On the Finite Time Convergence of Cyclic Coordinate Descent Methods

Ankan Saha
Department of Computer Science
University of Chicago
ankans@cs.uchicago.edu

Ambuj Tewari
Toyota Technological Institute
Chicago, USA
tewari@ttic.edu

Abstract

Cyclic coordinate descent is a classic optimization method that has witnessed a resurgence of interest in machine learning. Reasons for this include its simplicity, speed and stability, as well as its competitive performance on ℓ_1 regularized smooth optimization problems. Surprisingly, very little is known about its finite time convergence behavior on these problems. Most existing results either just prove convergence or provide asymptotic rates. We fill this gap in the literature by proving $O(1/k)$ convergence rates (where k is the iteration counter) for two variants of cyclic coordinate descent under an isotonicity assumption. Our analysis proceeds by comparing the objective values attained by the two variants with each other, as well as with the gradient descent algorithm. We show that the iterates generated by the cyclic coordinate descent methods remain better than those of gradient descent uniformly over time.

1 Introduction

The dominant paradigm in Machine Learning currently is to cast learning problems as optimization problems. This is clearly borne out by approaches involving empirical risk *minimization*, *maximum* likelihood, *maximum* entropy, *minimum* description length, etc. As machine learning faces ever increasing and high-dimensional datasets, we are faced with novel challenges in designing and analyzing optimization algorithms that can adapt efficiently to such datasets. A mini-revolution of sorts is taking place where algorithms that were "slow" or "old" from a purely optimization point of view are witnessing a resurgence of interest. This paper considers one such family of algorithms, namely the *coordinate descent* methods. There has been recent work demonstrating the potential of these algorithms for solving ℓ_1-regularized loss minimization problems:

$$\frac{1}{n}\sum_{i=1}^{n}\ell(x, Z_i) + \lambda\|x\|_1 \tag{1}$$

where x is possibly high dimensional predictor that is being learned from the samples $Z_i = (X_i, Y_i)$ consisting of input, output pairs, ℓ is a convex loss function measuring prediction performance, and $\lambda \geq 0$ is a "regularization" parameter. The use of the ℓ_1 norm $\|x\|_1$ (sum of absolute values of x_i) as a "penalty" or "regularization term" is motivated by its sparsity promoting properties and there is a large and growing literature studying such issues (see, e.g., Tropp (2006) and references therein). In this paper, we restrict ourselves to analyzing the behavior of coordinate descent methods on problems like (1) above. The general idea behind coordinate descent is to choose, at each iteration, an index j and change x_j such that objective F decreases. Choosing j can be as simple as cycling through the coordinates or a more sophisticated coordinate selection rule can be employed. Friedman et al. (2007) use the cyclic rule which we analyze in this paper.

Our emphasis is on obtaining *finite time* rates, i.e. guarantees about accuracy of iterative optimization algorithms that hold right from the first iteration. This is in contrast to asymptotic guarantees that only hold once the iteration count is "large enough" (and often, what is meant by "large enough", is left unspecified). We feel such an emphasis is in the spirit of Learning Theory that has distinguished itself by regarding finite sample generalization bounds as important. For our analysis, we abstract away the particulars of the setting above, and view (1) as a special case of the convex optimization problem:

$$\min_{x\in\mathbb{R}^d} \; F(x) := f(x) + \lambda\|x\|_1 \; . \tag{2}$$

[0] Eligible for Student Paper Award

In order to obtain finite time convergence rates, one must assume that f is "nice" is some sense. This can be quantified in different ways including assumptions of Lipschitz continuity, differentiability or strong convexity. We will assume that f is differentiable with a Lipschitz continuous gradient. In the context of problem (1), it amounts to assuming that the loss ℓ is differentiable. Many losses, such as squared loss and logistic loss, are differentiable. Our results therefore apply to ℓ_1 regularized squared loss ("Lasso") and to ℓ_1 regularized logistic regression.

For a method as old as cyclic coordinate descent, it is surprising that little is known about finite time convergence even under smoothness assumptions. As far as we know, finite time results are not available even when $\lambda = 0$. i.e. for unconstrained smooth convex minimization problem. Given recent empirical successes of the method, we feel that this gap in the literature needs to be filled urgently. In fact, this sentiment is shared in (Wu & Lange (2008)) by the authors who lamented, "Better understanding of the convergence properties of the algorithms is sorely needed." They were talking about greedy coordinate descent methods but their comment applies to cyclic methods as well.

The situation with gradient descent methods is much better. There are a variety of finite time convergence results available in the literature (Nesterov (2003)). Our strategy in this paper is to leverage these results to shed some light on the convergence of coordinate descent methods. We do this via a series of comparison theorems that relate variants of coordinate descent methods to each other and to the gradient descent algorithm. To do this, we make assumptions both on the starting point and an additional *isotonicity* assumption on the gradient of the function f. Since finite time $O(1/k)$ accuracy guarantees are available for gradient descent, we are able to prove the same rates for two variants of cyclic coordinate descent. Here k is the iteration count and the constants hidden in the $O(\cdot)$ notation are small and known. We feel it should be possible to relax, or even eliminate, the additional assumptions we make (these are detailed in section 4) and doing this is an important open problem left for future work.

We find it important to state at the outset that our aim here is not to give the best possible rates for the problem (2). For example, even among gradient-based methods, faster $O(1/k^2)$ finite time accuracy bounds can be achieved using Nesterov's celebrated 1983 method (Nesterov (1983)) or its later variants. Instead, our goal is to better understand cyclic coordinate descent methods and their relationship to gradient descent.

Related Work Coordinate descent methods are quite old and we cannot attempt a survey here. Instead, we refer the reader to Tseng (2001) and Tseng & Yun (2009b) that summarize previous work and also present analyses for coordinate descent methods. These consider cyclic coordinate descent as well as versions that use more sophisticated coordinate selection rules. However, as mentioned above, the analyses either establish convergence without rates or give asymptotic rates that hold after sufficiently many iterations have occurred. An exception is Tseng & Yun (2009a) that does give finite time rates but for a version of coordinate descent that is not cyclic. Finite time guarantees for a greedy version (choosing j to be the coordinate of the current gradient with the maximum value) also appear in Clarkson (2008). The author essentially considers minimizing a smooth convex function over the probability simplex and also surveys previous work on greedy coordinate descent in that setting. For finite time (expected) accuracy bounds for stochastic coordinate descent (choose j uniformly at random) for ℓ_1 regularization, see Shalev-Shwartz & Tewari (2009).

We mentioned that the empirical success reported in Friedman et al. (2007) was our motivation to consider cyclic coordinate descent for ℓ_1 regularized problems. They consider the Lasso problem:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2n} \|\mathbf{X}x - Y\|^2 + \lambda \|x\|_1 \,, \tag{3}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^n$. In this case, the smooth part f is a quadratic

$$f(x) = \tfrac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle \tag{4}$$

where $A = \mathbf{X}^\top \mathbf{X}$ and $b = -\mathbf{X}^\top Y$. Note that A is symmetric and positive semidefinite. Cyclic coordinate descent has also been applied to the ℓ_1-regularized logistic regression problem (Genkin et al., 2007). Since the logistic loss is differentiable, this problem also falls into the framework of this paper.

Outline Notation and necessary definitions are given in section 2. The gradient descent algorithm along with two variants of cyclic coordinate descent are presented in section 3. Section 4 spells out the additional assumptions on f that our current analysis needs. It also proves results comparing the iterates generated by the three algorithms considered in the paper when they are all started from the same point. Similar comparison theorems in the context of solving a system of non-linear equations using Jacobi and Gauss-Seidel methods appear in Rheinboldt (1970). The results in section 4 set the stage for the main results given in section 5. This section converts the comparison between iterates into a comparison between objective function values achieved by the iterates. The finite time convergence rates of cyclic coordinate descent are then inferred from rates for gradient descent. There are plenty of issues that are still unresolved. Section 6 discusses some of them and provides a conclusion.

2 Preliminaries and Notation

We use the lowercase letters x, y, z, g and γ to refer to vectors throughout the paper. Normally parenthesized superscripts, like $x^{(k)}$ refer to vectors as well, whereas subscripts refer to the components of the corresponding vectors. For any positive integer k, $[k] := \{1, \dots, k\}$. $\operatorname{sign}(a)$ is the interval-valued sign function, i.e. $\operatorname{sign}(a) = \{1\}$ or $\{-1\}$ corresponding to $a > 0$ or $a < 0$. For $a = 0$, $\operatorname{sign}(a) = [-1, 1]$. Unless otherwise specified, $\|\cdot\|$ refers to the Euclidean norm $\|x\| := \left(\sum_i x_i^2\right)^{\frac{1}{2}}$, $\|\cdot\|_1$ will denote the l_1 norm, $\|x\|_1 = (\sum_i |x_i|)$, $\langle \cdot, \cdot \rangle$ denotes the Euclidean dot product $\langle x, y \rangle = \sum_i x_i y_i$. Through out the paper inequalities between vectors are to be interpreted component wise *i.e.* $x \geq y$ means that $x_i \geq y_i$ for all $i \in [d]$. The following definition will be used extensively in the paper:

Definition 1 *Suppose a function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable on $\mathbb{R}^d$. Then f is said to have Lipschitz continuous gradient (l.c.g) with respect to a norm $\|\cdot\|$ if there exists a constant L such that*

$$\|\nabla f(x) - \nabla f(x')\| \leq L\|x - x'\| \qquad \forall\, x, x' \in \mathbb{R}^d. \tag{5}$$

An important fact (see, e.g., (Nesterov, 2003, Thm. 2.1.5)) we will use is that if a function f has Lipschitz continuous gradient with respect to a norm $\|\cdot\|$, then it satisfies the following generalized bounded Hessian property

$$f(x) \leq f(x') + \langle \nabla f(x'), x - x' \rangle + \frac{L}{2}\|x - x'\|^2. \tag{6}$$

An operator $T : \mathbb{R}^d \to \mathbb{R}$ is said to be *isotone* iff

$$x \geq y \quad \Rightarrow \quad T(x) \geq T(y). \tag{7}$$

An important isotone operator that we will frequently deal with is the *shrinkage* operator $\mathbf{S}_\tau : \mathbb{R}^d \to \mathbb{R}$ defined, for $\tau > 0$, as

$$[\mathbf{S}_\tau(x)]_i := S_\tau(x_i) \tag{8}$$

where $S_\tau(a)$ is the scalar shrinkage operator:

$$S_\tau(a) := \begin{cases} a - \tau & a > \tau \\ 0 & a \in [-\tau, \tau] \\ a + \tau & a < -\tau. \end{cases} \tag{9}$$

3 Algorithms

We will consider three iterative algorithms for solving the minimization problem (2). All of them enjoy the descent property: $F(x^{(k+1)}) \leq F(x^{(k)})$ for successive iterates $x^{(k)}$ and $x^{(k+1)}$.

Algorithm 1: Gradient Descent (GD)

```
Initialize: Choose an appropriate initial point $x^{(0)}$.
for $k = 0, 1, \dots$ do
    $x^{(k+1)} \leftarrow \mathbf{S}_{\lambda/L}(x^{(k)} - \frac{\nabla f(x^{(k)})}{L})$
end for
```

Algorithm 1, known as Gradient Descent (GD), is one of the most common iterative algorithms used for convex optimization (See Beck & Teboulle (2009), Duchi & Singer (2009) and references therein). It is based on the idea that using corollary (6) to generate a linear approximation of f at the current iterate $x^{(k)}$, we can come up with the following global upper approximation of F:

$$F(x) \leq f(x^{(k)}) + \left\langle \nabla f(x^{(k)}), x - x^{(k)} \right\rangle + \frac{L}{2}\|x - x^{(k)}\|^2 + \lambda\|x\|_1 \,.$$

It is easy to show that the above approximation is minimized at $x = \mathbf{S}_{\lambda/L}(x^{(k)} - \nabla f(x^{(k)})/L)$ (Beck & Teboulle (2009)). This is the next iterate for the GD algorithm. We call it "Gradient Descent" as it reduces to the following algorithm

$$x^{(k+1)} = x^{(k)} - \frac{\nabla f(x^{(k)})}{L}$$

when there is no regularization (i.e. $\lambda = 0$). Finite time convergence rate for the GD algorithm are well known.

Theorem 2 *Let $\{x^{(k)}\}$ be a sequence generated by the* GD *algorithm. Then, for any minimizer $x^\star$ of* (2)*, and $\forall k \geq 1$,*

$$F(x^{(k)}) - F(x^\star) \leq \frac{L\|x^\star - x^{(0)}\|^2}{2\,k}$$

The above theorem can be found in, e.g., (Beck & Teboulle, 2009, Thm. 3.1).

Algorithm 2: Cyclic Coordinate Descent (CCD)

```
Initialize: Choose an appropriate initial point $y^{(0)}$.
for $k = 0, 1, \dots$ do
  $y^{(k,0)} \leftarrow y^{(k)}$
  for $j = 1$ to $d$ do
    $y_j^{(k,j)} \leftarrow S_{\lambda/L}(y_j^{(k,j-1)} - [\nabla f(y^{(k,j-1)})]_j \,/\, L)$
    $\forall i \neq j,\ y_i^{(k,j)} \leftarrow y_i^{(k,j-1)}$
  end for
  $y^{(k+1)} \leftarrow y^{(k,d)}$
end for
```

The second algorithm, Cyclic Coordinate Descent (CCD), instead of using the current gradient to update all components simultaneously, goes through them in a cyclic fashion. The next "outer" iterate $y^{(k+1)}$ is obtained from $y^{(k)}$ by creating a series of d intermediate or "inner" iterates $y^{(k,j)}$, $j \in [d]$, where $y^{(k,j)}$ differs from $y^{(k,j-1)}$ only in the jth coordinate whose value can be found by minimizing the following one-dimensional over-approximation of F over the scalar α:

$$f(y^{(k,j-1)}) + \lambda \sum_{i \neq j} |y_i^{(k,j-1)}| + [\nabla f(y^{(k,j-1)})]_j \cdot (\alpha - y_j^{(k,j-1)}) + \frac{L}{2}(\alpha - y^{(k,j-1)})_j^2 + \lambda|\alpha| \,. \qquad (10)$$

It can again be verified that the above minimization has the closed form solution

$$\alpha = S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right)$$

which is what CCD chooses $y_j^{(k,j)}$ to be. Once all coordinates have been cycled through, $y^{(k+1)}$ is simply set to be $y^{(k,d)}$. Let us point out that in an actual implementation, the inner iterates $y^{(k,j)}$ would not be computed separately but $y^{(k)}$ would be updated "in place". For analysis purposes, it is convenient to give names to the intermediate iterates. Note that for all $j \in \{0, 1, \dots, d\}$, the inner iterate looks like

$$y^{(k,j)} = \left[y_1^{(k+1)}, \dots, y_j^{(k+1)}, y_{j+1}^{(k)}, \dots, y_d^{(k)}\right] \,.$$

In the CCD algorithm updating the jth coordinate uses the newer gradient value $\nabla f(y^{(k,j-1)})$ rather than $\nabla f(y^{(k)})$ which is used in GD. This makes CCD inherently sequential. In contrast, different coordinate updates in GD can easily be done by different processors in parallel. However, on a single processor, we might hope CCD converges faster than GD due to the use of "fresh" information. Therefore, it is natural to expect that CCD should enjoy the finite time convergence rate given in Theorem 2 (or better). We show this is indeed the case under an *isotonicity assumption* stated in Section 4 below. Under the assumption, we are actually able to show the correctness of the intuition that CCD should converge faster than GD.

The third and final algorithm that we consider is Cyclic Coordinate Minimization (CCM). The only way it differs from CCD is that instead of minimizing the one-dimensional over-approximation (10), it chooses $z_j^{(k,j)}$ to minimize,

$$F(z_1^{(k,j-1)}, \dots, z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \dots, z_d^{(k,j-1)})$$

over α. In a sense, CCM is not actually an algorithm as it does not specify how to minimize F for any arbitrary smooth function f. An important case when the minimum can be computed exactly is when f is quadratic as in (4). In that case, we have

$$z_j^{(k,j)} = S_{\lambda/A_{j,j}}\left(z_j^{(k,j-1)} - \frac{[Az^{(k,j-1)} + b]_j}{A_{j,j}}\right) \,.$$

If there is no closed form solution, then we might have to resort to numerical minimization in order to implement CCM. This is usually not a problem since one-dimensional convex functions can be minimized

Algorithm 3: Cyclic Coordinate Minimization

```
Initialize: Choose an appropriate initial point $z^{(0)}$.
for $k = 0, 1, \ldots$ do
  $z^{(k,0)} \leftarrow z^{(k)}$
  for $j = 1$ to $d$ do
    $z_j^{(k,j)} \leftarrow \text{argmin}_\alpha\ F(z_1^{(k,j-1)}, \ldots, z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \ldots, z_d^{(k,j-1)})$
    $\forall i \neq j,\ z_i^{(k,j)} \leftarrow z_i^{(k,j-1)}$
  end for
  $z^{(k+1)} \leftarrow z^{(k,d)}$
end for
```

numerically to an extremely high degree of accuracy in a few steps. For the purpose of analysis, we will assume that an exact minimum is found. Again, intuition suggests that the accuracy of CCM after any fixed number of iterations should be better than that of CCD since CCD only minimizes an over-approximation. Under the same isotonicity assumption that we mentioned above, we can show that this intuition is indeed correct.

We end this section with a cautionary remark regarding terminology. In the literature, CCM appears much more frequently than CCD and it is actually the former that is often referred to as "Cyclic Coordinate Descent" (See Friedman et al. (2007) and references therein). Our reasons for considering CCD are: (i) it is a nice, efficient alternative to CCM, and (ii) a stochastic version of CCD(where the coordinate to update is chosen randomly and not cyclically) is already known to enjoy finite time $O(1/k)$ expected convergence rate (Shalev-Shwartz & Tewari (2009)).

4 Analysis

We already mentioned the known convergence rate for GD (Theorem 2) above. Before delving into the analysis, it is necessary to state an assumption on f which accompanied by appropriate starting conditions results in particularly interesting properties of the convergence behavior of GD, as described in lemma 7. The GD algorithm generates iterates by applying the operator

$$T_{GD}(x) := \mathbf{S}_{\lambda/L}\left(x - \frac{\nabla f(x)}{L}\right) \tag{11}$$

repeatedly. It turns out that if T_{GD} is an isotone operator then the GD iterates satisfy lemma 7 which is essential for our convergence analysis. The above operator is a composition of $\mathbf{S}_{\lambda/L}$, an isotone operator, and $\mathbf{I} - \nabla f/L$ (where $\mathbf{I}$ denotes the identity operator). To ensure overall isotonicity, it suffices to assume that $\mathbf{I} - \nabla f/L$ is isotone. This is formally stated as:

Assumption 3 *The operator $x \mapsto x - \frac{\nabla f(x)}{L}$ is isotone.*

Similar assumptions appear in the literature comparing Jacobi and Gauss-Seidel methods for solving linear equations (Bertsekas & Tsitsiklis, 1989, Chap. 2). When the function f is quadratic as in (4), our assumption is equivalent to assuming that the off-diagonal entries in A are non-positive, i.e. $A_{i,j} \leq 0$ for all $i \neq j$. For a general smooth f, the following condition is sufficient to make the assumption true: f is twice-differentiable and the Hessian $\nabla^2 f(x)$ at any point x has non-positive off-diagonal entries.

In the next few subsections, we will see how the isotonicity assumption leads to an isotonically decreasing (or increasing) behavior of GD, CCD and CCM iterates under appropriate starting conditions. To specify what these starting conditions are, we need the notions of super- and subsolutions.

Definition 4 *A vector x is a supersolution iff $x \geq \mathbf{S}_\lambda(x - \nabla f(x))$. Analogously, x is a subsolution iff $x \leq \mathbf{S}_\lambda(x - \nabla f(x))$.*

Since the inequalities above are vector inequalities, an arbitrary x may neither be a supersolution nor a subsolution. The names "supersolution" and "subsolution" are justified because equality holds in the definitions above, *i.e.* $x = \mathbf{S}_\lambda(x - \nabla f(x))$ iff x is a minimizer of F. To see this, note that subgradient optimality conditions say that x is a minimizer of $F = f + \lambda\|\cdot\|_1$ iff for all $j \in [d]$

$$0 \in [\nabla f(x)]_j + \lambda\,\text{sign}(x_j)\ . \tag{12}$$

Further, it is easy to see that,

$$\forall a, b \in \mathbb{R},\ \tau > 0, \qquad 0 \in b + \lambda\,\text{sign}(a) \qquad \Leftrightarrow \qquad a = S_{\lambda/\tau}(a - b/\tau) \tag{13}$$

We prove a couple of properties of super- and subsolutions that will prove useful later. The first property refers to the scale invariance of the definition of super- and subsolutions and the second property is the monotonicity of a single variable function.

Lemma 5 *If for any $\tau > 0$,*

$$x \geq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right) \tag{14}$$

then x is a supersolution. If x is a supersolution then the above inequality holds for all $\tau > 0$.

Similarly, if for any $\tau > 0$,

$$x \leq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right)$$

then x is a subsolution. If x is a subsolution then the above inequality holds for all $\tau > 0$.

Proof: See Appendix B ■

Lemma 6 *If x is a supersolution (resp. subsolution) then for any j, the function*

$$\tau \mapsto S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right)$$

is monotonically nondecreasing (resp. nonincreasing).

Proof: See Appendix C ■

4.1 Gradient Descent

Lemma 7 *If $x^{(0)}$ is a supersolution and $\left\{x^{(k)}\right\}$ is the sequence of iterates generated by the* GD *algorithm then $\forall k \geq 0$,*

1) $x^{(k+1)} \leq x^{(k)}$ 2) $x^{(k)}$ *is a supersolution*

If $x^{(0)}$ is a subsolution and $\left\{x^{(k)}\right\}$ is the sequence of iterates generated by the GD *algorithm then $\forall k \geq 0$,*

1) $x^{(k+1)} \geq x^{(k)}$ 2) $x^{(k)}$ *is a subsolution*

Proof: We only prove the supersolution case. The proof for the subsolution case is analogous. We start with a supersolution $x^{(0)}$. Consider the operator

$$T_{GD}(x) := \mathbf{S}_{\lambda/L}\left(x - \frac{\nabla f(x)}{L}\right)$$

given by (11). By the isotonicity assumption, T_{GD} is an isotone operator. We will prove by induction that $T_{GD}(x^{(k)}) \leq x^{(k)}$. This proves that $x^{(k+1)} \leq x^{(k)}$ since $x^{(k+1)} = T_{GD}(x^{(k)})$. Using lemma 5, the second claim follows by the definition of the T_{GD} operator.

The base case $T_{GD}(x^{(0)}) \leq x^{(0)}$ is true by Lemma 5 since $x^{(0)}$ is given to be a supersolution. Now assume $T_{GD}(x^{(k)}) \leq x^{(k)}$. Applying the isotone operator T_{GD} on both sides we get $T_{GD}(T_{GD}(x^{(k)})) \leq T_{GD}(x^{(k)})$. This is the same as $T_{GD}(x^{(k+1)}) \leq x^{(k+1)}$ by definition of $x^{(k+1)}$ which completes our inductive claim.

■

4.2 Cyclic Coordinate Descent (CCD)

Lemma 8 *If $y^{(0)}$ is a supersolution and $\left\{y^{(k)}\right\}$ is the sequence of iterates generated by the* CCD *algorithm then $\forall k \geq 0$,*

1) $y^{(k+1)} \leq y^{(k)}$ 2) $y^{(k)}$ *is a supersolution*

If y_0 is a subsolution and $\left\{y^{(k)}\right\}$ is the sequence of iterates generated by the CCD *algorithm then $\forall k \geq 0$,*

1) $y^{(k+1)} \geq y^{(k)}$ 2) $y^{(k)}$ *is a subsolution*

Proof: We will only prove the supersolution case as the subsolution proof is analogous. We start with a supersolution $y^{(0)}$. We will prove the following: If $y^{(k)}$ is a supersolution then,

$$y^{(k+1)} \leq y^{(k)} \ , \tag{15}$$

$$y^{(k+1)} \text{ is a supersolution} \tag{16}$$

Then the lemma follows by induction on k. Let us make the induction assumption that $y^{(k)}$ is a supersolution and try to prove (15) and (16). To prove these, we will show that $y^{(k,j)} \leq y^{(k)}$ and $y^{(k,j)}$ is a supersolution by induction on $j \in \{0, 1, \dots, d\}$. This proves (15) and (16) for $y^{(k+1)}$ since $y^{(k+1)} = y^{(k,d)}$.

For the base case ($j = 0$) of the induction, note that $y^{(k,0)} \leq y^{(k)}$ is trivial since the two vectors are equal. For the same reason, $y^{(k,0)}$ is a supersolution since we have assumed $y^{(k)}$ to be a supersolution. Now assume $y^{(k,j-1)} \leq y^{(k)}$ and $y^{(k,j-1)}$ is a supersolution for some $j > 0$. We want to show that $y^{(k,j)} \leq y^{(k)}$ and $y^{(k,j)}$ is a supersolution.

Since $y^{(k,j-1)}$ and $y^{(k,j)}$ differ only in the jth coordinate, to show that $y^{(k,j)} \leq y^{(k)}$ given $y^{(k,j-1)} \leq y^{(k)}$, it suffices to show that $y^{(k,j)} \leq y^{(k,j-1)}$, i.e.

$$y_j^{(k,j)} \leq y_j^{(k,j-1)} = y_j^{(k)} \ . \tag{17}$$

Since $y^{(k,j-1)} \leq y^{(k)}$ applying the isotone operator $\mathbf{I} - \nabla f / L$ on both sides and taking the jth coordinate gives,

$$y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \leq y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}$$

Applying the scalar shrinkage operator on both sides gives,

$$S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L}\right) \leq y_j^{(k)}$$

The left hand side is $y_j^{(k,j)}$ by definition while the second inequality follows because $y^{(k)}$ is a supersolution. Thus, we have proved (17).

Now we prove that $y^{(k,j)}$ is a supersolution. Note that we have already shown $y^{(k,j)} \leq y^{(k,j-1)}$. Applying the isotone operator $\mathbf{I} - \frac{\nabla f}{L}$ on both sides gives,

$$y_j^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_j}{L} \leq y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \ , \tag{18}$$

$$\forall i \neq j, \ y_i^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_i}{L} \leq y_i^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_i}{L} \ . \tag{19}$$

Applying a scalar shrinkage on both sides of (18) gives,

$$S_{\lambda/L}\left(y_j^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) \ .$$

Since the right hand side is $y_j^{(k,j)}$ by definition, we have,

$$S_{\lambda/L}\left(y_j^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_j}{L}\right) \leq y_j^{(k,j)} \ . \tag{20}$$

For $i \neq j$, we have

$$\begin{aligned} y_i^{(k,j)} = y_i^{(k,j-1)} &\geq S_{\lambda/L}\left(y_i^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_i}{L}\right) \\ &\geq S_{\lambda/L}\left(y_i^{(k,j)} - \frac{[\nabla f(y^{(k,j)})]_i}{L}\right) \ . \end{aligned} \tag{21}$$

The first inequality above is true because $y^{(k,j-1)}$ is a supersolution (by Induction Assumption) (and Lemma 5). The second follows from (19) by applying a scalar shrinkage on both sides. Combining (20) and (21), we get

$$y^{(k,j)} \geq \mathbf{S}_{\lambda/L}\left(y^{(k,j)} - \frac{\nabla f(y^{(k,j)})}{L}\right)$$

which proves, using Lemma 5, that $y^{(k,j)}$ is a supersolution.

■

4.3 Comparison: GD vs. CCD

Theorem 9 *Suppose $\{x^{(k)}\}$ and $\{y^{(k)}\}$ are the sequences of iterates generated by the* GD *and* CCD *algorithms respectively when started from the same supersolution $x^{(0)} = y^{(0)}$. Then, $\forall k \geq 0$,*

$$y^{(k)} \leq x^{(k)} .$$

On the other hand, if they are started from the same subsolution $x^{(0)} = y^{(0)}$ then the sequences satisfy, $\forall k \geq 0$,

$$y^{(k)} \geq x^{(k)} .$$

Proof: We will prove lemma 9 only for the supersolution case by induction on k. The base case is trivial since $y^{(0)} = x^{(0)}$. Now assume $y^{(k)} \leq x^{(k)}$ and we will prove $y^{(k+1)} \leq x^{(k+1)}$. Fix a $j \in [d]$. Note that we have,

$$y_j^{(k+1)} = y_j^{(k,j)} = S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \right) .$$

By Lemma 8, $y^{(k,j-1)} \leq y^{(k)}$. Applying the isotone operator $S_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ on both sides and taking the jth coordinate gives,

$$S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L} \right) \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L} \right) .$$

Combining this with the previous equation gives,

$$y_j^{(k+1)} \leq S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L} \right) . \tag{22}$$

Since $y^{(k)} \leq x^{(k)}$ by induction hypothesis, applying the isotone operator $S_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ on both sides and taking the jth coordinate gives,

$$S_{\lambda/L}\left(y_j^{(k)} - \frac{[\nabla f(y^{(k)})]_j}{L} \right) \leq S_{\lambda/L}\left(x_j^{(k)} - \frac{[\nabla f(x^{(k)})]_j}{L} \right) .$$

By definition,

$$x_j^{(k+1)} = S_{\lambda/L}\left(x_j^{(k)} - \frac{[\nabla f(x^{(k)})]_j}{L} \right) . \tag{23}$$

Combining this with the previous inequality and (22) gives,

$$y_j^{(k+1)} \leq x_j^{(k+1)} .$$

Since j was arbitrary this means $y^{(k+1)} \leq x^{(k+1)}$ and the proof is complete. ■

4.4 Cyclic Coordinate Minimization (CCM)

Since CCM minimizes a one-dimensional restriction of the function F, let us define some notation for this subsection. Let,

$$\begin{aligned} f_{|j}(\alpha; x) &:= f(x_1, \ldots, x_{j-1}, \alpha, x_{j+1}, \ldots, x_d) \\ F_{|j}(\alpha; x) &:= F(x_1, \ldots, x_{j-1}, \alpha, x_{j+1}, \ldots, x_d) . \end{aligned}$$

With this notation, CCM update can be written as:

$$\begin{aligned} z_j^{(k,j)} &= \underset{\alpha}{\operatorname{argmin}}\ F_{|j}(\alpha; z^{(k,j-1)}) \qquad (24) \\ \forall i \neq j,\ z_i^{(k,j)} &= z_i^{(k,j-1)} . \end{aligned}$$

In order to avoid dealing with infinities in our analysis, we want to ensure that the minimum in (24) above is attained at a finite real number. This leads to the following assumption.

Assumption 10 *For any $x \in \mathbb{R}^d$ and any $j \in [d]$, the one-variable function $f_{|j}(\alpha; x)$ (and hence $F_{|j}(\alpha; x)$) is strictly convex.*

This is a pretty mild assumption: considerably weaker than assuming, for instance, that the function f itself is strictly convex. For example, when f is quadratic as in (4), then the above assumption is equivalent to saying that the diagonal entries $A_{j,j}$ of the positive semi definite matrix A are all strictly positive. This is much weaker than saying that f is strictly convex (which would mean A is invertible).

The next lemma shows that the CCM update can be represented in a way that makes it quite similar to the CCD update.

Lemma 11 *Fix $k \geq 0, j \in [d]$ and consider the* CCM *update* (24). *Let $g(\alpha) = f_{|j}(\alpha; z^{(k,j-1)})$. If the update is non-trivial, i.e. $z_j^{(k,j)} \neq z_j^{(k,j-1)}$, it can be written as*

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k-1,j)} - \frac{\left[\nabla f(z^{(k,j-1)})\right]_j}{\tau}\right)$$

for

$$\tau = \frac{g'(z_j^{(k,j)}) - g'(z_j^{(k,j-1)})}{z_j^{(k,j)} - z_j^{(k,j-1))}} \ . \tag{25}$$

Furthermore, we have $0 < \tau \leq L$.

Proof: See Appendix A ■

We point out that this lemma is useful only for the analysis of CCM and not for its implementation (as τ depends recursively on $z_j^{(k,j)}$) except in an important special case. In the quadratic example (4), $g(\alpha)$ is a one-dimensional quadratic function. In this case τ does not depend on $z_j^{(k,j)}$ and is simply $A_{j,j}$. This leads to an efficient implementation of CCM for quadratic f.

We are now equipped with everything to prove the following behavior of the CCM iterates.

Lemma 12 *If z_0 is a supersolution and $\left\{z^{(k)}\right\}$ is the sequence of iterates generated by the* CCM *algorithm then $\forall k \geq 0$,*

$$1) \quad z^{(k+1)} \leq z^{(k)} \qquad\qquad 2) \quad z^{(k)} \text{ is a supersolution}$$

If z_0 is a subsolution and $\left\{z^{(k)}\right\}$ is the sequence of iterates generated by the CCD *algorithm then $\forall k \geq 0$,*

$$1) \quad z^{(k+1)} \geq z^{(k)} \qquad\qquad 2) \quad z^{(k)} \text{ is a subsolution}$$

Proof: Again, we will only prove the supersolution case as the subsolution case is analogous. We are given that $z^{(0)}$ is a supersolution. We will prove the following: if $z^{(k)}$ is a supersolution then,

$$z^{(k+1)} \leq z^{(k)} \ , \tag{26}$$

$$z^{(k+1)} \text{ is a supersolution} \ . \tag{27}$$

Then the lemma follows by induction on k. Let us assume that $z^{(k)}$ is a supersolution and try to prove (26) and (27). To prove these we will show that $z^{(k,j)} \leq z^{(k)}$ and $z^{(k,j)}$ is a supersolution by induction on $j \in \{0, 1, \ldots, d\}$. This proves (26) and (27) for $z^{(k+1)}$ since $z^{(k+1)} = z^{(k,d)}$. .

The base case ($j = 0$) of the induction is trivial since $z^{(k,0)} \leq z^{(k)}$ since the two vectors are equal. For the same reason, $z^{(k,0)}$ is a supersolution since we have assumed $z^{(k)}$ to be a supersolution. Now assume $z^{(k,j-1)} \leq z^{(k)}$ and $z^{(k,j-1)}$ is a supersolution for some $j > 0$. We want to show that $z^{(k,j)} \leq z^{(k)}$ and $z^{(k,j)}$ is a supersolution. If the update to $z^{(k,j)}$ was trivial, i.e. $z^{(k,j-1)} = z^{(k,j)}$ then there is nothing to prove. Therefore, for the remainder of the proof assume that the update is non-trivial (and hence Lemma 11 applies).

Since $z^{(k,j-1)}$ and $z^{(k,j)}$ differ only in the jth coordinate, to show that $z^{(k,j)} \leq z^{(k)}$ given that $z^{(k,j-1)} \leq z^{(k)}$, it suffices to show that $z^{(k,j)} \leq z^{(k,j-1)}$, i.e.

$$z_j^{(k,j)} \leq z_j^{(k,j-1)} = z_j^{(k)} \ . \tag{28}$$

As in Lemma (11), let us denote $f_{|j}(\alpha; z^{(k,j-1)}$ by $g(\alpha)$. The lemma gives us a $\tau \in (0, L]$ such that,

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \ . \tag{29}$$

Since $z^{(k,j-1)}$ is a supersolution by induction hypothesis and $\tau \le L$, using Lemma 6 we get

$$z_j^{(k,j)} \le S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) \le S_{\lambda/L}\left(z_j^{(k)} - \frac{[\nabla f(z^{(k)})]_j}{L}\right) \le z_j^{(k)} \ .$$

where the second inequality above holds because $z^{(k,j-1)} \le z^{(k)}$ by induction hypothesis and since $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ is an isotone operator. The third holds since $z^{(k)}$ is a supersolution (coupled with Lemma 5). Thus, we have proved (28).

We now need to prove that $z^{(k,j)}$ is a supersolution. To this end, we first claim that

$$z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau} = z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \ . \tag{30}$$

This is true since

$$\begin{aligned} & z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau} - z_j^{(k,j)} + \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \\ &= z_j^{(k,j-1)} - z_j^{(k,j)} - \frac{1}{\tau}(g'(z_j^{(k,j-1)}) - g'(z_j^{(k,j)})) \\ &= z_j^{(k,j-1)} - z_j^{(k,j)} - (z_j^{(k,j-1)} - z_j^{(k,j)}) = 0 \ . \end{aligned}$$

The first equality is true by definition of g and the second by (25). Now, applying $S_{\lambda/\tau}$ to both sides of (30) and using (29), we get

$$\begin{aligned} z_j^{(k,j)} &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \\ &= S_{\lambda/\tau}\left(z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{\tau}\right) \ . \end{aligned} \tag{31}$$

For $i \ne j$, $z_i^{(k,j)} = z_i^{(k,j-1)}$ and thus we have

$$\begin{aligned} & z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau} - z_i^{(k,j)} + \frac{[\nabla f(z^{(k,j)})]_j}{\tau} \\ &= -\frac{1}{\tau}\left[[\nabla f(z^{(k,j-1)})]_i - [\nabla f(z^{(k,j)})]_i\right] \ge 0 \end{aligned}$$

The last inequality holds because we have already shown that $z^{(k,j-1)} \ge z^{(k,j)}$ and thus by isotonicity of $\mathbf{I} - \nabla f/L$, we have

$$[\nabla f(z^{(k,j-1)})]_i - [\nabla f(z^{(k,j)})]_i \le L(z_i^{(k,j-1)} - z_i^{(k,j)}) = 0 \ .$$

Using the monotonic scalar shrinkage operator we have

$$S_{\lambda/\tau}\left(z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau}\right) \ge S_{\lambda/\tau}\left(z_i^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_i}{\tau}\right)$$

which, using the inductive hypothesis that $z^{(k,j-1)}$ is a supersolution, further yields

$$z_i^{(k,j)} = z_i^{(k,j-1)} \ge S_{\lambda/\tau}\left(z_i^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_i}{\tau}\right) \ge S_{\lambda/\tau}\left(z_i^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_i}{\tau}\right) \ . \tag{32}$$

Combining (31) and (32), we get

$$z^{(k,j)} \ge \mathbf{S}_{\lambda/\tau}\left(z^{(k,j)} - \frac{\nabla f(z^{(k,j)})}{\tau}\right)$$

which proves, using Lemma 5, that $z^{(k,j)}$ is a supersolution. ■

4.5 Comparison: CCD vs. CCM

Theorem 13 *Suppose $\{y^{(k)}\}$ and $\{z^{(k)}\}$ are the sequences of iterates generated by the* CCD *and* CCM *algorithms respectively when started from the same supersolution $y^{(0)} = z^{(0)}$. Then, $\forall k \ge 0$,*

$$z^{(k)} \le y^{(k)} \ .$$

On the other hand, if they are started from the same subsolution $y^{(0)} = z^{(0)}$ then the sequences satisfy, $\forall k \ge 0$,

$$z^{(k)} \ge y^{(k)} \ .$$

Proof: We will only prove the supersolution case as the subsolution case is analogous. Given that $y^{(0)} = z^{(0)}$ is a supersolution, we will prove the following: if $z^{(k)} \leq y^{(k)}$ then,

$$z^{(k+1)} \leq y^{(k+1)} . \tag{33}$$

Then the lemma follows by induction on k. Let us assume $z^{(k)} \leq y^{(k)}$ and try to prove (33). To this end we will show that $z^{(k,j)} \leq y^{(k,j)}$ by induction on $j \in \{0, 1, \ldots, d\}$. This infers (33) since $z^{(k+1)} = z^{(k,d)}$ and $y^{(k+1)} = y^{(k,d)}$.

The base case ($j = 0$) is true by the given condition in the lemma since $z^{(k,0)} = z^{(k)}$ as well as $y^{(k,0)} = y^{(k)}$. Now, assume $z^{(k,j-1)} \leq y^{(k,j-1)}$ for some $j > 0$. We want to show that $z^{(k,j)} \leq y^{(k,j)}$.

Since $z^{(k,j-1)}, z^{(k,j)}$ and $y^{(k,j-1)}, y^{(k,j)}$ differ only in the jth coordinate, to show that $z^{(k,j)} \leq y^{(k,j)}$ given that $z^{(k,j-1)} \leq y^{(k,j-1)}$, it suffices to show that

$$z_j^{(k,j)} \leq y_j^{(k,j)} . \tag{34}$$

If the update to $z^{(k,j)}$ is non-trivial then using Lemma 11, there is a $\tau \in (0, L]$, such that

$$\begin{aligned} z_j^{(k,j)} &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{\tau}\right) \\ &\leq S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) , \end{aligned} \tag{35}$$

where the last inequality holds because of Lemma 6 and the fact that $z^{(k,j-1)}$ is a supersolution (Lemma 12). If the update is trivial, i.e. $z_j^{(k,j)} = z_j^{(k,j-1)}$ then using (24) and (12) we have

$$0 \in [\nabla f(z^{(k,j)})]_j + \lambda \operatorname{sign}(z_j^{(k,j)}) .$$

which coupled with (13) gives

$$z_j^{(k,j)} = S_{\lambda/L}\left(z_j^{(k,j)} - \frac{[\nabla f(z^{(k,j)})]_j}{L}\right) \leq S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right)$$

where the last inequality is obtained by applying the isotone operator $\mathbf{S}_{\lambda/L} \circ (\mathbf{I} - \nabla f/L)$ to the inequality $z^{(k,j)} \leq z^{(k,j-1)}$ which holds by lemma 12. Thus (35) holds irrespective of the triviality of the update.

Now applying the same isotone operator to the inequality $z^{(k,j-1)} \leq y^{(k,j-1)}$ and taking the jth coordinate gives,

$$S_{\lambda/L}\left(z_j^{(k,j-1)} - \frac{[\nabla f(z^{(k,j-1)})]_j}{L}\right) \leq S_{\lambda/L}\left(y_j^{(k,j-1)} - \frac{[\nabla f(y^{(k,j-1)})]_j}{L}\right) .$$

The right hand side above is, by definition, $y_j^{(k,j)}$. So, combining the above with (35) gives (34) and proves our inductive claim. ∎

5 Convergence Rates

Our results so far have given inequalities comparing the iterates generated by the three algorithms. We finally want to compare the function values obtained by these iterates. For doing that, the next lemma is useful.

Lemma 14 *If y is a supersolution and $y \leq x$ then $F(y) \leq F(x)$.*

Proof: Since F is convex, we have

$$F(y) - F(x) \leq \langle \nabla f(y) + \lambda\rho, y - x \rangle \tag{36}$$

for any $\rho \in \partial\|y\|_1$. We have assumed that $y \leq x$. Thus in order to prove $F(y) - F(x) \leq 0$, it suffices to show that

$$\forall i \in [d], \qquad \exists \rho_i \in \operatorname{sign}(y_i) \qquad \text{s.t.} \qquad \gamma_i + \lambda\rho_i \geq 0 \tag{37}$$

where, for convenience, we denote the gradient $\nabla f(y)$ by γ. Since y is a supersolution, Lemma 5 gives,

$$\forall i \in [d], \qquad y_i \geq S_{\lambda/L}\left(y_i - \frac{\gamma_i}{L}\right) \tag{38}$$

For any $i \in [d]$, there are three mutually exclusive and exhaustive cases.

Case (1) : $y_i > \frac{\gamma_i+\lambda}{L}$ Plugging this value in (38) and using the definition of scalar shrinkage (9), we get

$$y_i \geq y_i - \frac{\gamma_i + \lambda}{L}$$

which gives $\gamma_i + \lambda \geq 0$ and hence $y_i > 0$. Thus, we can choose $\rho_i = 1 \in \operatorname{sign}(y_i)$ and we indeed have $\gamma_i + \lambda\rho_i \geq 0$.

Case (2) : $y_i \in [\frac{\gamma_i-\lambda}{L}, \frac{\gamma_i+\lambda}{L}]$ In this case, we have $y_i \geq S_{\lambda/L}(y_i^{(k)} - \frac{\gamma_i}{L}) = 0$. Thus,

$$\frac{\gamma_i + \lambda}{L} \geq y_i \geq 0 \ .$$

Thus we can choose $\rho_i = 1 \in \operatorname{sign}(y_i)$ and we have $\gamma_i + \lambda\rho_i \geq 0$.

Case (3) : $y_i < \frac{\gamma_i-\lambda}{L}$ Plugging this value in (38) and using the definition of scalar shrinkage (9), we get

$$y_i \geq y_i - \frac{\gamma_i - \lambda}{L}$$

which gives $\gamma_i - \lambda \geq 0$. Now if $y_i \leq 0$, we can set $\rho = -1 \in \operatorname{sign}(y_i)$ and will have $\gamma_i + \lambda\rho_i \geq 0$. On the other hand, if $y_i > 0$, we need to choose $\rho_i = 1$ and thus $\gamma_i + \lambda \geq 0$ should hold if (37) is to be true. However, we know $\gamma_i - \lambda \geq 0$, and $\lambda \geq 0$ so $\gamma_i + \lambda \geq 0$ is also true.

Thus in all three cases we have that there is a $\rho_i \in \operatorname{sign}(y_i)$ such that (37) is true. ■

There is a similar lemma for subsolutions whose proof, being similar to the proof above, is skipped.

Lemma 15 *If y is a subsolution and $y \geq x$ then $F(y) \leq F(x)$.*

If we start from a supersolution, the iterates for CCD and CCM always maintain the supersolution property. Thus Lemma 14 ensures that starting from the same initial iterate, the function values of the CCD and CCM iterates always remain less than the corresponding GD iterates. Since the GD algorithm has $O(1/k)$ accuracy guarantees according to Theorem 2, the same rates must hold true for CCD and CCM. This is formalized in the following theorem.

Theorem 16 *Starting from the same super- or subsolution $x^{(0)} = y^{(0)} = z^{(0)}$, let $\{x^{(k)}\}$, $\{y^{(k)}\}$ and $\{z^{(k)}\}$ denote the* GD, CCD *and* CCM *iterates respectively. Then for any minimizer x^* of* (2)*, and $\forall k \geq 1$,*

$$F(z^{(k)}) \leq F(y^{(k)}) \leq F(x^{(k)}) \leq F(x^\star) + \frac{L\|x^\star - x^{(0)}\|^2}{2\,k}$$

6 Conclusion

Coordinate descent based methods have seen a resurgence of popularity in recent times in both the machine learning and the statistics community, due to the simplicity of the updates and implementation of the overall algorithms. Absence of finite time convergence rates is thus one of the most important theoretical issues to address.

In this paper, we provided a comparative analysis of GD, CCD and CCM algorithms to give the first known finite time guarantees on the convergence rates of cyclic coordinate descent methods. However, there still are a significant number of unresolved questions. Our comparative results require that the algorithms start from a supersolution so that the property is maintained for all the subsequent iterates. We also require an isotonicity assumption on the $\mathbf{I} - \nabla f/L$ operator. Although this is a fairly common assumption in numerical optimization (Bertsekas & Tsitsiklis, 1989), it is desirable to have a more generalized analysis without any restrictions. Since stochastic coordinate descent (Shalev-Shwartz & Tewari, 2009) converges at the same $O(1/k)$ rate as GD without additional assumptions, intuition suggests that same should be true for CCD and CCM. A theoretical proof of the same remains an open question.

Some greedy versions of the coordinate descent algorithm (e.g., (Wu & Lange, 2008)) still lack a theoretical analysis of their finite time convergence guarantees. Although Clarkson (2008) has a $O(1/k)$ rates for a greedy version, the analysis is restricted to a simplex domain and does not generalize to arbitrary domains. The phenomenal performance of greedy coordinate descent algorithms on real life datasets makes it all the more essential to validate these experimental results theoretically.

References

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 693–696. IEEE Computer Society.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN 0-13-648700-9.

Clarkson, K. L. (2008). Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, 922–931.

Duchi, J., & Singer, Y. (2009). Efficient learning using forward-backward splitting. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta, eds., *Advances in Neural Information Processing Systems 22*, 495–503.

Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. In *Annals of Applied Statistics*.

Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, *49*(3), 291–304.

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Soviet Math. Docl.*, *269*, 543–547.

Nesterov, Y. (2003). *Introductory Lectures On Convex Optimization: A Basic Course*. Springer.

Rheinboldt, W. C. (1970). On M-functions and their application to nonlinear Gauss–Seidel iterations and to network flows. *J. Math. Anal. Appl.*, *32*, 274–307.

Shalev-Shwartz, S., & Tewari, A. (2009). Stochastic methods for l_1 regularized loss minimization. In *Proceedings of the 26th International Conference on Machine Learning*, 929–936. ACM Press.

Tropp, J. A. (2006). Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, *52*(3), 1030–1051.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, *109*(3), 475–494.

Tseng, P., & Yun, S. (2009a). A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory and Applications*, *140*(3), 513–535.

Tseng, P., & Yun, S. (2009b). A coordinate gradient descent method for nonsmooth separable minimization. *Math. Prog. B*, *117*, 387–423.

Wu, T. T., & Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. In *Annals of Applied Statistics*, vol. 2, 224–244.

Appendix

A Proof of Lemma 11

Since $g(\alpha) = f_{|i}(\alpha; z^{(k,j-1)})$ we have

$$g'(\alpha) = \Big[\nabla f(z_1^{(k,j-1)}, z_2^{(k,j-1)}, \dots z_{j-1}^{(k,j-1)}, \alpha, z_{j+1}^{(k,j-1)}, \dots z_d^{(k,j-1)})\Big]_j$$

Therefore,

$$g'(z_j^{(k,j-1)}) = [\nabla f(z^{(k,j-1)})]_j \tag{39}$$

Since, by definition, $z_j^{(k,j)}$ is the minimizer of $g(\alpha) + \lambda|\alpha|$, we have

$$0 \in g'(z_j^{(k,j)}) + \lambda \operatorname{sign}(z_j^{(k,j)})$$

For notational convenience we denote $z_j^{(k,j)}$ as $\alpha^\star$, since it is the minimizer of $g(\alpha) + \lambda|\alpha|$. With this notation we have,

$$\tau = \frac{g'(\alpha^\star) - g'(z_j^{(k,j-1)})}{\alpha^\star - z_j^{(k,j-1)}} \; . \tag{40}$$

Note that τ is well defined since the denominator is non-zero by our assumption of a non-trivial update. Further, $\tau > 0$ by Assumption 10 and $\tau \le L$ since ∇f (and hence $g'(\alpha)$) is L-Lipschitz continuous.

Depending on the sign of $\alpha^\star$, there are three possible cases:

Case (1): $\alpha^\star > 0$: This implies that

$$g'(\alpha^\star) + \lambda = 0 \tag{41}$$

By (40),

$$g'(\alpha^\star) = g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)})$$

Plugging this in (41), we get

$$g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) + \lambda = 0 \; .$$

Using the definition of shrinkage operator (9) combined with the fact that $\alpha^\star > 0$, we have

$$\begin{aligned} \alpha^\star &= z_j^{(k,j-1)} - \frac{1}{\tau} g'(z_j^{(k,j-1)}) - \frac{\lambda}{\tau} \\ &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k)})}{\tau}\right) \end{aligned}$$

Case (2): $\alpha^\star = 0$: The corresponding condition is

$$0 \in [g'(\alpha^\star) - \lambda, g'(\alpha^\star) + \lambda]$$

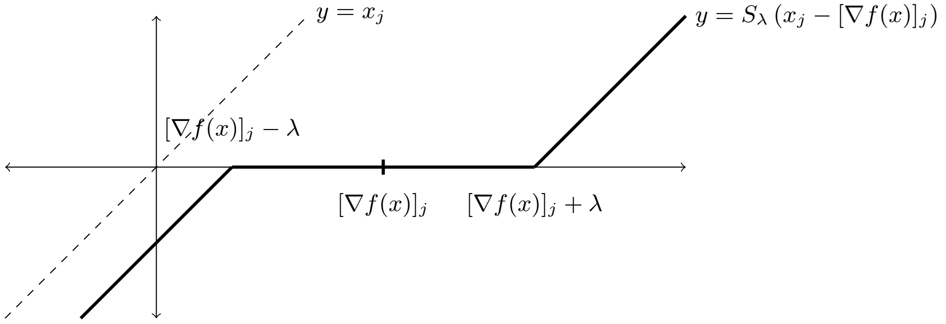
Again using (40), we have

$$\begin{aligned} & g'(\alpha^\star) = g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) = g'(z_j^{(k,j-1)}) - \tau(z_j^{(k,j-1)}) \qquad \text{[since } \alpha^\star = 0\text{]} \\ & \implies \alpha^\star = 0 \in \left[\frac{g'(z_j^{(k,j-1)})}{\tau} - z_j^{(k,j-1)} - \frac{\lambda}{\tau}, \frac{g'(z_j^{(k,j-1)})}{\tau} - z_j^{(k,j-1)} + \frac{\lambda}{\tau}\right] \\ & \implies \alpha^\star = 0 = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k,j-1)})}{\tau}\right) \end{aligned}$$

where the last step follows from the definition of the shrinkage operator (9).

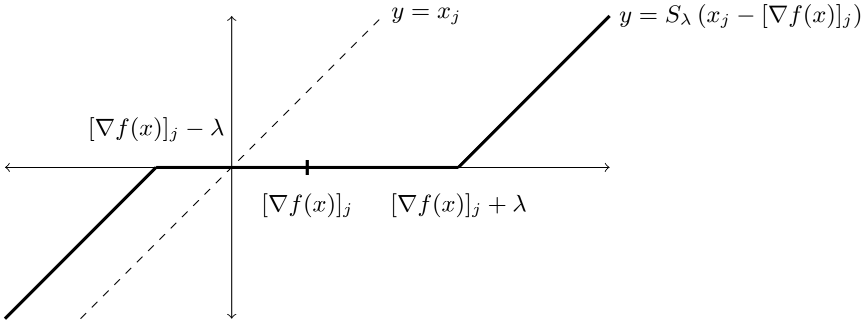
Case (3): $\alpha^\star < 0$: This implies that

$$g'(\alpha^\star) - \lambda = 0$$



$y = x_j$
$y = S_\lambda\left(x_j - [\nabla f(x)]_j\right)$
$[\nabla f(x)]_j - \lambda$
$[\nabla f(x)]_j$
$[\nabla f(x)]_j + \lambda$

Figure 1: Interval to right of zero



$y = x_j$
$y = S_\lambda\left(x_j - [\nabla f(x)]_j\right)$
$[\nabla f(x)]_j - \lambda$
$[\nabla f(x)]_j$
$[\nabla f(x)]_j + \lambda$

Figure 2: Interval crossing zero

Using (40) to substitute for $g'(\alpha^\star)$ as in the previous cases, we have,

$$g'(z_j^{(k,j-1)}) + \tau(\alpha^\star - z_j^{(k,j-1)}) - \lambda = 0$$

which yields

$$\begin{aligned}\alpha^\star &= z_j^{(k,j-1)} - \frac{1}{\tau} g'(z_j^{(k,j-1)}) + \frac{\lambda}{\tau} \\ &= S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{g'(z_j^{(k,j-1)})}{\tau}\right)\end{aligned}$$

where the last inequality follows because $\alpha^\star < 0$.

Combining these three cases and using (39) we get

$$z_j^{(k,j)} = S_{\lambda/\tau}\left(z_j^{(k,j-1)} - \frac{\left[\nabla f(z^{(k,j-1)})\right]_j}{\tau}\right) .$$

B Proof of lemma 5

We prove the supersolution case only as the subsolution case is analogous. Let for a particular $\tau > 0$, $x \geq \mathbf{S}_{\lambda/\tau}\left(x - \frac{\nabla f(x)}{\tau}\right)$. We prove the inequality for the scalar S operator on an arbitrary coordinate j. The subsequent proofs are divided into three disjoint cases related to the values taken by the shrinkage operator.

Case 1 $[\nabla f(x)]_j - \lambda > 0$**:**

This is illustrated in figure 1. Depending on whether $\tau > 1$ or not, the graph of the shrinkage operator shifts left or right, but clearly division by τ does not change the sign of the shrinkage operator value

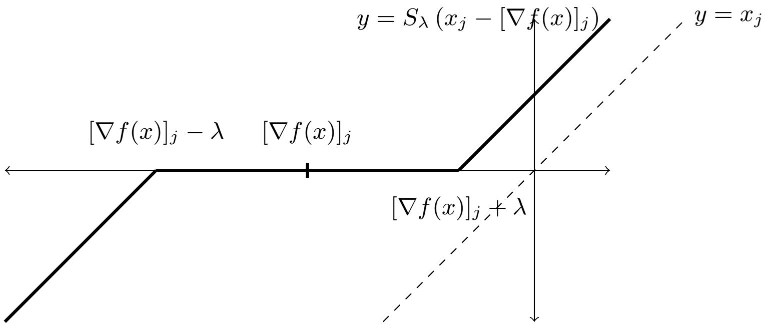

$y = S_\lambda\left(x_j - [\nabla f(x)]_j\right)$
$y = x_j$
$[\nabla f(x)]_j - \lambda$
$[\nabla f(x)]_j$
$[\nabla f(x)]_j + \lambda$

Figure 3: Interval to left of zero

at any point. As is evident from figure 1, the graph of $y = x_j$ always lies above that of the shrinkage operator. Thus

$$x_j \geq S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right) \tag{42}$$

for all values of τ and in particular for $\tau = 1$. Thus x is a supersolution.

Case 2 $0 \in [[\nabla f(x)]_j - \lambda, [\nabla f(x)]_j + \lambda]$**:**

The corresponding case is illustrated in figure 2. It is clear from the figure that $x_j \geq S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right)$ for positive τ, only when $x_j \geq 0$. Just as in the previous case, changing the value of τ shifts the graph by appropriate scale without changing its sign. Thus (42) holds for $x_j \geq 0$ irrespective of the value of τ. In particular, it should hold for $\tau = 1$ which proves that x is a supersolution.

Case 3 $[\nabla f(x)]_j + \lambda < 0$**:**

As illustrated in figure 3, in this case the graph of the shrinkage operator will always lie below the value of x_j. Thus (42) will not be satisfied for any value of τ which makes the case vacuous.

To prove the converse direction, we look at the same three exclusively disjoint cases for an arbitrary coordinate j.

Case 1 $[\nabla f(x)]_j - \lambda > 0$**:**

As seen from figure 1, x is always a supersolution since $[\nabla f(x)]_j + \lambda > [\nabla f(x)]_j - \lambda > 0$ and the graph of the shrinkage operator uniformly stays below the value of x_j. Since the sign of the shrinkage operator value does not change due to division by $\tau > 0$, (42) holds for arbitrary positive τ.

Case 2 $0 \in [[\nabla f(x)]_j - \lambda, [\nabla f(x)]_j + \lambda]$**:**

If x is a supersolution, it means that the value attained by the shrinkage operator lies below the value of x_j , which is true when $x_j \geq 0$ (Figure 2). In this subset of the domain, division by τ maintains the sign of the shrinkage value and thus (42) holds.

Case 3 $[\nabla f(x)]_j + \lambda < 0$**:**

In this case the graph of the shrinkage operator always lies above the value of x_j . Thus x can never be a supersolution if this condition holds true.

C Proof of lemma 6

Let

$$h(\tau) = S_{\lambda/\tau}\left(x_j - \frac{[\nabla f(x)]_j}{\tau}\right)$$

We again look at the three disjoint cases for arbitrary $\tau_1, \tau_2 \in (0, \infty)$ with $\tau_1 \geq \tau_2$ and show that $h(\tau_1) \geq h(\tau_2)$.

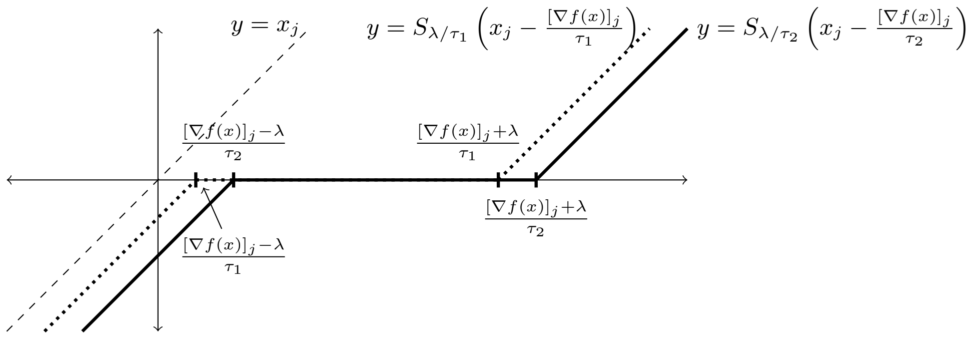

$y = x_j$
$y = S_{\lambda/\tau_1}\left(x_j - \frac{[\nabla f(x)]_j}{\tau_1}\right)$
$y = S_{\lambda/\tau_2}\left(x_j - \frac{[\nabla f(x)]_j}{\tau_2}\right)$
$\frac{[\nabla f(x)]_j-\lambda}{\tau_2}$
$\frac{[\nabla f(x)]_j+\lambda}{\tau_1}$
$\frac{[\nabla f(x)]_j+\lambda}{\tau_2}$
$\frac{[\nabla f(x)]_j-\lambda}{\tau_1}$

Figure 4: Interval to right of zero

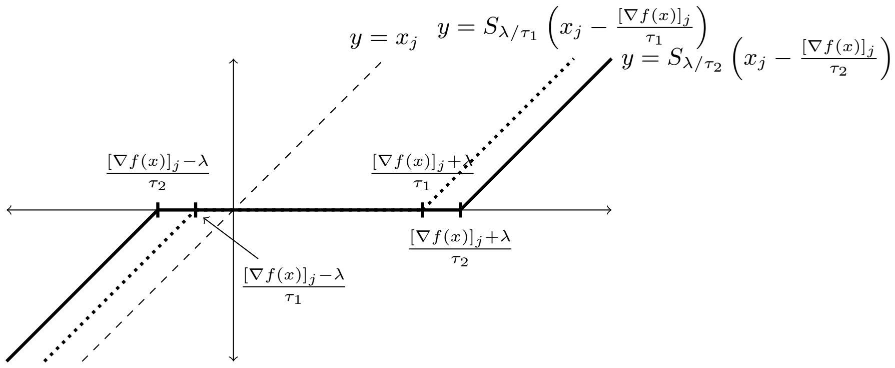

$y = x_j$
$y = S_{\lambda/\tau_1}\left(x_j - \frac{[\nabla f(x)]_j}{\tau_1}\right)$
$y = S_{\lambda/\tau_2}\left(x_j - \frac{[\nabla f(x)]_j}{\tau_2}\right)$
$\frac{[\nabla f(x)]_j-\lambda}{\tau_2}$
$\frac{[\nabla f(x)]_j+\lambda}{\tau_1}$
$\frac{[\nabla f(x)]_j+\lambda}{\tau_2}$
$\frac{[\nabla f(x)]_j-\lambda}{\tau_1}$

Figure 5: Interval crossing zero

Case 1 $[\nabla f(x)]_j - \lambda > 0$**:**

Since both the hinge points in the graph will be positive (figure 4), we have $\frac{[\nabla f(x)]_j-\lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j-\lambda}{\tau_2}$ and $\frac{[\nabla f(x)]_j+\lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j+\lambda}{\tau_2}$. Thus it is trivial to see that the graph of $h(\tau_1)$ is always greater than $h(\tau_2)$.

Case 2 $0 \in [[\nabla f(x)]_j - \lambda, [\nabla f(x)]_j + \lambda]$**:**

Since x needs to be a supersolution, we only need to consider the subset of the domain when $x_j \geq 0$. We still have $\frac{[\nabla f(x)]_j+\lambda}{\tau_1} \leq \frac{[\nabla f(x)]_j+\lambda}{\tau_2}$ and it is obvious from figure 5, that $h(\tau_1) \geq h(\tau_2)$.

Case3 $[\nabla f(x)]_j + \lambda < 0$:

Since x can never be a supersolution in this case as shown in the proof of lemma 5, this case is vacuous.