{1\leq i\leq N} \alpha_i^k p_i^k$ because the whole possible range of the right-hand side of (3.23) remains unchanged by manipulating λ alone. Since the right-hand side of (3.23) is a strongly convex function of λ (provided $x \notin \Omega$, in which case we have reached the optimal objective and there is nothing to prove), we can find the maximum by setting the derivative to zero, or its projection to the feasible set $[0,1]$ if the optimum occurs outside the range. Therefore,

$$\lambda = \min\left\{1, \frac{F\left(x^k\right) - F^*}{\left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1}\bar{\mathcal{A}}_k^{-1}H^k\left(x^k - P_\Omega\left(x^k\right)\right)\min_i \alpha_i^k p_i^k}\right\}. \tag{3.24}$$

By this choice, when

$$F\left(x^k\right) - F^* \geq \left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\left(x^k - P_\Omega\left(x^k\right)\right)\min_{1\leq i\leq N} \alpha_i^k p_i^k,$$

we have $\lambda = 1$ and (3.23) becomes

$$\mathbb{E}_{i_k}\left[\alpha_{i_k}^k \left(Q_{i_k}^k\right)^*\right] \leq \frac{1}{2}\min_{1\leq i\leq N} \alpha_i^k p_i^k \left(F^* - F\left(x^k\right)\right). \tag{3.25}$$

By combining (3.25) and (3.13), we have proven (3.16).

Now consider (3.23) again but with α_i^k replaced by $\bar{\alpha}_i$ and p_i^k replaced with p_i and take $k \geq k_0$. We define

$$\delta_k := \mathbb{E}_{i_{k_0},\ldots,i_{k-1}}\left[F\left(x^k\right) - F^* \middle| x^{k_0}\right]$$

and take expectation on both sides of (3.23) over $i_{k_0}, \ldots, i_{k-1}$ conditional on x^{k_0}, then by the definition (3.14) and the bound (3.15), we get

$$\mathbb{E}_{i_{k_0},\ldots,i_k}\left[\bar{\alpha}_{i_k}\left(Q_{i_k}^k\right)^* \middle| x^{k_0}\right] \leq -\theta\lambda\delta_k + \frac{\theta^2\lambda^2}{2}\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2. \tag{3.26}$$

Consider $\theta = \min_{1\leq i\leq N}\bar{\alpha}_i p_i$ in (3.26), we get that since Algorithm 3 is a descent method, $\delta_k < \min_{1\leq i\leq N}\bar{\alpha}_i p_i \left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2$, for all $k \geq k_0$. Therefore, we can use

$$\lambda = \frac{\delta_k}{\min_{1\leq i\leq N}\bar{\alpha}_i p_i \left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2}$$

in (3.26) to get

$$\begin{aligned}\mathbb{E}_{i_{k_0},\ldots,i_k}\left[\bar{\alpha}_{i_k}\left(Q_{i_k}^k\right)^* \middle| x^{k_0}\right] \leq & -\min_{1\leq i\leq N}\bar{\alpha}_i p_i \frac{\delta_k^2}{2\min_{1\leq i\leq N}\bar{\alpha}_i p_i \left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2} \\ = & -\frac{\delta_k^2}{2\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2}.\end{aligned} \tag{3.27}$$

Therefore, taking expectation on (3.13) over $i_{k_0}, \ldots, i_k$ and conditional on x^{k_0} and with (3.27) gives

$$\delta_{k+1} \leq \delta_k - \frac{(1-\eta)\gamma\delta_k^2}{2\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2}. \tag{3.28}$$

By dviding both sides of (3.28) by $\delta_k\delta_{k+1}$ and noting from Lemma 3.1 and (3.6) that $\{F(x_k)\}$ and therefore $\{\delta_k\}$ is descending, we get

$$\frac{1}{\delta_k} \leq \frac{1}{\delta_{k+1}} - \frac{(1-\eta)\gamma\delta_k}{2\delta_{k+1}\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2} \leq \frac{1}{\delta_{k+1}} - \frac{(1-\eta)\gamma}{2\left\|\mathcal{P}^{-1}\bar{\mathcal{A}}^{-1}M\right\| R_0^2}. \tag{3.29}$$

Further by telescoping, (3.29) leads to

$$\frac{1}{\delta_k} \geq \frac{1}{\delta_{k_0}} + (k - k_0) \frac{\gamma (1 - \eta)}{2 \left\| \mathcal{P}^{-1} \bar{\mathcal{A}}^{-1} M \right\| R_0^2}. \tag{3.30}$$

Finally, note that because $\bar{\alpha}_i \in [0, 1]$ for $i = 1, \ldots, N$, (3.17) implies that

$$\frac{1}{\delta_{k_0}} \geq \frac{1}{\min_i \bar{\alpha}_i p_i \left\| \mathcal{P}^{-1} \bar{\mathcal{A}}^{-1} M \right\| R_0^2} \geq \frac{1}{\min_i p_i \left\| \mathcal{P}^{-1} \bar{\mathcal{A}}^{-1} M \right\| R_0^2}. \tag{3.31}$$

Next, it is straightforward that the solution to

$$\begin{aligned} \min_{p_1, \ldots, p_N} \quad & \frac{1}{\min_{1 \leq i \leq N} p_i} \\ \text{subject to} \quad & \sum_{i=1}^{N} p_i = 1, \\ & p_i \geq 0, i = 1, \ldots, N, \end{aligned}$$

is $p_i \equiv 1/N$ and the corresponding objective value is N. Therefore, (3.31) is improved to

$$\frac{1}{\delta_{k_0}} \geq \frac{N}{\left\| \mathcal{P}^{-1} \bar{\mathcal{A}}^{-1} M \right\| R_0^2}.$$

Combining the inequality above with (3.30) then proves (3.19).

□

The first part of Theorem 3.3 has been observed frequently in practice, and some restricted special cases without a regularizer has been discussed in the literature Lee and Wright (2018b); Wright and Lee (2017). However, to our knowledge, this is the first time that a theoretical proof for BCD-type methods on general regularized problems (3.1) is given. Other works in the literature usually have global convergence bounds dependent on $R_0^2 + F(x^0) - F^*$, here our results significantly weaken the dependency on the initial objective value.

We can see from the second item of Theorem 3.3 that the optimal probability distribution after k_0 iterations is

$$p_i = \frac{M_i \bar{\alpha}_i^{-1}}{\sum_j M_j \bar{\alpha}_j^{-1}}. \tag{3.32}$$

It is also possible to replace $\bar{\alpha}_i$ and M_i with the real α_i^k's $\|H_i^k\|$'s to get adaptive probabilities and sharper rates. Note that although in Algorithm 3, p_i can change over iterations, we fix the probability over iterations for more succinct analysis, possibly sacrificing some sharper convergence rates.

We now consider the case that F satisfies the quadratic growth condition

$$F(x) - F^* \geq \frac{\mu}{2} \|x - P_\Omega(x)\|^2 \tag{3.33}$$

for some $\mu > 0$. This is a condition implied by (2.10) but not vice versa. We can get a global Q-linear convergence as shown in the following theorem.

Theorem 3.4. *Assume that f and ψ are convex and (3.2) and (3.33) hold for some $L_1, \ldots, L_N, \mu > 0$. If at the kth iteration of Algorithm 3, (3.5) is satisfied with some $\eta \in [0, 1)$ and H^k is chosen such that (3.15) holds for some $m_i > 0$ for all i so that the step sizes α_i^k are all lower bounded away from 0 as suggested by Lemma 3.1. Then given any probability distribution $\{p_i^k\} > 0$, we have*

$$\frac{\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \middle| x^k\right]}{F\left(x^k\right) - F^*} \tag{3.34}$$

$$\leq \begin{cases} 1 - \frac{(1-\eta)\gamma \min_i \alpha_i^k p_i^k}{2}, \\ \qquad \text{if} \quad \frac{F\left(x^k\right) - F^*}{\left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1} \mathcal{A}_k^{-1} H^k \left(x^k - P_\Omega\left(x^k\right)\right)} \geq \min_i \alpha_i^k p_i^k, \\ 1 - \frac{(1-\eta)\gamma\mu}{4\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\|}, \quad \text{if} \quad \frac{\mu}{2\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\| \min_i \alpha_i^k p_i^k} \leq 1, \\ 1 - (1-\eta)\gamma\left(\min_i \alpha_i^k p_i^k \left(\frac{\min_i \alpha_i^k p_i^k \|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\|}{\mu} - 1\right)\right), \quad \text{else}. \end{cases}$$

Proof. The first case is directly from Theorem 3.3. For the remaining two cases. By (3.13), (3.23), the Cauchy-Schwarz inequality, and (3.33), we have that

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right)-F\left(x^k\right)\mid x^k\right]\\
\leq\ &\gamma(1-\eta)\mathbb{E}_{i_k}\left[\alpha_{i_k}^k\left(Q_{i_k}^k\right)^*\right]\\
\leq\ &\gamma(1-\eta)\theta\left(F\left(x^k\right)-F^*\right)\left(-\lambda+\frac{\theta\lambda^2\left\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\right\|}{\mu}\right), \qquad (3.35)\\
&\forall\lambda\in[0,1],\forall\theta\in[0,\min_{1\leq i\leq N}\alpha_i p_i].
\end{aligned}$$

By the same argument in the previous proofs, we let $\theta=\min_{1\leq i\leq N}\alpha_i^k p_i^k$. To minimize the right-hand side of (3.35), we set its derivative with respect to λ to 0, which is guaranteed to be the minimizer because the right-hand side of (3.35) is strongly convex with respect to λ. This leads to

$$\lambda=\max\left\{\frac{\mu}{2\theta\left\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\right\|},1\right\},\quad \theta=\min_{1\leq i\leq N}\alpha_i^k p_i^k. \qquad (3.36)$$

Substituting (3.36) back to (3.35) and substracting F^* from both sides of (3.35) then proves (3.34).

□

Next, for problems satisfying (2.10), we have a faster convergence result.

Theorem 3.5. *Assume that f and ψ are convex and (3.2) and (2.10) hold for some $L_1,\ldots,L_N,\mu>0$. If at the kth iteration of Algorithm 3, (3.5) is satisfied with some $\eta\in[0,1)$ and H^k is chosen such that (3.15) holds for some $m_i>0$ for all i so that the step sizes α_i^k are all lower bounded away from 0 as suggested*

by Lemma 3.1. Then given any probability distribution $\{p_i^k\} > 0$*, we have*

$$\frac{\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \middle| x^k\right]}{F\left(x^k\right) - F^*} \leq \begin{cases} 1 - \frac{(1-\eta)\gamma \min_i \alpha_i^k p_i^k}{2}, \\ \qquad \textit{if} \quad \frac{F\left(x^k\right) - F^*}{\left(x^k - P_\Omega\left(x^k\right)\right)^\top \mathcal{P}_k^{-1} \mathcal{A}_k^{-1} H^k \left(x^k - P_\Omega\left(x^k\right)\right)} \geq \min_i \alpha_i^k p_i^k, \\ \left(1 - \frac{(1-\eta)\gamma}{\max_i \frac{1}{\alpha_i^k p_i^k} + \max_i \frac{\left\|H_i^k\right\|}{\mu \alpha_i^k p_i^k}}\right), \quad \textit{else}. \end{cases} \tag{3.37}$$

Proof. The first case in (3.37) is directly from Theorem 3.3. To prove the second case in (3.37), we let $\lambda = \mu/(\mu + \|\mathcal{P}^{-1}\mathcal{A}^{-1}H\|\theta)$ to cancel out the last two terms in (3.10), and this value of λ is clearly within the range $[0, 1]$. This together with setting $\theta = \min_{1 \leq i \leq N} \alpha_i^k p_i^k$ then implies

$$\begin{aligned} \mathbb{E}_{i_k}\left[\alpha_{i_k}\left(Q_{i_k}^k\right)^*\right] &\leq \frac{\mu\theta}{\mu + \left\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\right\|\theta}\left(F^* - F\left(x^k\right)\right) \\ &= \frac{1}{\frac{1}{\theta} + \frac{\left\|\mathcal{P}_k^{-1}\mathcal{A}_k^{-1}H^k\right\|}{\mu}}\left(F^* - F\left(x^k\right)\right) \\ &= \frac{1}{\max_i \frac{1}{\alpha_i^k p_i^k} + \max_i \frac{\left\|H_i^k\right\|}{\mu \alpha_i^k p_i^k}}\left(F^* - F\left(x^k\right)\right). \end{aligned} \tag{3.38}$$

By combining (3.38) and (3.13), we get the desired result.

□

As we have noted in the respective theorems, for problems on which Theorem 3.4 or 3.5 holds, Theorem 3.3 is also applicable, and the early linear convergence rate is significantly faster than the global ones from Theorems 3.4 and 3.5. Thus, we can sharpen the global iteration complexity for problems satisfying (2.10) with $\mu > 0$ by combining Theorems 3.5 and 3.3. The resulting complexity is therefore tighter than existing results from

global linear convergence rates. We also notice that the rate in Theorem 3.5 is faster than that in Theorem 3.4 and this is why we consider these two conditions separately.

Note too that with knowledge of α_i and $\|H_i^k\|$, we could in principle minimize the expected gap $\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \middle| x^k\right]$ by minimizing the denominator in the right-hand side of (3.37) and (3.34) with respect to p_i^k over $p_i^k > 0$ and $\sum_i p_i^k = 1$. (This is not generally a practical proposition except in the special cases discussed below, as it is usually expensive to get all α_i^k and all $\|H_i^k\|$.)

The results Theorems 3.3 and 3.5 suggest that larger step sizes α_i lead to faster convergence. When H_i^k incorporates curvature information of f, we tend to have much larger step sizes than the lower bound predicted in Lemma 3.1, and thus the practical performance of using the Hessian or its approximation usually outperforms using a multiple of the identity as H_i.

All the results here can be combined with Markov's inequality to get high-probability bounds for the objective value. The proofs are straightforward and therefore omitted.

Nonconvex Case

When f is not necessarily convex, we cannot use Lemma 3.2 anymore. Moreover, we cannot guarantee convergence to the global optima. Instead, we will show convergence of measures of stationarity of our method.

The first measure we consider is how fast the optimal objective of the subproblem (3.3) converges to zero. Since the subproblems are strongly convex, this measure is zero if and only if the optimal solution is the zero vector, implying the iterates will not change anymore. This is clear from the following lemma.

Lemma 3.6. *Assume $H_i \succeq m_i$ for some $m_i > 0$ for all i in (3.3)-(3.4), and (3.2) holds true. Then for any step sizes $\{\alpha_i\}_{i=1}^N > 0$ and any probability distribution*

$\{p_i\}_{i=1}^N > 0$ *for picking the blocks, we have*

$$\mathbb{E}_i[\alpha_i Q_i^*] = 0 \quad \Leftrightarrow \quad Q_i^* = 0, i = 1, \ldots, N \quad \Leftrightarrow \quad 0 \in \partial F(x), \tag{3.39}$$

where $\partial F(x) = \nabla f(x) + \partial\psi(x)$ *is the set of generalized gradient of* F *at* x*.*

Proof. From (3.8) in Lemma 3.1, by setting $\eta = 0$ we see that $Q_i^* \leq 0$ for all i, proving the first equivalence in (3.39).

To prove the second equivalence, we first notice that since Q_i are all strongly convex and $Q_i(0) \equiv 0$, $Q_i^* = 0$ if and only if $d_i^* = 0$, where

$$d_i^* := \arg\min_d Q_i(d).$$

Therefore, it suffices to prove that

$$d_i^* = 0 \quad \Leftrightarrow \quad -\nabla_i f(x) \in \partial\psi_i(x_i), i = 1, \ldots, N. \tag{3.40}$$

Now consider the optimality condition of (3.3). By setting the derivative to zero, we get

$$-(\nabla_i f(x) + H_i d_i^*) \in \partial\psi_i(x_i + d_i^*). \tag{3.41}$$

When $d_i^* = 0$, (3.41) implies that $-\nabla_i f(x) \in \partial\psi_i(x_i)$. Conversely, assume

$$-\nabla_i f(x) \in \partial\psi_i(x_i). \tag{3.42}$$

We have from the convexity of ψ_i that

$$\begin{cases} \psi_i(x_i + d_i^*) & \geq \psi_i(x_i) + h^\top d_i^*, \quad \forall h \in \partial\psi_i(x_i), \\ \psi_i(x_i) & \geq \psi_i(x_i + d_i^*) - r^\top d_i^*, \quad \forall r \in \partial\psi_i(x_i + d_i^*). \end{cases}$$

Combine the above inequalities with (3.41) and (3.42), we get that

$$\begin{cases} \psi_i\left(x_i + d_i^*\right) & \geq \psi_i\left(x_i\right) - \nabla_i f(x)^\top d_i^*, \\ \psi_i\left(x_i\right) & \geq \psi_i\left(x_i + d_i^*\right) + \nabla_i f(x)^\top d_i^* + \left(d_i^*\right)^\top H_i d_i^*. \end{cases}$$

By summing up these two inequalities, we obtain

$$0 \geq \left(d_i^*\right)^\top H_i d_i^*.$$

Since H_i is positive definite, we get that $d_i^* = 0$ as desired.

□

The second measure is how fast the gradient mapping vanishes, which we define as below.

$$G_k \coloneqq \arg\min_d \nabla f(x^k)^\top d + \frac{1}{2} d^\top d + \psi(x^k + d). \tag{3.43}$$

From Lemma 3.6, it is clear that $G_k = 0$ if and only if $0 \in \partial F(x^k)$, and this gradient mapping can serve as an indicator for closeness to stationarity.

Now we show convergence rates for these measures.

Theorem 3.7. *For Algorithm 3, assume there exists $\eta \in [0, 1)$ such that (3.5) holds for all iterations. Given any probability distributions $\{p_i^k\}_{i=1}^N > 0$ for picking the blocks at each k, and let $\{\alpha_i^k\}_{i=1}^N > 0$ be the step sizes generated by the line search procedure. If $H_i^k \succeq 0$ for all i and k, for any given x^0, we have*

$$\min_{0 \leq k \leq T} \left| \mathbb{E}_{i_0, \ldots, i_k} \left[\alpha_{i_k}^k Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \right| \leq \frac{F\left(x^0\right) - F^*}{\gamma\left(T + 1\right)}, \quad \forall T \geq 0. \tag{3.44}$$

Moreover, $\mathbb{E}_{i_0, \ldots, i_k} \left[\alpha_{i_k}^k Q_{i_k}^k (d_{i_k}^k) \middle| x^0 \right]$ converges to 0 as k approaches infinity.

Proof. Take expectation on (3.13) over i_k, we get that

$$\mathbb{E}_{i_k} \left[F\left(x^{k+1}\right) \middle| x^k \right] - F\left(x^k\right) \leq \gamma \mathbb{E}_{i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^k \right]. \tag{3.45}$$

By taking expectation on (3.45) over $i_0, \ldots, i_{k-1}$ conditional on x^0 and summing it from $k = 0$ to $k = T$, and noting from (3.5) and Lemma 3.1 that $Q_i^k(d_i^k) \leq 0$ for all k and all i, we get

$$\begin{aligned}
&\gamma \sum_{k=0}^{T} \left| \mathbb{E}_{i_0,\ldots,i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \right| \\
= & -\gamma \sum_{k=0}^{T} \mathbb{E}_{i_0,\ldots,i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \\
\leq & \sum_{k=0}^{T} \mathbb{E}_{i_0,\ldots,i_{k-1}} \left[F\left(x^k\right) \middle| x^0 \right] - \mathbb{E}_{i_0,\ldots,i_k} \left[F\left(x^{k+1}\right) \middle| x^0 \right] \\
= & F\left(x^0\right) - \mathbb{E}_{i_0,\ldots,i_T} \left[F\left(x^{T+1}\right) \middle| x^0 \right] \leq F\left(x^0\right) - F^*. \qquad (3.46)
\end{aligned}$$

The proof for the convergence rate is then concluded by the fact that

$$\sum_{k=0}^{T} \left| \mathbb{E}_{i_0,\ldots,i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \right| \geq (T+1) \min_{0 \leq k \leq T} \left| \mathbb{E}_{i_0,\ldots,i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \right|$$

That $\left| \mathbb{E}_{i_0,\ldots,i_k} \left[\alpha_{i_k} Q_{i_k}^k \left(d_{i_k}^k \right) \middle| x^0 \right] \right|$ converges to 0 is straightforward from its summability implied by (3.46).

□

Unlike previous results, the result above is independent of how accurate the subproblem is solved, the probability distributions for sampling the blocks, and the step sizes. From it we cannot see any difference in using different sampling strategies and using different η. We next take the second measure (3.43) and show that its convergence is relavant to these factors. We will need the following lemma from Tseng and Yun (2009).

Lemma 3.8 ((Tseng and Yun, 2009, Lemma 3)). *Given x^k and assume H^k satisfies (3.18) for all i, we have*

$$\left\| U_i^\top G_k \right\| \leq \frac{1 + \frac{1}{m_i} + \sqrt{1 - 2\frac{1}{M_i} + \frac{1}{m_i^2}}}{2} M_i \left\| d_i^{k*} \right\|, \text{ where } d_i^{k*} := \arg\min Q_i^k.$$

By using Lemma 3.8 and Theorem 3.7, we are able to show the convergence speed of $\min_{1\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_k}[\|G_k\||x^0]$ for any given x^0.

Corollary 3.9. *Assume that (3.5) holds at all iterations for some $\eta\in[0,1)$. For Algorithm 3, assume that H^k satisfies (3.18) for all $k\geq 0$. Given any probability distributions for sampling the blocks $\{p_i^k\}_{i=1}^N>0$ and let $\{\alpha_i^k\}_{i=1}^N>0$ be the step sizes generated by the line search procedure for all k, we have*

$$\begin{aligned}&\min_{0\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\|G_k\|^2\middle|x^0\right]\\ &\leq\frac{F\left(x^0\right)-F^*}{2(1-\eta)\gamma(T+1)}\max_{0\leq k\leq T,\,1\leq i\leq N}\frac{M_i^2\left(1+\frac{1}{m_i}+\sqrt{1-2\frac{1}{M_i}+\frac{1}{m_i^2}}\right)^2}{p_i^k\alpha_i^k m_i}.\end{aligned}\tag{3.47}$$

Proof. Consider Theorem 3.7 and denote $\bar{k}$ the iteration that corresponds to this smallest expected value. We therefore have from (3.5) and Theorem 3.7 that

$$\begin{aligned}\frac{F\left(x^0\right)-F^*}{\gamma\left(T+1\right)}&\geq\left|\mathbb{E}_{i_0,\ldots,i_{\bar{k}}}\left[\alpha_{i_{\bar{k}}}^{\bar{k}}Q_{i_{\bar{k}}}^{\bar{k}}\left(d_{i_{\bar{k}}}^{\bar{k}}\right)\middle|x^0\right]\right|\\&\geq-(1-\eta)\mathbb{E}_{i_0,\ldots,i_{\bar{k}}}\left[\alpha_{i_{\bar{k}}}^{\bar{k}}\left(Q_{i_{\bar{k}}}^{\bar{k}}\right)^*\middle|x^0\right].\end{aligned}\tag{3.48}$$

Now notice that since $H_i^k\succeq m_i$ from (3.18) and that ψ_i are convex, we have that for all i, Q_i^k for all k are m_i-strongly convex and therefore satisfy (3.33) with $\mu=m_i$. We thus have

$$-\left(Q_i^k\right)^*\geq\frac{m_i}{2}\left\|d_i^{k*}\right\|^2,\quad\forall k,i.\tag{3.49}$$

Apply (3.49) and Lemma 3.8 to (3.48), we get

$$\begin{aligned}
&\frac{F\left(x^0\right)-F^*}{(1-\eta)\gamma\left(T+1\right)} \\
&\geq \sum_{i=1}^{N} \frac{p_i^{\bar{k}}\alpha_i^{\bar{k}}m_i\mathbb{E}_{i_0,\ldots,i_{\bar{k}-1}}\left[\left\|d_i^{\bar{k}*}\right\|^2\middle| x^0\right]}{2} \qquad (3.50)\\
&\geq \sum_{i=1}^{N} \frac{p_i^{\bar{k}}\alpha_i^{\bar{k}}m_i}{M_i^2}\frac{2}{\left(1+\frac{1}{m_i}+\sqrt{1-2\frac{1}{M_i}+\frac{1}{m_i^2}}\right)^2}\mathbb{E}_{i_0,\ldots,i_{\bar{k}-1}}\left[\left\|U_i^\top G_{\bar{k}}\right\|^2\middle| x^0\right] \\
&\geq 2\mathbb{E}_{i_0,\ldots,i_{\bar{k}-1}}\left[\left\|G_{\bar{k}}\right\|^2\middle| x^0\right]\min_{1\leq i\leq N}\frac{p_i^{\bar{k}}\alpha_i^{\bar{k}}m_i}{M_i^2\left(1+\frac{1}{m_i}+\sqrt{1-2\frac{1}{M_i}+\frac{1}{m_i^2}}\right)^2}, \qquad (3.51)
\end{aligned}$$

where in (3.51), we use the fact that $\|x\|^2=\sum_{i=1}^N\left\|U_i^\top x\right\|^2$ for any $x\in\mathbb{R}^n$. Finally, (3.47) is proven by noticing that

$$\min_{0\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\left\|G_k\right\|^2\middle| x^0\right]\leq\mathbb{E}_{i_0,\ldots,i_{\bar{k}-1}}\left[\left\|G_{\bar{k}}\right\|^2\middle| x^0\right].$$

□

The result in Corollary 3.9 reveals as well that line search can help improve the convergence speed, and it is also possible to consider non-uniform sampling to obtain faster convergence.

3.4 Special Case: Traditional Randomized BCD

We now discuss how to extend the result of non-uniform sampling of the blocks for the non-regularized case ($\psi\equiv 0$) in Nesterov (2012) to the regularized problem (3.1), using results in Section 3.3. In the non-regularized case, the update for the ith block described in Nesterov (2012)

is $-\nabla_i f(x)/L_i$, which can be viewed as either the solution of

$$\min_d \quad \nabla_i f(x)^\top d + \frac{L_i d^\top d}{2}$$

with unit step size, or as the solution of

$$\min_d \quad \nabla_i f(x)^\top d + \frac{d^\top d}{2}$$

with step size $1/L_i$. Similarly, in this section, we do not use backtracking; an appropriate choice for α_i is available without performing this part of the algorithm, given the additional knowledge of L_i.

Both viewpoints above result in the same update in the non-regularized case, but with the presence of ψ in (3.1), the two interpretations lead to different updates rules.

We first use a more general setting to show that both approaches achieve a guaranteed degree of function value decrease.

Lemma 3.10. *Assume that* (3.2) *holds. If the* i*th block is selected, and* $H_i \succeq c_i I$ *in* (3.4) *for some* $c_i \in (0, L_i]$*, then* $\hat{\alpha}_i := c_i/L_i$ *satisfies*

$$F(x + \alpha U_i d) - F(x) \leq \alpha Q_{H_i}(d), \quad \forall d \in \mathbb{R}^{n_i}, \forall \alpha \in [0, \hat{\alpha}_i]. \tag{3.52}$$

Proof. Because $c_i \in (0, L_i]$, we have $\hat{\alpha}_i = c_i/L_i \in (0, 1]$. Thus we have from (3.2) and the convexity of ψ that for any $\alpha \in [0, \hat{\alpha}_i]$,

$$\begin{aligned}
& F\left(x + \alpha U_i d\right) \\
&= f\left(x + \alpha U_i d\right) + \psi\left(x + \alpha U_i d\right) \\
&\leq f\left(x\right) + \alpha \nabla_i f\left(x\right)^\top d + \frac{L_i \alpha^2}{2} \left\|d\right\|^2 + \alpha \psi\left(x + U_i d\right) + \left(1 - \alpha\right) \psi\left(x\right) \\
&= f\left(x\right) + \psi\left(x\right) + \alpha Q_{\alpha L_i I}\left(d\right) \\
&\leq F\left(x\right) + \alpha Q_{H_i}\left(d\right).
\end{aligned}$$

Algorithm 4 Inexact Randomized BCD with Unit Step Size for (3.1)

Given $\eta \in [0, 1)$ and $x = x^0 \in \mathbb{R}^n$;
for $k = 0, 1, 2, \ldots$ **do**
    Pick a probability distribution $p_1, \ldots, p_N > 0$, $\sum_i p_i = 1$, and sample i accordingly;
    Compute $\nabla_i f(x)$ and let $H_i = L_i I$;
    Approximately solve (3.3) to obtain a solution d_i satisfying (3.5);
    $x \leftarrow x + U_i d_i$;

Algorithm 5 Inexact Randomized BCD with Short Step Size for (3.1)

Given $\eta \in [0, 1)$ and $x = x^0 \in \mathbb{R}^n$;
Properly scale f such that $L_{\min} \geq 1$;
for $k = 0, 1, 2, \ldots$ **do**
    Pick a probability distribution $p_1, \ldots, p_N > 0$, $\sum_i p_i = 1$, and sample i accordingly;
    Compute $\nabla_i f(x)$ and let $H_i = I$;
    Approximately solve (3.3) to obtain a solution d_i satisfying (3.5);
    $x \leftarrow x + \frac{1}{L_i} U_i d_i$;

In the last inequality, we used the fact that

$$H_i \succeq c_i I = \hat{\alpha}_i L_i I \succeq \alpha L_i I.$$

□

We assume without loss of generality that $L_i \geq 1$ for all i, so that for $H = I$ we can directly apply $\alpha_i = 1/L_i$. (This assumption can be satisfied via scaling the whole problem by a constant factor $L_{\min}^{-1}$.) We summarize the two variants of BCD we described above in Algorithms 4-5. In those two cases, since the step sizes and the eigenvalues of H are all known in advance, we can use the probability given in (3.32) to get better convergence speed. This will be the focus of our discussion in this section.

With the help of Lemma 3.10, we can discuss the iteration complexities

of randomized BCD with different sampling strategies. We first consider the interpretation in Algorithm 4, starting from the case in which f is convex. The results below are direct applications of Theorem 3.3.

Corollary 3.11. *Consider Algorithm 4 and assume that (3.2) holds. If f is convex and (3.5) holds at every iteration for some $\eta \in [0, 1)$, the expected objective value satisfies the following.*

1. *When the uniform sampling $p_i \equiv 1/N$ is used:*

   1.1. *If $F\left(x^k\right) - F^* \geq \left(x^k - P_\Omega\left(x^k\right)\right)^\top L\left(x^k - P_\Omega\left(x^k\right)\right)$, where*

   $$L \coloneqq diag(L_1 I_{n_1}, \ldots, L_N I_{n_N}), \tag{3.53}$$

   we have

   $$\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \mid x^k\right] \leq \left(1 - \frac{(1-\eta)}{2N}\right)\left(F\left(x^k\right) - F^*\right).$$

   1.2. *For all $k \geq k_0$, where $k_0 \coloneqq \arg\min\{k : F\left(x^k\right) - F^* < L_{\max} R_0^2\}$,*

   $$\mathbb{E}_{i_{k_0}, \ldots, i_{k-1}}\left[F\left(x^k\right) \mid x^{k_0}\right] - F^* \leq \frac{2N L_{\max} R_0^2}{2N + (1-\eta)(k - k_0)}.$$

2. *When p_i are selected by*

   $$p_i = \frac{L_i}{N L_{avg}}, \quad i = 1, 2, \ldots, N, \tag{3.54}$$

   2.1. *If $F\left(x^k\right) - F^* \geq L_{\min}\left(x^k - P_\Omega\left(x^k\right)\right)^\top\left(x^k - P_\Omega\left(x^k\right)\right)$:*

   $$\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^* \mid x^k\right] \leq \left(1 - \frac{L_{\min}(1-\eta)}{2N L_{avg}}\right)\left(F\left(x^k\right) - F^*\right)$$

2.2. *For all* $k \geq k_0$, *where* $k_0 := \arg\min\{k : F\left(x^k\right) - F^* < L_{\min} R_0^2\}$,

$$\mathbb{E}_{i_{k_0},\ldots,i_{k-1}}\left[F\left(x^k\right) \mid x^{k_0}\right] - F^* \leq \frac{2NL_{avg}R_0^2}{2N + (1-\eta)(k-k_0)}.$$

The strategy (3.54) is referred to as "Lipschitz sampling" from now on. We notice that since in Algorithms 4 and 5, we always have that

$$\frac{\|H_i\|}{\alpha_i} = L_i,$$

(3.54) matches the optimal probability distribution (3.32) and it results in

$$\|\mathcal{P}^{-1}\mathcal{A}^{-1}M\| = NL_{\text{avg}}.$$

We next consider when (2.10) holds for some $\mu > 0$.

Corollary 3.12. *Consider Algorithm 4 and assume that* (3.2) *holds. For problems satisfying* (2.10) *with* $\mu \in (0, L_{\min}]$, *if* (3.5) *holds at every iteration for some* $\eta \in [0, 1)$, *the iteration complexity for reaching an expected* ϵ*-accurate solution is:*

1. $O\left(\frac{N}{(1-\eta)}\left(1 + \frac{L_{\max}}{\mu}\right)\log\left(1/\epsilon\right)\right)$, *if* $p_i \equiv \frac{1}{N}$.
2. $O\left(\frac{NL_{avg}}{(1-\eta)\mu}\log\left(1/\epsilon\right)\right)$, *if* (3.54) *is used.*

Proof. As shown in Lemma 3.10, this choice of H_i and α_i satisfies (3.13) with $\gamma = 1$. Thus the case of uniform sampling is directly obtained from Theorem 3.5 and the known fact that for Q-linear convergence rate of $1 - x$ with $x \in (0, 1)$, the iteration complexity for obtaining an ϵ-accurate solution is $O(x^{-1}\log(1/\epsilon))$.

For (3.54), we derive a different result from (3.10). We first get that $\|\mathcal{P}^{-1}\mathcal{A}^{-1}H\| = NL_{\text{avg}}$, and by letting $\lambda = 1/2$ and $\theta = \mu/(NL_{\text{avg}})$, (3.10)

leads to

$$\mathbb{E}_{i_k}\left[\alpha_{i_k} Q_{i_k}^* \middle| x^k\right] \leq \frac{\mu}{2NL_{\text{avg}}}\left(F^* - F\left(x^k\right)\right). \tag{3.55}$$

The rest then traces the same argument in the proof of Theorem 3.5 to get a Q-linear convergence rate.

□

When $\eta = 0$, the rates in Corollaries 3.11-3.12 are similar to the result in Nesterov (2012) for the non-regularized case with the same sampling strategies, if we interpret their result in the Euclidean norm. We can clearly see the advantage of the Lipschitz sampling over the uniform sampling. Therefore, our result here can be viewed as an extension of Nesterov's analysis for Lipschitz sampling to the regularized problem (3.1). Note that Nesterov has discussed the case of constrained optimization in Nesterov (2012), which can be treated as a special case of regularized optimization. In the constrained case, Nesterov shows a $O(1/k)$ convergence rate of the objective value when the objective is convex, but the rate depends on $(R_0^2/2 + F(x^0) - F^*)$. Here we provide another improvement to weaken the dependency on the initial objective value by showing the early linear convergence. The case that F satisfies (3.33) can also provide linear convergence for Algorithm 4 to be seen as an improvement of existing results, but the consequent rates do not suggest clear advantages of the Lipschitz sampling, and the derivations are trivial. We therefore omit these rates.

When f is not convex, Algorithm 4 still benefits from Lipschitz sampling.

Corollary 3.13. *Consider Algorithm 4 and assume that (3.2) holds. If (3.5) holds at every iteration for some $\eta \in [0, 1)$, then we have*

$$\min_{0\leq k\leq T} \mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\left\|G_k\right\|^2 \middle| x^0\right] \leq \frac{2(F(x^0) - F^*)}{(1-\eta)(T+1)} \max_{1\leq i\leq N} \frac{L_i}{p_i}.$$

Therefore, when uniform sampling is used, we get

$$\min_{0\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\left\|G_k\right\|^2\middle|x^0\right]\leq\frac{2NL_{\max}(F\left(x^0\right)-F^*)}{(1-\eta)(T+1)},$$

and when Lipschitz sampling is used, we get

$$\min_{0\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\left\|G_k\right\|^2\middle|x^0\right]\leq\frac{2NL_{avg}(F\left(x^0\right)-F^*)}{(1-\eta)(T+1)}.$$

Our result here for the case of uniform sampling is similar to that in Patrascu and Necoara (2015), but we show that Lipschitz sampling can improve the convergence rate by considering a slightly different measure of stationarity.

Now we turn to consider Algorithm 5. This can again be viewed as an extension of the sampling strategy in Nesterov (2012) to regularized problems (3.1).

Corollary 3.14. *Consider Algorithm 5 and assume that (3.2) holds. Assume (3.5) holds for some $\eta\in[0,1)$ for all iterations. Then*

1. *For $p_i=1/N$ we have*

$$\min_{0\leq k\leq T}\mathbb{E}_{i_0,\ldots,i_{k-1}}\left[\left\|G_k\right\|^2\middle|x^0\right]\leq\frac{2NL_{\max}(F\left(x^0\right)-F^*)}{(1-\eta)(T+1)}.$$

2. *If f is convex, for $p_i=1/N$ we have*

   2.1. *When $F(x^k)-F^*\geq(x^k-P_\Omega(x^k))^\top L(x^k-P_\Omega(x^k))/L_{\max}$, the convergence of the expected objective value is Q-linear:*

$$\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right)-F^*\middle|x^k\right]\leq\left(1-\frac{(1-\eta)}{2NL_{\max}}\right)\left(F\left(x^k\right)-F^*\right).$$

2.2. *For all* $k \geq k_0$, *where* $k_0 := \arg\min\{k : F\left(x^k\right) - F^* < R_0^2\}$, *the expected objective follows a sublinear convergence rate*

$$\mathbb{E}_{i_{k_0},\ldots,i_{k-1}}\left[F\left(x^k\right)\middle|\, x^{k_0}\right] - F^* \leq \frac{2NL_{\max}R_0^2}{2N + (1-\eta)(k-k_0)}.$$

3. *If* F *satisfies* (2.10) *for some* $\mu > 0$, *then for* $p_i = 1/N$ *we have*

$$\mathbb{E}_{i_k}\left[F(x^{k+1}) - F^*\middle|\, x^k\right] \leq \left(1 - \frac{(1-\eta)(1+1/\mu)^{-1}}{NL_{\max}}\right)\left(F(x^k) - F^*\right).$$

4. *With* p_i *chosen from* (3.54), *results in the the first and the thrid parts hold, with* $NL_{\max}$ *improved to* NL_{avg}. *The second part becomes the following for convex* f.

4.1. *When* $F(x^k) - F^* \geq (x^k - P_\Omega(x^k))^\top(x^k - P_\Omega(x^k))$, *the convergence of the expected objective value is Q-linear:*

$$\mathbb{E}_{i_k}\left[F\left(x^{k+1}\right) - F^*\middle|\, x^k\right] \leq \left(1 - \frac{(1-\eta)}{2NL_{avg}}\right)\left(F\left(x^k\right) - F^*\right).$$

4.2. *For all* $k \geq k_0$, *where* $k_0 := \arg\min\{k : F\left(x^k\right) - F^* < R_0^2\}$, *the expected objective follows a sublinear convergence rate*

$$\mathbb{E}_{i_{k_0},\ldots,i_{k-1}}\left[F\left(x^k\right)\middle|\, x^{k_0}\right] - F^* \leq \frac{2NL_{avg}R_0^2}{2N + (1-\eta)(k-k_0)}.$$

It is clear that no matter whether (2.10) holds, in terms of convergence rates, the bounds indicate a potential speedup of $L_{\max}/L_{\text{avg}}$ when (3.54) is used.

In comparing the Algorithms 4 and 5, an advantage of Algorithm 4 is when the solution exhibits some partly smooth structure, it may be able to identify the low-dimensional manifold. When it is not the case, we can compare the convergence speeds. We can see that the convergence

speeds for $\|G_k\|$ are identical in both algorithms, and the convergence in the general convex case after k_0 iterations is the same as well, although the definition of k_0 can be different and the early linear convergence conditions and rates also differ slightly. For the global linear convergent cases, the iteration complexities for Algorithm 5 to reach an ϵ-accurate solution in expected value are

$$O\left(\frac{N}{(1-\eta)}\left(L_{\max}+\frac{L_{\max}}{\mu}\right)\log\frac{1}{\epsilon}\right), \text{ and } O\left(\frac{N}{(1-\eta)}\left(L_{\text{avg}}+\frac{L_{\text{avg}}}{\mu}\right)\log\frac{1}{\epsilon}\right),$$

respectively for uniform sampling and Lipschitz sampling. We can see that when $L_{\max}/\mu$ (or L_{avg}/μ) dominates, which is usually the case, those complexities are similar to that in Corollary 3.12. Thus to conclude the comparison, the two algorithms seem to perform similarly in terms of iteration complexity when the problem does not exhibit partly smoothness. We will further confirm this empirically in Section 3.7.

Diagonal Scaling

It is possible to scale x by $L^{-1/2}$ so that in the scaled problem, $L_i \equiv 1, \forall i$, and Lipschitz sampling degenerates to uniform sampling. A pitfall is that for problems satisfying (2.10), the parameter μ changes by a factor of $1/L_{\max}$ under this scaling, so iteration complexity may not improve. Similarly, for general convex problems, R_0^2 as well as $F(x^0)-F^*$ can also be scaled by the same factor, leading to no convergence improvement. These observations suggest that Lipschitz sampling can still be more preferable for randomized BCD. The following result confirms these observations.

Theorem 3.15. *Consider* (3.1) *with* (3.2) *holds. Let* $\tilde{x} := \sqrt{L}x$, *where* L *is defined in* (3.53). *Then* (3.1) *is equivalent to*

$$\min_{\tilde{x}} \quad \tilde{f}(\tilde{x})+\tilde{\psi}(\tilde{x}) := f(\sqrt{L}^{-1}\tilde{x})+\psi(\sqrt{L}^{-1}\tilde{x}), \tag{3.56}$$

and $\nabla \tilde{f}$ is blockwise 1-Lipschitz continuous.

Moreover, updating the ith block of $\tilde{x}$ for (3.56) by solving (3.3) with $H_i = I$ and unit step size is equivalent to updating the ith block of x for (3.1) by solving (3.3) with $H_i = L_i I$ and unit step size.

Proof. The equivalence between (3.1) and (3.56) is straightforward from that $\sqrt{L}^{-1}\tilde{x} = x$. Regarding the blockwise Lipschitz continuity parameters of $\nabla \tilde{f}$, we have

$$\begin{aligned}
&\left\|\nabla_i \tilde{f}\left(\tilde{x} + U_i h\right) - \nabla_i \tilde{f}\left(\tilde{x}\right)\right\| \\
&= \left\|U_i^\top \left(\sqrt{L}^{-1} \nabla f\left(\sqrt{L}^{-1}\left(\tilde{x} + U_i h\right)\right) - \sqrt{L}^{-1} \nabla f\left(\sqrt{L}^{-1}\tilde{x}\right)\right)\right\| \\
&= \sqrt{L_i}^{-1} \left\|\nabla_i f(\sqrt{L}^{-1}\tilde{x} + \sqrt{L_i}^{-1} U_i h) - \nabla_i f(\sqrt{L}^{-1}\tilde{x}))\right\| \\
&\leq \sqrt{L_i}^{-1} L_i \left\|\sqrt{L_i}^{-1} h\right\| = \|h\| .
\end{aligned}$$

Next, for the different update rules, we show that the generated steps are equivalent. For (3.56), given any $\tilde{x}$, and any i, let the corresponding update direction be $\tilde{d}$, and define $d = \sqrt{L_i}^{-1}\tilde{d}$. The corresponding objective defined in (3.4) can then be written as

$$\begin{aligned}
&\nabla_i \tilde{f}(\tilde{x})^\top \tilde{d} + \frac{1}{2}\tilde{d}^\top \tilde{d} + \tilde{\psi}_i(\tilde{x}_i + \tilde{d}) \\
&= \sqrt{L_i}^{-1} \nabla_i f(\sqrt{L}^{-1}\tilde{x})^\top \tilde{d} + \frac{1}{2}\tilde{d}^\top \tilde{d} + \psi_i(\sqrt{L_i}^{-1}(\tilde{x}_i + \tilde{d})) \\
&= \nabla_i f(x)^\top d + \frac{L_i}{2}\|d\|^2 + \psi(x_i + d),
\end{aligned}$$

which is equivalent to using $H_i = L_i I$ in (3.4) for (3.1). Note that d directly corresponds to the update for the original $x = \sqrt{L}^{-1}\tilde{x}$ for (3.1).

□

3.5 Related Works

One of the (serial, deterministic) algorithms considered in Chapter 2 is a special case of Algorithm 3 with only one block ($N = 1$). The technique for measuring inexactness is also borrowed from Chapter 2, but our extension to stochastic BCD and arbitrary sampling probabilities is nontrivial, requiring novel convergence analysis. Moreover, our BCD algorithm is empirically much faster for certain problems.

The case in which (3.3) is solved exactly is discussed in Tseng and Yun (2009). This chapter uses the same boundedness condition for H^k as ours, and the blocks are selected either under a cyclic manner (with an arbitrary order), or a Gauss-Southwell fashion. For the cyclic variant, The convergence rate of the special case in which Q forms an upper bound of the objective improvement is further sharpened by Sun and Hong (2015); Li et al. (2017b). The relaxation to approximate subproblem solutions, with an inexactness criterion different from ours, is analyzed in Chouzenoux et al. (2016). In the latter paper, the coefficients in their convergence rates are unclear, only linear or a certain type of sublinear convergence is shown, but the relation between convergence rates and either the measure of inexactness or the choice H^k is unclear. We note to that the cyclic ordering of blocks is inefficient in certain cases: Sun and Ye (2016) showed that the worst case of cyclic BCD is $O(N^2)$ times slower than the expected rate for randomized BCD.

The Gauss-Southwell variant discussed in Tseng and Yun (2009) can be extended to the inexact case via straightforward modification of the analyses for inexact variable-metric methods in works such as Chapter 2 or Scheinberg and Tang (2016); Ghanbari and Scheinberg (2018); Bonettini et al. (2016), giving results close to what we obtain here with uniform sampling. It might be possible to utilize techniques for single-coordinate descent in Nutini et al. (2015) to obtain better rates by considering a norm other than the Euclidean norm, as was done in Nutini et al. (2017), but this

is beyond the scope of the current work.

The special case of Algorithm 4 we discussed in Section 3.4 has received much attention in the literature. The case for the non-regularized case of $\psi \equiv 0$ in (3.1) was first analyzed in Nesterov (2012) for convex and strongly convex f. That paper uses a quadratic approximation of f that is invariant over iterations, together with a fixed step size. Since it is relatively easy to solve the subproblem to optimality in the non-regularized case, inexactness is not considered. The sampling strategy of using the probability $p_i = L_i^\alpha / \sum_j L_j^\alpha$ for any $\alpha \in [0, 1]$ was analyzed in that work. The two extreme cases of $\alpha = 0$ and $\alpha = 1$ correspond to uniform sampling and (3.54), respectively. The ith block update is $d_i = -\nabla_i f(x)/L_i$, so we obtain from the blockwise Lipschitz continuity of ∇f that

$$\begin{aligned}\mathbb{E}_i \left[f\left(x + U_i d_i\right) - f\left(x\right)\right] &\leq \sum_i p_i f\left(x\right) - \frac{p_i}{2L_i} \left\|\nabla_i f(x)\right\|^2 - f\left(x\right) \\ &\leq -\min_i \frac{p_i}{2L_i} \left\|\nabla f(x)\right\|^2 .\end{aligned}$$

This bound suggests that if we use $p_i = 1/N$, the complexity will be related to $NL_{\max}$, whereas when p_i is proportional to L_i, the complexity is related to the smaller quantity NL_{avg} (consistent with our discussion in Section 3.4). The case that ψ is an indicator function of a convex set is also analyzed in Nesterov (2012), but the analysis is for uniform sampling of the blocks only. This analysis is extended to regularized problems (3.1) in Lu and Xiao (2015) for convex and strongly convex problems but still limited to uniform sampling. The case that f is not necessarily convex in (3.1) is analyzed in Patrascu and Necoara (2015) again under uniform sampling. These results for regularized problems are restricted to Algorithm 4 (with $\eta = 0$) and uniform sampling only, while our analysis covers broader algorithm choices and shows that non-uniform sampling can be used to accelerate the optimization process. The advantage of non-uniform sampling will be further confirmed empirically in Section 3.7.

In Zhao and Zhang (2015), the analysis for Algortihm 4 is extended to the special case of the dual of convex regularized ERM, where each ψ_i is strongly convex, with non-uniform samplings for the blocks. Some primal-dual properties of the convex regularized ERM problems are used in Zhao and Zhang (2015) to derive the optimal probability distribution for the primal suboptimality. However, generalization to other classes of problems is unclear. Our analysis fully extends Nesterov's result to regularized problems without assuming any particular primal-dual relations.

The paper Tappenden et al. (2016) describes inexact extensions of Nesterov (2012) to convex (3.1). This paper uses a different inexactness criterion from ours, and their framework fixes H_i over all iterations, and uses small steps based on L_i rather than a line search. (Thus, their algorithm requires knowledge of the parameters L_i, which is often expensive to calculate when $n_i > 1$.) In the regularized case of $\psi \neq 0$, their algorithm is compatible only with uniform sampling. Fountoulakis and Tappenden (2018) tries to address the limitations of Tappenden et al. (2016) by allowing variable H_i and backtracking line search, but under a different sampling strategy in which a predefined number of blocks is sampled at each iteration from a uniform distribution. Besides this different sampling strategy, the major difference between our algorithm and theirs is that their inexactness condition can be expensive to check except for special cases of ψ (see their Remark 5). In summary, our improvements over Fountoulakis and Tappenden (2018) include (1) a more practical framework that allows general ψ, (2) non-uniform sampling that may lead to significant acceleration when additional information is available, and (3) sharper convergence rates (4) convergence speed coverage for nonconvex f.

3.6 Efficient Implementation for Algorithm 3

The major concern for Algorithm 3 is its practicality. In particular, it is usually considered that linesearch can be as expensive as evaluating the objective or the full gradient of f, and it can also be hard to obtain a meaningful choice of variable metrics that can be updated efficiently. Fortunately, if f is of the form

$$f(x) = g(Ax) \tag{3.57}$$

with a given matrix $A \in \mathbb{R}^{\ell \times n}$ and a function $g : \mathbb{R}^{\ell} \to \mathbb{R}$ that is block-separable and the evaluation of $g(z)$ costs $O(\ell)$, we can implement Algorithm 3 with high efficiency. The problem class (3.57) includes many problems widely seen in real-world applications including the popular regularized ERM problem in machine learning and its Lagrange dual problem. We also discuss the practicality of non-uniform sampling in this section.

Line Search

Since ψ is assumed to be block-separable, when we update just one x_i, objective value evaluation of the regularization term is cheap as we just need to consider the change in $\psi_i(x_i)$. Therefore, we focus on the part of the smooth term.

We notice that for (3.57), we have

$$f\left(x + \alpha_i U_i d_i\right) = g\left(Ax + A\alpha_i U_i d_i\right) = g\left(Ax + \alpha_i \left(AU_i\right) d_i\right).$$

The computation of $\left(AU_i\right) d_i$ costs only $O(n_i \ell)$, so instead of conducting full matrix-vector products that costs $O(N\ell)$ each time, using $\left(AU_i\right) d_i$ to evaluate $f(x + \alpha_i U_i d_i)$ with different values of α_i costs only $O(l)$ each time, which is the same cost as updating Ax after the step size is finalized.

Therefore, the cost of linesearch is negligible as the dominant part will be the update of ∇f and solving the subproblem, just like the variant without linesearch. When A is sparse, the vector $(AU_i)\, d_i$ can be sparse as well and the cost of linesearch might be even lower if the sparsity pattern is properly exploited.

Pick for the Varible Metric

The reason of considering BCD in large-scale problems is its low computational cost per iteration. Therefore, if the cost of updating the quadratic term H is high, the algorithm becomes impractical. However, updating H can be much more expensive than updating x and $\nabla f(x)$ because it contains more variables. One possible choice of H that contains fewer variables but can potentially improve convergence over a fixed choice of H is to consider the diagonal entries of the (generalized) Hessian of f. As there are only n variables and we only need to use n_i of them at each iteration if the ith block is picked, this choice can be used in general.

On the other hand, when the smooth term is of the form (3.57), we can use the whole Hessian with low cost. The key is to observe that

$$\nabla^2 f(x) = A^\top \nabla^2 g(Ax) A, \tag{3.58}$$

and $\nabla^2 g$ is a block-diagonal matrix from the block-separable assumption of g. If we were to compute the whole updated $\nabla^2 f(x)$, even with the structure (3.58), the cost can still be prohibitively high. Fortunately, most of the algorithms for solving (3.3) do not need to explicitly form H_i. Instead, only the matrix-vector product $H_i d_i$ will be needed. In this case, we just need to update the corresponding block in $\nabla^2 g$ at each iteration so the implementation can be highly efficient.

Implementation of Nonuniform Samplings

There are two empirical concerns in implementing non-uniform samplings including the Lipschitz sampling. The first one is the cost of sampling from a non-uniform distribution. A naive implementation can easily cost $O(N)$ per sampling if it is not carefully designed. Fortunately, there are efficient methods such as that proposed in Walker (1977) for non-uniform sampling such that given a fixed distribution, after $O(N)$ cost of initialization, each time sampling a point from this distribution costs the same as sampling two points uniformly randomly. Note that this is also a reason that a fixed probability distribution over iterations is preferred: changing probability distributions over iterations nullify those efficient methods with $O(N)$ overhead, and thus the sampling of the blocks can become the bottleneck especially when the update itself is cheap. For completeness, details our implementation of non-uniform sampling is given in Appendix 3.A.

The second concern is that the cost per iteration is different under different sampling strategies. Especially when the data are sparse, the value of L_i may be positively correlated to the density of the corresponding data point. In this case, sampling according to L_i may increase the cost per iteration significantly. However, if one can estimate $\|H_i\|$, the step sizes, and the cost of updating each coordinate in advance, it is not hard to compare the expected cost increase and the expected convergence improvement to decide if non-uniform sampling should be considered. When the related information is unavailable or hard to get, uniform sampling is still the first choice.

3.7 Computational Results

We report on the empirical performance of our algorithm. We consider two sets of experiments. In the first one, we compare uniform sampling and the Lipschitz sampling for the traditional randomized BCD approaches

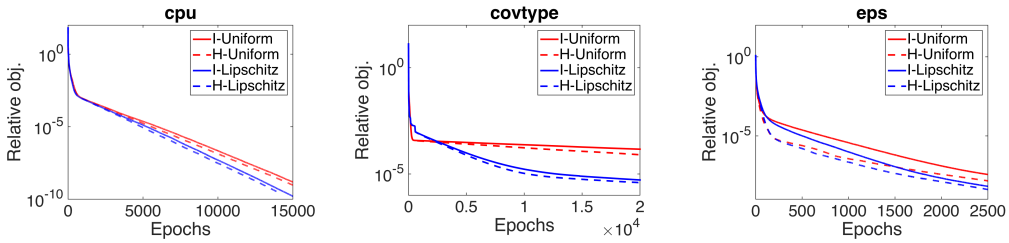


Figure 3.1: Comparison of different sampling strategies using fixed step sizes in terms of epochs. The prefix "H" refers to the choice $H_i = L_i I$, while "I" means $H = I$.

Data set	#instances	n	$L_{\max}/L_{\text{avg}}$
cpusmall_scale	$8,192$	12	1.29
covtype.binary.scale	$581,012$	54	8.58
epsilon_normalized	$400,000$	$2,000$	5.49

Table 3.1: Data sets used in the LASSO problems.

discussed in Section 3.4. We consider both Algorithm 4 and 5. In the second experiment, we show the empirical performance of the proposed variable metric approach with line search, using the corresponding block of the real Hessian as the quadratic term H.

In both experiments, we present the relative objective value difference to the optimum, defined as

$$\frac{F(x) - F^*}{F^*},$$

where F^* is approximately obtained by running our algorithm with a tight termination condition.

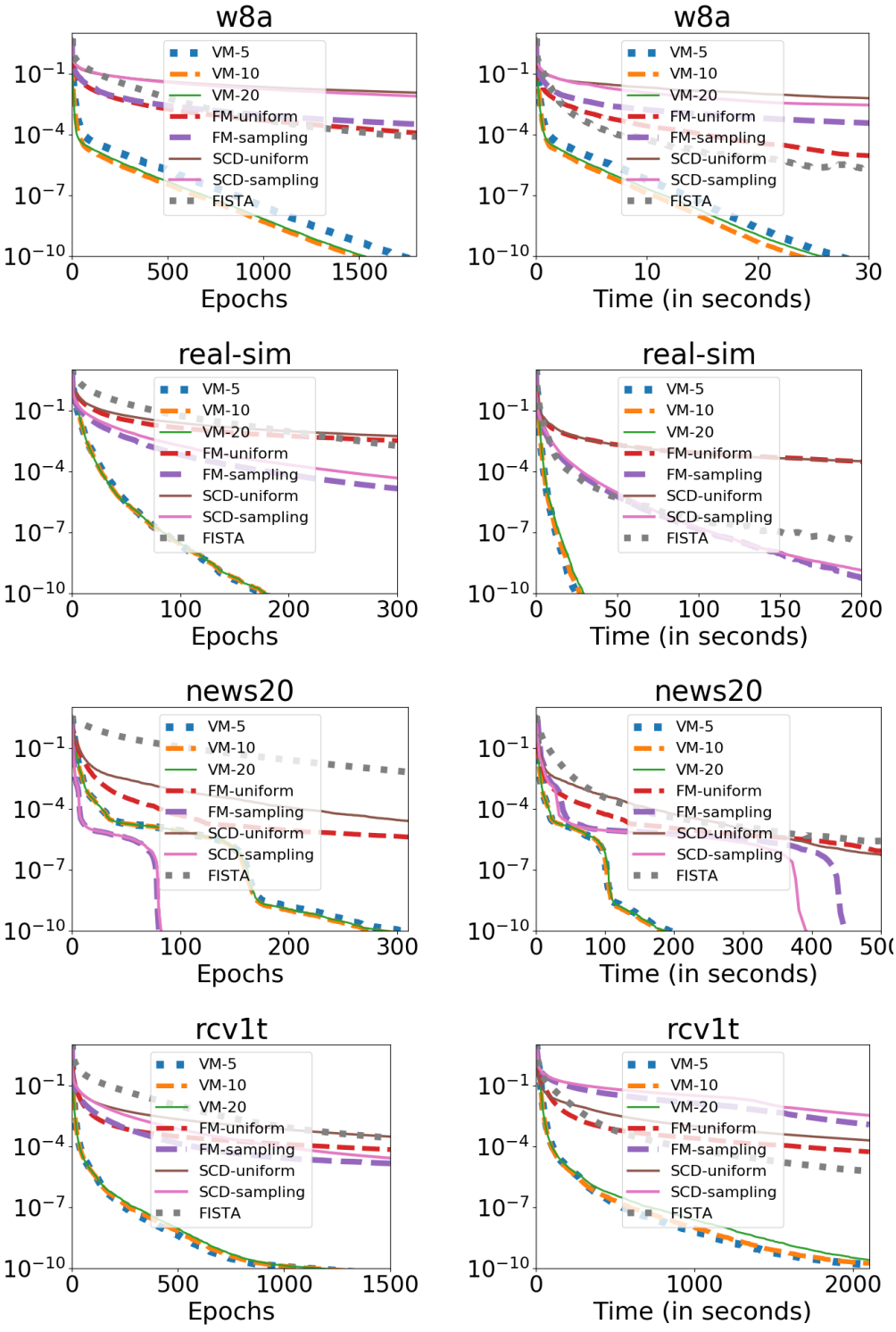


Figure 3.2: Comparison of fixed and variable quadratic terms for solving (3.60) with $C = 1$. Top row: epochs, bottom row: running time.

Comparison of Traditional BCD

We first illustrate the speedup of Lipschitz over uniform sampling using the rather simple LASSO problem (Tibshirani, 1996).

$$\min_{x\in\mathbb{R}^n} \quad \frac{C}{2}\sum_{i=1}^{l}(a_i^\top x - b_i)^2 + \|x\|_1, \tag{3.59}$$

where $(a_i, b_i) \in \mathbb{R}^n \times \mathbb{R}, i = 1, \ldots, l$, are the training data points and $C > 0$ is a parameter to balance the two terms. Note that the corresponding subproblem (3.3) has a closed-form solution when H is a multiple of identity so $\eta = 0$. Note that we do not aim at providing an algorithm for LASSO faster than state of the art, as these methods can significantly accelerate the optimization process by means of active set selection heuristics. Though our basic algorithm can indeed be combined with these techniques, the purpose of this experiment is to compare the two different samplings and the two algorithms, so we avoid a more sophisticated comparison.

To exclude possible speed difference caused by active set identification, we choose C large enough such that the optimal solutions are all nonzero. Statistics of the data sets are listed in Table 3.1. We test both Algorithms 4 and 5, and both uniform and Lipschitz samplings. We present convergence in terms of epochs to compare the iteration complexity difference, where one epoch means that N blocks are processed. The results in Figure 3.1 show a clear advantage for Lipschitz sampling, consistent with our convergence analysis. We also observe that Algorithms 4 and 5 seem to possess similar convergence behavior with the former being slightly faster when active set identification is not included. We can also clearly observe the early linear convergence result described in Theorem 3.3 in all variants on all data sets.

Empirical Performance of the Variable Metric Approach

We show the advantage of using variable quadratic terms in (3.3), in comparison with a fixed term. For this purpose, we consider the group-LASSO regularized squared-hinge loss problem defined by

$$\min_{x\in\mathbb{R}^n} \quad C\sum_{i=1}^{l}\max\left\{1-b_i a_i^\top x, 0\right\}^2 + \sum_{i=1}^{\lceil n/5\rceil}\sqrt{\sum_{j=1}^{\min\{5,n-5(i-1)\}} x_{5(i-1)+j}^2}, \tag{3.60}$$

where $(a_i, b_i) \in \mathbb{R}^n \times \{-1, 1\}, i = 1, \ldots, l$ are the training data points and $C > 0$ is a parameter to balance the two terms. Each set of five consecutive coordinates is grouped into a single block to form the regularizer. We compare the following approaches:

- VM-x: the proposed variable metric approach in Algorithm 3, with H being the real Hessian plus $10^{-10}I$ to ensure (3.18). We use uniform sampling of the blocks and the SpaRSA approach of Wright et al. (2009) to solve the subproblem, with x being the number of SpaRSA iterations each time, with $x \in \{5, 10, 20\}$.
- FM: the fixed metric approach considered in Tappenden et al. (2016). We use a global upper bound of the Hessian as the fixed metric. As H are precomputed, we consider both the sampling in (3.32) and uniform sampling for the blocks.
- RCD: Algorithm 4 with $\eta = 0$. We use both the Lipschitz sampling (3.54) and uniform sampling.
- FISTA Beck and Teboulle (2009): the accelerated proximal gradient approach that does not consider the block-separable nature of the problem.

The FISTA approach is included as a comparison with state of the art for problems without block separability. We do not include the approach

of Fountoulakis and Tappenden (2018) in our comparisons, as their subproblem solver inexactness condition can be expensive to check on this problem. We consider the data sets in Table 3.2, downloaded from the LIBSVM website,[3] with $C = 1$. Results are shown in Figure 3.2. We first observe that different SpaRSA iterations have little impact on the convergence in terms of both epochs and running time, showing that inexactly solving the subproblem is indeed advantageous in the overall efficiency. Next, we see that except for in terms of epochs for news20, the variable metric approach is always significantly faster than state of the art. For news20, the Lipschitz sampling approach for the fixed metric approach and Algorithm 4 are the fastest in terms of epochs, but the real running time of them are much slower than the proposed variable metric approach. The reason is that news20 is a very sparse data set, and the size of the Lipschitz constants are highly correlated to the density of each feature. Thus the computational cost of using Lipschitz sampling is much higher than uniform sampling.

We also observe that for both the fixed metric approach and Algorithm 4, using Lipschitz sampling is always faster than using uniform sampling, in terms of epochs, confirming our analysis. But it is not always the case when it comes to the real running time. We also observe that FISTA performs better in running time than in epochs, mainly because it updates the variables and the gradient less frequently, and its memory access is always sequential and therefore much faster. Finally, we can observe the early linear convergence in the variable metric approach, the fixed metric approach, and Algorithm 4, verifying the result in Theorem 3.3 empirically.

We also notice that the variable metric approach is the only one that requires line search, but it is still the fastest in terms of running time, showing that linesearch does not occupy a significant portion of the running

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Data set	#instances	n
w8a	$49,749$	300
real-sim	$72,309$	$20,958$
news20	$19,996$	$1,355,191$
rcv1_test	$677,399$	$47,236$

Table 3.2: Data sets used in the group-LASSO regularization experiment.

time.

3.8 Conclusions

Starting with a strategy for regularized optimization using regularized quadratic subproblems with variable quadratic terms, we have described a stochastic block-coordinate-descent scheme that is well suited to large scale problems with general structure. We provide detailed iteration complexity analysis, and our framework allows arbitrary sampling schemes. Special case of our theory extends theory for a sampling strategy based on blockwise Lipschitz constants for randomized gradient-coordinate descent from the non-regularized setting to the regularized problem (3.1). Computational experiments show empirical advantages for our approach.

Appendix

3.A Efficient Implementation of Nonuniform Sampling

We describe our implementation of non-uniform sampling. The $O(N)$ initialization step is described in Algorithm 6. After the initialization, each time to sample a point from the given probability distribution, it takes only 2 independent uniform sampling as described in Algorithm 7.

Algorithm 6 Initialization for non-uniform sampling

Given a probability distribution $p_1, \ldots, p_N > 0$;
$i \leftarrow 1$;
Construct $U \leftarrow \{u \mid p_u > 1/N\}$, $L \leftarrow \{l \mid p_l \leq 1/N\}$;
while $L \neq \phi$ **do**
    Pop an element l from L;
    Pop an element u from U;
    upper$_i \leftarrow u$, lower$_i \leftarrow l$, threshold$_i \leftarrow p_l/(1/N)$;
    $p_u \leftarrow p_u - (1/N - p_l)$;
    if $p_u > 1/N$ **then**
        $U \leftarrow U \cup \{u\}$;
    else
        $L \leftarrow L \cup \{u\}$;
    $i \leftarrow i + 1$;

Algorithm 7 Nonuniform sampling after initialization by Algorithm 6

Sample i and j independently and uniformly from $\{1, \ldots, N\}$;
if $j/N \geq$ threshold$_i$ **then**
    Output upper$_i$;
else
    Output lower$_i$;

4 FIRST-ORDER ALGORITHMS CONVERGE FASTER THAN $o(1/k)$ ON CONVEX PROBLEMS

4.1 Introduction

Consider the unconstrained optimization problem

$$\min_x f(x), \tag{4.1}$$

where f has domain in an inner-product space and is convex and L-Lipschitz continuously differentiable for some $L > 0$. We assume throughout that the solution set Ω is non-empty. (Elementary arguments based on the convexity and continuity of f show that Ω is a closed convex set.) Classical convergence theory for gradient descent on this problem indicates a $O(1/k)$ global convergence rate in the function value. Specifically, if

$$x_{k+1} \coloneqq x_k - \alpha_k \nabla f(x_k), \quad k = 0, 1, 2, \ldots, \tag{4.2}$$

and $\alpha_k \equiv \bar{\alpha} \in (0, 1/L]$, we have

$$f(x_k) - f^* \leq \frac{\text{dist}(x_0, \Omega)^2}{2\bar{\alpha}k}, \quad k = 1, 2, \ldots, \tag{4.3}$$

where f^* is the optimal objective value and $\text{dist}(x, \Omega)$ denotes the distance from x to the solution set. The proof of (4.3) relies on showing that

$$k(f(x_k) - f^*) \leq \sum_{T=1}^{k} (f(x_T) - f^*) \leq \frac{1}{2\bar{\alpha}} \text{dist}(x_0, \Omega)^2, \quad k = 1, 2, \ldots, \tag{4.4}$$

where the first inequality utilizes the fact that gradient descent is a descent method (yielding a nonincreasing sequence of function values $\{f(x_k)\}$). We claim that the bound (4.3) is not tight, in the sense that $k(f(x_k) - f^*) \to 0$,

and thus $f(x_k) - f^* = o(1/k)$. This result is a consequence of the following technical lemma.

Lemma 4.1. *Let $\{\Delta_k\}$ be a nonnegative sequence satisfying the following conditions:*

1. *$\{\Delta_k\}$ is monotonically decreasing;*
2. *$\{\Delta_k\}$ is summable, that is, $\sum_{k=0}^{\infty} \Delta_k < \infty$.*

Then $k\Delta_k \to 0$, so that $\Delta_k = o(1/k)$.

Proof. The proof uses simplified elements of the proofs of Lemmas 2 and 9 of Section 2.2.1 from Polyak (1987). Define $s_k := k\Delta_k$ and $u_k := s_k + \sum_{i=k}^{\infty} \Delta_i$. Note that

$$s_{k+1} = (k+1)\Delta_{k+1} \leq k\Delta_k + \Delta_{k+1} \leq s_k + \Delta_k. \tag{4.5}$$

From (4.5) we have

$$u_{k+1} = s_{k+1} + \sum_{i=k+1}^{\infty} \Delta_i \leq s_k + \Delta_k + \sum_{i=k+1}^{\infty} \Delta_i = s_k + \sum_{i=k}^{\infty} \Delta_i = u_k,$$

so that $\{u_k\}$ is a monotonically decreasing nonnegative sequence. Thus there is $u \geq 0$ such that $u_k \to u$, and since $\lim_{k\to\infty} \sum_{i=k}^{\infty} \Delta_i = 0$, we have $s_k \to u$ also.

Assuming for contradiction that $u > 0$, there exists $k_0 > 0$ such that $s_k \geq u/2 > 0$ for all $k \geq k_0$, so that $\Delta_k \geq u/(2k)$ for all $k \geq k_0$. This contradicts the summability of $\{\Delta_k\}$. Therefore we have $u = 0$, so that $k\Delta_k = s_k \to 0$, proving the result.

□

Our claim about the fixed-step gradient descent method follows immediately by setting $\Delta_k = f(x_k) - f^*$ in Lemma 4.1. We state the result formally as follows.

Theorem 4.2. *Consider (4.1) with f convex and L-Lipschitz continuously differentiable and nonempty solution set Ω. If the step sizes satisfy $\alpha_k \equiv \bar{\alpha} \in (0, 1/L]$ for all k, then gradient descent (4.2) generates objective values $f(x_k)$ that converge to f^* at an asymptotic rate of $o(1/k)$.*

This result shows that the $o(1/k)$ rate for gradient descent with a fixed short step size is universal on convex problems, without any additional requirements such as the boundedness of Ω assumed in (Bertsekas, 2016, Proposition 1.3.3). In the remainder of this chapter, we show that this faster rate holds for several other smooth optimization algorithms, including gradient descent with various line-search strategies and stochastic coordinate descent with arbitrary sampling strategies for the coordinates. We then extend the result to algorithms for regularized convex optimization problems, including proximal gradient and stochastic proximal coordinate descent with arbitrary sampling. Assumptions such as bounded solution set, bounded level set, or bounded distance to the solution set, which are commonly assumed in the literature, are all unnecessary.

In our description, the Euclidean norm is used for simplicity, but our results can be extended directly to any norms induced by an inner product,[1] provided that the definition of Lipschitz continuity of ∇f is with respect to the corresponding norm and its dual norm.

Related Work. Our work was inspired by (Peng et al., 2018, Corollary 2) and (Bertsekas, 2016, Proposition 1.3.3), which improve convergence for certain algorithms and problems on convex problems in a Euclidean space from $O(1/k)$ to $o(1/k)$ when the level set is compact. However, this chapter develops improved convergence rates of several algorithms on convex problems without the assumption on the level set, and some of our results apply to non-Euclidean Hilbert spaces. The main proof techniques in this

[1] We meant that given an inner product $< \cdot, \cdot >$, the norm $\| \cdot \|$ is defined as $\|x\| := \sqrt{< x, x >}$.

work are developed independently and different from that in the above works.

For an accelerated version of proximal gradient on convex problems, it is proved in Attouch and Peypouquet (2016) that the convergence rate can be improved from $O(1/k^2)$ to $o(1/k^2)$. Accelerated proximal gradient is a more complicated algorithm than the nonaccelerated versions we discuss, and thus Attouch and Peypouquet (2016) require a more complicated analysis that is quite different from ours.

We note that Deng et al. (2017) have stated a version of Lemma 4.1 with a proof different from the proof that we present, using it to show the convergence rate of the quantity $\|x_k - x_{k+1}\|$ of a version of the alternating-directions method of multipliers (ADMM). Our work differs in the range of algorithms considered and the nature of the convergence. We also provide a discussion of the tightness of the $o(1/k)$ convergence rate.

4.2 Main Results on Unconstrained Smooth Problems

We start by detailing the procedure for obtaining (4.4), to complete the proof of Theorem 4.2. Let us define

$$M(\alpha) := \alpha - \frac{L\alpha^2}{2}. \tag{4.6}$$

From the Lipschitz continuity of ∇f, we have for any point x and any real number α that

$$\begin{aligned} f(x - \alpha\nabla f(x)) &\leq f(x) - \nabla f(x)^\top (\alpha\nabla f(x)) + \frac{L}{2}\|\alpha\nabla f(x)\|^2 \\ &= f(x) - M(\alpha)\|\nabla f(x)\|^2. \end{aligned} \tag{4.7}$$

Clearly,

$$\alpha \in \left(0, \frac{1}{L}\right] \quad \Rightarrow \quad M(\alpha) \geq \frac{\alpha}{2} > 0, \tag{4.8}$$

so in this case, we have by rearranging (4.7) that

$$\begin{aligned} \|\nabla f(x)\|^2 &\leq \frac{1}{M(\alpha)} \left(f(x) - f(x - \alpha \nabla f(x))\right) \\ &\leq \frac{2}{\alpha} \left(f(x) - f(x - \alpha \nabla f(x))\right). \end{aligned} \tag{4.9}$$

Considering any solution $\bar{x} \in \Omega$ and any $T \geq 0$, we have for gradient descent (4.2) that

$$\begin{aligned} \|x_{T+1} - \bar{x}\|^2 &= \|x_T - \alpha_T \nabla f(x_T) - \bar{x}\|^2 \\ &= \|x_T - \bar{x}\|^2 + \alpha_T^2 \|\nabla f(x_T)\|^2 - 2\alpha_T \nabla f(x_T)^\top (x_T - \bar{x}). \end{aligned} \tag{4.10}$$

Since $\alpha_T \in (0, 1/L]$ in (4.10), we have from (4.9) and the convexity of f (which implies $\nabla f(x_T)^T(\bar{x} - x_T) \leq f^* - f(x_T)$), we obtain

$$\|x_{T+1} - \bar{x}\|^2 \leq \|x_T - \bar{x}\|^2 + 2\alpha_T \left(f\left(x_T\right) - f\left(x_{T+1}\right)\right) + 2\alpha_T \left(f^* - f\left(x_T\right)\right). \tag{4.11}$$

By rearranging (4.11) and using $\alpha_T \equiv \bar{\alpha} \in (0, 1/L]$, we obtain

$$f\left(x_{T+1}\right) - f^* \leq \frac{1}{2\bar{\alpha}} \left(\|x_T - \bar{x}\|^2 - \|x_{T+1} - \bar{x}\|^2\right). \tag{4.12}$$

We then obtain (4.4) by summing (4.12) from $T = 0$ to $T = k$ and noticing that $\bar{x}$ is arbitrary in Ω.

The argument above and Theorem 4.2 apply to arbitrary inner-product spaces. So, in particular, the $o(1/k)$ convergence result holds in Hilbert spaces. On the other hand, Theorem 4.2 applies to step sizes in the range $(0, 1/L]$ only, but it is known that gradient descent converges at the rate of $O(1/k)$ for both the fixed step size scheme with $\bar{\alpha} \in (0, 2/L)$ and line-search

schemes in (finite-dimensional) Euclidean spaces. We first show $o(1/k)$ rates for these variants, and then extend the result to stochastic coordinate descent with arbitrary sampling of coordinates, also in Euclidean spaces.

Gradient Descent with Line Search

In this section and the next, we consider the domain of f to be a Euclidean space. We consider two strategies for deciding α_k in (4.2). The first is again a fixed step size scheme

$$\alpha_k \equiv \bar{\alpha} \in \left(0, \frac{2}{L}\right). \tag{4.13}$$

The second one is a general line-search scheme that finds α_k satisfying

$$\alpha_k \in [C_2, C_1], \quad C_2 \in \left(0, \frac{2\,(1-\gamma)}{L}\right), \quad C_1 \geq C_2, \tag{4.14a}$$

$$f\left(x_k - \alpha_k \nabla f(x_k)\right) \leq f\left(x_k\right) - \gamma \alpha_k \|\nabla f(x_k)\|^2, \quad \gamma \in (0,1). \tag{4.14b}$$

From (4.7), the upper bound of C_2 ensures the existence of an α_k that satisfies conditions (4.14). The main result for this subsection is as follows.

Theorem 4.3. *Consider* (4.1) *with f convex and L-Lipschitz continuously differentiable and nonempty solution set Ω. Assume that the domain of f is the Euclidean space $\Re^n$. If the step sizes α_k are decided by either* (4.13) *or* (4.14)*, then gradient descent* (4.2) *generates objective values $f(x_k)$ converging to f^* at an asymptotic rate of $o(1/k)$.*

We will give a brief overview of $O(1/k)$ rates derived in the literature for step sizes chosen by (4.13) or (4.14), then improve on these rates to obtain the desired $o(1/k)$ rate. We first show that for both strategies, the iterates lie in a bounded set. Some parts of the results below are from (Nesterov, 2004, Section 2.1.5).

Before proving Theorem 4.3, we prove two technical lemmas.

Lemma 4.4. *Consider algorithm (4.2) with any initial point x_0, and assume that f is L-Lipschitz-continuously differentiable for some $L > 0$. Then when the sequence of steplengths α_k is chosen to satisfy either (4.13) or (4.14), all iterates x_k lie in a bounded set.*

Proof. Consider any solution $\bar{x} \in \Omega$. By convexity of f, and using the optimality condition $\nabla f(\bar{x}) = 0$, we have for any $T \geq 0$ that

$$\begin{aligned}\|x_{T+1} - \bar{x}\|^2 &= \|x_T - \bar{x}\|^2 + \alpha_T^2 \|\nabla f(x_k)\|^2 - 2\alpha_T \left(x_T - \bar{x}\right)^\top \left(\nabla f(x_T) - \nabla f(\bar{x})\right) \\ &\leq \|x_T - \bar{x}\|^2 + \alpha_T^2 \|\nabla f(x_T)\|^2.\end{aligned}$$

For both (4.13) and (4.14), there exists a constant $C > 0$ such that $\alpha_k^2 \leq C$ for all k. By summing the bound above for $T = 0, 1, \ldots, k-1$, and telescoping, we obtain

$$\|x_k - \bar{x}\|^2 - \|x_0 - \bar{x}\|^2 \leq C \sum_{T=0}^{k-1} \|\nabla f(x_T)\|^2 \leq C \sum_{T=0}^{\infty} \|\nabla f(x_T)\|^2. \tag{4.15}$$

For (4.13), note that $M(\bar{\alpha}) > 0$, so from (4.7) we obtain

$$\infty > f(x_0) - f^* \geq f(x_0) - \lim_{k \to \infty} f(x_k) \geq M\left(\bar{\alpha}\right) \sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2,$$

which implies that

$$\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - f^*}{M\left(\bar{\alpha}\right)} < \infty. \tag{4.16}$$

Similarly, for (4.14), we can sum (4.14b) from $k = 0, 1, 2, \ldots$ to obtain

$$\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - f^*}{C_2 \gamma} < \infty. \tag{4.17}$$

By combining (4.16) and (4.17) with (4.15), we obtain

$$\|x_k\| \leq \|\bar{x}\| + \sqrt{\|x_0 - \bar{x}\|^2 + C \sum_{T=0}^{\infty} \|\nabla f(x_T)\|^2} < \infty, \quad k = 0, 1, 2, \ldots,$$

proving that $\{x_k\}$ are in a bounded set.

□

When f has domain in a Euclidean space, from the Bolzano-Weierstrass theorem, a bounded and closed set is compact. In this case, Lemma 4.4 then implies that the sequence $\{x_k\}$ is in a compact set, thus there must be at least an accumulation point.

Denote $\delta_T := f(x_T) - f^*$ and let $\bar{x}_T$ be the projection of x_T onto Ω (which is well defined because Ω is nonempty, closed, and convex). We can utilize convexity to obtain

$$\delta_T \leq \nabla f(x_T)^\top (x_T - \bar{x}_T) \leq \|\nabla f(x_T)\| \text{dist}\,(x_T, \Omega)\,,$$

so that

$$\|\nabla f(x_T)\| \geq \frac{\delta_T}{\text{dist}(x_T, \Omega)}. \tag{4.18}$$

For the fixed step length (4.13), we have from (4.7) that

$$f(x_{T+1}) \leq f(x_T) - M(\bar{\alpha})\|\nabla f(x_T)\|^2,$$

with $M(\bar{\alpha}) > 0$ by definition of $\bar{\alpha}$. By subtracting f^* from both sides of this expression, and using (4.18), we obtain

$$\delta_{T+1} \leq \delta_T - M(\bar{\alpha}) \frac{\delta_T^2}{\text{dist}\,(x_T, \Omega)^2}.$$

By dividing both sides of this expression by $\delta_T \delta_{T+1}$, and using $\delta_{T+1} \leq \delta_T$,

we obtain

$$\frac{1}{\delta_{T+1}} \geq \frac{1}{\delta_T} + \frac{M\left(\bar{\alpha}\right)\delta_T}{\text{dist}\left(x_T,\Omega\right)^2\delta_{T+1}} \geq \frac{1}{\delta_T} + \frac{M\left(\bar{\alpha}\right)}{\text{dist}\left(x_T,\Omega\right)^2}. \tag{4.19}$$

Summing (4.19) over $T = 0, 1, \ldots, k$, we obtain

$$\frac{1}{\delta_{k+1}} \geq \frac{1}{\delta_0} + \sum_{T=0}^{k} \frac{M(\bar{\alpha})}{\text{dist}\left(x_T,\Omega\right)^2} \;\Rightarrow\; \delta_{k+1} \leq \frac{1}{\frac{1}{\delta_0} + \sum_{T=0}^{k} \frac{M(\bar{\alpha})}{\text{dist}(x_T,\Omega)^2}}. \tag{4.20}$$

A $O(1/k)$ rate is obtained by noting from Lemma 4.4 that $\text{dist}(x_T,\Omega) \leq R_0$ for some $R_0 > 0$ and all T, so that

$$\sum_{T=0}^{k} \frac{1}{\text{dist}\left(x_T,\Omega\right)^2} \geq \frac{k+1}{R_0^2}. \tag{4.21}$$

A similar rate is obtained for (4.14) by replacing $M(\bar{\alpha})$ above with γC_2.

We will show that the bound (4.21) is loose and an improved result can be obtained by working directly with (4.20). The key is to observe that $\text{dist}(x_k,\Omega)$ converges to zero asymptotically and to use the arithmetic-mean / harmonic-mean inequality. Convergence of $\text{dist}(x_k,\Omega)$ is shown in the following lemma, whose proof follows from a similar result in (Peng et al., 2018, Proposition 1).

Lemma 4.5. *When the method* (4.2) *is applied to a convex and L-Lipschitz-continuously differentiable function* $f : \Re^n \to \Re$*, with step sizes satisfying either* (4.13) *or* (4.14)*, then*

$$\lim_{k\to\infty} \textit{dist}\left(x_k,\Omega\right) = 0. \tag{4.22}$$

Proof. We prove the result for (4.13); the reasoning for (4.14) is nearly identical. Assume for contradiction that (4.22) does not hold. Then there

are $\epsilon > 0$ and an infinite increasing sequence $\{k_i\}$, $i = 1, 2, \ldots$, such that

$$\text{dist}\,(x_{k_i}, \Omega) \geq \epsilon, \quad i = 1, 2, \ldots . \tag{4.23}$$

From Lemma 4.4 and that $\{x_{k_i}\} \subset \Re^n$, we can use the Bolzano-Weierstrass theorem to deduce that the sequence $\{x_{k_i}\}$ lies in a compact set and therefore has an accumulation point x^*. From (4.19), we have $1/\delta_{k_{i+1}} \geq 1/\delta_{k_i} + M(\bar{\alpha})/\epsilon^2$, so since $\{1/\delta_k\}$ is an increasing sequence, we have $1/\delta_k \uparrow \infty$ and hence $\delta_k \downarrow 0$. By continuity, it follows that $f(x^*) = f^*$, so that $x^* \in \Omega$ by definition, contradicting (4.23).

□

We note that a result similar to Lemma 4.5 has been given in Burachik et al. (1995) using a more complicated argument with more restricted choices of α.

We are ready to prove Theorem 4.3.

Theorem 4.3. We start from (4.20) and show that

$$\lim_{k\to\infty} \frac{\frac{1}{\frac{1}{\delta_0} + M(\bar{\alpha}) \sum_{T=0}^{k} \frac{1}{\text{dist}_{(x_T,\Omega)^2}}}}{\frac{1}{k+1}} = 0,$$

which is implied by

$$\lim_{k\to\infty} \frac{k+1}{\sum_{T=0}^{k} \frac{1}{\text{dist}_{(x_T,\Omega)^2}}} = 0. \tag{4.24}$$

From the arithmetic-mean / harmonic-mean inequality,[2] we have that

$$0 \leq \frac{k+1}{\sum_{T=0}^{k} \frac{1}{\text{dist}(x_T,\Omega)^2}} \leq \frac{\sum_{T=0}^{k} \text{dist}(x_T,\Omega)^2}{k+1}. \tag{4.25}$$

Lemma 4.5 shows that $\text{dist}(x_T, \Omega) \to 0$, so by the Stolz-Cesàro theorem (see, for example, Mureşan (2009)), the right-hand side of (4.25) converges to 0. Therefore, from the sandwich lemma, (4.24) holds. □

Coordinate Descent

We now extend Theorem 4.2 to the case of randomized coordinate descent. Our results can extend immediately to block-coordinate descent with fixed blocks.

The standard short-step coordinate descent procedure requires knowledge of coordinate-wise Lipschitz constants. Denoting by e_i the ith unit vector, we denote by $L_i \geq 0$ the constants such that:

$$|\nabla_i f(x) - \nabla_i f(x + he_i)| \leq L_i |h|, \quad \text{for all } x \in \Re^n \text{ and all } h \in \Re, \tag{4.26}$$

where $\nabla_i f(\cdot)$ denotes the ith coordinate of the gradient. Note that if $\nabla f(x)$ is L-Lipschitz continuous, there always exist $L_1, \ldots, L_n \in [0, L]$ such that (4.26) holds. Without loss of generality, we assume $L_i > 0$ for all i. Given parameters $\{\bar{L}_i\}_{i=1}^n$ such that $\bar{L}_i \geq L_i$ for all i, the coordinate descent update is

$$x_{k+1} \leftarrow x_k - \frac{\nabla_{i_k} f(x_k)}{\bar{L}_{i_k}} e_i, \tag{4.27}$$

[2] This inequality says that for any real numbers $a_1, \ldots, a_n > 0$, their harmonic mean does not exceed their arithmetic mean. Namely,

$$\frac{n}{\sum_{i=1}^{n} a_i^{-1}} \leq \frac{\sum_{i=1}^{n} a_i}{n}.$$

where i_k is the coordinate selected for updating at the kth iteration. We consider the general case that each i_k is independently identically distributed following a fixed prespecified probability distribution $p_1, \ldots, p_n$ satisfying

$$p_i \geq p_{\min}, \quad i = 1, 2, \ldots, n; \quad \sum_{i=1}^{n} p_i = 1, \tag{4.28}$$

for some constant $p_{\min} > 0$. Nesterov Nesterov (2012) proves that stochastic coordinate descent has a $O(1/k)$ convergence rate (in expectation of f) on convex problems. We show below that this rate can be improved to $o(1/k)$.

Theorem 4.6. *Consider* (4.1) *with f convex and nonempty solution set Ω, and that componentwise-Lipschitz continuous differentiability* (4.26) *holds with some $L_1, \ldots, L_n > 0$. If we apply coordinate descent* (4.27) *and at each iteration, i_k is independently picked at random following a probability distribution satisfying* (4.28)*, then the expected objective $\mathbb{E}_{i_0, i_1, \ldots, i_k}[f(x_k)]$ converges to f^* at an asymptotic rate of $o(1/k)$.*

Proof. From (4.26) and that $\bar{L}_i \geq L_i$, by treating all other coordinates as non-variables, we have that for any $T \geq 0$,

$$f\left(x_T - \frac{\nabla_i f\left(x_T\right)}{\bar{L}_i} e_i\right) - f\left(x_T\right) \leq -\frac{1}{2\bar{L}_i} \|\nabla_i f\left(x_T\right)\|^2, \tag{4.29}$$

showing that the algorithm decreases f at each iteration. Consider any $\bar{x} \in \Omega$, by defining

$$r_T^2 \coloneqq \sum_{i=1}^{n} \frac{\bar{L}_i}{p_i} \|(x_T - \bar{x})_i\|^2, \tag{4.30}$$

we have from (4.27) that

$$r_{T+1}^2 = r_T^2 + \frac{1}{\bar{L}_{i_T} p_{i_T}} \left\|\nabla_{i_T} f\left(x_T\right)\right\|^2 - \frac{2}{p_{i_T}} \nabla_{i_T} f\left(x_T\right)^\top \left(x_T - \bar{x}\right)_{i_T}. \tag{4.31}$$

Taking expectation over i_T on both sides of (4.31), we obtain from the

convexity of f and (4.29) that

$$\begin{aligned}
&\mathbb{E}_{i_T}\left[r_{T+1}^2\right] - r_T^2 \\
&\overset{(4.29)}{\leq} \frac{1}{p_{\min}} \sum_{i=1}^{n} 2p_i \left(f\left(x_T\right) - f\left(x_T - \frac{\nabla_i f\left(x_T\right)}{\bar{L}_i} e_i\right)\right) - 2\nabla f\left(x_T\right)^\top \left(x_T - \bar{x}\right) \\
&\leq \frac{2}{p_{\min}} \left(f\left(x_T\right) - \mathbb{E}_{i_T}\left[f\left(x_{T+1}\right)\right]\right) + 2\left(f^* - f\left(x_T\right)\right). \qquad (4.32)
\end{aligned}$$

By taking expectation over $i_0, i_1, \ldots, i_k$ on (4.32), abbreviating $\mathbb{E}_{i_0,\ldots,i_k}$ as $\mathbb{E}$, and summing (4.32) over $T = 0, 1, \ldots, k$, we obtain

$$\begin{aligned}
2\sum_{T=0}^{k} \left(\mathbb{E} f(x_T) - f^*\right) &\leq r_0^2 - \mathbb{E} r_{k+1}^2 + \frac{2\left(f\left(x_0\right) - \mathbb{E} f\left(x_{k+1}\right)\right)}{p_{\min}} \\
&\leq r_0^2 + \frac{2\left(f\left(x_0\right) - f^*\right)}{p_{\min}}.
\end{aligned}$$

The result now follows from Lemma 4.1.

□

4.3 Regularized Problems

We turn now to regularized optimization:

$$\min_x \; F(x) \coloneqq f(x) + \psi(x), \qquad (4.33)$$

where both terms are convex, f is L-Lipschitz-continuously differentiable, and ψ is extended-valued, proper, and closed, but possibly nondifferentiable. We also assume that ψ is such that the prox-operator can be applied easily, by solving the following problem for given $y \in \Re^n$ and $\lambda > 0$:

$$\min_x \; \psi\left(x\right) + \frac{1}{2\lambda}\|x - y\|^2.$$

We assume further that the solution set Ω of (4.33) is nonempty, and denote by F^* the value of F for all $x \in \Omega$. We discuss two algorithms to show how our techniques can be extended to regularized problems. They are proximal gradient (both with and without line search) and stochastic proximal coordinate descent with arbitrary sampling.

Short-Step Proximal Gradient

Given $\bar{L} \geq L$, the kth step of the proximal gradient algorithm is defined as follows:

$$x_{k+1} \leftarrow x_k + d_k, \quad d_k := \arg\min_d \nabla f(x_k)^\top d + \frac{\bar{L}}{2}\|d\|^2 + \psi\left(x_k + d\right). \quad (4.34)$$

Note that d_k is uniquely defined here, since the subproblem is strongly convex. It is shown in Beck and Teboulle (2009); Nesterov (2013) that $F(x_k)$ converges to F^* at a rate of $O(1/k)$ for this algorithm, under our assumptions. We prove that a $o(1/k)$ rate can be attained.

Theorem 4.7. *Consider (4.33) with f convex and L-Lipschitz continuously differentiable, ψ convex, and nonempty solution set Ω. Given any $\bar{L} \geq L$, the proximal gradient method (4.34) generates iterates whose objective value converges to F^* at a $o(1/k)$ rate.*

Proof. The method (4.34) can be shown to be a descent method from the Lipschitz continuity of ∇f and the fact that $\bar{L} \geq L$. From the optimality of the solution to (4.34) and that $x_{k+1} = x_k + d_k$,

$$-\left(\nabla f(x_k) + \bar{L} d_k\right) \in \partial\psi\left(x_{k+1}\right), \quad (4.35)$$

where $\partial\psi$ denotes the subdifferential of ψ. Consider any $\bar{x} \in \Omega$. We have

from (4.34) that for any $T \geq 0$, the following chain of relationships holds:

$$\begin{aligned}
&\|x_{T+1} - \bar{x}\|^2 - \|x_T - \bar{x}\|^2 \\
&= 2d_T^\top (x_T - \bar{x}) + \|d_T\|^2 \\
&= 2d_T^\top (x_T + d_T - \bar{x}) - \|d_T\|^2 \\
&= 2\left(d_T + \frac{\nabla f(x_T)}{\bar{L}}\right)^\top (x_{T+1} - \bar{x}) - \frac{2}{\bar{L}} \nabla f(x_T)^\top (x_T + d_T - \bar{x}) - \|d_T\|^2 \\
&\overset{(4.35)}{\leq} 2\frac{\psi(\bar{x}) - \psi(x_{T+1})}{\bar{L}} - \frac{2}{\bar{L}} \nabla f(x_T)^\top (x_T - \bar{x}) - \frac{2}{\bar{L}} \nabla f(x_T)^\top d_T - \|d_T\|^2 \\
&\leq \frac{2}{\bar{L}} \left((\psi(\bar{x}) - \psi(x_{T+1})) + f(\bar{x}) - \left(f(x_T) + \nabla f(x_T)^\top d_T + \frac{\bar{L}\|d_T\|^2}{2} \right) \right) \\
&\leq \frac{2\left(F^* - F(x_{T+1})\right)}{\bar{L}}, \qquad (4.36)
\end{aligned}$$

where in the last inequality, we have used

$$f(x+d) \leq f(x) + \nabla f(x)^\top d + \frac{L}{2}\|d\|^2 \leq f(x) + \nabla f(x)^\top d + \frac{\bar{L}}{2}\|d\|^2. \quad (4.37)$$

By rearranging (4.36) we obtain

$$F(x_{T+1}) - F^* \leq \frac{\bar{L}}{2}\left(\|x_T - \bar{x}\|^2 - \|x_{T+1} - \bar{x}\|^2\right).$$

The result follows by summing both sides of this expression over $T = 0, 1, \ldots, k$ and applying Lemma 4.1. □

Proximal Gradient with Line Search

We discuss a line-search variant of proximal gradient, where the update is defined as follows:

$$x_{k+1} \leftarrow x_k + d_k, \quad d_k := \arg\min_d \nabla f(x_k)^\top d + \frac{1}{2\alpha_k}\|d\|^2 + \psi(x_k + d), \quad (4.38)$$

where α_k is chosen such that for given $\alpha_{\max} > \alpha_{\min} > 0$ and $\gamma \in (0, 1]$, we have

$$\alpha_k \in [\alpha_{\min}, \alpha_{\max}], \quad F(x_k + d_k) \leq F(x_k) - \frac{\gamma}{2\alpha_k}\|d_k\|^2. \tag{4.39}$$

This framework includes the SpaRSA algorithm Wright et al. (2009), which obtains an initial choice of α_k from a Barzilai-Borwein approach and adjusts it until (4.39) holds. The approach of the previous subsection can also be seen as a special case of (4.38)-(4.39) through the following lemma, whose proof is in the appendix.

Lemma 4.8. *Consider a convex function ψ, a positive scalar $a > 0$ and two vectors b and x. If d is the unique solution of the following problem:*

$$\min_d \; b^\top d + \frac{a}{2}\|d\|^2 + \psi(x + d), \tag{4.40}$$

then

$$b^\top d + \frac{a}{2}\|d\|^2 + \psi(x + d) - \psi(x) \leq -\frac{a}{2}\|d\|^2.$$

By setting $b = \nabla f(x)$, $1/\alpha_k \equiv a = \bar{L} > 0$ (where $\bar{L} \geq L$), this lemma together with (4.37) implies that (4.39) holds for any $\gamma \in (0, 1]$. Therefore, provided that $\alpha_{\min} \leq 1/L$, we can always find α_k such that (4.39) holds.

We show now that this approach also has a $o(1/k)$ convergence rate on convex problems.

Theorem 4.9. *Consider (4.33) with f convex and L-Lipschitz continuously differentiable, ψ convex, and nonempty solution set Ω. Given $\alpha_{\min}$ and $\alpha_{\max}$ such that $\alpha_{\max} > \alpha_{\min} > 0$ and $\alpha_{\min} \leq 1/L$, and given some $\gamma \in (0, 1]$, then the algorithm (4.38) with α_k satisfying (4.39) generates iterates $\{x_k\}$ whose objective values converge to F^* at a rate of $o(1/k)$.*

Proof. From optimality conditions in (4.38), we have

$$-\left(\nabla f(x_T) + \frac{1}{\alpha_T} d_T\right) \in \partial\psi\left(x_{T+1}\right). \tag{4.41}$$

Now consider any $\bar{x} \in \Omega$. We have from (4.38) that for any $T \geq 0$, the following chain of relationships holds:

$$\begin{aligned}
&\|x_{T+1} - \bar{x}\|^2 - \|x_T - \bar{x}\|^2 \\
&= 2d_T^\top \left(x_T + d_T - \bar{x}\right) - \|d_T\|^2 \\
&= 2\left(d_T + \alpha_T \nabla f(x_T)\right)^\top \left(x_{T+1} - \bar{x}\right) - 2\alpha_T \nabla f(x_T)^\top \left(x_T + d_T - \bar{x}\right) - \|d_T\|^2 \\
&\overset{(4.41)}{\leq} 2\alpha_T \left(\psi\left(\bar{x}\right) - \psi\left(x_{T+1}\right)\right) - 2\alpha_T \nabla f(x_T)^\top \left(x_T - \bar{x}\right) - 2\alpha_T \nabla f(x_T)^\top d_T - \|d_T\|^2 \\
&\leq 2\alpha_T \left(\psi\left(\bar{x}\right) - \psi\left(x_{T+1}\right)\right) - 2\alpha_T \nabla f(x_T)^\top \left(x_T - \bar{x}\right) - 2\alpha_T \nabla f(x_T)^\top d_T \\
&= 2\alpha_T \left(\psi\left(\bar{x}\right) - \psi\left(x_{T+1}\right)\right) - 2\alpha_T \nabla f(x_T)^\top \left(x_T - \bar{x}\right) - 2\alpha_T \nabla f(x_T)^\top d_T \\
&\qquad + \alpha_T L \|d_T\|^2 - \alpha_T L \|d_T\|^2 \\
&\leq 2\alpha_T \left(\psi\left(\bar{x}\right) - \psi\left(x_{T+1}\right) + f\left(\bar{x}\right) - \left(f\left(x_T\right) + \nabla f(x_T)^\top d_T + \frac{L}{2}\|d_T\|^2\right)\right) \\
&\qquad + \alpha_T L \|d_T\|^2 \\
&\overset{(4.39)}{\leq} 2\alpha_T \left(F^* - F\left(x_{T+1}\right)\right) + \frac{2L\alpha_T^2}{\gamma}\left(F(x_T) - F(x_{T+1})\right) \\
&\leq 2\alpha_{\min} \left(F^* - F\left(x_{T+1}\right)\right) + \frac{2L\alpha_{\max}^2}{\gamma}\left(F\left(x_T\right) - F\left(x_{T+1}\right)\right).
\end{aligned} \tag{4.42}$$

The result then follows from summing (4.42) from $T = 0$ to $T = k$ and applying Lemma 4.1. □

Theorems 4.7-4.9, like Theorem 4.2, are applicable to arbitrary inner-product spaces, while other results in this work require Euclidean spaces.

Proximal Coordinate Descent

We now discuss the extension of coordinate descent to (4.33), with the assumption (4.26) on f, sampling weighted according to (4.28) as in Section 4.2, and the additional assumption of separability of the regularizer ψ, that is,

$$\psi(x) = \sum_{i=1}^{n} \psi_i(x_i), \tag{4.43}$$

where each ψ_i is convex, extended valued, and possibly nondifferentiable. As in our discussion of Section 4.2, the results in this subsection can be extended directly to the case of block-coordinate descent.

Given the component-wise Lipschitz constants $L_1, L_2, \ldots, L_n$ and algorithmic parameters $\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_n$ with $\bar{L}_i \geq L_i$ for all i, proximal coordinate descent updates have the form

$$x_{k+1} \leftarrow x_k + d^k_{i_k} e_{i_k}, \tag{4.44a}$$

$$d^k_{i_k} := \arg\min_{d \in \Re} \nabla_{i_k} f(x_k) d + \frac{\bar{L}_{i_k}}{2} d^2 + \psi_{i_k}\left((x_k)_{i_k} + d\right). \tag{4.44b}$$

For (4.44) with $p_i \equiv 1/n$ for all i, Lu and Xiao (2015) showed that the expected objective value converges to F^* at a $O(1/k)$ rate. When arbitrary sampling (4.28) is considered, (4.44) is the same as Algorithm 4. For this algorithm, we have shown in Chapter 3 the same $O(1/k)$ rate for convex problems under the additional assumption that for any x_0,

$$\max_{x: F(x) \leq F(x_0)} \text{dist}\,(x, \Omega) < \infty. \tag{4.45}$$

We show here that with arbitrary sampling according to (4.28), (4.44) produces $o(1/k)$ convergence rates for the expected objective on convex problems, without the assumption (4.45).

Theorem 4.10. *Consider* (4.33) *with* f *and* ψ *convex and nonempty solution set* Ω. *Assume further than* ψ *is separable according to* (4.43) *is true, and that*

(4.26) *holds with some* $L_1, L_2, \ldots, L_n > 0$. *Given* $\{\bar{L}_i\}_{i=1}^n$ *with* $\bar{L}_i \geq L_i$ *for all* i, *suppose that proximal coordinate descent defines iterates according to* (4.44), *with* i_k *chosen i.i.d. according to a probability distribution satisfying* (4.28). *Then* $\mathbb{E}_{i_0,i_1,\ldots,i_k}[F(x_k)]$ *converges to* F^* *at an asymptotic rate of* $o(1/k)$.

Proof. From (4.26), we first notice that in the update (4.44),

$$F\left(x_k + d_{i_k}^k e_{i_k}\right) - F\left(x_k\right) \leq \nabla_{i_k} f(x_k) d_{i_k}^k + \frac{\bar{L}_{i_k}}{2}\left(d_{i_k}^k\right)^2 + \psi_{i_k}\left((x_k)_{i_k} + d_{i_k}^k\right) - \psi_{i_k}\left((x_k)_{i_k}\right). \quad (4.46)$$

From Lemma 4.8, the method defined by (4.44) is a descent method. The optimality condition of (4.44b) is

$$-\left(\nabla_{i_T} f\left(x_T\right) + \bar{L}_{i_T} d_{i_T}^T\right) \in \partial\psi_{i_T}\left(\left(x_T\right)_{i_T} + d_{i_T}^T\right). \quad (4.47)$$

Taking any $\bar{x} \in \Omega$, and using the definition (4.30), we have the following:

$$\begin{aligned}
& r_{T+1}^2 \\
&= r_T^2 + \frac{2\bar{L}_{i_T}}{p_{i_T}}\left(d_{i_T}^\top\right)^\top\left(x_T + d_{i_T}^T - \bar{x}\right)_{i_T} - \frac{\bar{L}_{i_T}}{p_{i_T}}\left(d_{i_T}^T\right)^2 \\
&= r_T^2 + \frac{2}{p_{i_T}}\left(\nabla_{i_T} f\left(x_T\right) + \bar{L}_{i_T} d_{i_T}^T\right)^\top\left(x_T + d_{i_T}^T - \bar{x}\right)_{i_T} - \frac{\bar{L}_{i_T}}{p_{i_T}}\left(d_{i_T}^T\right)^2 \\
&\quad - \frac{2}{p_{i_T}}\nabla_{i_T} f\left(x_T\right)^\top\left(x_T - \bar{x}\right)_{i_T} - \frac{2}{p_{i_T}}\nabla_{i_T} f\left(x_T\right)^\top d_{i_T}^T \\
&\overset{(4.47)}{\leq} r_T^2 + \frac{2}{p_{i_T}}\left(\psi_{i_T}\left(\bar{x}_{i_T}\right) - \psi_{i_T}\left(\left(x_T\right)_{i_T} + d_{i_T}^T\right) - \nabla_{i_T} f\left(x_T\right)^\top\left(x_T - \bar{x}\right)_{i_T}\right) \\
&\quad - \frac{2}{p_{i_T}}\left(\nabla_{i_T} f\left(x_T\right)^\top d_{i_T}^T + \frac{\bar{L}_{i_T}}{2}\left\|d_{i_T}^k\right\|^2\right) \\
&\leq r_T^2 + \frac{2}{p_{i_T}}\left(\psi_{i_T}\left(\bar{x}_{i_T}\right) - \psi_{i_T}\left(\left(x_T\right)_{i_T}\right) - \nabla_{i_T} f\left(x_T\right)^\top\left(x_T - \bar{x}\right)_{i_T}\right) \quad (4.48) \\
&\quad - \frac{2}{p_{i_T}}\left(\nabla_{i_T} f\left(x_T\right)^\top d_{i_T}^T + \frac{\bar{L}_{i_T}}{2}\left\|d_{i_T}^T\right\|^2 + \psi_{i_T}\left(\left(x_T\right)_{i_T} + d_{i_T}^T\right) - \psi_{i_T}\left(\left(x_T\right)_{i_T}\right)\right).
\end{aligned}$$

Taking expectation over i_T on both sides of (4.48), using convexity of f (which implies that $-\nabla f(x_T)^T(x_T - \bar{x}) \leq f(\bar{x}) - f(x_T)$), and using (4.46), we obtain

$$\begin{aligned}
&\mathbb{E}_{i_T}\left[r_{T+1}^2\right] - r_T^2 \\
&\leq 2\left(\psi\left(\bar{x}\right) - \psi\left(x_T\right) + f\left(\bar{x}\right) - f\left(x_T\right)\right) + 2\left(\sum_{i=1}^{n} F\left(x_T\right) - F\left(x_T + d_i^T e_i\right)\right) \\
&\leq 2\left(F^* - F\left(x_T\right)\right) + \frac{2}{p_{\min}} \sum_{i=1}^{n} p_i \left(F\left(x_T\right) - F\left(x_T + d_i^T e_i\right)\right) \qquad (4.49a) \\
&= 2\left(F^* - F\left(x_T\right)\right) + \frac{2}{p_{\min}} \left(F\left(x_T\right) - \mathbb{E}_{i_T}\left[F\left(x_{T+1}\right)\right]\right), \qquad (4.49b)
\end{aligned}$$

where in (4.49a) we used the fact that (4.44) is a descent method. By taking expectation over $i_0, i_1, \ldots, i_k$ on (4.49b), summing over $T = 0, 1, \ldots, k$, and applying Lemma 4.1, we obtain the result. □

Notice that our analysis here improves the rates in Lu and Xiao (2015) and Chapter 3 in terms of the dependency on k and removes the assumption of (4.14a). Even without the improvement from $O(1/k)$ to $o(1/k)$, Theorem 4.10 is the first time that a convergence rate of plain proximal stochastic coordinate descent with arbitrary sampling for the coordinates is proven without additional assumptions such as (4.45). However, we note that unlike the result in Chapter 3, without the additional assumption, we are not able to show faster convergence rate of nonuniform sampling for the coordinates.

4.4 Tightness of the $o(1/k)$ Estimate

We demonstrate that the $o(1/k)$ estimate of convergence of $\{f(x_k)\}$ is tight by showing that for any $\epsilon \in (0, 1]$, there is a convex smooth function for which the sequence of function values generated by gradient descent with a fixed step size converges slower than $O(1/k^{1+\epsilon})$. The example problem

we provide is a simple one-dimensional function, so it serves also as a special case of stochastic coordinate descent and the proximal methods (where $\psi \equiv 0$) as well. Thus, this example shows tightness of our analysis for all methods considered in this chapter.

Consider the one-dimensional real convex function

$$f(x) = x^p, \tag{4.50}$$

where p is an even integer greater than 2. The minimizer of this function is clearly at $x^* = 0$, for which $f(0) = f^* = 0$. Suppose that the gradient descent method is applied starting from $x_0 = 1$. For any descent method, the iterates x_k are confined to $[-1, 1]$ and we have

$$\|\nabla^2 f(x)\| \leq p(p-1) \;\text{ for all } x \text{ with } |x| \leq 1,$$

so we set $L = p(p-1)$. Suppose that $\bar{\alpha} \in (0, 2/L)$ as above. Then the iteration formula is

$$x_{k+1} = x_k - \bar{\alpha}\nabla f(x_k) = x_k \left(1 - p\bar{\alpha}x_k^{p-2}\right), \tag{4.51}$$

and by Lemma 4.4, all iterates lie in a bounded set: the level set $[-1, 1]$ defined by x_0. In fact, since $p \geq 4$ and $\bar{\alpha} \in (0, 2/L)$, we have that

$$\begin{aligned} x_k \in (0, 1] \;\Rightarrow\; 1 - p\bar{\alpha}x_k^{p-2} &\in \left(1 - \frac{2p}{p(p-1)}x_k^{p-2}, 1\right) \\ &\subseteq \left(1 - \frac{2}{p-1}, 1\right) \subseteq \left(\frac{2}{3}, 1\right), \end{aligned}$$

so that $x_{k+1} \in \left(\frac{2}{3}x_k, x_k\right)$ and the value of L remains valid for all iterates.

We show by an informal argument that there exists a constant C such that

$$f(x_k) \approx \frac{C}{k^{p/(p-2)}}, \quad \text{for all } k \text{ sufficiently large.} \tag{4.52}$$

From (4.51) we have

$$f(x_{k+1}) = x_{k+1}^p = x_k^p \left(1 - p\bar{\alpha} x_k^{p-2}\right)^p = f(x_k)\left(1 - p\bar{\alpha} f(x_k)^{(p-2)/p}\right)^p. \quad (4.53)$$

By substituting the hypothesis (4.52) into (4.53), and taking k to be large, we obtain the following sequence of equivalent approximate equalities:

$$\begin{aligned}
& \frac{C}{(k+1)^{p/(p-2)}} \approx \frac{C}{k^{p/(p-2)}} \left(1 - p\bar{\alpha}\frac{C^{(p-2)/p}}{k}\right)^p \\
\Leftrightarrow \quad & \left(\frac{k}{k+1}\right)^{p/(p-2)} \approx \left(1 - p\bar{\alpha}\frac{C^{(p-2)/p}}{k}\right)^p \\
\Leftrightarrow \quad & \left(1 - \frac{1}{k+1}\right)^{p/(p-2)} \approx \left(1 - p^2\bar{\alpha}\frac{C^{(p-2)/p}}{k}\right) \\
\Leftrightarrow \quad & \left(1 - \frac{p}{p-2}\frac{1}{k+1}\right) \approx \left(1 - p^2\bar{\alpha}\frac{C^{(p-2)/p}}{k}\right)
\end{aligned}$$

This last expression is approximately satisfied for large k if C satisfies the expression

$$\frac{p}{p-2} = p^2\bar{\alpha}C^{(p-2)/p}.$$

Stated another way, our result (4.52) indicates that a convergence rate faster than $O(1/k^{1+\epsilon})$ is not possible when steepest descent with fixed steplength is applied to the function $f(x) = x^p$ provided that

$$\frac{p}{p-2} \leq 1 + \epsilon,$$

that is,

$$p \geq 2\frac{1+\epsilon}{\epsilon} \quad \text{and } p \text{ is a positive even integer.}$$

We follow Attouch et al. (2018) to provide a continuous-time analysis of the same objective function, using a gradient flow argument. For the

function f defined by (4.50), consider the following differential equation:

$$x'(t) = -\alpha \nabla f(x(t)). \tag{4.54}$$

Suppose that

$$x(t) = t^{-\theta} \tag{4.55}$$

for some $\theta > 0$, which indicates that starting from any $t > 0$, $x(t)$ lies in a bounded area. Substituting (4.55) into (4.54), we obtain

$$-\theta t^{-\theta-1} = -\alpha p t^{-\theta(p-1)}, \tag{4.56}$$

which holds true if and only if the following equations are satisfied:

$$\begin{cases} \theta & = \alpha p, \\ -\theta - 1 & = -\theta p + \theta, \end{cases}$$

from which we obtain $\theta = (p-2)^{-1}$, $\alpha = (p(p-2))^{-1}$. Starting from $t = 1/\sqrt{2}$, we have $0 < x(t) \leq 2^{-1/(p-2)} \leq \sqrt{2}$ for $p = 4, 6, 8, \ldots$ and for all $t \geq 1/\sqrt{2}$, and $L = p(p-1)/2$ is an appropriate value for a bound on $\|\nabla^2 f(x)\|$ for all $x(t)$ in this range. For this value of L, we have $0 < \alpha \leq \frac{1}{L}$, making α a valid step size. The objective value is $f(x(t)) = t^{-p/(p-2)}$, matching the rate in (4.52).

Appendix

4.A Proof of Lemma 4.8

From the optimality of p for (4.40) and convexity of ψ, we have that for any $\lambda \in [0, 1]$,

$$b^\top d + \frac{a}{2}\|d\|^2 + \psi(x+d) - \psi(x) \leq b^\top \lambda d + \frac{a}{2}\|\lambda d\|^2 + \lambda\left(\psi(x+d) - \psi(x)\right).$$

By rearranging the terms, we get

$$(1-\lambda)\left(b^\top d + \frac{a}{2}\|d\|^2 + \psi(x+d) - \psi(x)\right) \leq -\frac{a\lambda(1-\lambda)}{2}\|d\|^2.$$

Dividing both sides by $1-\lambda$ and let $\lambda \to 1$, we get the desired result.

5 A DISTRIBUTED QUASI-NEWTON ALGORITHM FOR PRIMAL AND DUAL REGULARIZED EMPIRICAL RISK MINIMIZATION

5.1 Introduction

In this chapter, we adopt notations in the machine learning convention that are different from what we used in previous chapters. We consider using multiple machines to solve the following regularized problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad P(\boldsymbol{w}) \coloneqq \xi\left(X^\top \boldsymbol{w}\right) + g(\boldsymbol{w}), \tag{5.1}$$

where X is a d by n real-valued matrix, and g is a convex, closed, and extended-valued proper function that can be nondifferentiable, or its dual problem

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad D(\boldsymbol{\alpha}) \coloneqq g^*(X\boldsymbol{\alpha}) + \xi^*(-\boldsymbol{\alpha}), \tag{5.2}$$

where for any given function $f(\cdot)$, f^* denotes its convex conjugate

$$f^*(z) \coloneqq \max_y \quad z^\top y - f(y).$$

Each column of X represents a single data point or instance, and we assume that the set of data points is partitioned and spread across $K > 1$ machines (i.e. distributed *instance-wise*). We write X as

$$X \coloneqq [X_1, X_2, \ldots, X_K] \tag{5.3}$$

where X_k is stored exclusively on the kth machine. The dual variable $\boldsymbol{\alpha}$ is formed by concatenating $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_K$ where $\boldsymbol{\alpha}_k$ is the dual variable corresponding to X_k. We let $\mathcal{I}_1^X, \ldots, \mathcal{I}_K^X$ denote the indices of the columns of X corresponding to each of the X_k matrices. We further assume that ξ

shares the same block-separable structure and can be written as follows:

$$\xi\left(X^{\top}\boldsymbol{w}\right)=\sum_{k=1}^{K}\xi_{k}\left(X_{k}^{\top}\boldsymbol{w}\right), \tag{5.4}$$

and therefore in (5.2), we have

$$\xi^{*}\left(-\boldsymbol{\alpha}\right)=\sum_{k=1}^{K}\xi_{k}^{*}\left(-\boldsymbol{\alpha}_{k}\right). \tag{5.5}$$

For the ease of description and unification, when solving the primal problem, we also assume that there exists some partition $\mathcal{I}_1^g,\ldots,\mathcal{I}_K^g$ of $\{1,\ldots,d\}$ and g is block-separable according to the partition:

$$g\left(\boldsymbol{w}\right)=\sum_{k=1}^{K}g_{k}\left(\boldsymbol{w}_{\mathcal{I}_{k}^{g}}\right), \tag{5.6}$$

though our algorithm can be adapted for non-separable g with minimal modification.

When we solve the primal problem (5.1), ξ is assumed to be a differentiable function with Lipschitz continuous gradients, and is allowed to be nonconvex. On the other hand, when the dual problem (5.2) is considered, for recovering the primal solution, we require strong convexity on g and convexity on ξ, and ξ can be either nonsmooth but Lipschitz continuous (within the area of interest), or Lipschitz continuously differentiable. Note that strong convexity of g implies that g^* is Lipschitz-continuously differentiable (Hiriart-Urruty and Lemaréchal, 2001, Part E, Theorem 4.2.1 and Theorem 4.2.2), making (5.2) have the same structure as (5.1) such that both problems have one smooth and one nonsmooth term. There are several reasons for considering the alternative dual problem. First, when ξ is nonsmooth, the primal problem becomes hard to solve as both terms are nonsmooth, meanwhile in the dual problem, ξ^* is guaranteed to be smooth. Second, the number of variables in the primal and the dual

probelm are different. In our algorithm whose spatial and temporal costs are positively correlated to the number of variables, when the data set has much higher feature dimension than the number of data points, solving the dual problem can be more economical.

The bottleneck in performing distributed optimization is often the high cost of communication between machines. For (5.1) or (5.2), the time required to retrieve X_k over a network can greatly exceed the time needed to compute ξ_k or its gradient with locally stored X_k. Moreover, we incur a delay at the beginning of each round of communication due to the overhead of establishing connections between machines. This latency prevents many efficient single-core algorithms such as coordinate descent (CD) and stochastic gradient and their asynchronous parallel variants from being employed in large-scale distributed computing setups. Thus, a key aim of algorithm design for distributed optimization is to improve the communication efficiency while keeping the computational cost affordable. Batch methods are preferred in this context, because fewer rounds of communication occur in distributed batch methods.

When the objective is smooth, many batch methods can be used directly in distributed environments to optimize them. For example, Nesterov's accelerated gradient (AG) (Nesterov, 1983) enjoys low iteration complexity, and since each iteration of AG only requires one round of communication to compute the new gradient, it also has good communication complexity. Although its supporting theory is not particularly strong, the limited-memory BFGS (LBFGS) method (Liu and Nocedal, 1989) is popular among practitioners of distributed optimization. It is the default algorithm for solving ℓ_2-regularized smooth ERM problems in Apache Spark's distributed machine learning library (Meng et al., 2016), as it is empirically much faster than AG (see, for example, the experiments in Wang et al. (2016)). Other batch methods that utilize the Hessian of the objective in various ways are also communication-efficient under their

own additional assumptions (Shamir et al., 2014; Zhang and Lin, 2015; Lee et al., 2017; Zhuang et al., 2015; Lin et al., 2014).

However, when the objective is nondifferentiable, neither LBFGS nor Newton's method can be applied directly. Leveraging curvature information from the smooth part (ξ in the primal or g^* in the dual) can still be beneficial in this setting. For example, the orthant-wise quasi-Newton method OWLQN (Andrew and Gao, 2007) adapts the LBFGS algorithm to the special nonsmooth case in which $g(\cdot) \equiv \|\cdot\|_1$ for (5.1), and is popular for distributed optimization of ℓ_1-regularized ERM problems. Unfortunately, extension of this approach to other nonsmooth g is not well understood, and the convergence guarantees are only asymptotic, rather than global. Another example is that for (5.2), state of the art distributed algorithms (Yang, 2013; Lee and Chang, 2017; Zheng et al., 2017) utilize block-diagonal entries of the real Hessian of $g^*(X\boldsymbol{\alpha})$.

To the best of our knowledge, for ERMs with *general* nonsmooth regularizers in the instance-wise storage setting, proximal-gradient-like methods (Wright et al., 2009; Beck and Teboulle, 2009; Nesterov, 2013) are the only practical distributed optimization algorithms with convergence guarantees for the primal problem (5.1). Since these methods barely use the curvature information of the smooth part (if at all), we suspect that proper utilization of second-order information has the potential to improve convergence speed and therefore communication efficiency dramatically. As for algorithms solving the dual problem (5.2), computing $X\boldsymbol{\alpha}$ in the instance-wise storage setting requires communicating a d-dimensional vector, and only the block-diagonal part of $\partial^2_{\boldsymbol{\alpha}} g^*(X\boldsymbol{\alpha})$ can be obtained easily. Therefore, global curvature information is not utilized in existing algorithms, and we expect that utilizing global second-order information of g^* can also provide substantial benefits over the block-diagonal approximation approaches. We thus propose a practical distributed inexact variable-metric algorithm that can be applied to both (5.1) and (5.2). Our algorithm uses gradients

and updates information from previous iterations to estimate curvature of the smooth part in a communication-efficient manner. We describe construction of this estimate and solution of the corresponding subproblem. We also provide convergence rate guarantees, which also bound communication complexity. These rates improve on existing distributed methods, even those tailor-made for specific regularizers.

More specifically, We propose a distributed inexact proximal-quasi-Newton-like algorithm that can be used to solve both (5.1) and (5.2) under the instance-wise split setting that share the common structure of having a smooth term f and a nonsmooth term Ψ. At each iteration with the current iterate x, our algorithm utilizes the previous update directions and gradients to construct a second-order approximation of the smooth part f by the LBFGS method, and approximately minimizes this quadratic term plus the nonsmooth term Ψ to obtain an update iteration p.

$$p \approx \arg\min_{p} \; Q_H(p; x), \tag{5.7}$$

where H is the LBFGS approximation of the Hessian of f at x, and

$$Q_H(p; x) \coloneqq \nabla f(x)^\top p + \frac{1}{2} p^\top H p + \Psi(x + p) - \Psi(x). \tag{5.8}$$

For the primal problem (5.1), we believe that this work is the first to propose, analyze, and implement a practically feasible distributed optimization method for solving (5.1) with general nonsmooth regularizer g under the instance-wise storage setting. For the dual problem (5.2), our algorithm is the first to suggest an approach that utilizes global curvature information under the constraint of distributed data storage. This usage of non-local curvature information greatly improves upon state of the art for the distributed dual ERM problem which uses the block-diagonal parts of the Hessian only. An obvious drawback of the block-diagonal approach is that the convergence deteriorates with the number of ma-

chines, as more and more off-block-diagonal entries are ignored. In the extreme case, where there are n machines such that each machine stores only one column of X, the block-diagonal approach reduces to a scaled proximal-gradient algorithm and the convergence is expected to be extremely slow. On the other hand, our algorithm has convergence behavior independent of number of machines and data distribution over nodes, and is thus favorable when many machines are used. Our approach has both good communication and computational complexities, unlike certain approaches that focus only on communication at the expense of computation (and ultimately overall time).

Contributions

We summarize our main contributions as follows.

- The proposed method is the first real distributed second-order method for the dual ERM problem that utilizes global curvature information of the smooth part. Existing second-order methods use only the block-diagonal part of the Hessian and suffers from asymptotic convergence speed as slow as proximal gradient, while our method enjoys fast convergence throughout. Numerical results show that our inexact proximal-quasi-Newton method is magnitudes faster than state of the art for distributed optimizing the dual ERM problem.

- We propose the first distributed algorithm for primal ERMs with general nonsmooth regularizers (5.1) under the instance-wise split setting. Prior to our work, existing algorithms are either for a specific regularizer (in particular the ℓ_1 norm) or for the feature-wise split setting, which is often impractical. In particular, it is usually easier to generate new data points than to generate new features, and each time new data points are obtained from one location, one needs to

distribute their entries to different machines under the feature-wise setting.

- The proposed framework is applicable to both primal and dual ERM problems under the same instance-wise split setting, and the convergence speed is not deteriorated by the number of machines. Existing methods that applicable to both problems can deal with feature-wise split for the primal problem only, and their convergence degrades with the number of machines used, and are thus not suitable for large-scale applications where thousands of or more machines are used. This unification also reduces two problems into one and facilitates future development for them.

- Our analysis provides sharper convergence guarantees and therefore better communication efficiency. In particular, global linear convergence for a broad class of non-strongly convex problems that includes many popular ERM problems are shown, and an early linear convergence to rapidly reach a medium solution accuracy is proven for convex problems.

Organization

We first describe the general distributed algorithm in Section 5.2. Convergence guarantee, communication complexity, and the effect of the subproblem solution inexactness are analyzed in Section 5.3. Specific details for applying our algorithm respectively on the primal and the dual problem are given in Section 5.4. Section 5.5 discusses related works, and empirical comparisons are conducted in Section 5.6. Concluding observations appear in Section 5.7.

Notation and Assumptions

We use the following notation.

- $\|\cdot\|$ denotes the 2-norm, both for vectors and for matrices.
- Given any symmetric positive semi-definite matrix $H \in \mathbb{R}^{d\times d}$ and any vector $p \in \mathbb{R}^d$, $\|p\|_H$ denotes the semi-norm $\sqrt{p^\top H p}$.

In addition to the structural assumptions of distributed instance-wise storage of X in (5.3) and the block separability of ξ in (5.4), we also use the following assumptions throughout this chapter. When we solve the primal problem, we assume the following.

Assumption 3. *The regularization term $g(\boldsymbol{w})$ is convex, extended-valued, proper, and closed. The loss function $\xi(X^\top \boldsymbol{w})$ is L-Lipschitz continuously differentiable with respect to $\boldsymbol{w}$ for some $L > 0$. That is,*

$$\left\|X^\top \xi'\left(X^\top \boldsymbol{w}_1\right) - X^\top \xi'\left(X^\top \boldsymbol{w}_2\right)\right\| \leq L\left\|\boldsymbol{w}_1 - \boldsymbol{w}_2\right\|, \forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^d. \quad (5.9)$$

On the other hand, when we consider solving the dual problem, the following is assumed.

Assumption 4. *Both g and ξ are convex. $g^*(X\boldsymbol{\alpha})$ is L-Lipschitz continuously differentiable with respect to $\boldsymbol{\alpha}$. Either ξ^* is σ-strongly convex for some $\sigma > 0$, or the loss term $\xi(X^\top \boldsymbol{w})$ is ρ-Lipschitz continuous for some ρ.*

Because a function is ρ-Lipschitz continuously differentiable if and only if its conjugate is $(1/\rho)$-strongly convex (Hiriart-Urruty and Lemaréchal, 2001, Part E, Theorem 4.2.1 and Theorem 4.2.2), Assumption 4 implies that g is $\|X^\top X\|/L$-strongly convex. From the same reasoning, ξ^* is σ-strongly convex if only if ξ is $(1/\sigma)$ Lipschitz continuously differentiable. Convexity of the primal problem in Assumption 4 together with Slater's condition guarantee strong duality Boyd and Vandenberghe (2004, Section 5.2.3),

which then ensures (5.2) is indeed an alternative to (5.1). Moreover, from KKT conditions, any optimal solution $\boldsymbol{\alpha}^*$ for (5.2) gives us a primal optimal solution $\boldsymbol{w}^*$ for (5.1) through

$$\boldsymbol{w}^* = \nabla g^*(X\boldsymbol{\alpha}^*). \tag{5.10}$$

5.2 Algorithm

We describe and analyze a general algorithmic scheme that can be applied to solve both the primal (5.1) and dual (5.2) problems under the instance-wise distributed data storage scenario (5.3). In Section 5.4, we discuss how to efficiently implement particular steps of this scheme for (5.1) and (5.2).

Consider a general problem of the form

$$\min_{x\in\mathbb{R}^N} \quad F(x) \coloneqq f(x) + \Psi(x), \tag{5.11}$$

where f is L-Lipschitz continuously differentiable for some $L > 0$ and Ψ is convex, closed, proper, extended valued, and block-separable into K blocks. More specifically, we can write $\Psi(x)$ as

$$\Psi(x) = \sum_{k=1}^{K} \Psi_k(x_{\mathcal{I}_k}). \tag{5.12}$$

where $\mathcal{I}_1, \ldots, \mathcal{I}_K$ partitions $\{1, \ldots, N\}$.

We assume as well that for the kth machine, $\nabla_{\mathcal{I}_k} f(x)$ can be obtained easily after communicating a vector of size $O(d)$ across machines, and postpone the detailed gradient calculation until we discuss specific problem structures in later sections. Note that this d is the primal variable dimension in (5.1) and is independent of N.

The primal and dual problems are specific cases of the general form (5.11). For the primal problem (5.1) we let $N = d$, $x = \boldsymbol{w}$, $f(\cdot) = \xi(X^\top \cdot)$,

and $\Psi(\cdot) = g(\cdot)$. The block-separability of g (5.6) gives the desired block-separability of Ψ (5.12), and the Lipschitz-continuous differentiability of f comes from Assumption 3. For the dual problem (5.2), we have $N = n$, $x = \boldsymbol{\alpha}$, $f(\cdot) = g^*(X\cdot)$, and $\Psi(\cdot) = \xi^*(-\cdot)$. The separability follows from (5.5), where the partition (5.12) reflects the data partition in (5.3) and Lipschitz continuity from Assumption 4.

Each iteration of our algorithm has four main steps – (1) computing the gradient $\nabla f(x)$, (2) constructing an approximate Hessian H of f, (3) solving a quadratic approximation subproblem to find an update direction p, and finally (4) taking a step $x + \lambda p$ either via line search or trust-region approach. The gradient computation step and part of the line search process is dependent on whether we are solving the primal or dual problem, and we defer the details to Section 5.4. The approximate Hessian H comes from the LBFGS algorithm Liu and Nocedal (1989). To compute the update direction, we approximately solve (5.7), where Q_H consists of a quadratic approximation to f and the regularizer Ψ as defined in (5.8). We then use either a line search procedure to determine a suitable stepsize λ and perform the update $x \leftarrow x + \lambda p$, or use some trust-region-like techniques to decide whether to accept the update direction with unit step size.

We now discuss the following issues in the distributed setting: communication cost in distributed environments, the choice and construction of H that have low cost in terms of both communication and per machine computation, procedures for solving (5.7), and the line search and trust-region procedures for ensuring sufficient objective decrease.

Communication Cost Model

For the ease of description, we assume the *allreduce* model of MPI (Message Passing Interface Forum, 1994) throughout the chapter, but it is also straightforward to extend the framework to a master-worker platform. Under this *allreduce* model, all machines simultaneously fulfill master and

worker roles, and for any distributed operations that aggregate results from machines, the resultant is broadcast to all machines.

This can be considered as equivalent to conducting one map-reduce operation and then broadcasting the result to all nodes. The communication cost for the allreduce operation on a d-dimensional vector under this model is

$$\log(K)\, T_{\text{initial}} + dT_{\text{byte}}, \tag{5.13}$$

where T_{initial} is the latency to establish connection between machines, and T_{byte} is the per byte transmission time (see, for example, Chan et al. (2007, Section 6.3)).

The first term in (5.13) also explains why batch methods are preferable. Even if methods that frequently update the iterates communicate the same amount of bytes, it takes more rounds of communication to transmit the information, and the overhead of $\log(K)T_{\text{initial}}$ incurred at every round of communication makes this cost dominant, especially when K is large.

In subsequent discussion, when an allreduce operation is performed on a vector of dimension $O(d)$, we simply say that a round of $O(d)$ communication is conducted. We omit the latency term since batch methods like ours tend to have only a small constant number of rounds of communication per iteration. By contrast, non-batch methods such as CD or stochastic gradient require number of communication rounds per epoch equal to data size or dimension, and therefore face much more significant latency issues.

Constructing a good H efficiently

We use the Hessian approximation constructed by the LBFGS algorithm (Liu and Nocedal, 1989) as our H in (5.8), and propose a way to maintain it efficiently in a distributed setting. In particular, we show that most vectors involved can be stored perfectly in a distributed manner in accord with

the partition $\mathcal{I}_k$ in (5.12), and this distributed storage further facilitates parallelization of most computation. Note that the LBFGS algorithm works even if the smooth part is not twice-differentiable, see Lemma 5.1. In fact, Lipschitz continuity of the gradient implies that the function is twice-differentiable almost everywhere, and generalized Hessian can be used at the points where the smooth part is not twice-differentiable. In this case, the LBFGS approximation is for the generalized Hessian.

Using the compact representation in Byrd et al. (1994), given a prespecified integer $m > 0$, at the tth iteration for $t > 0$, let $m(t) := \min(m, t)$, and define

$$\boldsymbol{s}_i := x^{i+1} - x^i, \quad \boldsymbol{y}_i := \nabla f(x^{i+1}) - \nabla f(x^i), \quad \forall i.$$

The LBFGS Hessian approximation matrix is

$$H_t = \gamma_t I - U_t M_t^{-1} U_t^\top, \tag{5.14}$$

where

$$U_t := [\gamma_t S_t, Y_t], \quad M_t := \begin{bmatrix} \gamma_t S_t^\top S_t, & L_t \\ L_t^\top & -D_t \end{bmatrix}, \quad \gamma_t := \frac{\boldsymbol{y}_{t-1}^\top \boldsymbol{y}_{t-1}}{\boldsymbol{s}_{t-1}^\top \boldsymbol{y}_{t-1}}, \tag{5.15}$$

and

$$S_t := \left[\boldsymbol{s}_{t-m(t)}, \boldsymbol{s}_{t-m(t)+1}, \ldots, \boldsymbol{s}_{t-1}\right], \tag{5.16a}$$

$$Y_t := \left[\boldsymbol{y}_{t-m(t)}, \boldsymbol{y}_{t-m(t)+1}, \ldots, \boldsymbol{y}_{t-1}\right], \tag{5.16b}$$

$$D_t := \operatorname{diag}\left(\boldsymbol{s}_{t-m(t)}^\top \boldsymbol{y}_{t-m(t)}, \ldots, \boldsymbol{s}_{t-1}^\top \boldsymbol{y}_{t-1}\right), \tag{5.16c}$$

$$(L_t)_{i,j} := \begin{cases} \boldsymbol{s}_{t-m(t)-1+i}^\top \boldsymbol{y}_{t-m(t)-1+j}, & \text{if } i > j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.16d}$$

For $t = 0$ where no $\boldsymbol{s}_i$ and $\boldsymbol{y}_i$ are available, we either set $H_0 := a_0 I$ for some positive scalar a_0, or use some Hessian approximation constructed

using local data. More details are given in Section 5.4 when we discuss the primal and dual problems individually.

If f is not strongly convex, it is possible that (5.14) is only positive semi-definite, making the subproblem (5.7) ill-conditioned. In this case, we follow Li and Fukushima (2001), taking the m update pairs to be the most recent m iterations for which the inequality

$$\boldsymbol{s}_i^\top \boldsymbol{y}_i \geq \delta \boldsymbol{s}_i^\top \boldsymbol{s}_i \tag{5.17}$$

is satisfied, for some predefined $\delta > 0$. It can be shown that this safeguard ensures that H_t are always positive definite and the eigenvalues are bounded within a positive range. For a proof in the case that f is twice-differentiable, see, for example, the appendix of Lee and Wright (2017). For completeness, we provide a proof without the assumption of twice-differentiability of f in Lemma 5.1.

To construct and utilize this H_t efficiently, we store $(U_t)_{\mathcal{I}_k,:}$ on the kth machine, and all machines keep a copy of the whole M_t matrix as usually m is small and this is affordable. Under our assumption, on the kth machine, the local gradient $\nabla_{\mathcal{I}_k} f$ can be obtained, and we will show how to compute the update direction $p_{\mathcal{I}_k}$ locally in the next subsection. Thus, since $\boldsymbol{s}_i$ are just the update direction p scaled by the step size λ, it can be obtained without any additional communication. All the information needed to construct H_t is hence available locally on each machine.

We now consider the costs associated with the matrix M_t^{-1}. The matrix M_t, but not its inverse, is maintained for easier update. In practice, m is usually much smaller than N, so the $O(m^3)$ cost of inverting the matrix directly is insignificant compared to the cost of the other steps. On contrary, if N is large, the computation of the inner products $\boldsymbol{s}_i^\top \boldsymbol{y}_j$ and $\boldsymbol{s}_i^\top \boldsymbol{s}_j$ can be the bottleneck in constructing M_t^{-1}. We can significantly reduce this cost by computing and maintaining the inner products in parallel and assembling the results with $O(m)$ communication cost. At the tth iteration,

given the new $\boldsymbol{s}_{t-1}$, because U_t is stored disjointly on the machines, we compute the inner products of $\boldsymbol{s}_{t-1}$ with both S_t and Y_t in parallel via the summations

$$\sum_{k=1}^{K} \left((S_t)_{\mathcal{I}_k,:}^\top (\boldsymbol{s}_{t-1})_{\mathcal{I}_k} \right), \quad \sum_{k=1}^{K} \left((Y_t)_{\mathcal{I}_k,:}^\top (\boldsymbol{s}_{t-1})_{\mathcal{I}_k} \right),$$

requiring $O(m)$ communication of the partial sums on each machine. We keep these results until $\boldsymbol{s}_{t-1}$ and $\boldsymbol{y}_{t-1}$ are discarded, so that at each iteration, only $2m$ (not $O(m^2)$) inner products are computed.

Solving the Quadratic Approximation Subproblem to Find Update Direction

The matrix H_t is generally not diagonal, so there is no easy closed-form solution to (5.7). We will instead use iterative algorithms to obtain an approximate solution to this subproblem. In single-core environments, coordinate descent (CD) is one of the most efficient approaches for solving (5.7) (Yuan et al., 2012; Zhong et al., 2014; Scheinberg and Tang, 2016). When N is not too large, instead of the distributed approach we discussed in the previous section, it is possible to construct H_t on all machines. In this case, a local CD process can be applied on all machines to save communication cost, in the price that all machines conduct the same calculation and the additional computational power from multiple machines is wasted. The alternative approach of applying proximal-gradient methods to (5.7) may be more efficient in distributed settings, since they can be parallelized with little communication cost for large N.

The fastest proximal-gradient-type methods are accelerated gradient (AG) (Beck and Teboulle, 2009; Nesterov, 2013) and SpaRSA (Wright et al., 2009). SpaRSA is a basic proximal-gradient method with spectral initialization of the parameter in the prox term. SpaRSA has a few key advantages

over AG despite its weaker theoretical convergence rate guarantees. It tends to be faster in the early iterations of the algorithm (Yang and Zhang, 2011), thus possibly yielding a solution of acceptable accuracy in fewer iterations than AG. It is also a descent method, reducing the objective Q_H at every iteration, which ensures that the solution returned is at least as good as the original guess $p = 0$

In the rest of this subsection, we will describe a distributed implementation of SpaRSA for (5.7), with H as defined in (5.14). The major computation is obtaining the gradient of the smooth (quadratic) part of (5.8), and thus with minimal modification, AG can be used with the same per iteration cost. To distinguish between the iterations of our main algorithm (i.e. the entire process required to update x a single time) and the iterations of SpaRSA, we will refer to them by *main iterations* and *SpaRSA iterations* respectively.

Since H and x are fixed in this subsection, we will write $Q_H(\cdot; x)$ simply as $Q(\cdot)$. We denote the ith iterate of the SpaRSA algorithm as $p^{(i)}$, and we initialize $p^{(0)} = 0$ whenever there is no obviously better choice. We denote the smooth part of Q_H by $\hat{f}(p)$, and the nonsmooth $\Psi(x + p)$ by $\hat{\Psi}(p)$.

$$\hat{f}(p) \coloneqq \nabla f(x)^\top p + \frac{1}{2} p^\top H p, \quad \hat{\Psi}(p) \coloneqq \Psi(x + p) - \Psi(x). \tag{5.18}$$

At the ith iteration of SpaRSA, we define

$$u_{\psi_i}^{(i)} \coloneqq p^{(i)} - \frac{\nabla \hat{f}(p^{(i)})}{\psi_i}, \tag{5.19}$$

and solve the following subproblem:

$$p^{(i+1)} = \arg\min_p \frac{1}{2} \left\| p - u_{\psi_i}^{(i)} \right\|^2 + \frac{\hat{\Psi}(p)}{\psi_i}, \tag{5.20}$$

where ψ_i is defined by the following "spectral" formula:

$$\psi_i = \frac{\left(p^{(i)} - p^{(i-1)}\right)^\top \left(\nabla \hat{f}(p^{(i)}) - \nabla \hat{f}(p^{(i-1)})\right)}{\|p^{(i)} - p^{(i-1)}\|^2}. \tag{5.21}$$

When $i = 0$, we use a pre-assigned value for ψ_0 instead. (In our LBFGS choice for H_t, we use the value of γ_t from (5.15) as the initial estimate of ψ_0.) The exact minimizer of (5.20) can be difficult to compute for general Ψ. However, approximate solutions of (5.20) suffice to provide a convergence rate guarantee for solving (5.7) as discussed in Chapter 2. Since it is known (see Lemma 5.1) that the eigenvalues of H are upper- and lower-bounded in a positive range after the safeguard (5.17) is applied, we can guarantee that this initialization of ψ_i is bounded within a positive range; see Section 5.3. The initial value of ψ_i defined in (5.21) is increased successively by a chosen constant factor $\beta > 1$, and $p^{(i+1)}$ is recalculated from (5.20), until the following sufficient decrease criterion is satisfied:

$$Q\left(p^{(i+1)}\right) \leq Q\left(p^{(i)}\right) - \frac{\sigma_0 \psi_i}{2} \left\|p^{(i+1)} - p^{(i)}\right\|^2, \tag{5.22}$$

for some specified $\sigma_0 \in (0, 1)$. Note that the evaluation of $Q(p)$ needed in (5.22) can be done efficiently through a parallel computation of

$$\sum_{k=1}^{K} \frac{1}{2} \left(\nabla_{\mathcal{I}_k} \hat{f}\,(p) + \nabla_{\mathcal{I}_k} f\,(x)\right)^\top p_{\mathcal{I}_k} + \hat{\Psi}_k\,(p_{\mathcal{I}_k})\,.$$

From the boundedness of H, one can easily prove that (5.22) is satisfied after a finite number of increases of ψ_i, as we will show in Section 5.3. In our algorithm, SpaRSA runs until either a fixed number of iterations is reached, or when some certain inner stopping condition for optimizing (5.7) is satisfied.

For general H, the computational bottleneck of $\nabla \hat{f}$ would take $O(N^2)$ operations to compute the $Hp^{(i)}$ term. However, for our LBFGS choice of

H, this cost is reduced to $O(mN + m^2)$ by utilizing the matrix structure, as shown in the following formula:

$$\nabla \hat{f}\,(p) = \nabla f\,(x) + Hp = \nabla f(x) + \gamma p - U_t \left(M_t^{-1}\left(U_t^\top p\right)\right). \tag{5.23}$$

The computation of (5.23) can be parallelized, by first parallelizing computation of the inner product $U_t^\top p^{(i)}$ via the formula

$$\sum_{k=1}^{K} (U_t)_{\mathcal{I}_k,:}^\top \, p_{\mathcal{I}_k}^{(i)}$$

with $O(m)$ communication. (We implement the parallel inner products as described in Section 5.2.) We let each machine compute a subvector of u in (5.19) according to (5.12).

From the block-separability of Ψ, the subproblem (5.20) for computing $p^{(i)}$ can be decomposed into independent subproblems partitioned along $\mathcal{I}_1, \ldots, \mathcal{I}_K$. The kth machine therefore locally computes $p_{\mathcal{I}_k}^{(i)}$ without communicating the whole vector. Then at each iteration of SpaRSA, partial inner products between $(U_t)_{\mathcal{I}_k,:}$ and $p_{\mathcal{I}_k}^{(i)}$ can be computed locally, and the results are assembled with an allreduce operation of $O(m)$ communication cost. This leads to a round of $O(m)$ communication cost per SpaRSA iteration, with the computational cost reduced from $O(mN)$ to $O(mN/K)$ per machine on average. Since both the $O(m)$ communication cost and the $O(mN/K)$ computational cost are inexpensive when m is small, in comparison to the computation of ∇f, one can afford to conduct multiple iterations of SpaRSA at every main iteration. Note that the total latency incurred over all allreduce operations as discussed in (5.13) can be capped by setting a maximum iteration limit for SpaRSA.

The distributed implementation of SpaRSA for solving (5.7) is summarized in Algorithm 8.

Algorithm 8 Distributed SpaRSA for solving (5.7) with LBFGS quadratic approximation (5.14) on machine k

Given $\beta, \sigma_0 \in (0,1)$, M_t^{-1}, U_t, γ_t, and $\mathcal{I}_k$;
Set $p_{\mathcal{I}_k}^{(0)} \leftarrow 0$;
for $i = 0, 1, 2, \ldots$ **do**
  if $i = 0$ **then**
    $\psi = \gamma_t$;
  else
    Compute ψ in (5.21) through

$$\sum_{j=1}^{K} \left(p_{\mathcal{I}_j}^{(i)} - p_{\mathcal{I}_j}^{(i-1)}\right)^\top \left(\nabla_{\mathcal{I}_j} \hat{f}\left(p^{(i)}\right) - \nabla_{\mathcal{I}_j} \hat{f}\left(p^{(i-1)}\right)\right), \quad \text{and} \quad \sum_{j=1}^{K} \left\|p_{\mathcal{I}_j}^{(i)} - p_{\mathcal{I}_j}^{(i-1)}\right\|^2;$$

  Obtain

$$U_t^\top p^{(i)} = \sum_{j=1}^{K} (U_t)_{\mathcal{I}_j,:}^\top p_{\mathcal{I}_j}^{(i)};$$

  Compute

$$\nabla_{\mathcal{I}_k} \hat{f}\left(p^{(i)}\right) = \nabla_{\mathcal{I}_k} f(x) + \gamma p_{\mathcal{I}_k}^{(i)} - (U_t)_{\mathcal{I}_k,:} \left(M_t^{-1}\left(U_t^\top p^{(i)}\right)\right)$$

  by (5.23);
  while TRUE **do**
    Solve (5.20) on coordinates indexed by $\mathcal{I}_k$ to obtain $p_{\mathcal{I}_k}$;
    if (5.22) holds **then**
      $p_{\mathcal{I}_k}^{(i+1)} \leftarrow p_{\mathcal{I}_k}$; $\psi_i \leftarrow \psi$;
      Break;
    $\psi \leftarrow \beta^{-1}\psi$;
    Re-solve (5.20) with the new ψ to obtain a new $p_{\mathcal{I}_k}$;
  Break if some stopping condition is met;

Sufficient Function Decrease

After obtaining an update direction p by approximately solving (5.7), we need to ensure sufficient objective decrease. This is usually achieved by some line-search or trust-region procedure. In this section, we describe two such approaches, one based on backtracking line search for the step size,

and one based on a trust-region like approach that modifies H repeatedly until an update direction is accepted with unit step size.

For the line-search approach, we follow Tseng and Yun (2009) by using a modified-Armijo-type backtracking line search to find a suitable step size λ. Given the current iterate x, the update direction p, and parameters $\sigma_1, \theta \in (0, 1)$, we set

$$\Delta \coloneqq \nabla f(x)^\top p + \Psi(x+p) - \Psi(x) \tag{5.24}$$

and pick the step size as the largest of $\theta^0, \theta^1, \ldots$ satisfying

$$F(x + \lambda p) \leq F(x) + \lambda \sigma_1 \Delta. \tag{5.25}$$

The computation of Δ is negligible as all the terms are involved in $Q(p; x)$, and $Q(p; x)$ is evaluated in the line search procedure of SpaRSA. For the function value evaluation, the objective values of both (5.1) and (5.2) can be evaluated efficiently if we precompute Xp or $X^\top p$ in advance and conduct all reevaluations through this vector but not repeated matrix-vector products. Details are discussed in Section 5.4. Note that because H_t defined in (5.14) attempts to approximate the real Hessian, empirically the unit step $\lambda = 1$ frequently satisfies (5.25), so we use the value 1 as the initial guess.

For the trust-region-like procedure, we start from the original H, and use the same $\sigma_1, \theta \in (0, 1)$ as above. Whenever the sufficient decrease condition

$$F(x+p) - F(x) \leq \sigma_1 Q_H(p; x) \tag{5.26}$$

is not satisfied, we scale up H by $H \leftarrow H/\theta$, and resolve (5.7), either from 0 or from the previously obtained solution p if it gives an objective better than 0. We note that when Ψ is not present, both the backtracking approach and the trust-region one generate the same iterates. But when Ψ is incorporated, the two approaches may generate different updates.

Similar to the line-search approach, the evaluation of $Q_H(p; x)$ comes for free from the SpaRSA procedure, and usually the original H (5.14) generates update steps satisfying (5.26). Therefore, solving (5.7) multiple times per main iteration is barely encountered in practice.

The trust-region procedure may be more expensive than line search because solving the subproblem again is more expensive than trying a different step size, although both cases are empirically rare. But on the other hand, when there are additional properties of the regularizer such as sparsity promotion, a potential benefit of the trust-region approach is that it might be able to identify the sparsity pattern earlier because unit step size is always used.

Our distributed algorithm for (5.11) is summarized in Algorithm 9. We refer to the line search and trust-region variants of the algorithm as DPLBFGS-LS and DPLBFGS-TR respectively, and we will refer to them collectively as simply DPLBFGS.

Cost Analysis

We now describe the computational and communication cost of our algorithms. The computational cost for each machine depends on which X_k is stored locally and the size of $|\mathcal{I}_k|$, and for simplicity we report the computational cost *averaged over all machines*. The communication costs do not depend on X_k.

For the distributed version of Algorithm 8, each iteration costs

$$O\left(\frac{N}{K} + \frac{mN}{K} + m^2\right) = O\left(\frac{mN}{K} + m^2\right) \tag{5.27}$$

in computation, where the N/K term is for the vector additions in (5.23), and

$$O\left(m + \text{number of times (5.22) is evaluated}\right)$$

in communication. In the next section, we will show that (5.22) is accepted within a fixed number of times and thus the overall communication cost is $O(m)$.

For DPLBFGS, we will give details in Section 5.4 that for both (5.1) and (5.2), each gradient evaluation for f takes $O(\#\text{nnz}/K)$ per machine computation in average and $O(d)$ in communication, where #nnz is the number of nonzero elements in the data matrix X. As shown in the next section, in one main iteration, the number of function evaluations in the line search is bounded, and its cost is negligible if we are using the same p but just different step sizes; see Section 5.4. For the trust region approach, the number of times for modifying H and resolving (5.7) is also bounded, and thus the asymptotical cost is not altered. In summary, the computational cost per main iteration is therefore

$$O\left(\frac{\#\text{nnz}}{K} + \frac{mN}{K} + m^3 + \frac{N}{K}\right) = O\left(\frac{\#\text{nnz}}{K} + \frac{mN}{K} + m^3\right), \tag{5.28}$$

and the communication cost is

$$O\left(1 + d\right) = O\left(d\right),$$

where the $O(1)$ part is for function value evaluation and checking the safeguard (5.17). We note that the costs of Algorithm 8 are dominated by those of DPLBFGS if a fixed number of SpaRSA iterations is conducted every main iteration.

5.3 Convergence Rate and Communication Complexity Analysis

The use of an iterative solver for the subproblem (5.7) generally results in an inexact solution. We first show that running SpaRSA for any fixed

number of iterations guarantees a step p whose accuracy is sufficient to prove overall convergence.

Lemma 5.1. *Consider optimizing* (5.11) *by DPLBFGS. By using* H_t *as defined in* (5.14) *with the safeguard mechanism* (5.17) *in* (5.7)*, we have the following.*

1. *We have* $L^2/\delta \geq \gamma_t \geq \delta$ *for all* $t > 0$*, where* L *is the Lipschitz constant for* ∇f*. Moreover, there exist constants* $c_1 \geq c_2 > 0$ *such that* $c_1 I \succeq H_t \succeq c_2 I$ *for all* $t > 0$*.*
2. *At every SpaRSA iteration, the initial estimate of* ψ_i *is bounded within the range of*
$$\left[\min\left\{c_2, \delta\right\}, \max\left\{c_1, \frac{L^2}{\delta}\right\}\right],$$
and the final accepted value ψ_i *is upper-bounded.*
3. *SpaRSA is globally Q-linear convergent in solving* (5.7)*. Therefore, there exists* $\eta \in [0, 1)$ *such that if we run at least* S *iterations of SpaRSA for all main iterations for any* $S > 0$*, the approximate solution* p *satisfies*
$$-\eta^S Q^* = \eta^S \left(Q\left(0\right) - Q^*\right) \geq Q\left(p\right) - Q^*, \tag{5.29}$$
where Q^* *is the optimal objective of* (5.7)*.*

Lemma 5.1 establishes how the number of iterations of SpaRSA affects the inexactness of the subproblem solution. Given this measure, we can leverage the results developed in Chapter 2 to obtain iteration complexity guarantees for our algorithm. Since in our algorithm, communication complexity scales linearly with iteration complexity, this guarantee provides a bound on the amount of communication. In particular, our method communicates $O(d + mS)$ bytes per iteration (where S is the number of SpaRSA iterations used, as in Lemma 5.1) and the second term can usually be ignored for small m.

We show next that the step size generated by our line search procedure in DPLBFGS-LS is lower bounded by a positive value.

Lemma 5.2. *Consider* (5.11) *such that* f *is* L*-Lipschitz differentiable and* Ψ *is convex. If SpaRSA is run at least* S *iterations in solving* (5.7)*, the corresponding* Δ *defined in* (5.24) *satisfies*

$$\Delta \leq -\frac{c_2 \|p\|^2}{1+\eta^{\frac{S}{2}}}, \tag{5.30}$$

where η *and* c_2 *are the same as that defined in Lemma* 5.1*. Moreover, the backtracking subroutine in DPLBFGS-LS terminates in finite number of steps and produces a step size*

$$\lambda \geq \min\left\{1, \frac{2\theta\,(1-\sigma_1)\,c_2}{L\,(1+\eta^{S/2})}\right\} \tag{5.31}$$

satisfying (5.25)*.*

We also show that for the trust-region technique, at one main iteration, the number of times we solve the subproblem (5.7) until a step is accepted is upper-bounded by a constant.

Lemma 5.3. *For DPLBFGS-TR, suppose each time when we solve* (5.7) *we have guarantee that the objective value is no worse than* $Q(0)$*. Then when* (5.26) *is satisfied, we have that*

$$\|H_t\| \leq c_1 \max\left\{1, \frac{L}{c_2\theta}\right\}. \tag{5.32}$$

Moreover, at each main iteration, the number of times we solve (5.7) *with different* H *is upper-bounded by*

$$\max\left\{1, \left\lceil \log_\theta \frac{c_2}{L} \right\rceil\right\}$$

Note that the bound in Lemma 5.3 is independent to the number of SpaRSA iterations used. It is possible that one can incorporate the subproblem suboptimality to derive tighter but more complicated bounds, but for simplicity we use the current form of Lemma 5.3.

The results in Lemmas 5.2-5.3 are just worst-case guarantees; in practice we often observe that the line search procedure terminates with $\lambda = 1$ for our original choice of H, as we see in our experiments. This also indicates that in most of the cases, (5.26) is satisfied with the original LBFGS Hessian approximation without scaling H.

We now lay out the main theoretical results in Theorems 5.4 to 5.7, which describe the iteration and communication complexity under different conditions on the function F. In all these results, we assume the following setting:

> We apply DPLBFGS to solve the main problem (5.11), running Algorithm 8 for S iterations in each main iteration. Let x^t, λ_t, and H_t be respectively the x vector, the step size, and the final accepted quadratic approximation matrix at the tth iteration of DPLBFGS for all $t \geq 0$. Let M be the supremum of $\|H_t\|$ for all t (which is either c_1 or $c_1 L/(c_2\theta)$ according to Lemmas 5.1 and 5.3), and $\bar{\lambda}$ be the infimum of the step sizes over iterations (either 1 or the bound from Lemma 5.2). Let F^* be the optimal objective value of (5.11), Ω the solution set, and P_Ω the (convex) projection onto Ω.

Theorem 5.4. *If F is convex, given an initial point x^0, assume*

$$R_0 := \sup_{x:F(x)\leq F(x^0)} \|x - P_\Omega(x)\| \tag{5.33}$$

is finite, we obtain the following expressions for rate of convergence of the objective value.

1. *When*

$$F(x^t) - F^* \geq \left(x^t - P_\Omega\left(x^t\right)\right)^\top H_t \left(x^t - P_\Omega\left(x^t\right)\right),$$

the objective converges linearly to the optimum:

$$\frac{F(x^{t+1}) - F^*}{F(x_t)^- F^*} \leq 1 - \frac{\left(1 - \eta^S\right) \sigma_1 \lambda_t}{2}.$$

2. *For any $t \geq t_0$, where*

$$t_0 \coloneqq \arg\min\{t \mid MR_0^2 > F\left(x^t\right) - F^*\},$$

we have

$$F\left(x^t\right) - F^* \leq \frac{2MR_0^2}{\sigma_1(1-\eta^S)\sum_{i=t_0}^{t-1} \lambda_t + 2}.$$

Moreover,

$$t_0 \leq \max\left\{0, 1 + \frac{2}{\sigma_1(1-\eta^s)\bar{\lambda}} \log \frac{f\left(x^0\right) - f^*}{MR_0^2}\right\}.$$

Therefore, for any $\epsilon > 0$, the number of rounds of $O(d)$ communication required to obtain an x^t such that $F(x^t) - F^* \leq \epsilon$ is at most

$$\begin{cases} O\left(\max\left\{0, 1 + \frac{2}{\sigma_1(1-\eta^s)\bar{\lambda}} \log \frac{F\left(x^0\right) - F^*}{MR_0^2}\right\} + \frac{2MR_0^2}{\sigma_1 \bar{\lambda}(1-\eta^S)\epsilon}\right) & \text{if } \epsilon < MR_0^2, \\ O\left(\max\left\{0, 1 + \frac{2}{\sigma_1(1-\eta^S)\bar{\lambda}} \log \frac{F\left(x^0\right) - F^*}{\epsilon}\right\}\right) & \text{else.} \end{cases}$$

Theorem 5.5. *When F is convex and the quadratic growth condition*

$$F\left(x\right) - F^* \geq \frac{\mu}{2} \left\|x - P_\Omega\left(x\right)\right\|^2, \quad \forall x \in \mathbb{R}^N \tag{5.34}$$

holds for some $\mu > 0$, we get a global Q-linear convergence rate:

$$\frac{F\left(x^{t+1}\right) - F^*}{F\left(x^t\right) - F^*} \leq 1 - \lambda_t \sigma_1 \left(1 - \eta^S\right) \cdot \begin{cases} \frac{\mu}{4\|H_t\|}, & \text{if } \mu \leq 2\|H_t\|, \\ 1 - \frac{\|H_t\|}{\mu}, & \text{else.} \end{cases} \tag{5.35}$$

Therefore, the rounds of $O(d)$ communication needed for getting an ϵ-accurate objective is

$$\begin{cases} O\left(\max\left\{0, 1+\frac{2}{\sigma_1(1-\eta^s)\bar{\lambda}}\log\frac{F\left(x^0\right)-F^*}{MR_0^2}\right\}+\frac{4M}{\mu\bar{\lambda}\sigma_1(1-\eta^S)}\log\frac{MR_0^2}{\epsilon}\right) & \text{if } \epsilon < MR_0^2, \mu \leq 2M, \\ O\left(\max\left\{0, 1+\frac{2}{\sigma_1(1-\eta^s)\bar{\lambda}}\log\frac{F\left(x^0\right)-F^*}{MR_0^2}\right\}+\frac{\mu}{(\mu-M)\bar{\lambda}\sigma_1(1-\eta^S)}\log\frac{MR_0^2}{\epsilon}\right) & \text{if } \epsilon < MR_0^2, \mu > 2M, \\ O\left(0, 1+\frac{2}{\sigma_1(1-\eta^S)\bar{\lambda}}\log\frac{F\left(x^0\right)-F^*}{\epsilon}\right) & \text{if } \epsilon \geq MR_0^2. \end{cases}$$

Theorem 5.6. *Suppose that the following relaxation of strong convexity holds: There exists $\mu > 0$ such that for any $x \in \mathbb{R}^N$ and any $a \in [0,1]$, we have*

$$F\left(ax+(1-a)P_\Omega\left(x\right)\right) \leq aF\left(x\right)+(1-a)F^*-\frac{\mu a\left(1-a\right)}{2}\left\|x-P_\Omega\left(x\right)\right\|^2. \tag{5.36}$$

Then DPLBFGS converges globally at a Q-linear rate faster than (5.35). More specifically,

$$\frac{F\left(x^{t+1}\right)-F^*}{F\left(x^t\right)-F^*} \leq 1-\frac{\lambda_t\sigma_1\left(1-\eta^S\right)\mu}{\mu+\left\|H_t\right\|}.$$

Therefore, to get an approximate solution of (5.11) that is ϵ-accurate in the sense of objective value, we need to perform at most

$$\begin{cases} O\left(\max\left\{0, 1+\frac{2}{\sigma_1(1-\eta^s)\bar{\lambda}}\log\frac{F\left(x^0\right)-F^*}{MR_0^2}\right\}+\frac{\mu+M}{\mu\sigma_1\bar{\lambda}(1-\eta^S)}\log\frac{MR_0^2}{\epsilon}\right) & \text{if } \epsilon < MR_0^2, \\ O\left(0, 1+\frac{2}{\sigma_1(1-\eta^S)\bar{\lambda}}\log\frac{F\left(x^0\right)-F^*}{\epsilon}\right) & \text{else.} \end{cases}$$

rounds of $O(d)$ communication.

Theorem 5.7. *If F is non-convex, the norm of*

$$G_t \coloneqq \arg\min_p \quad \nabla f\left(x^t\right)^\top p+\frac{\|p\|^2}{2}+\Psi\left(x+p\right)$$

converges to zero at a rate of $O(1/\sqrt{t})$ *in the following sense:*

$$\min_{0\leq i\leq t} \|G_i\|^2 \leq \frac{F(x^0) - F^*}{\sigma_1 (t+1)} \frac{M^2 \left(1 + \frac{1}{c_2} + \sqrt{1 - \frac{2}{M} + \frac{1}{c_2^2}}\right)^2}{2c_2(1-\eta^S) \min_{0\leq i\leq t} \lambda_i}.$$

Moreover, if there are limit points in the sequence $\{x^0, x^1, \dots\}$*, then all limit points are stationary.*

Note that it is known that the norm of G_t is zero if and only if x^t is a stationary point, so this measure serves as an indicator for the first-order optimality condition. The class of quadratic growth (5.34) includes many non-strongly-convex ERM problems. Especially, it contains problems of the form

$$\min_{x\in\mathcal{X}} \quad g(Ax) + b^\top x, \tag{5.37}$$

where g is strongly convex, A is a matrix, b is a vector, and $\mathcal{X}$ is a polyhedron. Commonly seen non-strongly-convex ERM problems including ℓ_1-regularized logistic regression, LASSO, and the dual problem of support vector machines all fall in the form (5.37) and therefore our algorithm enjoys global linear convergence on them.

5.4 Solving the Primal and the Dual Problem

Now we discuss details on how to apply DPLBFGS described in the previous section to the specific problems (5.1) and (5.2) respectively. We discuss how to obtain the gradient of the smooth part f and how to conduct line search efficiently under distributed data storage. For the dual problem, we additionally describe how to recover a primal solution from our dual iterates.

Primal Problem

Recall that the primal problem is (5.1) $\min_{\boldsymbol{w}\in\mathbb{R}^d} \xi\left(X^\top \boldsymbol{w}\right) + g\left(\boldsymbol{w}\right)$, and is obtained from the general form (5.11) by having $N = d$, $x = \boldsymbol{w}$, $f(\cdot) = \xi(X^\top \cdot)$, and $\Psi(\cdot) = g(\cdot)$. The gradient of ξ with respect to $\boldsymbol{w}$ is

$$X\nabla\xi(X^\top \boldsymbol{w}) = \sum_{k=1}^{K} \left(X_k \nabla\xi_k(X_k^\top \boldsymbol{w})\right).$$

We see that, except for the sum over k, the computation can be conducted locally provided $\boldsymbol{w}$ is available to all machines. Our algorithm maintains $X_k^\top \boldsymbol{w}$ on the kth machine throughout, and the most costly steps are the matrix-vector multiplications between X_k and $\nabla\xi_k(X_k^\top \boldsymbol{w})$. Clearly, computing $X_k^\top \boldsymbol{w}$ and $X_k\nabla\xi_k(X_k^\top \boldsymbol{w})$ both cost $O(\#\text{nnz}/K)$ in average among the K machines. The local d-dimensional partial gradients are then aggregated through an allreduce operation using a round of $O(d)$ communication.

To initialize the approximate Hessian matrix H at $t = 0$, we set $H_0 \coloneqq a_0 I$ for some positive scalar a_0. In particular, we use

$$a_0 \coloneqq \frac{\left|\nabla f(\boldsymbol{w}_0)^\top \nabla^2 f(\boldsymbol{w}_0) \nabla f(\boldsymbol{w}_0)\right|}{\|\nabla f(\boldsymbol{w}_0)\|^2}, \tag{5.38}$$

where $\nabla^2 f(\boldsymbol{w}_0)$ denotes the generalized Hessian when f is not twice-differentiable.

For the function value evaluation part of line search, each machine will compute $\xi_k(X_k^\top \boldsymbol{w} + \lambda X_k^\top) + g_k(\boldsymbol{w}_{\mathcal{I}_k^g} + \lambda p_{\mathcal{I}_k^g})$ (the left-hand side of (5.25)) and send this scalar over the network. Once we have precomputed $X_k^\top \boldsymbol{w}$ and $X_k^\top p$, we can quickly obtain $X_k^\top(\boldsymbol{w} + \lambda p)$ for any value of λ without having to performing matrix-vector multiplications. Aside from the communication needed to compute the summation of the f_k terms in the evaluation of f, the only other communication needed is to share

the update direction p from subvectors $p_{\mathcal{I}_k^g}$. Thus, two rounds of $O(d)$ communication are incurred per main iteration.

Dual Problem

Now consider applying DPLBFGS to the dual problem (5.2). To fit it into the general form (5.11), we have $N = n$, $x = \boldsymbol{\alpha}$, $f(\cdot) = g^*(X\cdot)$, and $\Psi(\cdot) = -\xi^*(-\cdot)$. In this case, we need a way to efficiently obtain the vector

$$\boldsymbol{z} \coloneqq X\boldsymbol{\alpha}$$

on each machine in order to compute $g^*(X\boldsymbol{\alpha})$ and the gradient $X^\top \nabla g^*(X\boldsymbol{\alpha})$.

Since each machine has access to some columns of X, it is natural to split $\boldsymbol{\alpha}$ according to the same partition. Namely, we set $\mathcal{I}_k$ as described in (5.12) to $\mathcal{I}_k^X$. Every machine can then individually compute $X_k\boldsymbol{\alpha}_k$, and after one round of $O(d)$ communication, each machine has a copy of $\boldsymbol{z} = X\boldsymbol{\alpha} = \sum_{k=1}^K X_k\boldsymbol{\alpha}_k$. After using $\boldsymbol{z}$ to compute $\nabla_{\boldsymbol{z}} g^*(\boldsymbol{z})$, we can compute the gradient $\nabla_{\mathcal{I}_k^X} g^*(X\boldsymbol{\alpha}) = X_k^\top \nabla g^*(X\boldsymbol{\alpha})$ at a computation cost of $O(\#\text{nnz}/K)$ in average among the K machines, matching the cost of computing $X_k\boldsymbol{\alpha}_k$ earlier.

To construct the approximation matrix H_0 for the first main iteration, we make use of the fact that the (generalized) Hessian of $g^*(X\boldsymbol{\alpha})$ is

$$X^\top \nabla^2 g^*(\boldsymbol{z}) X. \tag{5.39}$$

Each machine has access to one X_k, so we can construct the block-diagonal proportion of this Hessian locally for the part corresponding to $\mathcal{I}_k^X$. Therefore, the block-diagonal part of the Hessian is a natural choice for H_0. Under this choice of H_0, the subproblem (5.7) is decomposable along the $\mathcal{I}_1^X, \ldots, \mathcal{I}_K^X$ partition and one can apply algorithms other than SpaRSA to solve this. For example, we can apply CD solvers on the independent local

subproblems, as done by Lee and Chang (2017); Yang (2013); Zheng et al. (2017). As it is observed in these works that the block-diagonal approaches tend to converge fast at the early iterations, we use it for initializing our algorithm. In particular, we start with the block-diagonal approach, until U_t has $2m$ columns, and then we switch to the LBFGS approach. This turns out to be much more efficient in practice than starting with the LBFGS matrix.

For the line search process, we can precompute the matrix-vector product Xp with the same $O(d)$ communication and $O(\#\text{nnz}/K)$ per machine average computational cost as computing $X\boldsymbol{\alpha}$. With $X\boldsymbol{\alpha}$ and Xp, we can now evaluate $X\boldsymbol{\alpha} + \lambda Xp$ quickly for different λ, instead of having to perform a matrix-vector multiplication of the form $X(\boldsymbol{\alpha} + \lambda p)$ for every λ. For most common choices of g, given $\boldsymbol{z}$, the computational cost of evaluating $g^*(\boldsymbol{z})$ is $O(d)$. Thus, the cost of this efficient implementation per backtracking iteration is reduced to $O(d)$, with an overhead of $O(\#\text{nnz}/K)$ per machine average per main iteration, while the naive implementation takes $O(\#\text{nnz}/K)$ per backtracking iteration. After the sufficient decrease condition holds, we locally update $\boldsymbol{\alpha}_k$ and $X\boldsymbol{\alpha}$ using $p_{\mathcal{I}_k^X}$ and Xp. For the trust region approach, the two implementations take the same cost.

Recovering a Primal Solution

In general, the algorithm only gives us an approximate solution to the dual problem (5.2), which means the formula

$$\boldsymbol{w}(\boldsymbol{\alpha}) \coloneqq \nabla g^*(X\boldsymbol{\alpha}). \tag{5.40}$$

used to obtain a primal optimal point from a dual optimal point (equation (5.10), derived from KKT conditions) is no longer guaranteed to even return a feasible point without further assumptions. Nonetheless, this is a common approach and under certain conditions (the ones we used in

Assumption 4), one can provide guarantees on the resulting point.

It can be shown from existing works (Bach, 2015; Shalev-Shwartz and Zhang, 2012) that when $\boldsymbol{\alpha}$ is not an optimum for (5.2), for (5.40), certain levels of primal suboptimality can be achieved, which depend on whether ξ is Lipschitz-continuously differentiable or Lipschitz continuous. This is the reason why we need the corresponding assumptions in Assumption 4. A summary of those results is available in Lee and Chang (2017). We restate their results here for completeness but omit the proof.

Theorem 5.8 (Lee and Chang (2017, Theorem 3))**.** *Given any $\epsilon > 0$, and any dual iterate $\boldsymbol{\alpha} \in \mathbb{R}^n$ satisfying*

$$D(\boldsymbol{\alpha}) - \min_{\bar{\boldsymbol{\alpha}} \in \mathbb{R}^n} \quad D(\bar{\boldsymbol{\alpha}}) \leq \epsilon.$$

If Assumption 4 holds, then the following results hold.

1. *If the part in Assumption 4 that ξ^* is σ-strongly convex holds, then $\boldsymbol{w}(\boldsymbol{\alpha})$ satisfies*
$$P\left(\boldsymbol{w}\left(\boldsymbol{\alpha}\right)\right) - \min_{\boldsymbol{w} \in \mathbb{R}^d} \quad P\left(\boldsymbol{w}\right) \leq \epsilon \left(1 + \frac{L}{\sigma}\right).$$
2. *If the part in Assumption 4 that ξ is ρ-Lipschitz continuous holds, then $\boldsymbol{w}(\boldsymbol{\alpha})$ satisfies*
$$P\left(\boldsymbol{w}\left(\boldsymbol{\alpha}\right)\right) - \min_{\boldsymbol{w} \in \mathbb{R}^d} \quad P\left(\boldsymbol{w}\right) \leq \min\left\{4\rho^2 L, \sqrt{8\epsilon\rho^2 L}\right\}.$$

One more issue to note from recovering the primal solution through (5.40) is that our algorithm only guarantees monotone decrease of the dual objective but not the primal objective. To ensure the best primal approximate solution, one can follow Lee and Chang (2017) to maintain the primal iterate that gives the best objective for (5.1) up to the current iteration as the output solution. The theorems above still apply to this iterate and we are guaranteed to have better primal performance.

5.5 Related Works

The framework of using the quadratic approximation subproblem (5.7) to generate update directions for optimizing (5.11) has been discussed in existing works with different choices of H, but always in the single-core setting. Lee et al. (2014) focused on using $H = \nabla^2 f$, and proved local convergence results under certain additional assumptions. In their experiment, they used AG to solve (5.7). However, in distributed environments, for (5.1) or (5.2), using $\nabla^2 f$ as H needs an $O(d)$ communication per AG iteration in solving (5.7), because computation of the term $\nabla^2 f(x)p$ involves either $XDX^\top p$ or $X^\top DXp$ for some diagonal matrix D, which requires one *allreduce* operation to calculate a weighted sum of the columns of X.

Scheinberg and Tang (2016) and Ghanbari and Scheinberg (2018) showed global convergence rate results for a method based on (5.7) with bounded H, and suggested using randomized coordinate descent to solve (5.7). In the experiments of these two works, they used the same choice of H as we do in this chapter, with CD as the solver for (5.7), which is well suited to their single-machine setting. Aside from our extension to the distributed setting and the use of SpaRSA, the third major difference between their algorithm and ours is how sufficient objective decrease is guaranteed. When the obtained solution with a unit step size does not result in sufficient objective value decrease, they add a multiple of the identity matrix to H and solve (5.7) again starting from $p^{(0)} = 0$. This is different from how we modify H and in some worst cases, the behavior of their algorithm can be closer to a first-order method if the identity part dominates, and more trials of different H might be needed. The cost of repeatedly solving (5.7) from scratch can be high, which results in an algorithm with higher overall complexity. This potential inefficiency is exacerbated further by the inefficiency of coordinate descent in the distributed setting.

For the dual problem (5.2), there are existing distributed algorithms under the instance-wise storage scheme (for example, Yang (2013); Lee and

Chang (2017); Zheng et al. (2017); Dünner et al. (2018) and the references therein). As we discussed in Section 5.4, it is easy to recover the block-diagonal part of the Hessian (5.39) under this storage scheme. Therefore, these works focus on using the block-diagonal part of the Hessian and use (5.7) to generate update directions. In this case, only blockwise curvature information is obtained, so the update direction can be poor if the data is distributed nonuniformly. In the extreme case in which each machine contains only one column of X, only the diagonal entries of the Hessian can be obtained, so the method reduces to a scaled version of proximal gradient. Indeed, we often observe in practice that these methods tend to converge quickly in the beginning, but after a while the progress appears to stagnate even for small K.

Zheng et al. (2017) give a primal-dual framework with acceleration that utilizes a distributed solver for (5.2) to optimize (5.1). Their algorithm is essentially the same as applying the Catalyst framework (Lin et al., 2018) on a strongly-convex primal problem to form an algorithm with an inner and an outer loop. In particular, their approach consists of the following steps per round to optimize a strongly-convex primal problem with the additional requirement that g being Lipschitz-continuously differentiable.

1. Add a quadratic term centered at a given point y to form a subproblem with better condition.
2. Approximately optimize the new problem by using a distributed dual problem solver, and
3. find the next y through extrapolation techniques similar to that of accelerated gradient (Nesterov, 2013; Beck and Teboulle, 2009).

A more detailed description of the Catalyst framework (without requiring both terms to be differentiable) is given in Appendix 5.B. We consider one round of the above process as one outer iteration of their algorithm, and the inner loop refers to the optimization process in the second step. The outer loop of their algorithm is conducted on the primal problem (5.1)

and a distributed dual solver is simply considered as a subproblem solver using results similar to Theorem 5.8. Therefore this approach is more a primal problem solver than a dual one, and it should be compared with other distributed primal solvers for smooth optimization but not with the dual algorithms. However, the Catalyst framework can be applied directly on the dual problem directly as well, and this type of acceleration can to some extent deal with the problem of stagnant convergence appeared in the block-diagonal approaches for the dual problem. Unfortunately, those parameters used in acceleration are not just global in the sense that the coordinate blocks are considered all together, but also global bounds for all possible $\boldsymbol{w} \in \mathbb{R}^d$ or $\boldsymbol{\alpha} \in \mathbb{R}^n$. This means that the curvature information around the current iterate is not considered, so the improved convergence can still be slow. By using the Hessian or its approximation as in our method, we can get much better empirical convergence.

A column-wise split of X in the dual problem (5.2) corresponds to a primal problem (5.1) where X is split row-wise. Therefore, existing distributed algorithms for the dual ERM problem (5.2) can be directly used to solve (5.1) in a distributed environment where X is partitioned feature-wise (i.e. along rows instead of columns). However, there are two potential disadvantages of this approach. First, new data points can easily be assigned to one of the machines in our approach, whereas in the feature-wise approach, the features of all new points would need to be distributed around the machines. Second, as we mentioned above, the update direction from the block-diagonal approximation of the Hessian can be poor if the data is distributed nonuniformly across machines, and data is more likely to be distributed evenly across instances than across features. Thus, those algorithms focusing on feature-wise split of X are excluded from our discussion and empirical comparison.

5.6 Numerical Experiments

We investigate the empirical performance of DPLBFGS for solving both the primal and dual problems (5.1) and (5.2) on binary classification problems with training data points $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ for $i = 1, \ldots, n$. For the primal problem, we consider solving ℓ_1-regularized logistic regression problems:

$$P(\boldsymbol{w}) = C \sum_{i=1}^{n} \log\left(1 + e^{-y_i \boldsymbol{x}_i^\top \boldsymbol{w}}\right) + \|\boldsymbol{w}\|_1, \tag{5.41}$$

where $C > 0$ is a parameter prespecified to trade-off between the loss term and the regularization term. Note that since the logarithm term is nonnegative, the regularization term ensures that the level set is bounded. Therefore, within the bounded set, the loss function is strongly convex with respect to $X^\top \boldsymbol{w}$ and the regularizer can be reformulated as a polyhedron constrained linear term. One can thus easily show that (5.41) satisfies the quadratic growth condition (5.34). Therefore, our algorithm enjoys global linear convergence on this problem.

For the dual problem, we consider ℓ_2-regularized squared-hinge loss problems, which is of the form

$$D(\boldsymbol{\alpha}) = \frac{1}{2} \|YX\boldsymbol{\alpha}\|_2^2 + \frac{1}{4C} \|\boldsymbol{\alpha}\|_2^2 - \mathbf{1}^\top \boldsymbol{\alpha} + \mathbb{1}_{\mathbb{R}^n_+}(\boldsymbol{\alpha}), \tag{5.42}$$

where Y is the diagonal matrix consists of the labels y_i, $\mathbf{1} = (1, \ldots, 1)$ is the vector of ones, given a convex set $\mathbf{X}$, $\mathbb{1}_{\mathbf{X}}$ is its indicator function such that

$$\mathbb{1}_{\mathbf{X}}(x) = \begin{cases} 0 & \text{if } x \in X, \\ \infty & \text{else}, \end{cases}$$

and $\mathbb{R}^n_+$ is the nonnegative orthant in $\mathbb{R}^n$. This strongly convex quadratic problem is considered for easier implementation of the Catalyst framework in comparison.

We consider the publicly available binary classification data sets listed

in Table 5.1,[1] and partitioned the instances evenly across machines. C is fixed to 1 in all our experiments for simplicity. We ran our experiments on a local cluster of 16 machines running MPICH2, and all algorithms are implemented in C/C++. The inversion of M defined in (5.15) is performed through LAPACK (Anderson et al., 1999). The comparison criteria are the relative objective error

$$\left|\frac{F(x) - F^*}{F^*}\right|$$

versus either the amount communicated (divided by d) or the overall running time, where F^* is the optimal objective value, and F can be either the primal objective $P(\boldsymbol{w})$ or the dual objective $D(\boldsymbol{\alpha})$, depending on which problem is being considered. The former criterion is useful in estimating the performance in environments in which communication cost is extremely high.

The parameters of our algorithm were set as follows: $\theta = 0.5$, $\beta = 2$, $\sigma_0 = 10^{-2}$, $\sigma_1 = 10^{-4}$, $m = 10$, $\delta = 10^{-10}$. The parameters in SpaRSA follow the setting in Wright et al. (2009), θ is set to halve the step size each time, the value of σ_0 follows the default experimental setting of Lee et al. (2017), δ is set to a small enough value, and $m = 10$ is a common choice for LBFGS. The code used in our experiments is available at `http://github.com/leepei/dplbfgs/`.

In all experiments, we show results of the backtracking variant only, as we do not observe significant difference in performance between the line-search approach and the trust-region approach in our algorithm.

In the subsequent experiments, we first use the primal problem (5.41) to examine how inexactness of the subproblem solution affects the communication complexity, overall running time, and step sizes. We then compare our algorithm with state of the art distributed solvers for (5.41). Finally, comparison on the dual problem (5.42) is conducted.

[1] Downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Table 5.1: Data statistics.

Data set	n (#instances)	d (#features)	#nonzeros
news	19,996	1,355,191	9,097,916
epsilon	400,000	2,000	800,000,000
webspam	350,000	16,609,143	1,304,697,446
avazu-site	25,832,830	999,962	387,492,144

Effect of Inexactness in the Subproblem Solution

We first examine how the degree of inexactness of the approximate solution of subproblems (5.7) affects the convergence of the overall algorithm. Instead of treating SpaRSA as a steadily linearly converging algorithm, we take it as an algorithm that sometimes decreases the objective much faster than the worst-case guarantee, thus an adaptive stopping condition is used. In particular, we terminate Algorithm 8 when the norm of the current update step is smaller than ϵ_1 times that of the first update step, for some prespecified $\epsilon_1 > 0$. From the proof of Lemma 5.1, the norm of the update step bounds the value of $Q(p) - Q^*$ both from above and from below (assuming exact solution of (5.20), which is indeed the case for the selected problems), and thus serves as a good measure of the solution precision. In Table 5.2, we compare runs with the values $\epsilon_1 = 10^{-1}, 10^{-2}, 10^{-3}$. For the datasets news20 and webspam, it is as expected that tighter solution of (5.7) results in better updates and hence lower communication cost, though it may not result in a shorter convergence time because of more computation per round. As for the dataset epsilon, which has a smaller data dimension d, the $O(m)$ communication cost per SpaRSA iteration for calculating $\nabla \hat{f}$ is significant in comparison. In this case, setting a tighter stopping criterion for SpaRSA can incur higher communication cost and longer running time.

In Table 5.3, we show the distribution of the step sizes over the main iterations, for the same set of values of ϵ_1. As we discussed in Section 5.3,

Table 5.2: Different stopping conditions of SpaRSA as an approximate solver for (5.7). We show required amount of communication (divided by d) and running time (in seconds) to reach $F(\boldsymbol{w}) - F^* \leq 10^{-3}F^*$.

Data set	ϵ_1	Communication	Time
news20	10^{-1}	28	11
	10^{-2}	25	11
	10^{-3}	23	14
epsilon	10^{-1}	144	45
	10^{-2}	357	61
	10^{-3}	687	60
webspam	10^{-1}	452	3254
	10^{-2}	273	1814
	10^{-3}	249	1419

Table 5.3: Step size distributions.

Data set	ϵ_1	percent of $\lambda = 1$	smallest λ
news20	10^{-1}	95.5%	2^{-3}
	10^{-2}	95.5%	2^{-4}
	10^{-3}	95.5%	2^{-3}
epsilon	10^{-1}	96.8%	2^{-5}
	10^{-2}	93.4%	2^{-6}
	10^{-3}	91.2%	2^{-3}
webspam	10^{-1}	98.5%	2^{-3}
	10^{-2}	97.6%	2^{-2}
	10^{-3}	97.2%	2^{-2}

although the smallest λ can be much smaller than one, the unit step is usually accepted. Therefore, although the worst-case communication complexity analysis is dominated by the smallest step encountered, the practical behavior is much better. This result also suggests that the difference between DPLBFGS-LS and DPLBFGS-TR should be negligible, as most of the times, the original H with unit step size is accepted.

Comparison with Other Methods for the Primal Problem

Now we compare our method with two state-of-the-art distributed algorithms for (5.11). In addition to a proximal-gradient-type method that can be used to solve general (5.11) in distributed environments easily, we also include one solver specifically designed for ℓ_1-regularized problems in our comparison. These methods are:

- DPLBFGS-LS: our Distributed Proximal LBFGS approach. We fix $\epsilon_1 = 10^{-2}$.
- SpaRSA (Wright et al., 2009): the method described in Section 5.2, but applied directly to (5.1) but not to the subproblem (5.7).
- OWLQN (Andrew and Gao, 2007): an orthant-wise quasi-Newton method specifically designed for ℓ_1-regularized problems. We fix $m = 10$ in the LBFGS approximation.

All methods are implemented in C/C++ and MPI. As OWLQN does not update the coordinates i such that $-X_{i,:}\nabla\xi(X^T\boldsymbol{w}) \in \partial g_i(\boldsymbol{w}_i)$ given any $\boldsymbol{w}$, the same preliminary active set selection is applied to our algorithm to reduce the subproblem dimension and the computational cost, but note that this does not reduce the communication cost as the gradient calculation still requires communication of a full d-dimensional vector.

The AG method (Nesterov, 2013) can be an alternative to SpaRSA, but its empirical performance has been shown to be similar to SpaRSA (Yang and Zhang, 2011) and it requires strong convexity and Lipschitz parameters to be estimated, which induces an additional cost.

A further examination on different values of m indicates that convergence speed of our method improves with larger m, while in OWLQN, larger m usually does not lead to better results. We use the same value of m for both methods and choose a value that favors OWLQN.

The results are provided in Figure 5.1. Our method is always the fastest in both criteria. For epsilon, our method is orders of magnitude faster, showing that correctly using the curvature information of the smooth part is indeed beneficial in reducing the communication complexity.

Comparison on the Dual Problem

Now we turn to solve the dual problem, considering the specific example (5.42). We compare the following algorithms.

- BDA (Lee and Chang, 2017): a distributed algorithm using Block-Diagonal Approximation of the real Hessian of the smooth part with line search.
- BDA with Catalyst: using the BDA algorithm within the Catalyst framework (Lin et al., 2018) for accelerating first-order methods.
- ADN (Dünner et al., 2018): a trust-region approach where the quadratic term is a multiple of the block-diagonal part of the Hessian, scaled adaptively as the algorithm progresses.
- DPLBFGS-LS: our Distributed Proximal LBFGS approach. We fix $\epsilon_1 = 10^{-2}$ and limit the number of SpaRSA iterations to 100. For the first ten iterations when $m(t) < m$, we use BDA to generate the update steps instead.

For BDA, we use the C/C++ implementation in the package MPI-LIBLINEAR.[2] We implement ADN by modifying the above implementation of BDA. In both BDA and ADN, following Lee and Chang (2017) we use random-permutation coordinate descent (RPCD) for the local subproblems, and for each outer iteration we perform one epoch of RPCD. For the line search step in both BDA and DPLBFGS-LS, since the objective (5.42) is quadratic,

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/distributed-liblinear/.

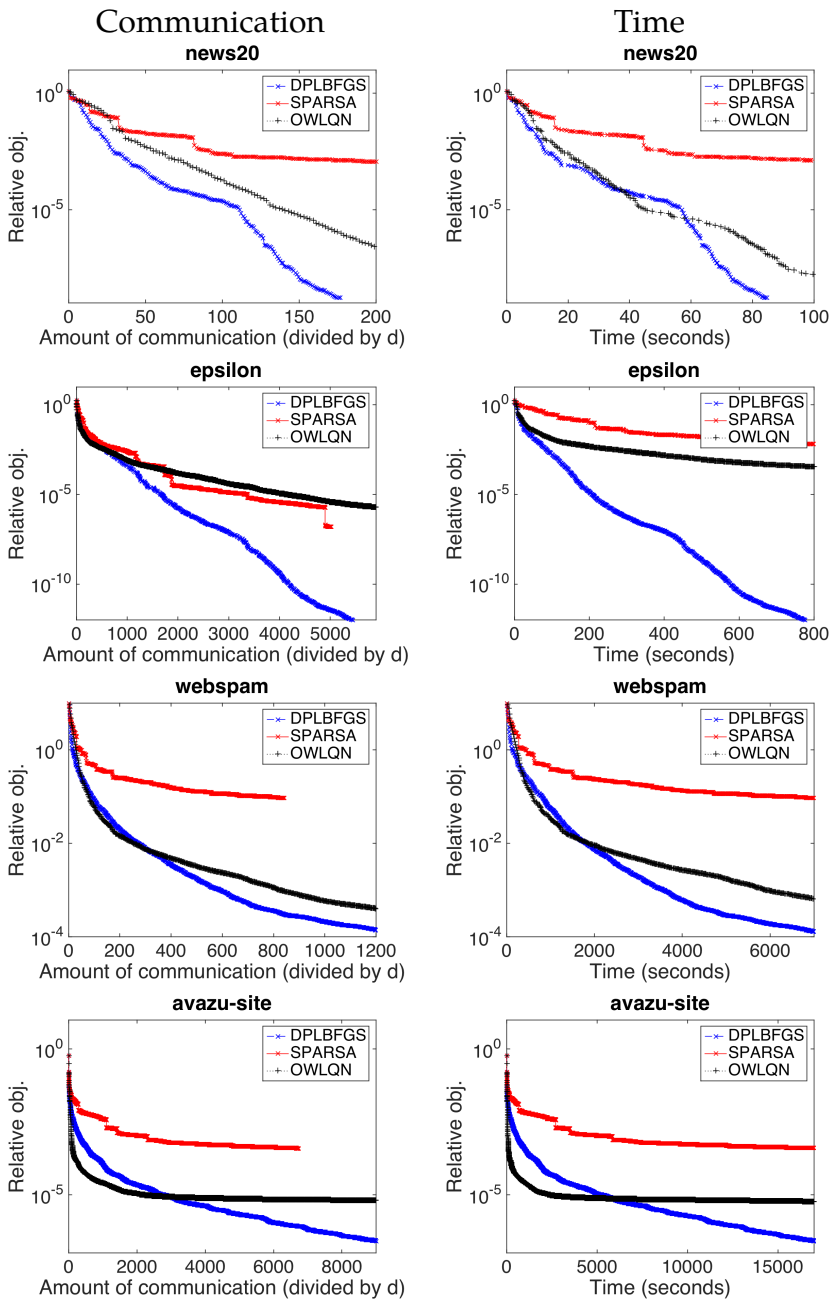


Figure 5.1: Comparison between different methods for (5.41) in terms of relative objective difference to the optimum. Left: communication (divided by d); right: running time (in seconds).

we can find the exact minimizer efficiently (in closed form). The convergence guarantees still holds for exact line search, so we use this here in place of the backtracking approach described earlier.

We also applied the Catalyst framework (Lin et al., 2018) for accelerating first-order methods to BDA to tackle the dual problem, especially for dealing with the stagnant convergence issue. This framework requires a good estimate of the convergence rate and the strong convexity parameter σ. From (5.42), we know that $\sigma = 1/(2C)$, but the actual convergence rate is hard to estimate as BDA interpolates between (stochastic) proximal coordinate descent (when only one machine is used) and proximal gradient (when n machines are used). After experimenting with different sets of parameters for BDA with Catalyst, we found the following to work most effectively: for every outer iteration of the Catalyst framework, K iterations of BDA is conducted with early termination if a negative step size is obtained from exact line search; for the next Catalyst iteration, the warm-start initial point is simply the iterate at the end of the previous Catalyst iteration; before starting Catalyst, we run the unaccelerated version of BDA for certain iterations to utilize its advantage of fast early convergence. Unfortunately, we do not find a good way to estimate the κ term in the Catalyst framework that works for all data sets. Therefore, we find the best κ by a grid search. We provide a detailed description of our implementation of the Catalyst framework on this problem and the related parameters used in this experiment in Appendix 5.B.

We focus on the combination of Catalyst and BDA (instead of with ADN) for a few reasons. Since both BDA and ADN are distributed methods that use the block-diagonal portion of the Hessian matrix, it should suffice to evaluate the application of Catalyst to the better performing of the two to represent this class of algorithms. In addition, dealing with the trust-region adjustment of ADN becomes complicated as the problem changes through the Catalyst iterations.

The results are shown in Figure 5.2. We do not present results on the avazu data set in this experiment as all methods take extremely long time to converge. We first observe that, contrary to what is claimed in Dünner et al. (2018), BDA outperforms ADN on news20 and webspam, though the difference is insignificant, and the two are competitive on epsilon. This also justifies that applying the Catalyst framework on BDA alone suffices. Comparing our DPLBFGS approach to the block-diagonal ones, it is clear that our method performs magnitudes better than the state of the art in terms of both communication cost and time. For webspam and epsilon, the block-diagonal approaches are faster at first, but the progress stalls after a certain accuracy level. In contrast, while the proposed DPLBFGS approach does not converge as rapidly initially, the algorithm consistently makes progress towards a high accuracy solution.

As the purpose of solving the dual problem is to obtain an approximate solution to the primal problem through the formulation (5.40), we are interested on how the methods compare in terms of the primal solution precision. This comparison is presented in Figure 5.3. Since these dual methods are not descent methods for the primal problem, we apply the pocket approach (Gallant, 1990) suggested in Lee and Chang (2017) to use the iterate with the smallest primal objective so far as the current primal solution. We see that the primal objective values have trends very similar to the dual counterparts, showing that our DPLBFGS method is also superior at generating better primal solutions.

A potentially more effective approach is a hybrid one that first uses a block-diagonal method and then switches over to our DPLBFGS approach after the block-diagonal method hits the slow convergence phase. Developing such an algorithm would require a way to determine when we reach such a stage, and we leave the development of this method to future work. Another possibility is to consider a structured quasi-Newton approach to construct a Hessian approximation only for the off-block-diagonal part so

that the block-diagonal part can be utilized simultaneously.

We also remark that our algorithm is partition-invariant in terms of convergence and communication cost, while the convergence behavior of the block-diagonal approaches depend heavily on the partition. This means when more machines are used, these block-diagonal approaches suffer from poorer convergence, while our method retains the same efficiency regardless of the number of machines begin used and how the data points are distributed (except for the initialization part).

5.7 Conclusions

In this chapter, we propose a practical and communication-efficient distributed algorithm for solving general regularized nonsmooth ERM problems. The proposed approach is the first one that can be applied both to the primal and the dual ERM problem under the instance-wise split scheme. Our algorithm enjoys fast performance both theoretically and empirically and can be applied to a wide range of ERM problems.

Appendix

5.A Proofs

In this appendix, we provide proof for Lemma 5.1. The rest of Section 5.3 directly follows the results in Chapter 2 and Peng et al. (2018) and are therefore omitted. Note that (5.36) implies (5.34), and (5.34) implies (5.33) because R_0^2 is upper-bounded by $2(F(x^0) - F^*)/\mu$. Therefore, we get improved communication complexity by the fast early linear convergence from the general convex case.

Lemma 5.1. We prove the three results separately.

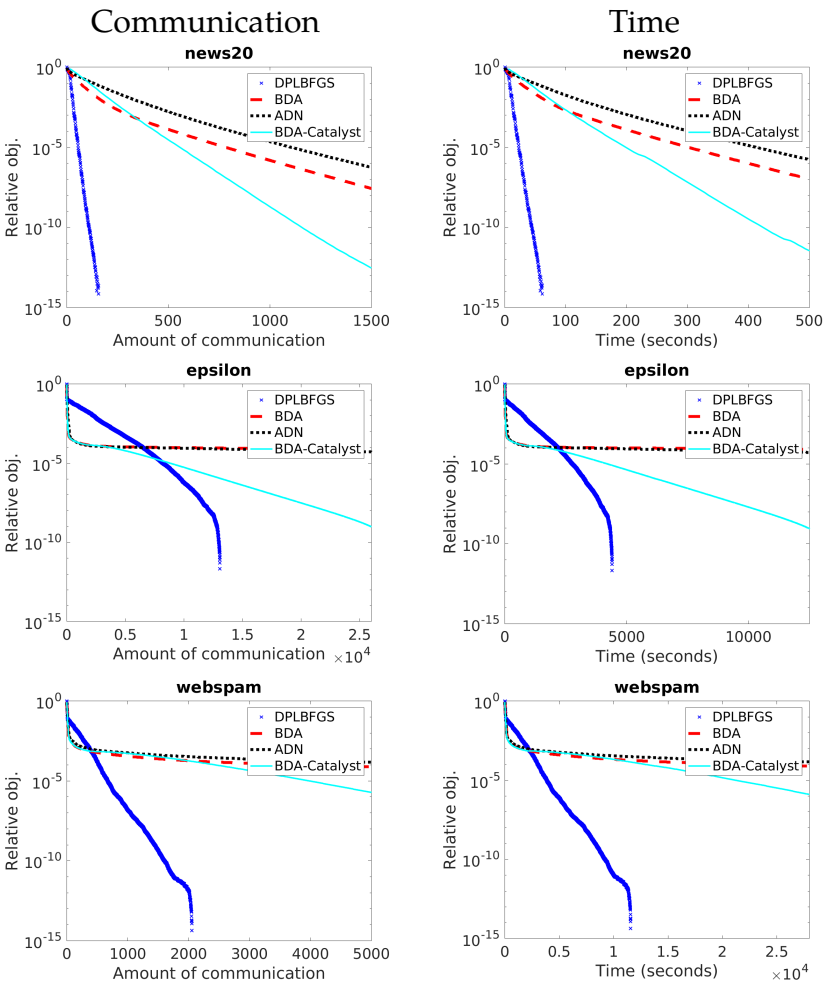


Figure 5.2: Comparison between different methods for (5.42) in terms of relative objective difference to the optimum. Left: communication (divided by d); right: running time (in seconds).

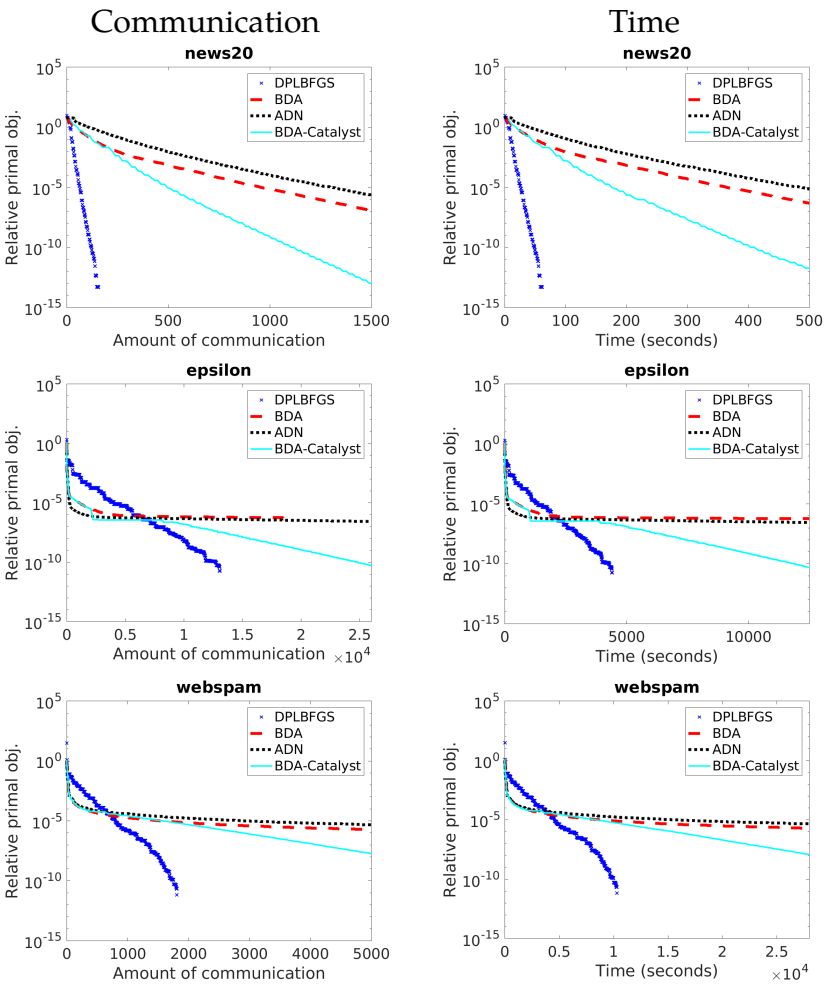


Figure 5.3: Comparison between different methods for (5.42) in terms of relative *primal* objective difference to the optimum. Left: communication (divided by d); right: running time (in seconds).

1. We assume without loss of simplicity that (5.17) is satisfied by all iterations. When it is not the case, we just need to shift the indices but the proof remains the same as the pairs of $(\boldsymbol{s}_t, \boldsymbol{y}_t)$ that do not satisfy (5.17) are discarded.
   We first bound γ_t defined in (5.15). From Lipschitz continuity of ∇f, we have that for all t,

$$\frac{\|\boldsymbol{y}_t\|^2}{\boldsymbol{y}_t^\top \boldsymbol{s}_t} \leq \frac{L^2\|\boldsymbol{s}_t\|^2}{\boldsymbol{y}_t^\top \boldsymbol{s}_t} \leq \frac{L^2}{\delta}, \tag{5.43}$$

   establishing the upper bound. For the lower bound, (5.17) implies that

$$\|\boldsymbol{y}_t\| \geq \delta\|\boldsymbol{s}_t\|, \quad \forall t. \tag{5.44}$$

   Therefore,

$$\frac{\boldsymbol{y}_t^\top \boldsymbol{s}_t}{\boldsymbol{y}_t^\top \boldsymbol{y}_t} \leq \frac{\|\boldsymbol{s}_t\|}{\|\boldsymbol{y}_t\|} \leq \frac{1}{\delta}, \quad \forall t.$$

   Following Liu and Nocedal (1989), H_t can be obtained equivalently by

$$\begin{aligned} H_t^{(0)} &= \gamma_t I, \\ H_t^{(k+1)} &= H_t^{(k)} - \frac{H_t^{(k)} \boldsymbol{s}_{t-m(t)+k} \boldsymbol{s}_{t-m(t)+k}^\top H_t^{(k)}}{\boldsymbol{s}_{t-m(t)+k}^\top H_t^{(k)} \boldsymbol{s}_{t-m(t)+k}} + \frac{\boldsymbol{y}_{t-m(t)+k} \boldsymbol{y}_{t-m(t)+k}^\top}{\boldsymbol{y}_{t-m(t)+k}^\top \boldsymbol{s}_{t-m(t)+k}}, \quad k = 0, \ldots, m(t)-1, \\ H_t &= H_t^{(m(t))}. \end{aligned} \tag{5.45}$$

   Therefore, we can bound the trace of $H_t^(k)$ and hence H_t through (5.43).

$$\text{trace}\left(H_t^{(k)}\right) \leq \text{trace}\left(H_t^{(0)}\right) + \sum_{j=t-m(t)}^{t-m(t)+k} \frac{\boldsymbol{y}_j^\top \boldsymbol{y}_j}{\boldsymbol{y}_j^\top \boldsymbol{s}_j} \leq \gamma_t N + \frac{kL^2}{\delta}, \quad \forall t, \tag{5.46}$$

where N is the matrix dimension. According to Byrd et al. (1994), the matrix $H_t^{(k)}$ is equivalent to the inverse of

$$B_t^{(k)} \coloneqq V_{t-m(t)+k}^\top \cdots V_{t-m(t))}^\top B_t^0 V_{t-m(t)} \cdots V_{t-m(t)+k} + \rho_{t-m(t)+k} \boldsymbol{s}_{t-m(t)+k} \boldsymbol{s}_{t-m(t)+k}^\top + \sum_{j=t-m(t)}^{t-m(t)-1+k} \rho_j V_{t-m(t)+k}^\top \cdots V_{j+1}^\top \boldsymbol{s}_j \boldsymbol{s}_j^\top V_{j+1} \cdots V_{t-m(t)+k}, \quad (5.47)$$

where for $j \geq 0$,

$$V_j \coloneqq I - \rho_j \boldsymbol{y}_j \boldsymbol{s}_j^\top, \quad \rho_j \coloneqq \frac{1}{\boldsymbol{y}_j^\top \boldsymbol{s}_j}, \quad B_t^0 = \frac{1}{\gamma_t} I.$$

From the form (5.47), it is clear that $B_t^{(k)}$ and hence H_t are all positive-semidefinite because $\gamma_t \geq 0, \rho_j > 0$ for all j and t. Therefore, from positive semidefiniteness, (5.46) implies the existence of $c_1 > 0$ such that

$$H_t^{(k)} \preceq c_1 I, \quad k = 0, \ldots, m(t), \quad \forall t.$$

Next, for its lower bound, from the formulation for (5.45) in Liu and Nocedal (1989), and the upper bound $\|H_t^{(k)}\| \leq c_1$, we have

$$\det(H_t) = \det\left(H_t^{(0)}\right) \prod_{k=t-m(t)}^{t-1} \frac{\boldsymbol{y}_k^\top \boldsymbol{s}_k}{\boldsymbol{s}_k^\top \boldsymbol{s}_k} \frac{\boldsymbol{s}_k^\top \boldsymbol{s}_k}{\boldsymbol{s}_k^\top H_t^{(k-t+m(t))} \boldsymbol{s}_k} \geq \gamma_t^N \left(\frac{\delta}{c_1}\right)^{m(t)} \geq M_1.$$

for some $M_1 > 0$. From that the eigenvalues of H_t are upper-bounded and nonnegative, and from the lower bound of the determinant, the eigenvalues of H_t are also lower-bounded by a positive value c_2, completing the proof.

2. By directly expanding $\nabla \hat{f}$, we have that for any p_1, p_2,

$$\nabla \hat{f}(p_1) - \nabla \hat{f}(p_2) = \nabla f(x) + H p_1 - (\nabla f(x) + H p_2) = H(p_1 - p_2).$$

Therefore, we have

$$\frac{\left(\nabla \hat{f}(p_1) - \nabla \hat{f}(p_2)\right)^\top (p_1 - p_2)}{\|p_1 - p_2\|^2} = \frac{\|p_1 - p_2\|_H^2}{\|p_1 - p_2\|^2} \in [c_2, c_1]$$

for bounding ψ_i for $i > 0$, and the bound for ψ_0 is directly from the bounds of γ_t. The combined bound is therefore $[\min\{c_2, \delta\}, \max\{c_1, L^2/\delta\}]$. Next, we show that the final ψ_i is always upper-bounded. The right-hand side of (5.20) is equivalent to the following:

$$\arg\min_{\boldsymbol{d}} \hat{Q}_{\psi_i}(\boldsymbol{d}) := \nabla \hat{f}\left(p^{(i)}\right)^\top \boldsymbol{d} + \frac{\psi_i \|\boldsymbol{d}\|^2}{2} + \hat{\Psi}(\boldsymbol{d} + p) - \hat{\Psi}(p). \quad (5.48)$$

Denote the solution by $\boldsymbol{d}$, then we have $p^{(i+1)} = p^{(i)} + \boldsymbol{d}$. Note that we allow $\boldsymbol{d}$ to be an approximate solution. Because H is upper-bounded by c_1, we have that $\nabla \hat{f}$ is c_1-Lipschitz continuous. Therefore,

$$\begin{aligned} Q\left(p^{(i+1)}\right) - Q\left(p^{(i)}\right) &\leq \nabla \hat{f}(p^{(i)})^\top \left(p^{(i+1)} - p^{(i)}\right) + \frac{c_1}{2}\left\|p^{(i+1)} - p^{(i)}\right\|^2 + \hat{\Psi}\left(p^{(i+1)}\right) - \hat{\Psi}\left(p^{(i)}\right) \\ &\overset{(5.48)}{=} \hat{Q}_{\psi_i}(\boldsymbol{d}) - \frac{\psi_i}{2}\|\boldsymbol{d}\|^2 + \frac{c_1}{2}\|\boldsymbol{d}\|^2. \end{aligned} \quad (5.49)$$

As $\hat{Q}_{\psi_i}(0) = 0$, provided that the approximate solution $\boldsymbol{d}$ is better than the point 0, we have

$$\hat{Q}(\boldsymbol{d}) \leq \hat{Q}(0) = 0. \quad (5.50)$$

Putting (5.50) into (5.49), we obtain

$$Q\left(p^{(i+1)}\right) - Q\left(p^{(i)}\right) \leq \frac{c_1 - \psi_i}{2}\|\boldsymbol{d}\|^2.$$

Therefore, whenever

$$\frac{c_1 - \psi_i}{2} \leq -\frac{\sigma_0 \psi_i}{2},$$

(5.22) holds. This is equivalent to

$$\psi_i \geq \frac{c_1}{1-\sigma_0},$$

Note that the initialization of ψ_i is upper-bounded by c_1 for all $i > 1$, so the final ψ_i is indeed upper-bounded. Together with the first iteration where we start with $\psi_0 = \gamma_t$, we have that ψ_i for all i are always bounded from the boundedness of γ_t.

3. From the results above, at every iteration, SpaRSA finds the update direction by constructing and optimizing a quadratic approximation of $\hat{f}(x)$, where the quadratic term is a multiple of identity, and its coefficient is bounded in a positive range. Therefore, the theory developed in Chapter 2 can be directly used to show the desired result even if (5.20) is solved only approximately. For completeness, we provide a simple proof for the case that (5.20) is solved exactly. We note that since Q is c_2-strongly convex, the following condition holds.

$$\frac{\min_{\boldsymbol{s}\in\nabla\hat{f}\left(p^{(i+1)}\right)+\partial\hat{g}\left(p^{(i+1)}\right)} \|\boldsymbol{s}\|^2}{2c_2} \geq Q\left(p^{(i+1)}\right) - Q^*. \tag{5.51}$$

On the other hand, from the optimality condition of (5.48), we have that for the optimal solution $\boldsymbol{d}^*$ of (5.48),

$$-\psi_i \boldsymbol{d}^* = \nabla\hat{f}\left(p^{(i)}\right) + \boldsymbol{s}_{i+1}, \tag{5.52}$$

for some

$$\boldsymbol{s}_{i+1} \in \partial\hat{\Psi}\left(p^{(i+1)}\right).$$

Therefore,

$$
\begin{aligned}
Q\left(p^{(i+1)}\right) - Q^* &\overset{(5.51)}{\leq} \frac{1}{2c_2}\left\|\nabla \hat{f}\left(p^{(i+1)}\right) - \nabla \hat{f}\left(p^{(i)}\right) + \nabla \hat{f}\left(p^{(i)}\right) + \boldsymbol{s}_{i+1}\right\|^2 \\
&\overset{(5.52)}{\leq} \frac{1}{c_2}\left\|\nabla \hat{f}\left(p^{(i+1)}\right) - \nabla \hat{f}\left(p^{(i)}\right)\right\|^2 + \left\|\psi_i \boldsymbol{d}^*\right\|^2 \\
&\leq \frac{1}{c_2}\left(c_1^2 + \psi_i^2\right)\left\|\boldsymbol{d}^*\right\|^2. \qquad (5.53)
\end{aligned}
$$

By combining (5.22) and (5.53), we obtain

$$
Q\left(p^{(i+1)}\right) - Q\left(p^{(i)}\right) \leq -\frac{\sigma_0 \psi_i}{2}\left\|\boldsymbol{d}^*\right\|^2 \leq -\frac{\sigma_0 \psi_i}{2}\frac{c_2}{c_1^2 + \psi_i^2}\left(Q\left(p^{(i+1)}\right) - Q^*\right).
$$

Rearranging the terms, we obtain

$$
\left(1 + \frac{c_2 \sigma_0 \psi_i}{2(c_1^2 + \psi^2)}\right)\left(Q\left(p^{(i+1)}\right) - Q^*\right) \leq Q\left(p^{(i)}\right) - Q^*,
$$

showing Q-linear convergence of SpaRSA, with

$$
\eta = \sup_{i=0,1,\ldots} \left(1 + \frac{c_2 \sigma_0 \psi_i}{2\left(c_1^2 + \psi_i^2\right)}\right)^{-1} \in [0, 1).
$$

Note that since ψ_i are bounded in a positive range, we can find this supremum in the desired range.

□

5.B Implementation Details and Parameter Selection for the Catalyst Framework

We first give an overview to the version of Catalyst framework for strongly-convex problems (Lin et al., 2018) for accelerating convergence rate of first-order methods, then describe our implementation details in the experiment

in Section 5.6. The Catalyst framework is described in Algorithm 10.

According to Lin et al. (2018), when $\mathcal{M}$ is the proximal gradient method, the ideal value of κ is $\max(L - 2\mu, 0)$, and when $L > 2\mu$, the convergence speed can be improved to the same order as accelerated proximal gradient (up to a logarithm factor difference). Similarly, when $\mathcal{M}$ is stochastic proximal coordinate descent with uniform sampling, by taking $\kappa = \max(L_{\max} - 2\mu, 0)$, where $L_{\max}$ is the largest block Lipschitz constant, one can obtain convergence rate similar to that of accelerated coordinate descent. Since when using proximal coordinate descent as the local solver, both BDA and ADN interpolate between proximal coordinate descent and proximal gradient,[3] depending on the number of machines, it is intuitive that acceleration should work for them.

Considering (5.42), the problem is clearly strongly convex with parameter $1/(2C)$, thus we take $\mu = 1/(2C)$. For the stopping condition, we use the simple fixed iteration choice suggested in Lin et al. (2018) (called (C3) in their notation). Empirically we found a very effective way is to run K iterations of BDA with early termination whenever a negative step size is obtained from exact line search. For the warm-start part, although (5.42) is a regularized problem, the objective part is smooth, so we take their suggestion for smooth problem to use $x_0^k = x^{k-1}$. Note that they suggested that for general regularized problems, one should take one proximal gradient step of the original F at x^{k-1} to obtain x_0^k. We also experimented with this choice, but preliminary results show that using x^{k-1} gives better initial objective value for (5.54).

The next problem is how to select κ. We observe that for webspam and epsilon, the convergence of both BDA and ADN clearly falls into two stages. Through some checks, we found that the first stage can barely be improved.

[3] Although we used RPCD but not stochastic coordinate descent, namely sampling with replacement, it is commonly considered that RPCD behaves similar to, and usually outperforms slightly, the variant that samples without replacement; see, for example, analyses in Lee and Wright (2018b); Wright and Lee (2017) and experiment in Shalev-Shwartz and Zhang (2013).

Table 5.B.1: Catalyst parameters.

Data set	#BDA iterations before starting Catalyst	κ
news	0	17
epsilon	$2,000$	$12,000$
webspam	400	$2,000$

On the other hand, if we pick a value of κ that can accelerate convergence at the later stage, the fast early convergence behavior is not present anymore, thus it takes a long time for the accelerated approach to outperform the unaccelerated version. To get better results, we take an approach from the hindsight: first start with the unaccelerated version with a suitable number of iterations, and then we switch to Catalyst with κ properly chosen by grid search for accelerating convergence at the later stage. The parameters in this approach is recorded in Table 5.B.1. We note that this way of tuning from the hindsight favors the accelerated method unfairly, as it takes information obtained through running other methods first. In particular, it requires the optimal objective (obtained by first solving the problem through other methods) and running the unaccelerated method to know the turning point of the convergence stages (requires the optimal objective to compute). Parameter tuning for κ is also needed. These additional efforts are not included in the running time comparison, so our experimental result does not suggest that the accelerated method is better than the unaccelerated version. The main purpose is to show that our proposed approach also outperforms acceleration methods with careful parameter choices.

Algorithm 9 DPLBFGS: A distributed proximal variable-metric LBFGS method for (5.11)

Given $\theta, \sigma_1 \in (0,1)$, $\delta > 0$, an initial point $x = x^0$, a partition $\{\mathcal{I}_k\}_{k=1}^K$ satisfying (5.12);
for Machines $k = 1, \ldots, K$ in parallel **do**
    Obtain $F(x)$;
    for $t = 0, 1, 2, \ldots$ **do**
        Compute $\nabla f(x)$;
        Initialize H;
        if $t \neq 0$ and (5.17) holds for $(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1})$ **then**
            Update $U_{\mathcal{I}_k,:}$, M, and γ by (5.15)-(5.16);
            Compute M^{-1};
            Implicitly form a new H from (5.14);
        if U is empty **then**
            Solve (5.7) using some existing distributed algorithm to obtain $p_{\mathcal{I}_k}$;
        else
            Solve (5.7) using Algorithm 8 in a distributed manner to obtain $p_{\mathcal{I}_k}$;
        if Line search **then**
            Compute Δ defined in (5.24);
            for $i = 0, 1, \ldots$ **do**
                $\lambda = \theta^i$;
                Compute $F(x + \lambda p)$;
                if $F(x + \lambda p) \leq F(x) + \sigma_1 \lambda \Delta$ **then**
                    Break;
        else if Trust region **then**
            $\lambda = 1$;
            Compute $Q_H(p; x)$;
            while $F(x + p) - F(x) > \sigma_1 Q_H(p; x)$ **do**
                $H \leftarrow H/\theta$;
                Re-solve (5.7) to obtain update $p_{\mathcal{I}_k}$;
                Compute $Q_H(p; x)$;
        $x_{\mathcal{I}_k} \leftarrow x_{\mathcal{I}_k} + \lambda p_{\mathcal{I}_k}$, $F(x) \leftarrow F(x + \lambda p)$;
        $x^{t+1} := x$;
        $(\boldsymbol{s}_t)_{\mathcal{I}_k} \leftarrow x^{t+1}_{\mathcal{I}_k} - x^t_{\mathcal{I}_k}$, $(\boldsymbol{y}_t)_{\mathcal{I}_k} \leftarrow \nabla_{\mathcal{I}_k} f(x^{t+1} - \nabla_{\mathcal{I}_k} f(x^t)$;

Algorithm 10 Catalyst Framework for optimizing strongly-convex (5.11).

1: Input: $x^0 \in \mathbb{R}^N$, a smoothing parameter κ, the strong convexity parameter μ, an optimization method $\mathcal{M}$, and a stopping criterion for the inner optimization.
2: Initialize $y^0 = x^0$, $q = \mu/(\mu + \kappa)$, $\beta = (1 - \sqrt{q})/(1 + \sqrt{q})$.
3: **for** $k = 1, 2, \ldots,$ **do**
4: Use $\mathcal{M}$ with the input stopping condition to approximately optimize

$$\min_x \quad F(x) + \frac{\kappa}{2}\|x - y^{k-1}\|^2 \tag{5.54}$$

from a warm-start point x_0^k to obtain the iterate x^k.
5: $y_k = x^k + \beta(x^k - x^{k-1})$.
6: Output x^k.

REFERENCES

Anderson, Edward, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. 1999. *LAPACK users' guide*. SIAM.

Andrew, Galen, and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the international conference on machine learning*, 33–40.

Attouch, Hedy, Zaki Chbani, Juan Peypouquet, and Patrick Redont. 2018. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming* 168(1-2):123–175.

Attouch, Hedy, and Juan Peypouquet. 2016. The rate of convergence of nesterov's accelerated forward-backward method is actually faster than 1/k^2. *SIAM Journal on Optimization* 26(3):1824–1834.

Bach, Francis. 2015. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization* 25(1):115–129.

Baraniuk, Richard G. 2007. Compressive sensing. *IEEE signal processing magazine* 24(4).

Beck, Amir, and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202.

Bertsekas, Dimitri P. 1982. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization* 20(2):221–246.

Bertsekas, Dimitri P. 2016. *Nonlinear programming*. 3rd ed. Athena scientific Belmont.

Bonettini, Silvia, Ignace Loris, Federica Porta, and Marco Prato. 2016. Variable metric inexact line-search-based methods for nonsmooth optimization. *SIAM Journal on Optimization* 26(2):891–921.

Bonettini, Silvia, Ignace Loris, Federica Porta, Marco Prato, and Simone Rebegoldi. 2017. On the convergence of a linesearch based proximal-gradient method for nonconvex optimization. *Inverse Problems* 33(5).

Boyd, Stephen, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge University Press.

Burachik, Regina, LM Graña Drummond, Alfredo N Iusem, and BF Svaiter. 1995. Full convergence of the steepest descent method with inexact line searches. *Optimization* 32(2):137–146.

Burke, James V., and Michael C. Ferris. 1993. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization* 31(5):1340–1359.

Burke, James V, Jorge J. Moré, and Gerardo Toraldo. 1990. Convergence properties of trust region methods for linear and convex constraints. *Mathematical Programming* 47(1-3):305–336.

Byrd, Richard H., Peihung Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16:1190–1208.

Byrd, Richard H., Jorge Nocedal, and Figen Oztoprak. 2016. An inexact successive quadratic approximation method for $L - 1$ regularized optimization. *Mathematical Programming* 157(2):375–396.

Byrd, Richard H., Jorge Nocedal, and Robert B. Schnabel. 1994. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming* 63(1-3):129–156.

Chan, Ernie, Marcel Heimlich, Avi Purkayastha, and Robert Van De Geijn. 2007. Collective communication: theory, practice, and experience. *Concurrency and Computation: Practice and Experience* 19(13):1749–1783.

Chouzenoux, Emilie, Jean-Christophe Pesquet, and Audrey Repetti. 2014. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications* 162(1):107–132.

———. 2016. A block coordinate variable metric forward–backward algorithm. *Journal of Global Optimization* 66(3):457–485.

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

Conn, Andrew R., Nicholas I. M. Gould, and Philippe L. Toint. 1988. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis* 25(2):433–460.

Crammer, Koby, and Yoram Singer. 2002. On the learnability and design of output codes for multiclass problems. *Machine Learning* 47(2–3):201–233.

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

Donoho, David L, et al. 2006. Compressed sensing. *IEEE Transactions on information theory* 52(4):1289–1306.

Drusvyatskiy, Dmitriy, and Adrian S Lewis. 2018. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*.

Dünner, Celestine, Aurelien Lucchi, Matilde Gargiani, An Bian, Thomas Hofmann, and Martin Jaggi. 2018. A distributed second-order algorithm you can trust. In *Proceedings of the international conference on machine learning*.

Fletcher, Roger. 1987. *Practical methods of optimization*. John Wiley and Sons.

Fountoulakis, Kimon, and Rachael Tappenden. 2018. A flexible coordinate descent method. *Computational Optimization and Applications* 70(2): 351–394.

Gallant, Stephen I. 1990. Perceptron-based learning algorithms. *Neural Networks, IEEE Transactions on* 1(2):179–191.

Ghanbari, Hiva, and Katya Scheinberg. 2018. Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates. *Computational Optimization and Applications* 69(3):597–627.

Hiriart-Urruty, Jean-Baptiste, and Claude Lemaréchal. 2001. *Fundamentals of convex analysis*. Springer Science & Business Media.

Hiriart-Urruty, Jean-Baptiste, Jean-Jacques Strodiot, and V Hien Nguyen. 1984. Generalized hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Applied Mathematics & Optimization* 11(1): 43–56.

Jiang, Kaifeng, Defeng Sun, and Kim-Chuan Toh. 2012. An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp. *SIAM Journal on Optimization* 22(3):1042–1064.

Lebanon, Guy, and John D Lafferty. 2002. Boosting and maximum likelihood for exponential models. In *Advances in neural information processing systems*, 447–454.

Lee, Ching-pei, and Kai-Wei Chang. 2017. Distributed block-diagonal approximation methods for regularized empirical risk minimization. Tech. Rep.

Lee, Ching-pei, Cong Han Lim, and Stephen J. Wright. 2018. A distributed quasi-Newton algorithm for empirical risk minimization with nonsmooth regularization. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, 1646–1655. New York, NY, USA: ACM.

Lee, Ching-pei, and Chih-Jen Lin. 2013. A study on L2-loss (squared hinge-loss) multi-class SVM. *Neural Computation* 25(5):1302–1323.

Lee, Ching-pei, and Dan Roth. 2015. Distributed box-constrained quadratic optimization for dual linear SVM. In *Proceedings of the international conference on machine learning*.

Lee, Ching-pei, Po-Wei Wang, Weizhu Chen, and Chih-Jen Lin. 2017. Limited-memory common-directions method for distributed optimization and its application on empirical risk minimization. In *Proceedings of the siam international conference on data mining*.

Lee, Ching-pei, and Stephen J. Wright. 2017. Using neural networks to detect line outages from PMU data. Tech. Rep.

———. 2018a. Inexact variable metric stochastic block-coordinate descent for regularized optimization. Tech. Rep.

———. 2018b. Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*.

———. 2019a. First-order algorithms converge faster than $O(1/k)$ on convex problems. In *Proceedings of the international conference on machine learning*.

———. 2019b. Inexact successive quadratic approximation for regularized optimization. *Computational Optimization and Applications* 72:641–674.

Lee, Jason D., Yuekai Sun, and Michael A. Saunders. 2014. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization* 24(3):1420–1443.

Li, Dong-Hui, and Masao Fukushima. 2001. On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization* 11(4):1054–1064.

Li, Jinchao, Martin S. Andersen, and Lieven Vandenberghe. 2017a. Inexact proximal Newton methods for self-concordant functions. *Mathematical Methods of Operations Research* 85(1):19–41.

Li, Xingguo, Tuo Zhao, Raman Arora, Han Liu, and Mingyi Hong. 2017b. On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *Journal of Machine Learning Research* 18(1):6741–6764.

Lin, Chieh-Yen, Cheng-Hao Tsai, Ching-Pei Lee, and Chih-Jen Lin. 2014. Large-scale logistic regression and linear support vector machines using Spark. In *Proceedings of the ieee international conference on big data*, 519–528.

Lin, Chih-Jen, and Jorge J. Moré. 1999. Newton's method for large-scale bound constrained problems. *SIAM Journal on Optimization* 9:1100–1127.

Lin, Hongzhou, Julien Mairal, and Zaid Harchaoui. 2018. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research* 18(212):1–54.

Lions, Pierre-Louis, and Bertrand Mercier. 1979. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis* 16(6):964–979.

Liu, Dong C., and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45(1): 503–528.

Lu, Zhaosong, and Lin Xiao. 2015. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming* 152(1-2):615–642.

Meier, Lukas, Sara Van De Geer, and Peter Bühlmann. 2008. The group LASSO for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70(1):53–71.

Meng, Xiangrui, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. 2016. MLlib: Machine learning in Apache Spark. *Journal of Machine Learning Research* 17(1):1235–1241.

Message Passing Interface Forum. 1994. MPI: a message-passing interface standard. *International Journal on Supercomputer Applications* 8(3/4).

Moré, Jorge J, and Danny C Sorensen. 1983. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing* 4(3):553–572.

Mureşan, Marian. 2009. *A concrete approach to classical analysis*, vol. 14. Springer.

Necoara, Ion, Yurii Nesterov, and Francois Glineur. 2018. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming* 1–39.

Nesterov, Yurii. 1983. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady* 27:372–376.

———. 2004. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers.

———. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* 22(2):341–362.

———. 2013. Gradient methods for minimizing composite functions. *Mathematical Programming* 140(1):125–161.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical optimization*. 2nd ed. Springer.

Nutini, Julie, Issam Laradji, and Mark Schmidt. 2017. Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. Tech. Rep. ArXiv:1712.08859.

Nutini, Julie, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. 2015. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International conference on machine learning*, 1632–1641.

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

Patrascu, Andrei, and Ion Necoara. 2015. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization* 61(1):19–46.

Peng, Wei, Hui Zhang, and Xiaoya Zhang. 2018. Global complexity analysis of inexact successive quadratic approximation methods for regularized optimization under mild assumptions. Tech. Rep.

Polyak, Boris T. 1987. *Introduction to optimization*. Translation Series in Mathematics and Engineering.

Rasmussen, Carl Edward. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*, 63–71. Springer.

Rodomanov, Anton, and Dmitry Kropotov. 2016. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *Proceedings of the international conference on machine learning*, 2597–2605.

Rosen, Jo Bo. 1960. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the society for industrial and applied mathematics* 8(1):181–217.

Scheinberg, Katya, and Xiaocheng Tang. 2016. Practical inexact proximal quasi-Newton method with global complexity analysis. *Mathematical Programming* 160(1-2):495–529.

Schmidt, Mark, Nicolas Roux, and Francis Bach. 2011. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, 1458–1466.

Shalev-Shwartz, Shai, and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shalev-Shwartz, Shai, and Tong Zhang. 2012. Proximal stochastic dual coordinate ascent. Tech. Rep.

———. 2013. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research* 14(Feb):567–599.

Shamir, Ohad, Nati Srebro, and Tong Zhang. 2014. Communication-efficient distributed optimization using an approximate Newton-type method. In *Proceedings of the international conference on machine learning*.

Sun, Ruoyu, and Mingyi Hong. 2015. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in neural information processing systems*, 1306–1314.

Sun, Ruoyu, and Yinyu Ye. 2016. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. Technical Report, Department of Management Science and Engineering, Stanford University, Stanford, California. ArXiv:1604.07130.

Tappenden, Rachael, Peter Richtárik, and Jacek Gondzio. 2016. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications* 170(1):144–176.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society Series B* 58:267–288.

Tikhonov, Andrey Nikolayevich. 1943. On the stability of inverse problems. In *Dokl. akad. nauk sssr*, vol. 39, 195–198.

Tran-Dinh, Quoc, Anastasios Kyrillidis, and Volkan Cevher. 2014. An inexact proximal path-following algorithm for constrained convex minimization. *SIAM Journal on Optimization* 24(4):1718–1745.

Tseng, Paul, and Sangwoon Yun. 2009. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming* 117(1):387–423.

Tsochantaridis, Ioannis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of machine learning research* 6(Sep): 1453–1484.

Villa, Silvia, Saverio Salzo, Luca Baldassarre, and Alessandro Verri. 2013. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization* 23(3):1607–1633.

Walker, Alastair J. 1977. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software* 3(3):253–256.

Wang, Po-Wei, Ching-pei Lee, and Chih-Jen Lin. 2016. The common directions method for regularized empirical loss minimization. Tech. Rep.

Wright, Stephen J., and Ching-pei Lee. 2017. Analyzing random permutations for cyclic coordinate descent. Tech. Rep., Department of Computer Sciences, University of Wisconsin-Madison. ArXiv:1706:00908.

Wright, Stephen J., Robert D. Nowak, and Mário A. T. Figueiredo. 2009. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing* 57(7):2479–2493.

Yang, Junfeng, and Yin Zhang. 2011. Alternating direction algorithms for $\ell 1$-problems in compressive sensing. *SIAM Journal on Scientific Computing* 33(1):250–278.

Yang, Tianbao. 2013. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in neural information processing systems*, 629–637.

Yuan, Guo-Xun, Chia-Hua Ho, and Chih-Jen Lin. 2012. An improved GLMNET for $L1$-regularized logistic regression. *Journal of Machine Learning Research* 13:1999–2030.

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

Yun, Sangwoon. 2014. On the iteration complexity of cyclic coordinate gradient descent methods. *SIAM Journal on Optimization* 24(3):1567–1580.

Zhang, Yuchen, and Xiao Lin. 2015. DiSCO: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, 362–370.

Zhao, Peilin, and Tong Zhang. 2015. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd international conference on machine learning*.

Zheng, Shun, Jialei Wang, Fen Xia, Wei Xu, and Tong Zhang. 2017. A general distributed dual coordinate optimization framework for regularized loss minimization. *Journal of Machine Learning Research* 18(115):1–52.

Zhong, Kai, Ian En-Hsu Yen, Inderjit S. Dhillon, and Pradeep K. Ravikumar. 2014. Proximal quasi-newton for computationally intensive $l1$-regularized M-estimators. In *Advances in neural information processing systems*.

Zhuang, Yong, Wei-Sheng Chin, Yu-Chin Juan, and Chih-Jen Lin. 2015. Distributed Newton method for regularized logistic regression. In *Proceedings of the pacific-asia conference on knowledge discovery and data mining*.