

Randomized Coordinate Update Methods

Ernest K. Ryu and Wotao Yin

Large-Scale Convex Optimization via Monotone Operators

Outline

Randomized coordinate fixed-point iteration

Coordinate and extended coordinate friendly operators

Examples

Discussion

Coordinate partitioning

Partition $x \in \mathbb{R}^n$ into m non-overlapping blocks of sizes n_1, \dots, n_m .

¹some researchers would write $x = [x_1; \dots; x_m]$ instead

Coordinate partitioning

Partition $x \in \mathbb{R}^n$ into m non-overlapping blocks of sizes n_1, \dots, n_m .

Write¹ $x = (x_1, \dots, x_m)$, so $x_i \in \mathbb{R}^{n_i}$.

Partition $\mathbf{T}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ into $(\mathbf{T}(x))_i \in \mathbb{R}^{n_i}$ and define $\mathbf{T}_i(x)$ as follows:

$$\mathbf{T}(x) = \begin{bmatrix} (\mathbf{T}(x))_1 \\ \vdots \\ (\mathbf{T}(x))_i \\ \vdots \\ (\mathbf{T}(x))_m \end{bmatrix} \quad \mathbf{T}_i(x) = \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ (\mathbf{T}(x))_i \\ x_{i+1} \\ \vdots \\ x_m \end{bmatrix},$$

i.e., \mathbf{T}_i is \mathbf{T} on the i -th block and is identity on the other blocks.

We say “block” and “coordinate” interchangeably.

¹some researchers would write $x = [x_1; \dots; x_m]$ instead

Coordinate-update fixed-point iteration

For $\mathbb{T}: \mathbb{R}^n \rightarrow \mathbb{R}^n$, consider

$$\underset{x \in \mathbb{R}^n}{\text{find}} \quad x = \mathbb{T}x.$$

Coordinate-update fixed-point iteration (C-FPI) is

$$\begin{aligned} \text{select } & i(k) \in \{1, \dots, m\}, \\ & x^{k+1} = \mathbb{T}_{i(k)}(x^k). \end{aligned}$$

At the k -th iteration, C-FPI updates only the $i(k)$ -th block.

Specifying the selection rule for $i(k)$ fully specifies the method.

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

- ▶ Cyclic rule. Select the blocks in a fixed cyclic order.

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

- ▶ Cyclic rule. Select the blocks in a fixed cyclic order.
- ▶ Cyclic shuffle rule. Shuffle the ordering of blocks and select each of them sequentially, and repeat. A.k.a. random sample without replacement.

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

- ▶ Cyclic rule. Select the blocks in a fixed cyclic order.
- ▶ Cyclic shuffle rule. Shuffle the ordering of blocks and select each of them sequentially, and repeat. A.k.a. random sample without replacement.
- ▶ Essential cyclic rule. Each block appears once or more in each “cycle”. (Both cyclic and cyclic shuffle rules are special cases.)

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

- ▶ Cyclic rule. Select the blocks in a fixed cyclic order.
- ▶ Cyclic shuffle rule. Shuffle the ordering of blocks and select each of them sequentially, and repeat. A.k.a. random sample without replacement.
- ▶ Essential cyclic rule. Each block appears once or more in each “cycle”. (Both cyclic and cyclic shuffle rules are special cases.)
- ▶ Greedy rule. Select block that leads to the most immediate progress, measured in many different ways.

Block selection rules

There are many ways to select $i(k)$ with different advantages and disadvantages.

Common selection rules:

- ▶ Cyclic rule. Select the blocks in a fixed cyclic order.
- ▶ Cyclic shuffle rule. Shuffle the ordering of blocks and select each of them sequentially, and repeat. A.k.a. random sample without replacement.
- ▶ Essential cyclic rule. Each block appears once or more in each “cycle”. (Both cyclic and cyclic shuffle rules are special cases.)
- ▶ Greedy rule. Select block that leads to the most immediate progress, measured in many different ways.
- ▶ Randomized rule. Select blocks randomly. Each selection is independent of others. A.k.a. random sample with replacement.

Randomized coordinate-update fixed-point iteration

We focus on the randomized rule (i.e. select $i(k) \in \{1, \dots, m\}$ independently uniformly at random) since its analysis is simplest.

We get randomized coordinate-update fixed-point iteration (RC-FPI):

$$\begin{aligned}i(k) &\sim \text{IID Uniform}\{1, \dots, m\} \\x^{k+1} &= \mathbf{T}_{i(k)}(x^k).\end{aligned}$$

Randomized coordinate-update fixed-point iteration

Theorem 2.

Assume $\mathbb{T}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is θ -averaged with $\theta \in (0, 1)$ and $\text{Fix } \mathbb{T} \neq \emptyset$. Assume the random indices $i(0), i(1), \dots \in \{1, \dots, m\}$ are i.i.d. with uniform probability. Then $x^{k+1} = \mathbb{T}_{i(k)}x^k$ with any starting point $x^0 \in \mathbb{R}^n$ converges to one fixed point with probability 1, i.e., the event

$$x^k \rightarrow x^*$$

holds with probability 1 for some $x^* \in \text{Fix } \mathbb{T}$. The quantities $\mathbb{E} \text{dist}^2(x^k, \text{Fix } \mathbb{T})$ and $\mathbb{E} \|x^k - x^*\|^2$ for any $x^* \in \text{Fix } \mathbb{T}$ decrease monotonically with k . Finally, we have

$$\text{dist}(x^k, \text{Fix } \mathbb{T}) \rightarrow 0$$

with probability 1.

Prepare to prove Theorem 2

Proving Theorem 2 requires the following standard result from probability theory.

Theorem (supermartingale convergence theorem).

Let $(V^k)_{k=0,1,\dots}$ and $(S^k)_{k=0,1,\dots}$ be two sequences of random variables satisfying $V^k \geq 0$ and $S^k \geq 0$ almost surely for $k = 0, 1, \dots$. Assume

$$\mathbb{E}[V^{k+1} | V^0, \dots, V^k] \leq V^k - S^k$$

holds for $k = 0, 1, \dots$. Then

1. $\sum_{k=0}^{\infty} S^k < \infty$
2. $V^k \rightarrow V^\infty$

hold with probability 1. (Note that the limit V^∞ is a random variable.)

Prepare to prove Theorem 2

Define \mathbf{S} with $\mathbf{T} = \mathbf{I} - \theta\mathbf{S}$ and \mathbf{S}_i with $\mathbf{T}_i = \mathbf{I} - \theta\mathbf{S}_i$. So we have

$$\mathbf{S}_i(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (\mathbf{S}(x))_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

for $i = 1, \dots, m$, and $\sum_{i=1}^m \mathbf{S}_i = \mathbf{S}$.

Prepare to prove Theorem 2

Define \mathbf{S} with $\mathbf{T} = \mathbf{I} - \theta\mathbf{S}$ and \mathbf{S}_i with $\mathbf{T}_i = \mathbf{I} - \theta\mathbf{S}_i$. So we have

$$\mathbf{s}_i(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (\mathbf{S}(x))_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

for $i = 1, \dots, m$, and $\sum_{i=1}^m \mathbf{S}_i = \mathbf{S}$.

Alternately express $x^{k+1} = \mathbf{T}_{i(k)}x^k$ as

$$x^{k+1} = x^k - \theta\mathbf{S}_{i(k)}x^k.$$

Prepare to prove Theorem 2

Fact: \mathbf{T} is θ -averaged if and only if \mathbf{S} is $(1/2)$ -cocoercive. Proof:

$$\begin{aligned} \mathbf{T} \text{ is } \theta\text{-averaged} &\Leftrightarrow \frac{1}{\theta}\mathbf{T} - \left(\frac{1}{\theta} - 1\right)\mathbf{I} \text{ is nonexpansive} \\ &\Leftrightarrow \mathbf{I} - \mathbf{S} \text{ is nonexpansive} \\ &\Leftrightarrow \|x - \mathbf{S}x - y + \mathbf{S}y\|^2 \leq \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n \\ &\Leftrightarrow \frac{1}{2}\|\mathbf{S}x - \mathbf{S}y\|^2 \leq \langle x - y, \mathbf{S}x - \mathbf{S}y \rangle \quad \forall x, y \in \mathbb{R}^n \\ &\Leftrightarrow \mathbf{S} \text{ is } (1/2)\text{-cocoercive.} \end{aligned}$$

Prepare to prove Theorem 2

Fact: \mathbf{T} is θ -averaged if and only if \mathbf{S} is $(1/2)$ -cocoercive. Proof:

$$\begin{aligned}\mathbf{T} \text{ is } \theta\text{-averaged} &\Leftrightarrow \frac{1}{\theta}\mathbf{T} - \left(\frac{1}{\theta} - 1\right)\mathbf{I} \text{ is nonexpansive} \\ &\Leftrightarrow \mathbf{I} - \mathbf{S} \text{ is nonexpansive} \\ &\Leftrightarrow \|x - \mathbf{S}x - y + \mathbf{S}y\|^2 \leq \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n \\ &\Leftrightarrow \frac{1}{2}\|\mathbf{S}x - \mathbf{S}y\|^2 \leq \langle x - y, \mathbf{S}x - \mathbf{S}y \rangle \quad \forall x, y \in \mathbb{R}^n \\ &\Leftrightarrow \mathbf{S} \text{ is } (1/2)\text{-cocoercive.}\end{aligned}$$

Clearly, $\text{Fix } \mathbf{T} = \text{Zer } \mathbf{S}$. For any $x^* \in \text{Fix } \mathbf{T} = \text{Zer } \mathbf{S}$ and $x \in \mathbb{R}^n$,

$$\frac{1}{2}\|\mathbf{S}x\|^2 \leq \langle \mathbf{S}x, x - x^* \rangle \tag{1}$$

Prepare to prove Theorem 2

x^{k+1} is a random variable depending on $i(k), i(k-1), \dots, i(0)$.
 x^0 is arbitrary but not random. Write \mathbb{E} for the full expectation. Write \mathbb{E}_k for the conditional expectation with respect to $i(k)$ conditioned on the past random variables $i(k-1), i(k-2), \dots, i(0)$.

Prepare to prove Theorem 2

x^{k+1} is a random variable depending on $i(k), i(k-1), \dots, i(0)$.
 x^0 is arbitrary but not random. Write \mathbb{E} for the full expectation. Write \mathbb{E}_k for the conditional expectation with respect to $i(k)$ conditioned on the past random variables $i(k-1), i(k-2), \dots, i(0)$.

Under these definitions, $\mathbb{E}_k[x^k] = x^k$ and

$$\mathbb{E}_k[\mathbf{S}_{i(k)}x^k] = \sum_{i=1}^m \frac{1}{m} \mathbf{S}_i x^k = \frac{1}{m} \mathbf{S} x^k, \quad (2)$$

$$\mathbb{E}_k \|\mathbf{S}_{i(k)}x^k\|^2 = \sum_{i=1}^m \frac{1}{m} \|\mathbf{S}_i x^k\|^2 = \frac{1}{m} \|\mathbf{S} x^k\|^2. \quad (3)$$

Proof of Theorem 2

Stage 1. For any $x^* \in \text{Fix } \mathbb{T}$,

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &= \|x^k - \theta \mathbf{S}_{i(k)} x^k - x^*\|^2 \\ &= \|x^k - x^*\|^2 - 2\theta \langle \mathbf{S}_{i(k)} x^k, x^k - x^* \rangle + \theta^2 \|\mathbf{S}_{i(k)} x^k\|^2.\end{aligned}$$

Take conditional expectation \mathbb{E}_k and use (2) and (3):

$$\begin{aligned}\mathbb{E}_k \|x^{k+1} - x^*\|^2 &= \|x^k - x^*\|^2 - 2\theta \langle \mathbb{E}_k[\mathbf{S}_{i(k)} x^k], x^k - x^* \rangle + \theta^2 \mathbb{E}_k \|\mathbf{S}_{i(k)} x^k\|^2 \\ &= \|x^k - x^*\|^2 - \frac{2\theta}{m} \langle \mathbf{S} x^k, x^k - x^* \rangle + \frac{\theta^2}{m} \|\mathbf{S} x^k\|^2 \\ &\leq \|x^k - x^*\|^2 - (1 - \theta) \frac{\theta}{m} \|\mathbf{S} x^k\|^2,\end{aligned}\tag{4}$$

where the inequality follows from (1).

So $(\|x^k - x^*\|^2)_{k=0,1,\dots}$ a nonnegative supermartingale.

Proof of Theorem 2

Take the full expectation on both ends of (4):

$$\mathbb{E}\|x^{k+1} - x^*\|^2 \leq \mathbb{E}\|x^k - x^*\|^2 - (1 - \theta) \frac{\theta}{m} \mathbb{E}\|\mathbf{S}x^k\|^2.$$

Therefore, $\mathbb{E}\|x^k - x^*\|^2$ decreases monotonically with k and, by minimizing over $x^* \in \text{Fix } \mathbf{T}$, so does $\mathbb{E} \text{dist}^2(x^k, \text{Fix } \mathbf{T})$.

Proof of Theorem 2

Stage 2. We prove convergence of the iterates. Apply the *supermartingale convergence theorem* to (4) to get:

(i) $\sum_{k=0}^{\infty} \|\mathbf{S}x^k\|^2 < \infty$ and

(ii) $\lim_{k \rightarrow \infty} \|x^k - x^*\|$ exists

with probability 1. Note (i) implies $\|\mathbf{S}x^k\|^2 \rightarrow 0$ and (ii) implies x^k is bounded with probability 1.

Proof of Theorem 2

Stage 2. We prove convergence of the iterates. Apply the *supermartingale convergence theorem* to (4) to get:

(i) $\sum_{k=0}^{\infty} \|\mathbb{S}x^k\|^2 < \infty$ and

(ii) $\lim_{k \rightarrow \infty} \|x^k - x^*\|$ exists

with probability 1. Note (i) implies $\|\mathbb{S}x^k\|^2 \rightarrow 0$ and (ii) implies x^k is bounded with probability 1.

For all $x^* \in \text{Fix } \mathbb{T}$, $\lim_{k \rightarrow \infty} \|x^k - x^*\|$ exists with probability 1. Apply Proposition 1 (stated and proved later) to conclude with probability 1, $\lim_{k \rightarrow \infty} \|x^k - x^*\|$ exists for all $x^* \in \text{Fix } \mathbb{T}$. Apply the same argument of Theorem 1 to get: with probability 1, $x^k \rightarrow x^*$. \square

Measurability argument

In the proof, we first choose an arbitrary $x^* \in \text{Fix } \mathbb{T}$ and then apply the supermartingale convergence theorem, so $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ holds with probability 1}]$ applies to each x^* .

We need a stronger statement before we can the argument of Theorem 1: $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ for all } x^* \in \text{Fix } \mathbb{T}]$ holds with probability 1.

Measurability argument

In the proof, we first choose an arbitrary $x^* \in \text{Fix } \mathbb{T}$ and then apply the supermartingale convergence theorem, so $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ holds with probability 1}]$ applies to each x^* .

We need a stronger statement before we can the argument of Theorem 1: $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ for all } x^* \in \text{Fix } \mathbb{T}]$ holds with probability 1.

The subtlety:

1. the intersection of *countably many* measure-1 sets has measure 1;
2. the intersection of *uncountably many* measure-1 sets has measure *less than or equal to* 1.

As set $\text{Fix } \mathbb{T}$ is possibly uncountable, case 2 is applicable.

Measurability argument

In the proof, we first choose an arbitrary $x^* \in \text{Fix } \mathbb{T}$ and then apply the supermartingale convergence theorem, so $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ holds with probability 1}]$ applies to each x^* .

We need a stronger statement before we can the argument of Theorem 1: $[\lim_{k \rightarrow \infty} \|x^k - x^*\| \text{ for all } x^* \in \text{Fix } \mathbb{T}]$ holds with probability 1.

The subtlety:

1. the intersection of *countably many* measure-1 sets has measure 1;
2. the intersection of *uncountably many* measure-1 sets has measure *less than or equal to* 1.

As set $\text{Fix } \mathbb{T}$ is possibly uncountable, case 2 is applicable.

Proposition 1 (next slide) shows the stronger statement actually holds.

Proposition 1

Proposition 1.

Let $Y \subseteq \mathbb{R}^n$ and let x^0, x^1, \dots be a random sequence. Then statement 1 implies statement 2.

1. For all $y \in Y$ [with probability 1, $\lim_{k \rightarrow \infty} \|x^k - y\|$ exists].
2. With probability 1 [for all $y \in Y$, $\lim_{k \rightarrow \infty} \|x^k - y\|$ exists].

Proof outline. (i) $Y \subseteq \mathbb{R}^n$ is separable (i.e., has a countable dense subset), (ii) sequence of functions $\{\|x^k - \cdot\|\}_{k \in \mathbb{N}}$ has a limit on the countable dense subset of Y , and (iii) if an equicontinuous sequence of functions has limits on the dense subset of Y , limits exist on all of Y . \square

Understand “equicontinuous” as “uniformly (over sequence) uniformly (over function domain) continuous”

Outline

Randomized coordinate fixed-point iteration

Coordinate and extended coordinate friendly operators

Examples

Discussion

Coordinate friendly operators

While Theorem 2 is true, RC-FPI is useful only if it is computationally efficient.

RC-FPI is computationally effective when \mathbb{T} is coordinate friendly (defined below) or extended coordinate friendly (defined later).

Coordinate friendly operators

While Theorem 2 is true, RC-FPI is useful only if it is computationally efficient.

RC-FPI is computationally effective when \mathbb{T} is coordinate friendly (defined below) or extended coordinate friendly (defined later).

Let $z = (z_1, \dots, z_m) \in \mathbb{R}^n$. Then $x \mapsto z$ is coordinate friendly if

$$\max_{i=1, \dots, m} \mathcal{F}[x \mapsto z_i] \ll \mathcal{F}[x \mapsto z].$$

(Meaning of \ll depends on context.)

\mathbb{T} is coordinate friendly if $x \mapsto \mathbb{T}x$ is coordinate friendly.

Coordinate friendly \Rightarrow parallelizable

If $x \mapsto z$ is coordinate friendly, then for $p \geq m$

$$\mathcal{F}_p[x \mapsto z] = \max_{i=1, \dots, m} \mathcal{F}[x \mapsto z_i] \ll \mathcal{F}[x \mapsto z].$$

So $x \mapsto z$ is parallelizable.

Affine operators

Affine operator $\mathbb{T}x = Ax + b$, where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, is coordinate friendly if $n_i \ll n$ for $i = 1, \dots, m$, since

$$\mathcal{F}[x \mapsto \mathbb{T}_i x] \sim 2nn_i \ll \mathcal{F}[x \mapsto \mathbb{T}x] \sim 2n^2.$$

Separable operators

$\mathbf{T}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a separable operator if

$$\mathbf{T}(x) = (\mathbf{U}_1(x_1), \dots, \mathbf{U}_m(x_m)),$$

where $\mathbf{U}_i: \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$ for $i = 1, \dots, m$. Separable operators are coordinate friendly if $\max_{i=1, \dots, m} \mathcal{F}[x_i \mapsto \mathbf{U}_i(x_i)] \ll \mathcal{F}[x \mapsto \mathbf{T}(x)]$.

Common example: multiplication by a (block) diagonal matrix.

$f: \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is a separable function if

$$f(x) = \sum_{i=1}^m f_i(x_i),$$

where $f_i: \mathbb{R}^{n_i} \rightarrow \overline{\mathbb{R}}$ for $i = 1, \dots, m$. If f is separable and differentiable, then ∇f is separable. If f is separable and CCP, then Prox_f is separable.

Separable operators

A separable constraint is of the form

$$x_i \in C_i \quad \text{for } i = 1, \dots, m.$$

Projection onto a separable constraint is separable.

Common example: box constraint

$$a_i \leq x_i \leq b_i \quad \text{for } i = 1, \dots, m.$$

Extended coordinate-friendly

\mathbb{T} is extended coordinate-friendly if there is an auxiliary quantity $y(x)$ such that

$$\max_{i=1,\dots,m} \mathcal{F} [\{x, y(x)\} \mapsto \{\mathbb{T}_i x, y(\mathbb{T}_i x)\}] \ll \mathcal{F} [x \mapsto \mathbb{T}x].$$

In other words, computing $\mathbb{T}_i(x)$ is efficient if $y(x)$ is maintained.

More coordinate notation

Use notation $x = (x_1, \dots, x_m)$ with $x_i \in \mathbb{R}^{n_i}$. For $A \in \mathbb{R}^{r \times n}$, write

$$A_{:,i} \in \mathbb{R}^{r \times n_i}$$

for the submatrix, i.e.,

$$A = [A_{:,1} \quad \cdots \quad A_{:,m}]$$

and

$$Ax = A_{:,1}x_1 + \cdots + A_{:,m}x_m.$$

When f is differentiable, write

$$\nabla f(x) = \begin{bmatrix} \nabla_1 f(x) \\ \vdots \\ \nabla_m f(x) \end{bmatrix}.$$

Example: Gradient descent on least squares

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|Ax - b\|^2,$$

where $A \in \mathbb{R}^{r \times n}$ and $b \in \mathbb{R}^r$, and

$$\mathbb{T}(x) = x - \alpha A^\top (Ax - b).$$

\mathbb{T} is parallelizable, *not* coordinate friendly, but extended coordinate friendly.

Evaluation of \mathbb{T} costs

$$\mathcal{F}[x \mapsto \mathbb{T}x] = \mathcal{O}(rn).$$

Parallelizable (assuming $p < \min\{r, n\}$):

$$\begin{aligned}\mathcal{F}_p[x \mapsto \mathbb{T}x] &= \mathcal{F}_p[\{A, x\} \mapsto Ax] + \mathcal{F}_p[\{A^\top, Ax\} \mapsto A^\top(Ax)] \\ &= \mathcal{O}(rn/p)\end{aligned}$$

Evaluation of \mathbb{T} costs

$$\mathcal{F}[x \mapsto \mathbb{T}x] = \mathcal{O}(rn).$$

Parallelizable (assuming $p < \min\{r, n\}$):

$$\begin{aligned}\mathcal{F}_p[x \mapsto \mathbb{T}x] &= \mathcal{F}_p[\{A, x\} \mapsto Ax] + \mathcal{F}_p[\{A^\top, Ax\} \mapsto A^\top(Ax)] \\ &= \mathcal{O}(rn/p)\end{aligned}$$

Not coordinate friendly:

$$\begin{aligned}\mathcal{F}[x \mapsto \mathbb{T}_i x] &= \mathcal{F}[x \mapsto Ax] + \mathcal{F}[Ax \mapsto \mathbb{T}_i x] \\ &= \mathcal{O}(rn) + \mathcal{O}(rn_i) \\ &= \mathcal{O}(rn)\end{aligned}$$

However, extended coordinate friendly with auxiliary quantity Ax :

$$\mathcal{F} [\{x, Ax\} \mapsto \{\mathbb{T}_i x, A(\mathbb{T}_i x)\}] = \mathcal{O}(rn_i)$$

if we use the formula

$$A(\mathbb{T}_i x) = Ax + A_{:,i}((\mathbb{T}x)_i - x_i).$$

However, extended coordinate friendly with auxiliary quantity Ax :

$$\mathcal{F} [\{x, Ax\} \mapsto \{\mathbb{T}_i x, A(\mathbb{T}_i x)\}] = \mathcal{O}(rn_i)$$

if we use the formula

$$A(\mathbb{T}_i x) = Ax + A_{:,i}((\mathbb{T}x)_i - x_i).$$

Therefore the C-FPI with \mathbb{T}

$$\begin{aligned}x_{i(k)}^{k+1} &= x_{i(k)}^k - \alpha A_{:,i(k)}^\top (y^k - b) \\x_j^{k+1} &= x_j^k \quad \text{for } j \neq i(k) \\y^{k+1} &= y^k + A_{:,i(k)} (x_{i(k)}^{k+1} - x_{i(k)}^k)\end{aligned}$$

costs $\mathcal{O}(rn_{i(k)})$ flops per iteration. ($x_j^{k+1} = x_j^k$ costs no operations.)
Initialize $x^0 = 0$ and $y = Ax^0 = 0$.

If $r \leq n$ or $r \sim n$, using $\mathbb{T}(x) = x - \alpha((A^\top A)x - A^\top b)$ is not effective.

Precomputing

$$\mathcal{F}[\{A, b\} \mapsto \{A^\top A, A^\top b\}] = \mathcal{O}(rn^2)$$

can be prohibitively expensive when $r \ll n$, and

$$\mathcal{F}[\{x^k, A^\top A, A^\top b\} \mapsto x_i^{k+1}] = \mathcal{O}(nn_i),$$

is larger than $\mathcal{O}(rn_i)$.

If $r \leq n$ or $r \sim n$, using $\mathbb{T}(x) = x - \alpha((A^\top A)x - A^\top b)$ is not effective.

Precomputing

$$\mathcal{F}[\{A, b\} \mapsto \{A^\top A, A^\top b\}] = \mathcal{O}(rn^2)$$

can be prohibitively expensive when $r \ll n$, and

$$\mathcal{F}[\{x^k, A^\top A, A^\top b\} \mapsto x_i^{k+1}] = \mathcal{O}(nn_i),$$

is larger than $\mathcal{O}(rn_i)$.

If $r \gg n$ and precomputing $A^\top A$ is feasible, then

$$\mathbb{T}(x) = (\mathbb{I} - \alpha A^\top A)x + \alpha A^\top b$$

is coordinate friendly.

Outline

Randomized coordinate fixed-point iteration

Coordinate and extended coordinate friendly operators

Examples

Discussion

Example: Coordinate gradient descent

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where f is differentiable. RC-FPI applied to $\mathbb{I} - \alpha \nabla f$

$$x_{i(k)}^{k+1} = x_{i(k)}^k - \alpha \nabla_{i(k)} f(x^k),$$

is randomized/stochastic coordinate gradient method or randomized/stochastic coordinate gradient descent. Converges if a minimizer exists, f is L -smooth, and $\alpha \in (0, 2/L)$.

In general, $\mathbb{I} - \alpha \nabla f$ may not be extended coordinate friendly.

However, the following machine learning setup is extended coordinate friendly

$$f(x) = \sum_{j=1}^r \ell_j(a_j^\top x - b_j),$$

where $a_1, \dots, a_r \in \mathbb{R}^n$, $b_1, \dots, b_r \in \mathbb{R}$, and ℓ_1, \dots, ℓ_r are differentiable CCP functions on \mathbb{R} .

Write

$$A = \begin{bmatrix} - & a_1^\top & - \\ & \vdots & \\ - & a_r^\top & - \end{bmatrix} \in \mathbb{R}^{r \times n}, \quad \ell(y) = \sum_{j=1}^r \ell_j(y_j).$$

Then

$$\nabla \ell(x) = (\ell'_1(x_1), \dots, \ell'_r(x_r)).$$

Randomized coordinate gradient descent with $y^k = Ax^k$

$$\begin{aligned}x_{i(k)}^{k+1} &= x_{i(k)}^k - \alpha A_{:,i(k)}^\top \nabla \ell(y^k - b) \\y^{k+1} &= y^k + A_{:,i(k)}(x_{i(k)}^{k+1} - x_{i(k)}^k)\end{aligned}$$

has cost per iteration of $\mathcal{O}(rn_{i(k)})$, if $\max_{j=1,\dots,r} \mathcal{F}[x \mapsto \ell'_j(x)] = \mathcal{O}(1)$.

Initialize $x^0 = 0$ and $y = Ax^0 = 0$.

Example: Coordinate GD with block-wise stepsize

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where f is L -smooth. For any diagonal matrix

$$D = \begin{bmatrix} \beta_1 I_{n_1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \beta_m I_{n_m} \end{bmatrix}$$

where $\beta_i > 0$ and I_{n_i} is the $n_i \times n_i$ identity matrix, the problem is equivalent to

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(Dx).$$

Randomized coordinate gradient method on equivalent problem is

$$x_{i(k)}^{k+1} = x_{i(k)}^k - \alpha_{i(k)} \nabla_{i(k)} f(x^k),$$

where $\alpha_{i(k)} = \alpha \beta_{i(k)}$. Non-uniform block-wise stepsize is often necessary for a speedup compared to the (full deterministic) gradient method.

Example: Coordinate proximal-gradient descent

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \sum_{i=1}^m g_i(x_i)$$

where f is differentiable. So we minimize sum of a differentiable function and a separable function. Write

$$g(x) = \sum_{i=1}^m g_i(x_i).$$

Example: Coordinate proximal-gradient descent

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \sum_{i=1}^m g_i(x_i)$$

where f is differentiable. So we minimize sum of a differentiable function and a separable function. Write

$$g(x) = \sum_{i=1}^m g_i(x_i).$$

RC-FPI with FBS operator $\text{Prox}_{\alpha g}(I - \alpha \nabla f)$

$$x_{i(k)}^{k+1} = \text{Prox}_{\alpha g_{i(k)}}(x_{i(k)}^k - \alpha \nabla_{i(k)} f(x^k)),$$

is coordinate proximal-gradient (descent) method. Converges if a minimizer exists, f is L -smooth, and $\alpha \in (0, 2/L)$.

Example: Coordinate proximal-gradient descent

With block-wise argument, we get

$$x_{i(k)}^{k+1} = \text{Prox}_{\alpha_{i(k)} g_{i(k)}} \left(x_{i(k)}^k - \alpha_{i(k)} \nabla_{i(k)} f(x^k) \right).$$

Non-uniform block-wise stepsizes important for speedup.

When g is not separable, $\text{Prox}_{\alpha g}(I - \alpha \nabla f)$ is in general not extended coordinate friendly and RC-FPI not efficient.

Example: Stochastic dual coordinate ascent

Consider

$$\underset{x \in \mathbb{R}^r}{\text{minimize}} \quad g(x) + \sum_{i=1}^n \ell_i(a_i^\top x - b_i),$$

where g is a strongly convex CCP function on \mathbb{R}^r (so g^* is smooth) and ℓ_i is a CCP function on \mathbb{R} . Write

$$A = \begin{bmatrix} - & a_1^\top & - \\ & \vdots & \\ - & a_n^\top & - \end{bmatrix} \in \mathbb{R}^{n \times r}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n.$$

Primal problem generated by

$$\mathbf{L}(x, u) = g(x) + \langle u, Ax - b \rangle - \sum_{i=1}^n \ell_i^*(u_i)$$

and corresponding dual problem is

$$\underset{u \in \mathbb{R}^n}{\text{maximize}} \quad -g^*(-A^\top u) - b^\top u - \sum_{i=1}^n \ell_i^*(u_i).$$

Example: Stochastic dual coordinate ascent

Randomized coordinate proximal-gradient applied to dual

$$\begin{aligned}u_{i(k)}^{k+1} &= \text{Prox}_{\alpha_{i(k)} \ell_{i(k)}^*} \left(u_{i(k)}^k + \alpha_{i(k)} (A_{i(k),:} \nabla g^*(y^k) - b_{i(k)}) \right) \\y^{k+1} &= y^k - A_{i(k),:}^\top (u_{i(k)}^{k+1} - u_{i(k)}^k)\end{aligned}$$

is a variation of stochastic dual coordinate ascent. Assume $\mathcal{F}[y \mapsto \nabla g^*(y)] = \mathcal{O}(r)$ and $\max_{i=1, \dots, n} \mathcal{F}[u \mapsto \text{Prox}_{\alpha_i \ell_i^*}(u)] = \mathcal{O}(1)$. Extended coordinate friendly with $y^k = -A^\top u^k$ maintained. We have

$$\mathcal{F}[\{y^k, u^k\} \mapsto \{y^{k+1}, u^{k+1}\}] = \mathcal{O}(rn_{i(k)}).$$

(One can recover the primal solution with $\nabla g^*(y^k)$.)

Note on splitting data

Iteration of primal coordinate GD accesses $A_{:,i(k)}$, a block of columns.
Iteration of dual coordinate GD accesses $A_{i(k),:}$, a block of rows.

In machine learning, a row of A is a training sample, and we may not want to split it into parts. In so, dual approach is preferred.

Example: MISO/Finito

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad r(x) + \frac{1}{m} \sum_{i=1}^m f_i(x),$$

where f_1, \dots, f_m are differentiable. Use consensus technique to get

$$\underset{\mathbf{x} \in \mathbb{R}^{nm}}{\text{minimize}} \quad \delta_C(\mathbf{x}) + \sum_{i=1}^m (r(x_i) + f_i(x_i)),$$

where $\mathbf{x} = (x_1, \dots, x_m)$ and C is the consensus set.

Write $f(\mathbf{x}) = \sum_{i=1}^m f_i(x_i)$ and $g(\mathbf{x}) = \delta_C(\mathbf{x}) + \sum_{i=1}^m r(x_i)$, so

$$\text{Prox}_{\alpha g}(y_1, \dots, y_m) = (x, \dots, x), \quad x = \text{Prox}_{\alpha r} \left(\frac{1}{m} \sum_{i=1}^m y_i \right).$$

(See Exercise 2.28.)

Both FBS and BFS are extended coordinate friendly with \bar{z}^k maintained.

Both FBS and BFS are extended coordinate friendly with \bar{z}^k maintained.
RC-FPI with BFS operator $(\mathbf{I} - \alpha \nabla f) \text{Prox}_{\alpha g}$ is

$$\begin{aligned}x^k &= \text{Prox}_{\alpha r}(\bar{z}^k) \\z_{i(k)}^{k+1} &= x^k - \alpha \nabla f_{i(k)}(x^k) \\ \bar{z}^{k+1} &= \bar{z}^k + \frac{1}{m} \left(z_{i(k)}^{k+1} - z_{i(k)}^k \right).\end{aligned}$$

RC-FPI with FBS operator $\text{Prox}_{\alpha g}(\mathbf{I} - \alpha \nabla f)$ is

$$\begin{aligned}x_{i(k)}^{k+1} &= \text{Prox}_{\alpha r}(\bar{z}^k) \\ \bar{z}^{k+1} &= \bar{z}^k + \frac{1}{m} \left(x_{i(k)}^{k+1} - x_{i(k)}^k - \alpha (\nabla f_i(x_{i(k)}^{k+1}) - \nabla f_i(x_{i(k)}^k)) \right),\end{aligned}$$

where $\bar{z}^k = \frac{1}{m} \sum_{i=1}^m (x_i^k - \alpha \nabla f_{i(k)}(x_i^k))$ is maintained.

Both FBS and BFS are extended coordinate friendly with \bar{z}^k maintained.
RC-FPI with BFS operator $(\mathbf{I} - \alpha \nabla f) \text{Prox}_{\alpha g}$ is

$$\begin{aligned}x^k &= \text{Prox}_{\alpha r}(\bar{z}^k) \\z_{i(k)}^{k+1} &= x^k - \alpha \nabla f_{i(k)}(x^k) \\ \bar{z}^{k+1} &= \bar{z}^k + \frac{1}{m} \left(z_{i(k)}^{k+1} - z_{i(k)}^k \right).\end{aligned}$$

RC-FPI with FBS operator $\text{Prox}_{\alpha g}(\mathbf{I} - \alpha \nabla f)$ is

$$\begin{aligned}x_{i(k)}^{k+1} &= \text{Prox}_{\alpha r}(\bar{z}^k) \\ \bar{z}^{k+1} &= \bar{z}^k + \frac{1}{m} \left(x_{i(k)}^{k+1} - x_{i(k)}^k - \alpha (\nabla f_i(x_{i(k)}^{k+1}) - \nabla f_i(x_{i(k)}^k)) \right),\end{aligned}$$

where $\bar{z}^k = \frac{1}{m} \sum_{i=1}^m (x_i^k - \alpha \nabla f_{i(k)}(x_i^k))$ is maintained.

These two equivalent methods are called minimization by incremental surrogate optimization (MISO) or Finito. Converges if a solution exists and $\alpha \in (0, 2/L)$.

Among the two, BFS has a minor and subtle advantage.

For BFS, one can use $(z_1^0, \dots, z_m^0) = (0, \dots, 0)$ and $\bar{z}^0 = 0$ as the starting point.

For FBS, the starting point $(x_1^0, \dots, x_m^0) \in \mathbb{R}^{nm}$ can be arbitrary, but

$$\bar{z}^0 = \frac{1}{m} \sum_{i=1}^m (x_i^0 - \alpha \nabla f_i(x_i^0))$$

needs to be computed before starting the iterations in order to establish convergence via Theorem 2.

Example: Conic programs with many small cones

Consider

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^\top x \\ \text{subject to} & Ax = b \\ & x \in Q_1 \times \cdots \times Q_m, \end{array}$$

where $Q_i \subseteq \mathbb{R}^{n_i}$ is a nonempty closed convex set, $A \in \mathbb{R}^{r \times n}$ has rank r , and $b \in \mathbb{R}^r$. Assume $\mathcal{F}[x_i \mapsto \Pi_{Q_i} x_i] = C_i$.

Note: $[x \in Q_1 \times \cdots \times Q_m] \Leftrightarrow [x_i \in Q_i \text{ for } i = 1, \dots, m]$

When Q_1, \dots, Q_m are convex cones, problem called a conic program.

Naive RC-FPI with DRS applied to

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{c^\top x + \delta_{\{x \mid Ax=b\}}(x)}_{=f(x)} + \underbrace{\delta_{Q_1 \times \dots \times Q_m}(x)}_{=g(x)}$$

becomes

$$\begin{aligned} x_i^{k+1/2} &= \Pi_{Q_i}(z_i^k) \quad \text{for } i = 1, \dots, m \\ z_{i(k)}^{k+1} &= z_{i(k)}^k + D_{i(k),:}(2x^{k+1/2} - z^k) + v_{i(k)} - x_{i(k)}^{k+1/2}, \end{aligned}$$

where $D = I - A^\top(AA^\top)^{-1}A$ and $v = A^\top(AA^\top)^{-1}b - \alpha Dc$. (Exercise 2.24.) Costs $\mathcal{O}(C_1 + \dots + C_n + nn_{i(k)})$ per iteration.

Utilize the extended coordinate friendly structure with

$$y^k = D2x^{k+1/2} - z^k:$$

$$x_{i(k)}^{k+1/2} = \Pi_{Q_{i(k)}}(z_{i(k)}^k)$$

$$z_{i(k)}^{k+1} = z_{i(k)}^k + y_{i(k)}^k + v_{i(k)} - x_{i(k)}^{k+1/2}$$

$$y^{k+1} = D_{:,i(k)} \left(2\Pi_{Q_{i(k)}}(z_{i(k)}^{k+1}) - 2x_{i(k)}^{k+1/2} - z_{i(k)}^{k+1} + z_{i(k)}^k \right),$$

which costs $\mathcal{O}(C_{i(k)} + nn_{i(k)})$ per iteration.

Example: Conic programs with large second-order cone

Second-order cone programming (SOCP) is a generalization linear programming with constraints in the the second-order cone

$$Q = \left\{ x \in \mathbb{R}^n \mid \sqrt{x_1^2 + \cdots + x_{n-1}^2} \leq x_n \right\}.$$

Write $x = (u, t)$ where $u \in \mathbb{R}^{n-1}$ and $t \in \mathbb{R}$. Projection $x = (u, t)$ to Q scales u and t :

$$\Pi_Q(x) = (\rho u, \sigma t), \quad (\rho, \sigma) = \begin{cases} (0, 0) & \text{if } t \leq -\|u\| \\ (1, 1) & \text{if } \|u\| \leq t \\ \frac{1}{2}(\|u\| + t) \left(\frac{1}{\|u\|}, \frac{1}{t} \right) & \text{otherwise.} \end{cases}$$

See Exercise 5.2.

Conic programs with large second-order cone

Consider

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^T x \\ \text{subject to} & Ax = b \\ & x \in Q \end{array}$$

where Q is the second-order cone, $A \in \mathbb{R}^{r \times n}$ with rank r , and $b \in \mathbb{R}^r$. For simplicity, assume all blocks are single coordinates, i.e., $m = n$ and $n_1 = \dots n_m = 1$.

Apply DRS to

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{c^T x + \delta_{\{x \mid Ax=b\}}(x)}_{=f(x)} + \underbrace{\delta_Q(x)}_{=g(x)}.$$

Define $D = I - A^\top(AA^\top)^{-1}A$ and $v = A^\top(AA^\top)^{-1}b - \alpha Dc$ and write

$$z^k = (u^k, t^k), \quad D = \begin{bmatrix} G & f \end{bmatrix},$$

where $G \in \mathbb{R}^{n \times (n-1)}$ and $f \in \mathbb{R}^n$. The DRS operator is extended coordinate friendly, and we can implement the RC-FPI with DRS as

$$\begin{aligned} x_{i(k)}^{k+1/2} &= \Pi_Q(z_{i(k)}^k) \\ z_{i(k)}^{k+1} &= z_{i(k)}^k + v_{i(k)} - x_{i(k)}^{k+1/2} \\ &\quad + (2\rho(\|u^k\|, t^k) - 1)(Gu^k)_{i(k)} + (2\sigma(\|u^k\|, t^k) - 1)t^k f_{i(k)} \\ \|u^{k+1}\|^2 &= \begin{cases} \|u^k\|^2 + (z_{i(k)}^{k+1})^2 - (z_{i(k)}^k)^2 & \text{if } i(k) < n \\ \|u^k\|^2 & \text{if } i(k) = n \end{cases} \\ Gu^{k+1} &= \begin{cases} Gu^k + G_{:,i(k)}(u_{i(k)}^{k+1} - u_{i(k)}^k) & \text{if } i(k) < n \\ Gu^k & \text{if } i(k) = n. \end{cases} \end{aligned}$$

This implementation costs $\mathcal{O}(n)$ flops per iteration.

Outline

Randomized coordinate fixed-point iteration

Coordinate and extended coordinate friendly operators

Examples

Discussion

Discussion

- ▶ RC-FPI may provide a speedup but no guarantees.

Discussion

- ▶ RC-FPI may provide a speedup but no guarantees.
- ▶ Larger stepsizes often, but not always, translate to speedup.

Discussion

- ▶ RC-FPI may provide a speedup but no guarantees.
- ▶ Larger stepsizes often, but not always, translate to speedup.
- ▶ RC-FPI is a precursor to asynchronous FPI in Chapter 6.