Inexact Coordinate Descent: Complexity and Preconditioning*

Rachael Tappenden Peter Richtárik Jacek Gondzio

School of Mathematics
University of Edinburgh
United Kingdom

April 18, 2013

Abstract

In this paper we consider the problem of minimizing a convex function using a randomized block coordinate descent method. One of the key steps at each iteration of the algorithm is determining the update to a block of variables. Existing algorithms assume that in order to compute the update, a particular subproblem is solved *exactly*. In his work we relax this requirement, and allow for the subproblem to be solved *inexactly*, leading to an inexact block coordinate descent method. Our approach incorporates the best known results for exact updates as a special case. Moreover, these theoretical guarantees are complemented by practical considerations: the use of iterative techniques to determine the update as well as the use of preconditioning for further acceleration.

Keywords: inexact methods, block coordinate descent, convex optimization, iteration complexity, preconditioning, conjugate gradients.

AMS: 65F08; 65F10; 65F15; 65Y20; 68Q25; 90C25

1 Introduction

Due to a dramatic increase in the size of optimization problems being encountered, first order methods are becoming increasingly popular. These large-scale problems are often highly structured and it is important for any optimization method to take advantage of the underlying structure. Applications where such problems arise and where first order methods proved successful include machine learning [16, 32], compressive sensing [7, 43], group lasso [25, 36], matrix completion [4, 26], truss topology design [27].

Block coordinate descent methods seem a natural choice for these very large-scale problems due to their low memory requirements and low per-iteration computational cost. Furthermore, they are often designed to take advantage of the underlying structure of the optimization problem [41, 42] and many of these algorithms are supported by high probability iteration complexity results [23, 24, 27, 28, 38].

*This work was supported by the EPSRC grant EP/I017127/1 "Mathematics for vast digital resources". Peter Richtárik was also supported by the Centre for Numerical Algorithms and Intelligent Software (funded by EPSRC grant EP/G036136/1 and the Scottish Funding Council).

1.1 The Problem

If the block size is larger than one, determining the update to use at a particular iteration in a block coordinate descent method can be computationally expensive. The purpose of this work is to reduce the cost of this step. To achieve this, we extend the work in [28] to include the case of an *inexact* update.

In this work we study randomized block coordinate descent methods applied to the problem of minimizing a composite objective function. That is, a function formed as the sum of a smooth convex and a simple nonsmooth convex term:

$$\min_{x\in\mathbb{R}^N}\{F(x) := f(x) + \Psi(x)\}. \tag{1}$$

We assume that the problem has a minimum ($F^* > -\infty$), f has (block) coordinate Lipschitz gradient, and Ψ is a (block) separable proper closed convex extended real valued function (all these concepts will be defined precisely in Section 2).

Our algorithm (namely, the Inexact Coordinate Descent (ICD) method) is supported by high probability iteration complexity results. That is, confidence level $\rho \in (0, 1)$ and error tolerance $\epsilon > 0$, we give an explicit expression for the number of iterations k that guarantee that the method produces a random iterate x_k for which

$$\mathbb{P}(F(x_k) - F^* \leq \epsilon) \geq 1 - \rho.$$

We will show that in the inexact case it is not always possible to achieve a solution with small error and/or high confidence.

Our theoretical guarantees are complemented by practical considerations. Because an inexact update is allowed, it is sensible to use iterative methods to solve the subproblems; this can be especially beneficial for quadratic minimization. This motivates us to study the smooth quadratic case where the problem exhibits block angular structure, and to solve the subproblems using conjugate gradients [12]. The benefits of preconditioning the conjugate gradients method are well known, so we introduce and analyze a preconditioner that can be used to speed up the update step, and ultimately reduce the overall ICD algorithm running time. Finally, we present some encouraging computational results.

1.2 Literature Review

As problem sizes increase, first order methods are benefiting from revived interest. On very large problems however, the computation of a single gradient step is expensive, and methods are needed which would be able to make progress before a standard gradient algorithm takes a single step. For instance, a randomized variant of the Kaczmarz method for solving linear systems has recently been studied, equipped with iteration complexity bounds [20, 21, 15, 37], and found surprisingly efficient. This method can be seen as a special case of a more general class of decomposition algorithms, block coordinate descent methods, which have recently gained much popularity [19, 23, 24, 28, 29, 30, 40]. One of the main differences between various (serial) coordinate descent schemes is the way in which the coordinate is chosen at each iteration. Traditionally cyclic schemes [31] and greedy schemes [27] were studied. More recently, a popular alternative is to select coordinates randomly, because the coordinate can be selected cheaply, and useful iteration complexity results can be obtained [18, 28, 29, 30, 39, 34].

Another current trend in this area is to consider methods that incorporate some kind of 'inexactness', perhaps using approximate gradients, or using inexact updates. For example, [17] considers methods based on inexact dual gradient information, while [32] considers the minimization of an unconstrained convex composite function where error is present in the gradient of the smooth term, or in the proximity operator for the non-smooth term. Other works study methods that use inexact updates when the objective function is convex, smooth and unconstrained [1], smooth and constrained [3] or for ℓ_1-regularized quadratic least squares problem [14].

1.3 Contribution

In this paper we extend the work of Richtárik and Takáč [28] and present a block coordinate descent method that employs inexact updates having the potential to reduce the overall algorithm running time. Furthermore, we focus in detail on the quadratic case, which benefits greatly from inexact updates, and show how preconditioning can be used to complement the inexact update strategy.

F	Exact Method [28]	Inexact Method [this paper]	Theorem
C-N	$\frac{c_1}{\epsilon}\left(1+\log\frac{1}{\rho}\right)+2$	$\frac{c_1}{\epsilon}+\frac{c_1}{\epsilon-\alpha c_1}\log\left(\frac{\epsilon-\frac{\beta c_1}{\epsilon-\alpha c_1}}{\epsilon\rho-\frac{\beta c_1}{\epsilon-\alpha c_1}}\right)+2$	7(i)
C-N	$c_2\log\left(\frac{F(x_0)-F^*}{\epsilon\rho}\right)$	$\frac{c_2}{1-\alpha c_2}\log\left(\frac{F(x_0)-F^*-\frac{\beta c_2}{1-\alpha c_2}}{\epsilon\rho-\frac{\beta c_2}{1-\alpha c_2}}\right)$	7(ii)
SC-N	$\frac{n}{\mu}\log\left(\frac{F(x_0)-F^*}{\epsilon\rho}\right)$	$\frac{n}{\mu-\alpha n}\log\left(\frac{F(x_0)-F^*-\frac{\beta n}{\mu-\alpha n}}{\epsilon\rho-\frac{\beta n}{\mu-\alpha n}}\right)$	9
C-S	$\frac{\hat{c}_1}{\epsilon}\left(1+\log\frac{1}{\rho}\right)+2$	$\frac{\hat{c}_1}{\epsilon}+\frac{\hat{c}_1}{\epsilon-\alpha\hat{c}_1}\log\left(\frac{\epsilon-\frac{\beta\hat{c}_1}{\epsilon-\alpha\hat{c}_1}}{\epsilon\rho-\frac{\beta\hat{c}_1}{\epsilon-\alpha\hat{c}_1}}\right)+2$	10
SC-S	$\frac{1}{\mu_f}\log\left(\frac{f(x_0)-f^*}{\epsilon\rho}\right)$	$\frac{1}{\mu_f-\alpha}\log\left(\frac{f(x_0)-f^*-\frac{\beta}{\mu_f-\alpha}}{\epsilon\rho-\frac{\beta}{\mu_f-\alpha}}\right)$	11

Table 1: Comparison of the iteration complexity results for coordinate descent methods using an inexact update and using an exact update (C=Convex, SC=Strongly Convex, N=Nonsmooth, S = Smooth).

Table 1 compares *some* of the new complexity results obtained in this paper for an inexact update with the complexity results for an exact update presented in [28]. The following notation is used in the table: by μ_ϕ we denote the strong convexity parameter of function ϕ (with respect to a certain norm specified later), $\mu=(\mu_f+\mu_\Psi)/(1+\mu_\Psi)$ and $\mathcal{R}_w(x_0)$ can be roughly considered to be distance from x_0 to a solution of (1) measured in a specific weighted norm parameterized by the vector w (to be defined precisely in (14)). The constants are $c_1=2n\max\{\mathcal{R}_w^2(x_0),F(x_0)-F^*\}$, $\hat{c}_1=$

$2\mathcal{R}_w^2(x_0)$ and $c_2 = 2n\mathcal{R}_w^2(x_0)/\epsilon$, and n is the number of blocks. Parameters $\alpha, \beta \geq 0$ control the level of inexactness (to be defined precisely in Section 3.2).

Table 1 shows that for fixed ϵ and ρ, an inexact method will require more iterations than an exact one. However, it is expected that in certain situations an inexact update will be significantly cheaper to compute than an exact update, leading to better overall running time. Moreover, the new complexity results for the inexact method generalize those for the exact method. Specifically, for inexactness parameters $\alpha = \beta = 0$ we recover the complexity results in [28].

1.4 Outline

The first part of this paper focuses on the theoretical aspects of a block coordinate descent method when an inexact update is employed. In Section 2 the assumptions and notation are laid out and in Section 3 the ICD method is presented. In Section 4 iteration complexity results for ICD applied to (1) are presented in both the convex and strongly convex cases. Iteration complexity results for ICD applied to a convex smooth minimization problem ($\Psi = 0$ in (1)) are presented in Section 5, in both the convex and strongly convex cases.

The second part of the paper considers the practicality of an inexact update. Section 6 describes how the ICD method can be effectively implemented in the quadratic case when block angular structure is present. The update step is computed using conjugate gradients and we describe a preconditioner for the update step. Section 7 provides a detailed analysis of the spectrum of the preconditioned matrix, numerical experiments are presented in Section 8 and concluding remarks are given in Section 9.

2 Assumptions and Notation

In this section we introduce the notation and definitions that are used throughout the paper.

2.1 Block structure of $\mathbb{R}^N$

The problem under consideration is assumed to have block structure and this is modelled by decomposing the space $\mathbb{R}^N$ into n subspaces as follows. Let $U \in \mathbb{R}^{N\times N}$ be a column permutation of the $N \times N$ identity matrix and further let $U = [U_1, U_2, \ldots, U_n]$ be a decomposition of U into n submatrices, where U_i is $N \times N_i$ and $\sum_{i=1}^n N_i = N$. It is clear (e.g., see [29] for a brief proof) that any vector $x \in \mathbb{R}^N$ can be written uniquely as

$$x = \sum_{i=1}^{n} U_i x^{(i)}, \tag{2}$$

where $x^{(i)} \in \mathbb{R}_i \equiv \mathbb{R}^{N_i}$. Moreover, these vectors are given by

$$x^{(i)} := U_i^T x. \tag{3}$$

For simplicity we will sometimes write $x = (x^{(1)}, x^{(2)}, \ldots, x^{(n)})$ instead of (2). We equip $\mathbb{R}_i$ with a pair of conjugate Euclidean norms:

$$\|t\|_{(i)} := \langle B_i t, t\rangle^{\frac{1}{2}}, \qquad \|t\|_{(i)}^* = \langle B_i^{-1} t, t\rangle^{\frac{1}{2}}, \qquad t \in \mathbb{R}_i, \tag{4}$$

where $B_i \in \mathbb{R}^{N_i\times N_i}$ is positive definite and $\langle\cdot,\cdot\rangle$ is the standard Euclidean dot product.

2.2 Smoothness of f

Throughout this paper we assume that the gradient of f is block Lipschitz, uniformly in x, with positive constants $l_1, \dots, l_n$. This means that, for all $x \in \mathbb{R}^N$, $i \in \{1, 2, \dots, n\}$ and $t \in \mathbb{R}_i$ we have

$$\|\nabla_i f(x + U_i t) - \nabla_i f(x)\|_{(i)}^* \leq l_i \|t\|_{(i)}, \tag{5}$$

where

$$\nabla_i f(x) := (\nabla f(x))^{(i)} \overset{(3)}{=} U_i^T \nabla f(x) \in \mathbb{R}_i. \tag{6}$$

An important consequence of (5) is the following standard inequality [22, p.57]:

$$f(x + U_i t) \leq f(x) + \langle \nabla_i f(x), t \rangle + \tfrac{l_i}{2} \|t\|_{(i)}^2. \tag{7}$$

2.3 Block separability of Ψ

The function $\Psi : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is assumed to be block separable. That is, we assume that it can be decomposed as:

$$\Psi(x) = \sum_{i=1}^n \Psi_i(x^{(i)}), \tag{8}$$

where the functions $\Psi_i : \mathbb{R}_i \to \mathbb{R} \cup \{+\infty\}$ are convex and closed.

2.4 Norms on $\mathbb{R}^N$

For fixed positive scalars $w_1, w_2, \dots, w_n$, let $w = (w_1, \dots, w_n)$ and define a pair of conjugate norms in $\mathbb{R}^N$ by

$$\|x\|_w^2 := \sum_{i=1}^n w_i \|x^{(i)}\|_{(i)}^2, \quad (\|y\|_w^*)^2 := \max_{\|x\|_w \leq 1} \langle y, x \rangle^2 = \sum_{i=1}^n w_i^{-1} (\|y^{(i)}\|_{(i)}^*)^2. \tag{9}$$

In the subsequent analysis we will use $w = l$ (for $\Psi \neq 0$) and $w = lp^{-1}$ (for $\Psi = 0$), where $l = (l_1, \dots, l_n)$ is a vector of Lipschitz constants, $p = (p_1, \dots, p_n)$ is a vector of positive probabilities and lp^{-1} denotes the vector $(l_1/p_1, \dots, l_n/p_n)$.

2.5 Strong convexity of F

In some of the results presented in this work we assume that F is strongly convex with respect to the norm $\|\cdot\|_w$ for some w, with (strong) convexity parameter $\mu_F(w) > 0$. A function $\phi : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is strongly convex w.r.t. $\|\cdot\|_w$ with convexity parameter $\mu_\phi(w) > 0$ if for all $x, y \in \operatorname{dom} \phi$,

$$\phi(y) \geq \phi(x) + \langle \phi'(x), y - x \rangle + \tfrac{\mu_\phi(w)}{2} \|y - x\|_w^2, \tag{10}$$

where ϕ' is any subgradient of ϕ at x. The case with $\mu_\phi(w) = 0$ reduces to convexity.

Strong convexity of F may come from f or Ψ or both and we will write $\mu_f(w)$ (resp. $\mu_\Psi(w)$) for the strong convexity parameter of f (resp. Ψ). Following from (10)

$$\mu_F(w) \geq \mu_f(w) + \mu_\Psi(w). \tag{11}$$

Using (7) and (10) it can be shown that

$$\mu_f(l) \leq 1, \quad \text{and} \quad \mu_f(lp^{-1}) < 1. \tag{12}$$

We will also make use of the following characterisation of strong convexity. For all $x, y \in \operatorname{dom}\phi$ and $\lambda \in [0,1]$,

$$\phi\big(\lambda x + (1-\lambda)y\big) \;\leq\; \lambda\phi(x) + (1-\lambda)\phi(y) - \tfrac{\mu_\phi(w)\lambda(1-\lambda)}{2}\|x-y\|_w^2. \tag{13}$$

2.6 Level set radius

The set of optimal solutions of (1) is denoted by X^* and x^* is any element of that set. We define

$$\mathcal{R}_w(x) := \max_y \max_{x^* \in X^*} \{\|y - x^*\|_w : F(y) \leq F(x)\}, \tag{14}$$

which is a measure of the size of the level set of F given by x. We assume that $\mathcal{R}_w(x_0)$ is finite for the initial iterate x_0.

3 The Algorithm

Let us start by presenting the algorithm; a more detailed description will follow.

Algorithm 1 ICD: Inexact Coordinate Descent

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Choose $\delta_k \in \mathbb{R}^n$ such that $\sum_i p_i \delta_k^{(i)} \leq \alpha(F(x_k) - F^*) + \beta$
3: Choose block $i \in \{1, 2, \ldots, n\}$ with probability $p_i > 0$
4: Compute the inexact update $T_{\delta_k}^{(i)}(x_k)$ to block i of x_k
5: Update block i of x_k: $x_{k+1} = x_k + U_i T_{\delta_k}^{(i)}(x_k)$
6: **end for**

3.1 Generic description

Given iterate $x_k \in \mathbb{R}^N$, Algorithm 1 picks block $i \in \{1, 2, \ldots, n\}$ with probability p_i, computes the vector $T_{\delta_k}^{(i)}(x_k) \in \mathbb{R}_i$ and then adds it to the ith block of x_k, producing the new iterate x_{k+1}. The iterates $\{x_k\}$ are random vectors and the values $\{F(x_k)\}$ are random variables. The update vector depends on x_k, the current iterate, and on δ_k, a vector of parameters controlling the "level of inexactness" with which the update is computed. The rest of this section is devoted to giving a precise definition of $T_{\delta_k}^{(i)}(x_k)$. Note that from (1) and (7) we have, for all $x \in \mathbb{R}^N$, $i \in \{1, 2, \ldots, n\}$ and $t \in \mathbb{R}_i$:

$$F(x + U_i t) = f(x + U_i t) + \Psi(x + U_i t) \leq f(x) + V_i(x, t) + \psi_i(x), \tag{15}$$

where

$$V_i(x,t) := \langle \nabla_i f(x), t\rangle + \tfrac{l_i}{2}\|t\|_{(i)}^2 + \Psi_i(x^{(i)} + t), \tag{16}$$

$$\psi_i(x) := \sum_{j \neq i} \Psi_j(x^{(j)}). \tag{17}$$

That is, (15) gives an upper bound on $F(x+U_it)$, viewed as a function of $t \in \mathbb{R}_i$.

The inexact update applied in step 4 of Algorithm 1 is the *inexact* minimizer of the upper bound (15) on $F(x_k+U_it)$ (to be defined precisely below). However, since only the second term of this bound depends on t, the update is computed by minimizing, *inexactly*, $V_i(x,t)$ in t.

3.2 Inexact update

The approach of this paper best applies to situations in which it is much easier to approximately minimize $t \mapsto V_i(x,t)$ than to both (i) approximately minimize $t \mapsto F(x+U_it)$ and (ii) exactly minimize $t \mapsto V_i(x,t)$. For $x \in \mathbb{R}^N$ and[1] $\delta = (\delta^{(1)},\dots,\delta^{(n)}) \geq 0$ we define $T_\delta(x) = (T_\delta^{(1)}(x),\dots,T_\delta^{(n)}(x)) \in \mathbb{R}^N$ to be any vector satisfying

$$V_i(x, T_\delta^{(i)}(x)) \leq \min\left\{V_i(x,0), \delta^{(i)} + \min_{t\in\mathbb{R}_i} V_i(x,t)\right\}, \quad i=1,\dots,n. \tag{18}$$

That is, we require that the update $T_\delta^{(i)}(x)$ of the ith block of x is (i) no worse than a nil update, and that it is (ii) close to the optimal update $T_0^{(i)}(x) = \arg\min_t V_i(x,t)$, where the degree of suboptimality/inexactness is bounded by $\delta^{(i)}$. In particular, we consider, and give iteration complexity results for, the situation where at iteration k of the ICD method we choose $\delta_k = (\delta_k^{(1)},\dots,\delta_k^{(n)})$ so that the expected suboptimality is bounded above by a linear function of the residual $F(x_k)-F^*$, i.e.,

$$\bar{\delta}_k := \sum_{i=1}^n p_i \delta_k^{(i)} \leq \alpha(F(x_k)-F^*) + \beta, \tag{19}$$

where α and β are nonnegative constants. Note that, for instance, (19) holds if we require $\delta_k^{(i)} \leq \alpha(F(x_k)-F^*)+\beta$ for all i.

As the following lemma shows, the update (18) leads to a monotonic algorithm.

Lemma 1. *For all $x \in \mathbb{R}^N, \delta \in \mathbb{R}^n_+$ and $i \in \{1,2,\dots,n\}$,*

$$F(x+U_iT_\delta^{(i)}(x)) \leq F(x). \tag{20}$$

Proof:

$$\begin{aligned} F(x+U_iT_\delta^{(i)}(x)) &\overset{(15)}{\leq} f(x) + V_i(x, T_\delta^{(i)}(x)) + \psi_i(x) \\ &\overset{(18)}{\leq} f(x) + V_i(x,0) + \psi_i(x) \overset{(16)+(17)}{=} F(x). \end{aligned}$$ □

3.3 Technical result

The following result plays a key role in the complexity analysis of ICD.

Theorem 2. *Fix $x_0 \in \mathbb{R}^N$ and let $\{x_k\}_{k\geq 0}$ be a sequence of random vectors in $\mathbb{R}^N$ with x_{k+1} depending on x_k only. Let $\varphi : \mathbb{R}^N \to \mathbb{R}$ be a nonnegative function, define $\xi_k := \varphi(x_k)$ and assume that $\{\xi_k\}_{k\geq 0}$ is nonincreasing. Further, let $\rho \in (0,1)$, $\epsilon > 0$ and $\alpha, \beta \geq 0$ be such that one of the following two conditions holds:*

[1] We allow here for an abuse of notation ($\delta^{(i)}$ is a scalar, rather than a vector in $\mathbb{R}_i$ as $x^{(i)}$ for $x \in \mathbb{R}^N$) as we wish to emphasize that the scalar $\delta^{(i)}$ is associated with the i-th block.

(i) $\mathbb{E}[\xi_{k+1} \mid x_k] \leq (1+\alpha)\xi_k - \frac{\xi_k^2}{c_1} + \beta$, *for all* $k \geq 0$,
where $c_1 > 0$, $\frac{c_1}{2}\left(\alpha + \sqrt{\alpha^2 + \frac{4\beta}{c_1 \rho}}\right) < \epsilon < \xi_0$ *and* $\sigma := \sqrt{\alpha^2 + \frac{4\beta}{c_1}} < 1$;

(ii) $\mathbb{E}[\xi_{k+1} \mid x_k] \leq \left(1 + \alpha - \frac{1}{c_2}\right)\xi_k + \beta$, *for all* $k \geq 0$ *for which* $\xi_k \geq \epsilon$,
where $c_2 > 1$, $\alpha c_2 < 1$ *and* $\frac{\beta c_2}{\rho(1-\alpha c_2)} < \epsilon < \xi_0$.

If (i) *holds and we define* $u := \frac{c_1}{2}(\alpha + \sigma)$ *and choose*[2]

$$K \geq \frac{c_1}{\epsilon - \alpha c_1} \log\left(\frac{\epsilon - \frac{\beta c_1}{\epsilon - \alpha c_1}}{\epsilon\rho - \frac{\beta c_1}{\epsilon - \alpha c_1}}\right) + \min\left\{\frac{1}{\sigma}\log\left(\frac{\xi_0 - u}{\epsilon - u}\right), \frac{c_1}{\epsilon} - \frac{c_1}{\xi_0 - u}\right\} + 2, \tag{21}$$

or if (ii) *holds and we choose*

$$K \geq \frac{c_2}{1 - \alpha c_2} \log\left(\frac{\xi_0 - \frac{\beta c_2}{1-\alpha c_2}}{\epsilon\rho - \frac{\beta c_2}{1-\alpha c_2}}\right), \tag{22}$$

then $\mathbb{P}(\xi_K \leq \epsilon) \geq 1 - \rho$.

Proof. First notice that the thresholded sequence $\{\xi_k^\epsilon\}_{k \geq 0}$ defined by

$$\xi_k^\epsilon = \begin{cases} 0, & \text{if } \xi_k < \epsilon, \\ \xi_k, & \text{otherwise,} \end{cases} \tag{23}$$

satisfies $\xi_k^\epsilon > \epsilon \Leftrightarrow \xi_k > \epsilon$. Therefore, by Markov's inequality, $\mathbb{P}(\xi_k > \epsilon) = \mathbb{P}(\xi_k^\epsilon > \epsilon) \leq \frac{\mathbb{E}[\xi_k^\epsilon]}{\epsilon}$. Letting $\theta_k := \mathbb{E}[\xi_k^\epsilon]$, it thus suffices to show that

$$\theta_K \leq \epsilon\rho. \tag{24}$$

The rationale behind this "thresholding trick" is that the sequence $\mathbb{E}[\xi_k^\epsilon]$ decreases faster than $\mathbb{E}[\xi_k]$ and hence will reach $\epsilon\rho$ sooner. Assume now that (i) holds. It can be shown (for example, see Theorem 1 of [28] for the case $\alpha = \beta = 0$) that

$$\mathbb{E}[\xi_{k+1}^\epsilon \mid x_k] \leq (1+\alpha)\xi_k^\epsilon - \frac{(\xi_k^\epsilon)^2}{c_1} + \beta, \quad \mathbb{E}[\xi_{k+1}^\epsilon \mid x_k] \leq \left(1 + \alpha - \frac{\epsilon}{c_1}\right)\xi_k^\epsilon + \beta. \tag{25}$$

By taking expectations in (25) (in x_k) and using Jensen's inequality, we get

$$\theta_{k+1} \leq (1+\alpha)\theta_k - \frac{\theta_k^2}{c_1} + \beta, \quad k \geq 0, \tag{26}$$

$$\theta_{k+1} \leq \left(1 + \alpha - \frac{\epsilon}{c_1}\right)\theta_k + \beta, \quad k \geq 0. \tag{27}$$

Notice that (26) is better than (27) precisely when $\theta_k > \epsilon$. It is easy to see that the inequality $(1+\alpha)\theta_k - \frac{\theta_k^2}{c_1} + \beta \leq \theta_k$ holds if and only if $\theta_k \geq u$. In other words, (26) leads to θ_{k+1} that is better than θ_k only for $\theta_k \geq u$. We will now compute $k = k_1$ for which $u < \theta_k \leq \epsilon$. Inequality (26) can be equivalently written as[3]

$$\theta_{k+1} - u \leq (1 - \sigma)(\theta_k - u) - \frac{(\theta_k - u)^2}{c_1}, \quad k \geq 0. \tag{28}$$

[2] We ignore the first term in the minimum if $\sigma = 0$; or, equivalently, treat $1/\sigma$ as ∞.

[3] We do this to eliminate the constant term β; this allows us to provide a simple analysis. Moreover, this "shifted" form leads to a better result; see the remarks after the Theorem for details.

where $\sigma < 1$. Letting $\hat{\theta}_k := \theta_k - u$, by monotonicity we have $\hat{\theta}_{k+1}\hat{\theta}_k \le \hat{\theta}_k^2$, whence

$$\frac{1-\sigma}{\hat{\theta}_{k+1}} - \frac{1}{\hat{\theta}_k} = \frac{(1-\sigma)\hat{\theta}_k - \hat{\theta}_{k+1}}{\hat{\theta}_{k+1}\hat{\theta}_k} \ge \frac{(1-\sigma)\hat{\theta}_k - \hat{\theta}_{k+1}}{\hat{\theta}_k^2} \overset{(28)}{\ge} \frac{1}{c_1}. \tag{29}$$

If we choose $r \in \{1, \frac{1}{1-\sigma}\}$, then

$$\frac{1}{\hat{\theta}_k} \overset{(29)}{\ge} r\left(\frac{1}{\hat{\theta}_{k-1}} + \frac{1}{c_1}\right) \ge r^k \frac{1}{\hat{\theta}_0} + \frac{1}{c_1}\sum_{j=1}^{k} r^j = \begin{cases} r^k\left(\frac{1}{\xi_0 - u} + \frac{1}{c_1\sigma}\right) - \frac{1}{c_1\sigma}, & r = \frac{1}{1-\sigma}, \\ \frac{1}{\xi_0 - u} + \frac{k}{c_1}, & r = 1. \end{cases}$$

In particular, using the above estimate with $r = 1$ *and* $r = \frac{1}{1-\sigma}$ gives

$$\hat{\theta}_{k_1} \le \epsilon - u \qquad \text{(and hence } \theta_{k_1} \le \epsilon\text{)} \tag{30}$$

for

$$k_1 := \min\left\{\left\lceil \log\left(\frac{\frac{1}{\epsilon-u} + \frac{1}{c_1\sigma}}{\frac{1}{\xi_0-u} + \frac{1}{c_1\sigma}}\right) / \log\left(\frac{1}{1-\sigma}\right)\right\rceil, \left\lceil \frac{c_1}{\epsilon} - \frac{c_1}{\xi_0 - u}\right\rceil\right\}, \tag{31}$$

where the left term in (31) applies when $\sigma > 0$ only.

Applying the inequalities (i) $\lceil t \rceil \le 1 + t$; (ii) $\log(\frac{1}{1-t}) \ge t$ (holds for $0 < t < 1$; we use the inverse version, which is surprisingly tight for small t); and (iii) the fact that $t \mapsto \frac{C+t}{D+t}$ is decreasing on $[0, \infty)$ if $C \ge D > 0$, we arrive at the following bound

$$k_1 \le 1 + \min\left\{\frac{1}{\sigma}\log\left(\frac{\xi_0 - u}{\epsilon - u}\right), \frac{c_1}{\epsilon} - \frac{c_1}{\xi_0 - u}\right\}. \tag{32}$$

Letting $\gamma := 1 - \frac{\epsilon - \alpha c_1}{c_1}$ (notice that $\gamma \in (0,1)$), for any $k_2 \ge 0$ we have

$$\begin{aligned} \theta_{k_1+k_2} &\overset{(27)}{\le} \gamma\theta_{k_1+k_2-1} + \beta \le \gamma^{k_2}\theta_{k_1} + \beta(\gamma^{k_2-1} + \gamma^{k_2-2} + \cdots + 1) \\ &\overset{(30)}{\le} \gamma^{k_2}\epsilon + \beta\frac{1-\gamma^{k_2}}{1-\gamma} = \gamma^{k_2}\left(\epsilon - \frac{\beta}{1-\gamma}\right) + \frac{\beta}{1-\gamma}. \end{aligned} \tag{33}$$

In (33), notice that the second to last term can be made as small as we like (by taking k_2 large), but we can never force $\theta_{k_1+k_2} \le \frac{\beta}{1-\gamma}$. Therefore, in order to establish (24), we need to ensure that $\frac{\beta c_1}{\epsilon - \alpha c_1} < \epsilon\rho$. Rearranging this gives the condition $\frac{c_1}{2}(\alpha + \sqrt{\alpha^2 + \frac{4\beta}{c_1\rho}}) < \epsilon$, which holds by assumption. Now we can find k_2 for which the right hand side in (33) is at most $\epsilon\rho$:

$$k_2 := \left\lceil \log\left(\frac{\epsilon - \frac{\beta}{1-\gamma}}{\epsilon\rho - \frac{\beta}{1-\gamma}}\right) / \log\left(\frac{1}{\gamma}\right)\right\rceil \le 1 + \frac{c_1}{\epsilon - \alpha c_1}\log\left(\frac{\epsilon - \frac{\beta c_1}{\epsilon - \alpha c_1}}{\epsilon\rho - \frac{\beta c_1}{\epsilon - \alpha c_1}}\right), \tag{34}$$

where we have used the inequality $t \le \log(\frac{1}{1-t})$ (which holds for $t \in [0,1)$ and is a good approximation for small t) with $t = 1 - \gamma$.

In view of (24), it is enough to take $K = k_1 + k_2$ iterations. The expression in (21) is obtained by adding the upper bounds on k_1 and k_2 in (32) and (34).

Now assume that property (ii) holds. By a similar argument as that leading to (25), we obtain

$$\begin{aligned}\theta_K \le \left(1-\tfrac{1-\alpha c_2}{c_2}\right)\theta_{K-1}+\beta &\le \left(1-\tfrac{1-\alpha c_2}{c_2}\right)^K\theta_0+\beta\sum_{j=0}^{K-1}\left(1-\tfrac{1-\alpha c_2}{c_2}\right)^j \\ &\le \left(1-\tfrac{1-\alpha c_2}{c_2}\right)^K\left(\theta_0-\tfrac{\beta c_2}{1-\alpha c_2}\right)+\tfrac{\beta c_2}{1-\alpha c_2} \overset{(22)}{\le} \epsilon\rho.\end{aligned}$$

The proof follows by taking K given by (22). □

Let us now comment on several aspects of the above result:

1. *Usage.* We will use Theorem 2 to finish the proofs of the complexity results in Section 4; with $\xi_k=\varphi(x_k):=F(x_k)-F^*$, where $\{x_k\}$ is the random process generated by ICD.

2. *Monotonicity and Nonnegativity.* Note that the monotonicity assumption in Theorem 2 is for the choice of x_k and φ described in 1) satisfied due to (20). Nonnegativity is satisfied automatically since $F(x_k)\ge F^*$ for all x_k.

3. *Best of two.* In (31), we notice that the first term applies when $\sigma>0$ only. If $\sigma=0$, then $u=0$, and subsequently the second term in (31) applies, which corresponds to the exact case. Notice that if $\sigma>0$ is very small (so $u\neq 0$), the iteration complexity result still may be better if the second term is used.

4. *Generalization.* Note that for $\alpha=\beta=0$, (21) recovers $\frac{c_1}{\epsilon}(1+\log\frac{1}{\rho})+2-\frac{c_1}{\xi_0}$, which is the result proved in Theorem 1(i) in [28], while (22) recovers $c_2\log((F(x_0)-F^*)/\epsilon\rho)$, which is the result proved in Theorem 1(ii) in [28]. Since the last term in (21) is negative, the theorem holds also if we ignore it. This is what we have done, for simplicity, in Table 1.

5. *High accuracy with high probability.* In the exact case, the iteration complexity results hold for any error tolerance $\epsilon>0$ and confidence $\rho\in(0,1)$. However, in the inexact case, there are restrictions on the choice of ρ and ϵ for which we can guarantee the result $\mathbb{P}(F(x_k)-F^*\le\epsilon)\ge 1-\rho$. Table 5 gives conditions on α and β under which arbitrary confidence level (i.e., small ρ) and accuracy (i.e., small ϵ) is achievable. For instance, if Theorem 1(ii) is used, then one can achieve arbitrary accuracy only if $\beta=0$, but arbitrary confidence under no assumptions on α and β. The situation with part (i) is worse: ϵ is lower bounded by a positive expression that involves ρ, unless $\alpha=\beta=0$.

	Theorem 2(i)	Theorem 2(ii)
ϵ can be arbitrarily small if	$\alpha=\beta=0$	$\beta=0$
ρ can be arbitrarily small if	$\beta=0$	any α, β

Table 2: The conditions under which arbitrary confidence ρ and accuracy ϵ are attainable.

6. *Two lower bounds on ϵ.* The inequality $\epsilon>\frac{c_1}{2}\left(\alpha+\sqrt{\alpha^2+\frac{4\beta}{\rho c_1}}\right)$ (see part (i) of Theorem 2) is equivalent to $\epsilon>\frac{\beta c_1}{\rho(\epsilon-\alpha c_1)}$. Note the similarity of the last expression and the lower bound on ϵ in part (ii) of the theorem. We can see that the lower bound on ϵ is smaller (and hence, is less restrictive) in (ii) than in (i), provided that $c_1=c_2$.

7. *Two analyses.* It can be seen that analyzing the "shifted" form (28) leads to a better result than analyzing (26) directly, even when $\beta = 0$. Consider the case $\beta = 0$, so that $\sigma = \alpha$ and $u = \alpha c_1$. From equation (29) $\theta_{k+1} \leq A := \alpha c_1 + (1-\alpha)/(\frac{1}{\theta_k - \alpha c_1} + \frac{1}{c_1})$, whereas analyzing equation (26) directly yields $\theta_{k+1} \leq B := (1+\alpha)/(\frac{1}{\theta_k} + \frac{1}{c_1})$. It can be shown that $A \leq B$, with equality if $\alpha = 0$.

4 Complexity Analysis: Convex Composite Objective

The following function plays a central role in our analysis:

$$H(x,T) := f(x) + \langle \nabla f(x), T \rangle + \tfrac{1}{2}\|T\|_l^2 + \Psi(x+T). \tag{35}$$

Comparing (35) with (16) using (2), (3), (6), (8) and (9) we get

$$H(x,T) = f(x) + \sum_{i=1}^{n} V_i(x, T^{(i)}). \tag{36}$$

It will be useful to establish inequalities relating H evaluated at the vector of exact updates $T_0(x)$ and H evaluated at the vector of inexact updates $T_\delta(x)$.

Lemma 3. *For all $x \in \mathbb{R}^N$ and $\delta \in \mathbb{R}^n_+$,*

$$H(x, T_0(x)) \leq H(x, T_\delta(x)) \leq H(x, T_0(x)) + \sum_{i=1}^{n} \delta^{(i)}. \tag{37}$$

Proof:

$$\begin{aligned}
H(x, T_0(x)) &\overset{(36)}{=} f(x) + \sum_{i=1}^{n} V_i(x, T_0^{(i)}(x)) \overset{(18)}{=} f(x) + \sum_{i=1}^{n} \min_{t \in \mathbb{R}_i} V_i(x,t) \\
&\leq f(x) + \sum_{i=1}^{n} V_i(x, T_\delta^{(i)}(x)) \overset{(36)}{=} H(x, T_\delta(x)) \\
&\overset{(18)}{\leq} f(x) + \sum_{i=1}^{n} \left(\delta^{(i)} + \min_{t \in \mathbb{R}_i} V_i(x,t) \right) \overset{(36)}{=} H(x, T_0(x)) + \sum_{i=1}^{n} \delta^{(i)}. \qquad \square
\end{aligned}$$

The following Lemma provides an upper bound on the expected distance between the current and optimal objective value in terms of the function H.

Lemma 4. *For $x, T \in \mathbb{R}^N$, let $x_+(x,T)$ be the random vector equal to $x + U_i T^{(i)}$ with probability $\frac{1}{n}$ for each $i \in \{1, 2, \ldots, n\}$. Then*

$$\mathbb{E}[F(x_+(x,T)) - F^* \mid x] \leq \tfrac{1}{n}(H(x,T) - F^*) + \tfrac{n-1}{n}(F(x) - F^*).$$

Proof:

$$\begin{aligned}
\mathbb{E}[F(x_+(x,T)) \mid x] &= \sum_{i=1}^{n} \tfrac{1}{n} F(x + U_i T^{(i)}) \\
&\overset{(15)}{\leq} \tfrac{1}{n} \sum_{i=1}^{n} [f(x) + V_i(x, T^{(i)}) + \psi_i(x)] \\
&\overset{(36)+(17)}{=} \tfrac{1}{n} H(x,T) + \tfrac{n-1}{n} f(x) + \tfrac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i} \Psi_j(x^{(j)}) \\
&= \tfrac{1}{n} H(x,T) + \tfrac{n-1}{n} F(x). \qquad \square
\end{aligned}$$

Note that if $x = x_k$ and $T = T_\delta(x_k)$, then $x_+(x,T) = x_{k+1}$, as produced by Algorithm 1. The following Lemma, which provides an upper bound on H, will be used repeatedly throughout the remainder of this paper.

Lemma 5. *For all $x \in \operatorname{dom} F$ and $\delta \in \mathbb{R}^n_+$ (letting $\Delta = \sum_i \delta^{(i)}$), we have*

$$H(x, T_\delta(x)) \leq \Delta + \min_{y \in \mathbb{R}^N} \left\{ F(y) + \tfrac{1-\mu_f(l)}{2} \|y - x\|_l^2 \right\}. \tag{38}$$

Proof:

$$\begin{aligned}
H(x, T_\delta(x)) &\overset{(37)}{\leq} \Delta + \min_{T \in \mathbb{R}^N} H(x,T) \\
&= \Delta + \min_{y \in \mathbb{R}^N} H(x, y - x) \qquad \text{(where } y = x + T\text{)} \\
&\overset{(35)}{=} \Delta + \min_{y \in \mathbb{R}^N} \{ f(x) + \langle \nabla f(x), y - x \rangle + \Psi(y) + \tfrac{1}{2} \|y - x\|_l^2 \} \\
&\overset{(10)}{\leq} \Delta + \min_{y \in \mathbb{R}^N} \{ f(y) - \tfrac{\mu_f(l)}{2} \|y - x\|_l^2 + \Psi(y) + \tfrac{1}{2} \|y - x\|_l^2 \}. \qquad \square
\end{aligned}$$

4.1 Convex case

Now we need to estimate $H(x, T_\delta(x)) - F^*$ from above in terms of $F(x) - F^*$.

Lemma 6. *Fix $x^* \in X^*$, $x \in \operatorname{dom} F$, $\delta \in \mathbb{R}^n_+$ and let $R = \|x - x^*\|_l$ and $\Delta = \sum_i \delta^{(i)}$. Then*

$$H(x, T_\delta(x)) - F^* \leq \Delta + \begin{cases} (1 - \frac{F(x) - F^*}{2R^2})(F(x) - F^*), & \text{if } F(x) - F^* \leq R^2, \\ \frac{1}{2} R^2 < \frac{1}{2}(F(x) - F^*), & \text{otherwise.} \end{cases} \tag{39}$$

Proof: Because strong convexity is not assumed, $\mu_f(l) = 0$, so

$$\begin{aligned}
H(x, T_\delta(x)) &\overset{(38)}{\leq} \Delta + \min_{y \in \mathbb{R}^N} \{ F(y) + \tfrac{1}{2} \|y - x\|_l^2 \} \\
&\leq \Delta + \min_{\lambda \in [0,1]} \{ F(\lambda x^* + (1 - \lambda) x) + \tfrac{\lambda^2}{2} \|x - x^*\|_l^2 \} \\
&\leq \Delta + \min_{\lambda \in [0,1]} \{ F(x) - \lambda (F(x) - F^*) + \tfrac{\lambda^2}{2} R^2 \}.
\end{aligned}$$

Minimizing in λ gives $\lambda^* = \min\{1, (F(x) - F^*)/R^2\}$ and the result follows. □

We now state the main complexity result of this section, which gives the number of iterations sufficient for ICD used with uniform probabilities to push the value of the objective within ϵ of the optimal value with probability at least $1 - \rho$.

Theorem 7. *Choose an initial point $x_0 \in \mathbb{R}^N$ and let $\{x_k\}_{k\geq 0}$ be the random iterates generated by ICD applied to problem (1), using uniform probabilities $p_i = \frac{1}{n}$ and inexactness parameters $\delta_k^{(1)}, \ldots, \delta_k^{(n)} \geq 0$ that satisfy (19) for $\alpha, \beta \geq 0$. Choose target confidence $\rho \in (0,1)$ and error tolerance $\epsilon > 0$ so that one of the following two conditions hold:*

(i) $\frac{c_1}{2}(\alpha + \sqrt{\alpha^2 + \frac{4\beta}{c_1\rho}}) < \epsilon < F(x_0) - F^*$ *and* $\alpha^2 + \frac{4\beta}{c_1} < 1$, *where* $c_1 = 2n\max\{\mathcal{R}_l^2(x_0), F(x_0) - F^*\}$,

(ii) $\frac{\beta c_2}{\rho(1-\alpha c_2)} < \epsilon < \min\{\mathcal{R}_l^2(x_0), F(x_0) - F^*\}$, *where* $c_2 = \frac{2n\mathcal{R}_l^2(x_0)}{\epsilon}$ *and* $\alpha c_2 < 1$.

If (i) holds and we choose K as in (21), or if (ii) holds and we choose K as in (22), then $\mathbb{P}(F(x_K) - F^* \leq \epsilon) \geq 1 - \rho$.

Proof. Since $F(x_k) \leq F(x_0)$ for all k by (20), we have $\|x_k - x^*\|_l \leq \mathcal{R}_l(x_0)$ for all k and $x^* \in X^*$. Using Lemma 4 and Lemma 6, we have

$$\begin{aligned}
\mathbb{E}[\xi_{k+1} \mid x_k] &\leq \bar{\delta}_k + \tfrac{1}{n}\max\left\{1 - \tfrac{\xi_k}{2\|x_k - x^*\|_l^2}, \tfrac{1}{2}\right\}\xi_k + \tfrac{n-1}{n}\xi_k \qquad (40)\\
&= \bar{\delta}_k + \max\left\{1 - \tfrac{\xi_k}{2n\|x_k - x^*\|_l^2}, 1 - \tfrac{1}{2n}\right\}\xi_k\\
&\leq \bar{\delta}_k + \max\left\{1 - \tfrac{\xi_k}{2n\mathcal{R}_l^2(x_0)}, 1 - \tfrac{1}{2n}\right\}\xi_k, \qquad (41)
\end{aligned}$$

where $\xi_k := F(x_k) - F^*$. Consider case (i). From (41) and (19) we obtain

$$\mathbb{E}[\xi_{k+1} \mid x_k] \leq \bar{\delta}_k + \left(1 - \tfrac{\xi_k}{c_1}\right)\xi_k \leq (1+\alpha)\xi_k - \tfrac{\xi_k^2}{c_1} + \beta, \qquad (42)$$

and the result follows by applying Theorem 2(i). Now consider case (ii). Notice that if $\xi_k \geq \epsilon$, then (41) together with (19), imply that

$$\mathbb{E}[\xi_{k+1} \mid x_k] \leq \bar{\delta}_k + \max\left\{1 - \tfrac{\epsilon}{2n\mathcal{R}_l^2(x_0)}, 1 - \tfrac{1}{2n}\right\}\xi_k \leq \left(1 + \alpha - \tfrac{1}{c_2}\right)\xi_k + \beta.$$

The result follows by applying Theorem 2(ii). □

4.2 Strongly convex case

Let us start with an auxiliary result.

Lemma 8. *Let F be strongly convex with respect to $\|\cdot\|_l$ with $\mu_f(l) + \mu_\Psi(l) > 0$. Then for all $x \in \operatorname{dom} F$ and $\delta \in \mathbb{R}_+^n$, with $\Delta = \sum_i \delta^{(i)}$, we have*

$$H(x, T_\delta(x)) - F^* \leq \Delta + \left(\tfrac{1-\mu_f(l)}{1+\mu_\Psi(l)}\right)(F(x) - F^*).$$

Proof: Let $\mu_f = \mu_f(l)$, $\mu_\Psi = \mu_\Psi(l)$ and $\lambda^* = (\mu_f + \mu_\Psi)(1 + \mu_\Psi) \leq 1$. Then using (38) we can upper-bound $H(x, T_\delta(x))$ as follows:

$$\begin{aligned} & \Delta + \min_{y \in \mathbb{R}^N} \{F(y) + \tfrac{1-\mu_f}{2}\|y - x\|_l^2\} \\ \leq \quad & \Delta + \min_{\lambda \in [0,1]} \{F(\lambda x^* + (1-\lambda)x) + \tfrac{(1-\mu_f)\lambda^2}{2}\|x - x^*\|_l^2\} \\ \overset{(11)+(13)}{\leq} \quad & \Delta + \min_{\lambda \in [0,1]} \{\lambda F^* + (1-\lambda)F(x) + \tfrac{(1-\mu_f)\lambda^2 - (\mu_f + \mu_\Psi)\lambda(1-\lambda)}{2}\|x - x^*\|_l^2\} \\ \leq \quad & \Delta + F(x) - \lambda^*(F(x) - F^*). \end{aligned}$$

The last inequality follows from the fact that $(\mu_f + \mu_\Psi)(1 - \lambda^*) - (1 - \mu_f)\lambda^* = 0$. It remains to subtract F^* from both sides of the final inequality. □

We can now estimate the number of iterations needed to push a strongly convex objective F within ϵ of the optimal value with high probability.

Theorem 9. *Let F be strongly convex with respect to the norm $\|\cdot\|_l$ with $\mu_f(l) + \mu_\Psi(l) > 0$ and let $\mu := \frac{\mu_f(l) + \mu_\Psi(l)}{1 + \mu_\Psi(l)}$. Choose an initial point $x_0 \in \mathbb{R}^N$ and let $\{x_k\}_{k \geq 0}$, be the random iterates generated by ICD applied to problem (1), used with uniform probabilities $p_i = \frac{1}{n}$ for $i = 1, 2, \ldots, n$ and inexactness parameters $\delta_k^{(1)}, \ldots, \delta_k^{(n)} \geq 0$ satisfying (19), for $0 \leq \alpha < \frac{\mu}{n}$ and $\beta \geq 0$. Choose confidence level $\rho \in (0, 1)$ and error tolerance ϵ satisfying $\frac{\beta n}{\rho(\mu - \alpha n)} < \epsilon < F(x_0) - F^*$. Then for K given by (22), we have $\mathbb{P}(F(x_K) - F^* \leq \epsilon) \geq 1 - \rho$.*

Proof. Letting $\xi_k = F(x_k) - F^*$, we have

$$\begin{aligned} \mathbb{E}[\xi_{k+1} \mid x_k] \quad & \overset{\text{(Lemma 4)}}{\leq} \quad \tfrac{1}{n}(H(x_k, T_{\delta_k}(x_k)) - F^*) + \tfrac{n-1}{n}\xi_k \\ & \overset{\text{(Lemma 8)}}{\leq} \quad \bar{\delta}_k + \tfrac{1}{n}\left(\tfrac{1-\mu_f(l)}{1+\mu_\Psi(l)}\xi_k\right) + \tfrac{n-1}{n}\xi_k \\ & \overset{(19)}{\leq} \quad \left(1 + \alpha - \tfrac{\mu}{n}\right)\xi_k + \beta. \end{aligned}$$

By (12), $\mu < 1$, and the result follows from Theorem 2(ii) with $c_2 = \frac{n}{\mu} > 1$. □

5 Complexity Analysis: Smooth Objective

In this section we provide simplified iteration complexity results when the objective function is smooth ($\Psi \equiv 0$ so $F \equiv f$). Furthermore, we will present complexity results for arbitrary (rather than uniform) probabilities $p_i > 0$.

5.1 Convex case

In the smooth exact case, we can write down a closed-form expression for the update:

$$T_0^{(i)}(x) \overset{(18)}{=} \arg\min_{t \in \mathbb{R}_i} V_i(x, t) \overset{(16)}{=} \arg\min_{t \in \mathbb{R}_i} \{\langle \nabla_i f(x), t\rangle + \tfrac{l_i}{2}\|t\|_{(i)}^2\} = -\tfrac{1}{l_i} B_i^{-1} \nabla_i f(x).$$

Substituting this into $V_i(x,\cdot)$ yields

$$V_i(x,T_0^{(i)}(x)) = \langle \nabla_i f(x), T_0^{(i)}(x)\rangle + \tfrac{l_i}{2}\|T_0^{(i)}(x)\|_{(i)}^2 = -\tfrac{1}{2l_i}(\|\nabla_i f(x)\|_{(i)}^*)^2. \tag{43}$$

We can now estimate the decrease in f during one iteration of ICD:

$$\begin{aligned} f(x+U_iT_\delta^{(i)}(x)) - f(x) &\overset{(7)}{\leq} \langle \nabla_i f(x), T_\delta^{(i)}(x)\rangle + \tfrac{l_i}{2}\|T_\delta^{(i)}(x)\|_{(i)}^2 \\ &\overset{(16)}{=} V_i(x,T_\delta^{(i)}(x)) \\ &\overset{(18)}{\leq} \min\{0, \delta^{(i)} + V_i(x,T_0^{(i)}(x))\} \\ &\overset{(43)}{=} \min\{0, \delta^{(i)} - \tfrac{1}{2l_i}\big(\|\nabla_i f(x)\|_{(i)}^*\big)^2\}. \end{aligned} \tag{44}$$

The main iteration complexity result of this section can be now established.

Theorem 10. *Choose an initial point $x_0 \in \mathbb{R}^N$ and let $\{x_k\}_{k\geq 0}$ be the random iterates generated by ICD applied to the problem of minimizing f, used with probabilities $p_1,\dots,p_n > 0$ and inexactness parameters $\delta_k^{(1)},\dots,\delta_k^{(n)} \geq 0$ satisfying (19) for $\alpha,\beta \geq 0$, where $\alpha^2 + \frac{4\beta}{c_1} < 1$ and $c_1 = 2\mathcal{R}_{lp^{-1}}^2(x_0)$. Choose target confidence $\rho \in (0,1)$, error tolerance ϵ satisfying $\frac{c_1}{2}(\alpha + \sqrt{\alpha^2 + \frac{4\beta}{c_1\rho}}) < \epsilon < f(x_0) - f^*$, and let the iteration counter K be given by (21). Then $\mathbb{P}(f(x_K) - f^* \leq \epsilon) \geq 1-\rho$.*

Proof. We first estimate the expected decrease of the objective function during one iteration of the method:

$$\begin{aligned} \mathbb{E}[f(x_{k+1}) \mid x_k] &= f(x_k) + \sum_{i=1}^n p_i [f(x_k + U_i T_{\delta_k}^{(i)}(x_k)) - f(x_k)] \\ &\overset{(44)}{\leq} f(x_k) + \sum_{i=1}^n p_i \left(\delta_k^{(i)} - \tfrac{1}{2l_i}\big(\|\nabla_i f(x_k)\|_{(i)}^*\big)^2\right) \\ &\overset{(9)}{=} f(x_k) - \tfrac{1}{2}\big(\|\nabla f(x_k)\|_{lp^{-1}}^*\big)^2 + \sum_{i=1}^n p_i \delta_k^{(i)} \\ &\leq f(x_k) - \tfrac{1}{2}\big(\|\nabla f(x_k)\|_{lp^{-1}}^*\big)^2 + \alpha(f(x_k) - f^*) + \beta. \end{aligned} \tag{45}$$

Since $f(x_k) \leq f(x_0)$ for all k,

$$f(x_k) - f^* \leq \max_{x^* \in X^*} \langle \nabla f(x_k), x_k - x^*\rangle \leq \|\nabla f(x_k)\|_{lp^{-1}}^* \mathcal{R}_{lp^{-1}}(x_0). \tag{46}$$

Substituting (46) into (45) we obtain

$$\mathbb{E}[f(x_{k+1}) - f^* \mid x_k] \leq f(x_k) - f^* - \tfrac{1}{2}\left(\tfrac{f(x_k)-f^*}{\mathcal{R}_{lp^{-1}}(x_0)}\right)^2 + \alpha(f(x_k) - f^*) + \beta. \tag{47}$$

It remains to apply Theorem 2(i). ∎

5.2 Strongly convex case

In this section we assume that f is strongly convex with respect to $\|\cdot\|_{lp^{-1}}$ with convexity parameter $\mu_f(lp^{-1})$. Using (10) with $x = x_k$ and $y = x^*$, and letting $h = x^* - x_k$, we obtain

$$\begin{aligned} f^* - f(x_k) &\geq \langle \nabla f(x_k), h\rangle + \tfrac{\mu_f(lp^{-1})}{2}\|h\|^2_{lp^{-1}} \\ &= \mu_f(lp^{-1})\left(\langle \tfrac{1}{\mu_f(lp^{-1})}\nabla f(x_k), h\rangle + \tfrac{1}{2}\|h\|^2_{lp^{-1}}\right). \end{aligned} \tag{48}$$

By minimizing the right hand side of (48), and rearranging, we obtain

$$f(x_k) - f^* \leq \frac{1}{2\mu_f(lp^{-1})}(\|\nabla f(x_k)\|^*_{lp^{-1}})^2. \tag{49}$$

We can now give an efficiency estimate for the case of a strongly convex objective.

Theorem 11. *Let f be strongly convex with respect to the norm $\|\cdot\|_{lp^{-1}}$ with convexity parameter $\mu_f(lp^{-1}) > 0$. Choose an initial point $x_0 \in \mathbb{R}^N$ and let $\{x_k\}_{k\geq 0}$ be the random iterates generated by ICD applied to the problem of minimizing f, used with probabilities $p_1, \ldots, p_n > 0$ and inexactness parameters $\delta_k^{(1)}, \ldots, \delta_k^{(n)} \geq 0$ that satisfy (19) for $0 \leq \alpha < \mu_f(lp^{-1})$ and $\beta \geq 0$. Choose the target confidence $\rho \in (0,1)$, let the target accuracy ϵ satisfy $\frac{\beta}{\rho(\mu_f(lp^{-1})-\alpha)} < \epsilon < f(x_0) - f^*$, let $c_2 = 1/\mu_f(lp^{-1})$ and let iteration counter K be as in (22). Then $\mathbb{P}(f(x_K) - f^* \leq \epsilon) \geq 1 - \rho$.*

Proof. The expected decrease of the objective function during one iteration of the method can be estimated as follows:

$$\begin{aligned} \mathbb{E}[f(x_{k+1}) - f^*|x_k] &\overset{(45)}{\leq} (1+\alpha)(f(x_k) - f^*) - \frac{1}{2}(\|\nabla f(x_k)\|^*_{lp^{-1}})^2 + \beta \\ &\overset{(49)}{\leq} (1 + \alpha - \mu_f(lp^{-1}))(f(x_k) - f^*) + \beta \end{aligned}$$

It remains to apply Theorem 2(ii) with $\varphi(x_k) = f(x_k) - f^*$ (and notice that $c_2 > 1$ by (12)). □

6 Inexact Updates in the Quadratic Case

The goal of the second part of this paper is to demonstrate the importance of employing an inexact update in the block coordinate descent method through the use of a specific example: the special case when f is quadratic, $\Psi = 0$, and the problem exhibits block angular structure. We motivate this choice now.

Suppose that $\Psi = 0$, so the function $F(x) = f(x)$ is smooth and convex. Then the overapproximation (7) becomes

$$F(x + U_i t) = f(x + U_i t) \leq f(x) + \langle \nabla_i f(x), t\rangle + \frac{L_i}{2}\langle B_i t, t\rangle. \tag{50}$$

At every iteration of the block coordinate descent method, the update t is found by determining the minimizer of the overapproximation (50), and this is equivalent to solving the system of equations

$$B_i t = -\frac{1}{L_i}\nabla_i f(x). \tag{51}$$

Clearly, solving systems of equations is central to the block coordinate descent method in the smooth case because a system of the form (51) must be solved at each iteration to determine the update to apply to the ith block. Furthermore, minimizing a quadratic function is equivalent to solving a system of equations, which is why it is a natural choice to study the case $\Psi = 0$ and f quadratic.

Further, this method is designed to accommodate blocks of data. Matrices with block structure arise in many areas, and often the structure is dictated by the application under consideration. In particular, there is a broad class of problems that involve matrices with block angular structure, originating from loosely coupled systems and producing nearly separable structures; see for example Dantzig [6]. Matrices with this structure commonly arise in optimization, from optimal control, scheduling and planning problems to stochastic optimization problems, and exploiting this structure is an active area of research [5, 10, 33]. This important example motivates the work here and we demonstrate that our theoretical developments provide a framework to specialize first order methods for such problems.

6.1 Problem setup

Suppose we have the following unconstrained quadratic minimization problem

$$\min_{x\in\mathbb{R}^N} f(x) = \frac{1}{2}\|Ax-b\|_2^2 \tag{52}$$

where $A \in \mathbb{R}^{M\times N}$, $b \in \mathbb{R}^M$ and $x \in \mathbb{R}^N$. Recall that U_i is defined in Section 2.1 so that $A_i = AU_i$, and consider the following

$$\begin{aligned} f(x+U_it) &= \frac{1}{2}\|A(x+U_it)-b\|_2^2 \\ &= \frac{1}{2}\|Ax-b\|_2^2 + \langle U_i^TA^T(Ax-b),t\rangle + \frac{1}{2}\|AU_it\|_2^2 \\ &= f(x) + \langle \nabla_i f(x),t\rangle + \frac{1}{2}\langle A_i^TA_it,t\rangle. \end{aligned} \tag{53}$$

Comparing (53) with (50), we see that in the quadratic case (53) is an exact upper bound on $f(x+U_it)$ if we choose $L_i = 1$ and $B_i = A_i^TA_i$ for all blocks $i = 1,\dots,n$. The matrix B_i is required to be (strictly) positive definite so A_i is assumed to have full (column) rank.[4]

Now, substituting $L_i = 1$ and $B_i = A_i^TA_i$ into (51) gives

$$A_i^TA_it = -A_i^T(Ax-b). \tag{54}$$

Therefore, at each iteration of ICD applied to problem (52), the update t is found by solving the system (54). The user defined matrix B_i is positive definite, (it must be so because it defines a norm; recall Section 2.1), so the exact solution to (51) is $(T_0^{(i)} \equiv)\, t = -\frac{1}{L_i}B_i^{-1}\nabla_i f(x)$ and a standard approach to solving (51) is to form the Cholesky factors of B_i followed by two triangular solves. However, armed with the iteration complexity results for ICD presented in the first part of this work, it is now possible to apply an iterative technique to (51) to find an *inexact* update

[4] If a block A_i does not have full column rank then we simply adjust our choice of L_i and B_i accordingly, although this means that we have an overapproximation to $f(x+U_it)$, rather than equality as in (53).

$(T_\delta^{(i)}(x) \equiv) t$. Because B_i is positive definite, a natural choice in the inexact case is to solve the system (54) using conjugate gradients [12, 35], and this is the method we adopt in what follows.

It is expected that an iterative technique will be faster than a direct method, so the update t can be determined more quickly, and subsequently the overall algorithm time reduces.

6.2 Block angular structure

Now suppose that $A \in \mathbb{R}^{M \times N}$ has block angular structure. Define

$$A = \left[\frac{C}{D}\right], \tag{55}$$

where the columns of A are partitioned into n blocks

$$C = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_n \end{bmatrix} \in \mathbb{R}^{m \times N} \tag{56}$$

and

$$D = \begin{bmatrix} D_1 & D_2 & \dots & D_n \end{bmatrix} \in \mathbb{R}^{\ell \times N}. \tag{57}$$

Furthermore, assume that each block C_i has size $M_i \times N_i$, and the linking blocks D_i have size $\ell \times N_i$, respectively. We assume that $\ell \leq N_i$ and subsequently $\ell \ll N$ and that there are n blocks with $m = \sum_{i=1}^n M_i$ so $M = m + \ell$, and $N = \sum_{i=1}^n N_i$.

Remark: Notice that if $D = \mathbf{0}$, where $\mathbf{0}$ is the $\ell \times N$ matrix of all zeros, then problem (52) is completely (block) separable so it can be solved easily. The linking constraints D make problem (52) nonseparable, which makes it difficult to solve.

A block of columns A_i has the form

$$A_i = \begin{bmatrix} \\ C_i \\ \\ D_i \end{bmatrix} \in \mathbb{R}^{M \times N_i} \tag{58}$$

so

$$B_i = A_i^T A_i = C_i^T C_i + D_i^T D_i, \tag{59}$$

and we can rewrite the system (54) as

$$(C_i^T C_i + D_i^T D_i)t = -A_i^T (Ax - b). \tag{60}$$

The system of equations (60) must be solved at each iteration of ICD because it determines the update to apply to the ith block. As previously mentioned, the system (60) can be solved inexactly using an iterative method and we use the conjugate gradients method in the numerical experiments presented in Section 8.

6.3 Preconditioning the update step

It is important that (60) be solved quickly, and one way to speed up an iterative method is to apply a preconditioner. It is commonly accepted that the conjugate gradient method often needs a good preconditioner to enable fast convergence and finding effective preconditioners for conjugate gradients is an active area of research; see for example [2, 9, 11].

In general, a sensible choice for a preconditioner is one that approximates the Hessian, in this case $B_i = A_i^T A_i$. We propose the preconditioner (for the ith system)

$$\mathcal{P}_i := C_i^T C_i. \tag{61}$$

If $M_i \geq N_i$ and $\text{rank}(C_i) = N_i$, then the block $C_i^T C_i$ is positive definite and therefore is nonsingular, so the preconditioner (61) is also nonsingular. Applying $\mathcal{P}_i^{-1}$ to B_i gives:

$$\mathcal{M}_i := \mathcal{P}_i^{-1} B_i = \mathcal{P}_i^{-1}(\mathcal{P}_i + D_i^T D_i) = I + \mathcal{P}_i^{-1} D_i^T D_i. \tag{62}$$

However, if $M_i < N_i$ then $\mathcal{P}_i$ defined in (61) is rank deficient and is therefore singular. To remedy this, when $M_i < N_i$ we perturb (61) by adding a positive multiple of the identity matrix, and propose the nonsingular preconditioner

$$\hat{\mathcal{P}}_i = \mathcal{P}_i + \rho I = C_i^T C_i + \rho I, \tag{63}$$

where $\rho > 0$. We have

$$\hat{\mathcal{M}}_i = \hat{\mathcal{P}}_i^{-1} B_i = \hat{\mathcal{P}}_i^{-1} \mathcal{P}_i + \hat{\mathcal{P}}_i^{-1} D_i^T D_i, \tag{64}$$

where

$$\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i = (C_i^T C_i + \rho I)^{-1} C_i^T C_i. \tag{65}$$

Applying the preconditioners (defined in (61) for $M_i \geq N_i$, and (63) for $M_i < N_i$) to (60), should result in the system having better spectral properties than the original, and this will lead to faster convergence of the conjugate gradient algorithm. Specifically, the preconditioner should shift the spectrum so that the eigenvalues of the preconditioned system are clustered around one, with few outliers. Studying the eigenvalues of the preconditioned matrix is the topic of the next section.

7 Eigenvalues of the preconditioned matrix

To investigate the quality of a preconditioner, we study the eigenvalues of the preconditioned matrices $\mathcal{M}_i$ and $\hat{\mathcal{M}}_i$ defined in (62) and (64), respectively. We will make use of the following simple result.

Theorem 12 (Theorem 2.8 in [44])**.** *Let A and B be $m \times n$ and $n \times m$ complex matrices, respectively. Then AB and BA have the same nonzero eigenvalues, counting multiplicity.*

Therefore, the nonzero eigenvalues of the $N_i \times N_i$ matrix $\mathcal{P}_i^{-1} D_i^T D_i$ are the same as the nonzero eigenvalues of $D_i \mathcal{P}_i^{-1} D_i^T$. We prefer to work with $D_i \mathcal{P}_i^{-1} D_i^T$ because it is symmetric and positive semidefinite, so it has real, nonnegative eigenvalues. (Furthermore, if D_i has full (row) rank, then $D_i \mathcal{P}_i^{-1} D_i^T$ is positive definite.)

Lemma 13. *Let $r_i = \text{rank}(D_i)$ and $r_i \leq N_i$. Then $\mathcal{P}_i^{-1} D_i^T D_i \in \mathbb{R}^{N_i \times N_i}$ has*

(i) r_i eigenvalues that are strictly positive,

(ii) $N_i - r_i$ eigenvalues equal to zero.

Consequently, we have the following result.

Theorem 14. *Let $r_i = \text{rank}(D_i)$ and $r_i \leq N_i$. Then $\mathcal{M}_i = I + \mathcal{P}_i^{-1} D_i^T D_i$ has*

(i) r_i eigenvalues that are strictly greater than one.

(ii) $N_i - r_i$ eigenvalues equal to one.

We can say more about the eigenvalues of $\mathcal{M}_i$ by considering the blocks of A and investigating the relationship between the matrices C and D, defined in (56) and (57), respectively. (Note that the eigenvalues of $\mathcal{P}_i^{-1} D_i^T D_i$ can be determined exactly by solving the generalized eigenvalue problem $D_i^T D_i v = \lambda \mathcal{P}_i v$.)

Recall that the blocks along the diagonal of C are $C_i \in \mathbb{R}^{M_i \times N_i}$. The remainder of this section is broken into two parts. The first part considers the case when $M_i \geq N_i$ while the second part considers the case when $M_i < N_i$. In each case C_i is assumed to have full rank.

7.1 Skyscraper shaped blocks

Here it is assumed that the blocks C_i have size $M_i \times N_i$ where $M_i \geq N_i$, and that $N_i = \text{rank}(C_i)$ (i.e., C_i has full column rank).

The preconditioner (61) is applied to the system (60) and we are interested in the distribution of the eigenvalues of the preconditioned matrix $\mathcal{M}_i = \mathcal{P}_i^{-1} B_i$. Subsequently, we study the eigenvalues of $\mathcal{P}_i^{-1} D_i^T D_i$.

We consider the general case when the matrix $D_i \in \mathbb{R}^{\ell \times N_i}$ with $1 \leq \ell < N_i$. The matrix C_i is assumed to have full rank, so the rows of C_i contain a basis for $\mathbb{R}^{N_i}$. Subsequently, each row in the linking matrix D_i is a linear combination of the rows of C_i; i.e., for $Z_i \in \mathbb{R}^{\ell \times M_i}$ we can write

$$D_i = Z_i C_i. \tag{66}$$

Lemma 15. *Let $\mathcal{P}_i \in \mathbb{R}^{N_i \times N_i}$ and $D_i \in \mathbb{R}^{\ell \times N_i}$ be the matrices defined in* (61) *and* (66) *respectively and let z_j^T denote the jth row of Z_i. Let $C_i = Y_i R_i$ denote the thin QR factorization of C_i, so $Y_i \in \mathbb{R}^{M_i \times N_i}$ has orthonormal columns and $R_i \in \mathbb{R}^{N_i \times N_i}$ is upper triangular [8]. Then*

$$\text{trace}(D_i \mathcal{P}_i^{-1} D_i^T) \;=\; \sum_{j=1}^{\ell} \|z_j^T Y_i\|_2^2 \;\leq\; \|Z_i\|_F^2. \tag{67}$$

Proof. The trace is simply the sum of the diagonal entries of a (square) matrix, so consider the diagonal elements of $D_i \mathcal{P}_i^{-1} D_i^T$.

$$D_i \mathcal{P}_i^{-1} D_i = Z_i C_i (C_i^T C_i)^{-1} C_i^T Z_i^T = Z_i Y_i R_i (R_i^T Y_i^T Y_i R_i)^{-1} R_i^T Y_i^T Z_i^T = (Z_i Y_i)(Z_i Y_i)^T$$

The jth diagonal element of $D_i \mathcal{P}_i^{-1} D_i^T$ can be written as

$$(D_i \mathcal{P}_i^{-1} D_i^T)_{jj} = \|Y_i^T z_j\|_2^2.$$

Furthermore, $\|Z_i\|_F^2 = \sum_{j=1}^{\ell} \|z_j\|_2^2$. Because $Y_i Y_i^T$ is a projection matrix,

$$\|Y_i^T z_j\|_2^2 = \|Y_i Y_i^T z_j\|_2^2 \leq \|z_j\|_2^2,$$

and the result follows. □

Remark: When C_i is square and has full rank, Y_i is an orthogonal matrix, and subsequently $\text{trace}(D_i \mathcal{P}_i^{-1} D_i^T) = \sum_{j=1}^{\ell} \|z_j\|_2^2 = \|Z_i\|_F^2$.

Theorem 16. *Suppose that the matrix $A \in \mathbb{R}^{M \times N}$ has primal block angular structure, with rectangular blocks $C_i \in \mathbb{R}^{M_i \times N_i}$ ($M_i \geq N_i$) of full rank ($N_i = \text{rank}\,(C_i)$) along the diagonal. Suppose that B_i, D_i and $\mathcal{P}_i$ are defined in* (59), (66) *and* (61) *respectively, and let $r_i = \text{rank}(D_i)$ where $r_i \leq N_i$. Then $\mathcal{P}_i^{-1} B_i$ has*

(i) $N_i - r_i$ eigenvalues equal to one,

(ii) r_i eigenvalues that are strictly greater than 1, and sum to $r_i + \sum_{j=1}^{\ell} \|Y_i^T z_j\|_2^2$.

7.2 Warehouse shaped blocks

Now it is assumed that the blocks C_i have size $M_i \times N_i$ where $M_i < N_i$, and that each block has full (row) rank, $M_i = \text{rank}(C_i)$.

For the block coordinate descent method, the matrix $B_i = A_i^T A_i$ must have full rank because it defines a norm (see Section 2.1). However, when C_i is warehouse-shaped, $\mathcal{P}_i$ defined in (61) is rank deficient so we use the preconditioner $\hat{\mathcal{P}}_i$ defined in (63).

Recall that in this case, the preconditioned matrix is defined in (64). We study the eigenvalues of $\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i$ and $\hat{\mathcal{P}}_i^{-1} D_i^T D_i$ separately, before stating the main result of this section, which describes the eigenvalues of $\hat{\mathcal{M}}_i$.

We begin by describing the eigenvalues of $\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i$ (65).

Theorem 17. *Let C_i be a real $M_i \times N_i$ matrix with $M_i < N_i$ and full row rank $M_i = \text{rank}(C_i)$. Let $\mathcal{P}_i$ and $\hat{\mathcal{P}}_i$ be defined in* (61) *and* (63) *respectively, and let $M_i = \text{rank}(\mathcal{P}_i)$. Then $\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i$ has $N_i - M_i$ zero eigenvalues and M_i positive eigenvalues that tend to 1 as $\rho \to 0$.*

Proof. The matrix $\hat{\mathcal{P}}_i$ has full rank so

$$\text{rank}(\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i) = \text{rank}(\mathcal{P}_i) = M_i.$$

Therefore, $\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i$ has M_i nonzero eigenvalues and $N_i - M_i$ zero eigenvalues. Furthermore, the M_i nonzero eigenvalues are positive. Indeed, $\hat{\mathcal{P}}_i^{-1}$ and $\mathcal{P}_i$ are both symmetric positive semidefinite matrices, and by Theorem 12, the nonzero eigenvalues of $\hat{\mathcal{P}}_i^{-1} \mathcal{P}_i$ are the same as the nonzero eigenvalues of $\hat{\mathcal{P}}_i^{-\frac{1}{2}} \mathcal{P}_i \hat{\mathcal{P}}_i^{-\frac{1}{2}}$. The latter matrix is clearly positive semidefinite so its eigenvalues are nonnegative.

Let $C_i = U \Sigma V^T$ denote the singular value decomposition of C_i, and let $\lambda_1, \ldots, \lambda_{M_i}$ denote the M_i nonzero eigenvalues of $\mathcal{P}_i$. Then we can write $\mathcal{P}_i = C_i^T C_i = V \Lambda V^T$, where

$$\Lambda = \Sigma^T \Sigma = \begin{bmatrix} \Lambda_1 & \\ & \mathbf{0} \end{bmatrix} \qquad \text{and} \qquad \Lambda_1 = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{M_i} \end{bmatrix},$$

and $\mathbf{0}$ denotes the $(N_i - M_i) \times (N_i - M_i)$ matrix of all zeros.

Define the matrix $\hat{\Lambda}_1 \in R^{M_i \times M_i}$ and its inverse as follows:

$$\hat{\Lambda}_1 = \Lambda_1 + \rho I = \begin{bmatrix} \lambda_1 + \rho & & \\ & \ddots & \\ & & \lambda_{M_i} + \rho \end{bmatrix}, \quad \hat{\Lambda}_1^{-1} = \begin{bmatrix} \frac{1}{\lambda_1+\rho} & & \\ & \ddots & \\ & & \frac{1}{\lambda_{M_i}+\rho} \end{bmatrix}. \tag{68}$$

Now consider the preconditioner $\hat{\mathcal{P}}_i = \mathcal{P}_i + \rho I$, which has the following singular value decomposition:

$$\hat{\mathcal{P}}_i = \mathcal{P}_i + \rho I = V \hat{\Lambda} V^T \qquad \text{and} \qquad \hat{\mathcal{P}}_i^{-1} = (\mathcal{P}_i + \rho I)^{-1} = V \hat{\Lambda}^{-1} V^T \tag{69}$$

where

$$\hat{\Lambda} = \begin{bmatrix} \hat{\Lambda}_1 & \\ & \rho I \end{bmatrix} \quad \text{and} \quad \hat{\Lambda}^{-1} = \begin{bmatrix} \hat{\Lambda}_1^{-1} & \\ & \frac{1}{\rho} I \end{bmatrix}$$

and $\hat{\Lambda}_1$ and $\hat{\Lambda}_1^{-1}$ are defined in (68). Then

$$\begin{aligned} \hat{\mathcal{P}}_i^{-1}\mathcal{P}_i = V \hat{\Lambda}^{-1} V^T V \Lambda V^T &= V \begin{bmatrix} \frac{1}{\lambda_1+\rho} & & & \\ & \ddots & & \\ & & \frac{1}{\lambda_{M_i}+\rho} & \\ & & & \frac{1}{\rho} I \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_{M_i} & \\ & & & \mathbf{0} \end{bmatrix} V^T \\ &= V \begin{bmatrix} \frac{\lambda_1}{\lambda_1+\rho} & & & \\ & \ddots & & \\ & & \frac{\lambda_{M_i}}{\lambda_{M_i}+\rho} & \\ & & & \mathbf{0} \end{bmatrix} V^T \end{aligned}$$

and as $\rho \to 0$, $\frac{\lambda_j}{\lambda_j+\rho} \to 1$ for $j = 1, \ldots, M_i$. □

Remark: This is a very useful result because it also shows that $\hat{\mathcal{P}}_i^{-1}\mathcal{P}_i$ is a symmetric matrix. (We know that, in general, the product of two symmetric matrices does not have to be symmetric. However, this result shows that, for a general matrix A, if A is a symmetric matrix then $(A+\rho I)^{-1}A$ is also symmetric.)

In what follows, we make use of the following simple idea. We assume that the matrix C_i has full (row) rank, so the rows of C_i form a basis for a subspace $\mathcal{W} = \text{span}\{c_1^{(i)}, \ldots, c_{M_i}^{(i)}\} \subset \mathbb{R}^{N_i}$, where $(c_j^{(i)})^T$ is the jth row of C_i. Let $\mathcal{W}^\perp$ denote the orthogonal complement of $\mathcal{W}$. Any vector $v \in \mathbb{R}^{N_i}$ can be expressed as

$$v = w + w^\perp, \qquad \text{where} \qquad w \in \mathcal{W}, \;\; w^\perp \in \mathcal{W}^\perp. \tag{70}$$

Recall that the blocks $C_i \in \mathbb{R}^{M_i \times N_i}$ are warehouse-shaped ($M_i < N_i$) and have full (row) rank: $M_i = \text{rank}(C_i)$. Suppose that the matrix $D_i \in \mathbb{R}^{\ell \times N_i}$ where $\ell \geq N_i - M_i$ and that $N_i = \text{rank}(A_i)$ to ensure that B_i has full rank. Furthermore, let W be an $\ell \times N_i$ matrix whose rows $w_j^T \in \mathcal{W}$, for $j = 1, \ldots, \ell$, and let $W^\perp$ be an $\ell \times N_i$ matrix whose rows $(w_j^\perp)^T \in \mathcal{W}^\perp$, for $j = 1, \ldots, \ell$. Then one can write

$$D_i = W + W^\perp. \tag{71}$$

Recall the preconditioned matrix $\mathcal{M}_i$ in (64) and notice that it remains to study the eigenvalues of $\hat{\mathcal{P}}_i^{-1} D_i^T D_i$.

By Theorem 12, the nonzero eigenvalues of $\hat{\mathcal{P}}_i^{-1} D_i^T D_i$ are equivalent to the eigenvalues of $D_i \hat{\mathcal{P}}_i^{-1} D_i^T$ and we prefer to work with this small, symmetric, positive definite matrix (D_i has full row rank by assumption). Now we study the diagonal elements of $D_i \hat{\mathcal{P}}_i^{-1} D_i^T$.

$$\begin{aligned}(D_i \hat{\mathcal{P}}_i^{-1} D_i^T)_{jj} &= (w_j + w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} (w_j + w_j^\perp) \\ &= w_j^T \hat{\mathcal{P}}_i^{-1} w_j + 2(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} w_j + (w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} (w_j^\perp). \end{aligned} \tag{72}$$

Recall the eigenvalue decomposition (69) and let $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ be a partitioning of V, where $V_1 \in R^{N_i \times M_i}$, and $V_2 \in R^{N_i \times (N_i - M_i)}$. Now consider the first term on the right hand side of (72):

$$w_j^T \hat{\mathcal{P}}_i^{-1} w_j = w_j^T V \hat{\Lambda}^{-1} V^T w_j = \left[\begin{array}{c|c} w_j^T V_1 & w_j^T V_2 \end{array}\right] \left[\begin{array}{c|c} \hat{\Lambda}_1^{-1} & \\ \hline & \frac{1}{\rho} I \end{array}\right] \left[\begin{array}{c} V_1^T w_j \\ \hline V_2^T w_j \end{array}\right].$$

The vector w is a linear combination of the rows of C_i and the columns of V_2 form a basis for the null space of C_i, so

$$w_j^T \hat{\mathcal{P}}_i^{-1} w_j = \left[\begin{array}{c|c} w_j^T V_1 & 0 \end{array}\right] \left[\begin{array}{c|c} \hat{\Lambda}_1^{-1} & \\ \hline & \frac{1}{\rho} I \end{array}\right] \left[\begin{array}{c} V_1^T w_j \\ \hline 0 \end{array}\right] = w_j^T V_1 \hat{\Lambda}_1^{-1} V_1^T w_j. \tag{73}$$

This is important because $\frac{1}{\rho} \to \infty$ as $\rho \to 0$, but because $V_2^T w_j = 0$, none of the components in $V^T w_j$ grow too large, so neither does the term $w_j^T \hat{\mathcal{P}}_i^{-1} w_j$. Similarly

$$\begin{aligned}(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} w_j &= \left[\begin{array}{c|c} (w_j^\perp)^T V_1 & (w_j^\perp)^T V_2 \end{array}\right] \left[\begin{array}{c|c} \hat{\Lambda}_1^{-1} & \\ \hline & \frac{1}{\rho} I \end{array}\right] \left[\begin{array}{c} V_1^T w_j \\ \hline 0 \end{array}\right] \\ &= (w_j^\perp)^T V_1 \hat{\Lambda}_1^{-1} V_1^T w_j \end{aligned} \tag{74}$$

so $(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} w_j$ also remains independent of the parameter ρ. However, the last term in (72) becomes

$$\begin{aligned}(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} (w_j^\perp) &= \left[\begin{array}{c|c} (w_j^\perp)^T V_1 & (w_j^\perp)^T V_2 \end{array}\right] \left[\begin{array}{c|c} \hat{\Lambda}_1^{-1} & \\ \hline & \frac{1}{\rho} I \end{array}\right] \left[\begin{array}{c} V_1^T (w_j^\perp) \\ \hline V_2^T (w_j^\perp) \end{array}\right] \\ &= (w_j^\perp)^T V_1 \hat{\Lambda}_1^{-1} V_1^T (w_j^\perp) + \frac{1}{\rho} (w_j^\perp)^T V_2 V_2^T (w_j^\perp) \end{aligned} \tag{75}$$

and as $\rho \to 0$, $(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} (w_j^\perp) \to \infty$.

Combining (73), (74) and (75) we can rewrite (72) in the following way

$$\begin{aligned}(D_i \hat{\mathcal{P}}_i^{-1} D_i^T)_{jj} &= (w_j + w_j^\perp)^T V_1 \hat{\Lambda}_1^{-1} V_1^T (w_j + w_j^\perp) + \frac{1}{\rho} (w_j^\perp)^T V_2 V_2^T (w_j^\perp) \\ &= \|\hat{\Lambda}_1^{-\frac{1}{2}} V_1^T (w_j + w_j^\perp)\|_2^2 + \frac{1}{\rho} \|V_2^T w_j^\perp\|_2^2. \end{aligned} \tag{76}$$

This demonstrates that the choice of the parameter ρ is very important. There is a trade-off here: ρ should not be too small or $(w_j^\perp)^T \hat{\mathcal{P}}_i^{-1} (w_j^\perp)$ will become arbitrarily large, but a small value of ρ will lead to a good clustering of the eigenvalues around one (Theorem 17),

Before we say more about the eigenvalues of $\hat{\mathcal{P}}_i^{-1} D_i^T D_i$, we present the following result.

Theorem 18 (Theorem 4.3.1 in [13]). *Let A and B be $N \times N$ Hermitian matrices and let the eigenvalues $\lambda_i(A)$, $\lambda_i(B)$ and $\lambda_i(A+B)$ be arranged in increasing order ($\lambda_{\min} = \lambda_1 \leq \cdots \leq \lambda_N = \lambda_{\max}$). For each $k = 1, 2, \ldots, N$ we have*

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_N(B).$$

Now we state the main result of this section.

Theorem 19. *Let C_i be an $M_i \times N_i$ matrix with $M_i < N_i$ and $M_i = \operatorname{rank}(C_i)$ and let D_i be an $\ell \times N_i$ matrix with $r_i = \operatorname{rank}(D_i)$. Let $\hat{\mathcal{P}}_i$ be the preconditioner defined in* (63) *and let A_i and B_i be defined in* (58) *and* (59) *respectively with $N_i \geq s_i = \operatorname{rank}(A_i)$. Then $\mathcal{M}_i = \hat{\mathcal{P}}_i^{-1} B_i$ has*

(i) $N_i - s_i$ eigenvalues equal to zero.

(ii) $s_i - r_i$ eigenvalues in the interval $(0, 1)$

(iii) r_i eigenvalues in the interval

$$\left(1, 1 + \sum_{j=1}^{\ell} \left(\|\hat{\Lambda}_1^{-\frac{1}{2}} V_1^T (w_j + w_j^{\perp})\|_2^2 + \frac{1}{\rho} \|V_2^T (w_j^{\perp})\|_2^2 \right) \right)$$

Proof. Part (i) holds because B_i is $N_i \times N_i$ with $\operatorname{rank}(B_i) = \operatorname{rank}(A_i) = s_i$. Part (ii) follows from Theorem 17 and Theorem 18. For part (iii), notice that $\lambda_{\max}(D_i \hat{\mathcal{P}}_i^{-1} D_i^T) \leq \operatorname{trace}(D_i \hat{\mathcal{P}}_i^{-1} D_i^T)$. Now using (76) and Theorem 18 gives the result. □

Remark: For the ICD method, we require that $\operatorname{rank}(A_i) = N_i$, because this ensures that B_i is a positive definite matrix. Notice that in this case, Theorem 19 explains that all eigenvalues of $\mathcal{M}_i = \hat{\mathcal{P}}_i^{-1} B_i$ are strictly greater than zero (i.e., $N_i = s_i$).

Figure 1 shows the distribution of eigenvalues before and after application of the preconditioner. In the plot on the left the eigenvalues of both B_i and $\mathcal{P}_i^{-1} B_i$ are shown, where C_i is 200×100 (skyscraper-shaped) and D_i is 10×100. In the plot on the right, the eigenvalues of both B_i and $\hat{\mathcal{P}}_i^{-1} B_i$ are shown, where C_i is 450×500 (warehouse-shaped) and D_i is 50×500 with $\rho = 10^{-4}$. In both cases the distribution of the eigenvalues is greatly improved after application of the preconditioner. The eigenvalues are clustered around one with few outliers. Further, in the right hand plot, notice that the largest eigenvalue is of the order $\frac{1}{\rho}$, as expected from Theorem 19.

8 Numerical Experiments

In this section we present small-scale preliminary numerical results to demonstrate the practical performance of Inexact Coordinate Descent applied to the problem described in Section 6.1. Specifically, we assume that the function $F = f$ is quadratic, $\Psi = 0$ and the matrix A has block angular structure with $n = 10$ blocks. The vector x was constructed and $b = Ax$ was computed and i.i.d noise (at a level of 1%) was added to the last ℓ components of b.

Each experiment (to be described shortly) was implemented in MATLAB and was performed 100 times, with the average result presented in the Tables 3 and 4.

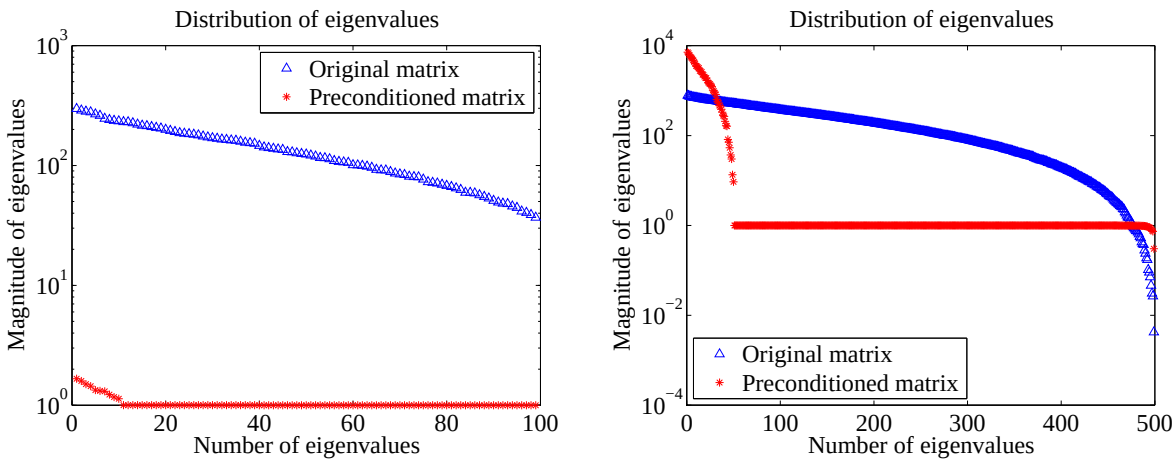

Figure 1: Plots showing the distribution of eigenvalues before and after preconditioning. In the left plot the matrix C_i is 200×100 (skyscraper-shaped) and D_i is 10×100, while in the right plot C_i is 450×500 (warehouse-shaped) and D_i is 50×500. The distribution of eigenvalues improves greatly after preconditioning with clustering around 1.

The stopping criterion was chosen to be $\|Ax - b\|_2^2 / \|b\|_2^2 <$ tol where 'tol' is a user defined tolerance that was set to tol$= 10^{-4}$.

Each of the matrices involved is sparse with the density of C_i set to approximately $10^{-3}\, M_i\, N_i + N_i$, and the density of the linking constraints D_i set to approximately $0.5\, \ell\, N_i$.

The purpose of each experiment was to study the use of an iterative technique (with and without preconditioning) to determine the update used at each iteration of the block coordinate descent method.

In the first experiment each of the blocks C_i is skyscraper shaped with $M_i = 1250$, $N_i = 1000$ and D_i has 1, 10 or 100 rows. The results are shown in the Table 3.

Table 3: Experiment 1: Skyscraper shaped blocks. All times and iteration counts are average over 100 runs.

	ℓ	Time(s)	Its	Inner its
PCG	1	2.8	854.4	2.2
	10	12.5	2136.8	4.8
	100	49.0	2756.5	8.2
CG	1	4.6	857.0	4.6
	10	18.4	2146.3	6.2
	100	65.9	2769.4	6.9
Direct solve	1	15.0	853.3	—
	10	81.0	2134.2	—
	100	135.5	2754.2	—

In the second experiment the blocks C_i are warehouse shaped with $N_i = 1000$ and M_i and ℓ varying (while ensuring that $M_i + \ell \geq N_i$ and A_i has full rank so that B_i is positive definite). For the preconditioner (63), $\rho = 10^{-2}$. The results are shown in the Table 4.

Table 4: Experiment 2: Warehouse shaped blocks. Times and iteration counts are average over 100 runs.

	ℓ	Time	Its	Inner its	$\|r\|_2^2/\|b\|_2^2$
PCG	2	2.31	1000	1.37	0.18×10^{-4}
	50	5.62	1000	1.01	0.18×10^{-4}
	100	39.78	1000	1.07	0.10×10^{-4}
CG	2	2.28	1000	1.01	0.21×10^{-4}
	50	7.20	1000	1	0.48×10^{-4}
	100	42.37	1000	1.01	0.11×10^{-4}
Direct	2	6.20	1000	—	0.07×10^{-6}
solve	50	6.22	1000	—	0.20×10^{-6}
	100	27.21	1000	—	0.23×10^{-6}

9 Conclusion

In this work we introduce Inexact Coordinate Descent, which is a block coordinate descent method that uses an inexact update. Iteration complexity results are presented to show that the algorithm is guaranteed to converge with high probability when applied to a convex composite function (1). The theoretical results were complemented by practical considerations in the second half of this paper. Because an inexact update is allowed, iterative techniques can be used to determine the update to apply at each iteration and the advantages were highlighted by studying the quadratic case where block angular structure was present. The numerical results presented at the end of this work strongly support ICD.

References

[1] G. C. Bento, J. X. Da Cruz Neto, P. R. Oliveira, and A. Soubeyran. The self regulation problem as and inexact steepest descent method for multicriteria optimization. Technical report, July 2012. arXiv:1207.0775v1 [math.OC].

[2] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, pages 1–137, 2005.

[3] S. Bonettini. Inexact block coordinate descent methods with application to non-negative matrix factorization. *IMA Journal of Numerical Analysis*, 31:1431–1452, 2011.

[4] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6):111–119, 2012.

[5] J. Castro and J. Cuesta. Quadratic regularizations in an interior-point method for primal block-angular problems. *Math. Program., Ser. A*, 130:415–445, 2011.

[6] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[7] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289 – 1306, April 2006.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1986.

[9] G. H. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comput.*, 21(4):1305–1320, 1999.

[10] J. Gondzio and R. Sarkissian. Parallel interior-point solver for structured linear programs. *Mathematical Programming*, 96(3):561–584, 2003.

[11] S. Gratton, A. Sartenaer, and J. Tshimanga. On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. *SIAM J. Optim.*, 21(3):912–935, 2011.

[12] M. R. Hestenes and E. Steifel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1952.

[13] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[14] X. Hua and N. Yamashita. An inexact coordinate descent method for the weighted l_1-regularized convex optimization problem. Technical report, School of Mathematics and Physics, Kyoto University, Kyoto 606-8501, Japan, September 2012.

[15] D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: Convergence rates and conditioning. *Mathematics of Operations Research*, 2010.

[16] P. Machart, S. Anthoine, and L. Baldassarre. Optimal computational trade-off of inexact proximal methods. Technical report, INRIA, 00704398, October 2012. Version 3.

[17] I. Necoara and V. Nedelcu. Inexact dual gradient methods with guaranteed primal feasibility: application to distributed MPC. Technical report, Politehnica University of Bucharest, 060042 Bucharest, Romania, September 2012.

[18] I. Necoara, Y. Nesterov, and F. Glineur. Efficiency of randomized coordinate descent methods on optimization problems with linearly coupled constraints. Technical report, June 2012. pp.1–21.

[19] I. Necoara and A. Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Technical report, University Politehnica Bucharest, Spl. Independentei 313, Romania, June 2012. pp.1–20.

[20] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50:395–403, 2010.

[21] D. Needell and J. Tropp. Paved with good intentions: Analysis of a randomized Kaczmarz method. Technical report, August 2012. ArXiv:1208.3805vl [math.NA].

[22] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Kluwer Academic Publishers, 2004.

[23] Y. Nesterov. Gradient methods form minimizing composite objective function. Technical report, Core discussion paper #2007/76, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), September 2007.

[24] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22(2):341–362, 2012.

[25] Z. Qin, K. Scheinberg, and D. Goldfarb. Efficient block-coordinate descent algorithms for the group lasso. Technical report, Department of Industrial Engineering and Operations Research, Columbia University, 2010.

[26] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. Technical report, Computer Sciences Department, University of Wisconsin-Madison, 1210 W Dayton St, Madison, WI 53706, April 2011.

[27] P. Richtárik and M. Takáč. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In Diethard Klatte, Hans-Jakob Lūthi, and Karl Schmedders, editors, *Operations Research Proceedings 2011*, Operations Research Proceedings, pages 27–32. Springer Berlin Heidelberg, 2012.

[28] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming, DOI: 10.1007/s10107-012-0614-z*, pages 1–38, 2012.

[29] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. Technical report, November 2012. arXiv:1212.0873.

[30] P. Richtárik and M. Takáč. Efficiency of randomized coordinate descent methods on minimization problems with a composite objective function. In *4th Workshop on Signal Processing with Adaptive Sparse Structured Representations*, June 2011.

[31] A. Saha and A. Tewari. On the finite time convergence of cyclic coordinate descent methods. Technical report, May 2010. arXiv:1005.2146v1 [cs.LG].

[32] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. Technical report, INRIA, 00618152, December 2011.

[33] G. L. Schultz and R. R. Meyer. An interior point method for block angular optimization. *SIAM J. Optim.*, 1(4):583–602, 1991.

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

[35] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonising pain. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, August 1994.

[36] N. Simon and R. Tibshirani. Standardization and the group lasso penalty. Technical report, Stanford University, March 2011.

[37] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15:262–278, 2009.

[38] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, 2013.

[39] Q. Tao, K. Kong, D. Chu, and G. Wu. Stochastic coordinate descent methods for regularized smooth and nonsmooth losses. In P. A. Flach, T. De Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *Lecture Notes in Computer Science*, pages 537–552. Springer, 2012.

[40] P. Tseng. Convergence of block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, June 2001.

[41] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program., Ser. B*, 117:387–423, 2009.

[42] S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM J. Optim.*, 22(1):159–186, 2012.

[43] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *Trans. Sig. Proc.*, 57:2479–2493, July 2009.

[44] F. Zhang. *Matrix Theory: Basic Results and Techniques*. Springer, 1999.