Overview

1 Coordinate Descent

2 Variations of Coordinate Descent
   - CD with different indexing
   - BCD
   - CD with different updates
   - Inexact BCD / BSUM

3 Summary
An unconstrained optimization problem

\[
\min_x f(x) = f(x_1, x_2, \ldots, x_n)
\]

- \( x \in \mathbb{R}^n \) is the vector variable of \( f \)
- \( x_i \) are scalar coordinates of \( x \)
- function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is continuous
- no further assumption on the structure of \( f \) (e.g. convex, Lipschitz, differentiability)
Algorithm 1 CD framework

1: OUTPUT: $x \in \mathbb{R}^n$ that minimizes $f(x)$
2: INITIALIZATION: $x^0 \in \mathbb{R}^n$
3: for $k = 1, 2, ...$ until convergence condition is satisfied do
4:   Indexing: Pick a coordinate index $i_k$
5:   Updating: Update the selected coordinate $x_{i_k}^k$ using $f$ and the previous iterate $x^k$ while holding other coordinates fix:

$$x_j^k = \begin{cases} \text{Update}(f, x^{k-1}) & \text{if } j = i_k \\ x_j^{k-1} & \text{if } j \neq i_k \end{cases}$$

6: end for

The Update$(f, x^{k-1})$ itself is to solve a optimization sub-problem. So CD itself is not really an algorithm but an conceptual algorithmic framework.
The Update \( (f, \mathbf{x}^{k-1}) \)

It is to renew the selected component \( x_{i_k} \) by minimizing the objective function \( f \) with respect to \( x_{i_k} \) while holding all other coordinates fix

\[
x_{i_k}^k = \arg \min_{x_{i_k}} f(x_{i_k}^{k-1}, x_2^{k-1}, \ldots, x_{i_k-1}^{k-1}, x_{i_k}, x_{i_k+1}^{k-1}, \ldots, x_n^{k-1})
\]

Notational rearrange,

\[
x_{i_k}^k = \arg \min_{x_{i_k}} f(x_{i_k}; x_{1}^{k-1}, x_2^{k-1}, \ldots, x_{i_k-1}^{k-1}, x_{i_k+1}^{k-1}, \ldots, x_n^{k-1})
\quad \text{ where } \mathbf{x}^{k-1} \neq i_k
\]

Short hand notation

\[
x_{i_k}^k = \arg \min_{x_{i_k}} f(x_{i_k}; \mathbf{x}^{k-1})
\]
Algorithm 2 CD framework for solving $\min_x f(x)$

1: INITIALIZATION : $x^0 \in \mathbb{R}^n$
2: for $k = 1, 2, ...$ until convergence condition is satisfied do
3:   Indexing : Pick the coordinate index $i_k$
4:   Updating : $x_{i_k} = \text{Update}(f, x^{k-1}) = \arg\min_{x_{i_k}} f(x_{i_k}; x^{k-1})$
5: end for

CD is an conceptual framework on design of algorithm to solve optimization problem. Thus variations can be introduced on

- Indexing : the way to select $i_k$
- Updating : the way to formulate $\text{Update}(f, x^{k-1})$

The next pages will be on the variations of CD.
Algorithm 3 CD framework for solving $\min_x f(x)$

1: INITIALIZATION : $x^0 \in \mathbb{R}^n$

2: for $k = 1, 2, ...$ until convergence condition is satisfied do

3: Cyclic indexing : $i_{k+1} = (k \mod n) + 1$

4: Updating : $x_{i_k} = \text{Update}(f, x^{k-1}) = \arg \min_{x_{i_k}} f(x_{i_k}; x^{k-1}_{\neq i_k})$

5: end for

- simple indexing scheme
- what it does : select index in cyclic manner

\[ i_k = \underbrace{1, 2, 3, \ldots, n}_\text{one cycle}, \underbrace{1, 2, 3, \ldots, n}_\text{one cycle}, \ldots \]

- In general : as long as it cycle through all indices then ok. Can be irregular as :

\[ i_k = \underbrace{7, 2, 3, \ldots, 6, 8, \ldots, n}_\text{one cycle}, \underbrace{7, 2, 3, \ldots, 6, 8, \ldots, n}_\text{one cycle}, 1, \ldots \]
Variations of CD : random indexing

Algorithm 4 CD framework for solving $\min_x f(x)$

1: INITIALIZATION : $x^0 \in \mathbb{R}^n$

2: for $k = 1, 2, \ldots$ until convergence condition is satisfied do

3: Random indexing : pick $i_k$ according to $\mathbb{P}(i_k = j) = p_j$, where $\{p_i\}_{i=1}^n$ are some assigned probability

4: Updating : $x_{i_k} = \text{Update}(f, x^{k-1}) = \arg\min_{x_{i_k}} f(x_{i_k}; x^{k-1} \neq i_k)$

5: end for

Ways to assign probability :

- Uniform : index is chosen with equal chance $p_j = \frac{1}{n}$
- Importance : important coordinate has a higher chance being selected. "Importance" can be defined in various ways. For example, $p_j$ can be defined as the portion of the coordinate-wise Lipschitz constant among all coordinate $p_j = \frac{L_j}{\sum_j L_j}$ (in fact, this indexing achieve a faster convergence than using uniform sampling if any $L_i$ differs)
Variations of CD: greedy indexing

Algorithm 5 CD framework for solving $\min_x f(x)$

1: INITIALIZATION: $x^0 \in \mathbb{R}^n$
2: for $k = 1, 2, ...$ until convergence condition is satisfied do
3: Pick $i_k$ greedily
4: Updating: $x_{i_k} = \text{Update}(f, x^{k-1}) = \arg\min_{x_{i_k}} f(x_{i_k}; x^{k-1}_{i_k})$
5: end for

Greedy can be defined by maximum improvement

$$i_k = \arg\min_j f(x_j, x^{k-1})$$

or, if the function is differentiable (non-differentiable), Greedy can be defined as picking the index with largest gradient (sub-gradient)

$$i_k = \arg\min_j \|\nabla_j f(x^{k-1})\|$$

or gradient normalized by Lipschitz constant

$$i_k = \arg\min_j \frac{\|\nabla_j f(x^{k-1})\|}{\sqrt{L_j}}$$
Variations of CD : Block Coordinate Descent

If variable \( x \) is decomposed into \( s \) blocks that each block \( x_i \) is a collection of coordinate, then CD becomes BCD, which share the same algorithmic structure as CD :

\[
\text{Algorithm 6 CD framework for solving } \min_x f(x)
\]

1: INITIALIZATION : \( x^0 \in \mathbb{R}^n \)
2: for \( k = 1, 2, \ldots \) until convergence condition is satisfied do
3: Indexing : Pick the coordinate index \( i_k \in \{1, 2, \ldots, s\} \)
4: Updating : \( x_{i_k} = \text{Update}(f, x^{k-1}) = \arg \min_{x_{i_k}} f(x_{i_k}; x^{k-1}_{i_k} \neq i_k) \)
5: end for

Difference between BCD and CD :
- CD on coordinate (scalar component of \( x \))
- BCD on block of coordinate (vector component of \( x \))
- \( n \) coordinates in \( x \) for CD, \( s \) blocks for BCD

CD can be seen as a special case of BCD.
Variations of CD: coordinate minimization update

Now consider the variation on updating. The coordinate minimization update is:

**Algorithm 7** CD framework for solving \( \min_x f(x) \)

1: **INITIALIZATION**: \( x^0 \in \mathbb{R}^n \)
2: **for** \( k = 1, 2, ... \) until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index \( i_k \in \{1, 2, ..., s\} \)
4: **Updating**: \( x_{i_k} = \arg\min_{x_{i_k}} f(x_{i_k}; x^{k-1}_{\neq i_k}) \)
5: **end for**

Line 4 itself is also an optimization problem: if (computable) close form solution exists for line 4, done.
Otherwise numerical optimization method such as gradient descent can be applied on line 4:

\[
x_{i_k} \leftarrow x_{i_k} - t_{i_k} \nabla_{i_k} f(x^{k-1})
\]

\( t_{i_k} \) step size, \( \nabla_{i_k} \) partial gradient. Such CD framework is called **coordinate gradient descent** which requires \( f(x_{i_k}; x^{k-1}_{\neq i_k}) \) to be differentiable. If \( f \) is not differentiable, we get **coordinate sub-gradient algorithm** by replacing \( \nabla f \) with sub-gradient.
Adding a quadratic term on the sub-problem of coordinate minimization update, we get coordinate proximal point algorithm:

**Algorithm 8** CD framework for solving $\min_x f(x)$

1. **INITIALIZATION** : $x^0 \in \mathbb{R}^n$
2. **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3. **Indexing** : Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4. **Updating** : $x_{i_k} = \arg\min_{x_{i_k}} f(x_{i_k}; x_{k-1}^{\neq i_k}) + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2$
5. **end for**

- $x_{i_k}^-$ the previous iterate of $x_{i_k}$
- $\|x_{i_k} - x_{i_k}^-\|^2_2$ the proximal term
- $\alpha_{i_k}$ a positive constant (proximal point parameter)

The addition of the quadratic proximal term "gives" certain advantage for solving the sub-problem.

E.g. if $f$ not differentiable/smooth, the addition of the proximal term (with a suitable $\alpha_{i_k}$) makes it differentiable/smooother
Variations of CD: proximal gradient update

For structured $f$ as $f(x) = g(x) + h(x)$, proximal gradient update can be used:

**Algorithm 9** CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Updating**:
   
   $x_{i_k} = \arg\min_{x_{i_k}} g(x^{k-1}) + \left< \nabla_{i_k} g(x_{i_k}^k; x_{\neq i_k}^{k-1}), x_{i_k} - x_{i_k}^- \right> + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2 + h(x_{i_k})$
5: **end for**

What it does: minimizes the local quadratic model of $g(x)$ plus the non-differentiable term $h(x)$. 

---

Variations of CD: proximal gradient update

For structured $f$ as $f(x) = g(x) + h(x)$, proximal gradient update can be used:

**Algorithm 9** CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Updating**:
   
   $x_{i_k} = \arg\min_{x_{i_k}} g(x^{k-1}) + \left< \nabla_{i_k} g(x_{i_k}^k; x_{\neq i_k}^{k-1}), x_{i_k} - x_{i_k}^- \right> + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2 + h(x_{i_k})$
5: **end for**

What it does: minimizes the local quadratic model of $g(x)$ plus the non-differentiable term $h(x)$.

---

Variations of CD: proximal gradient update

For structured $f$ as $f(x) = g(x) + h(x)$, proximal gradient update can be used:

**Algorithm 9** CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Updating**:
   
   $x_{i_k} = \arg\min_{x_{i_k}} g(x^{k-1}) + \left< \nabla_{i_k} g(x_{i_k}^k; x_{\neq i_k}^{k-1}), x_{i_k} - x_{i_k}^- \right> + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2 + h(x_{i_k})$
5: **end for**

What it does: minimizes the local quadratic model of $g(x)$ plus the non-differentiable term $h(x)$. 

---

Variations of CD: proximal gradient update

For structured $f$ as $f(x) = g(x) + h(x)$, proximal gradient update can be used:

**Algorithm 9** CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Updating**:
   
   $x_{i_k} = \arg\min_{x_{i_k}} g(x^{k-1}) + \left< \nabla_{i_k} g(x_{i_k}^k; x_{\neq i_k}^{k-1}), x_{i_k} - x_{i_k}^- \right> + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2 + h(x_{i_k})$
5: **end for**

What it does: minimizes the local quadratic model of $g(x)$ plus the non-differentiable term $h(x)$. 

---

Variations of CD: proximal gradient update

For structured $f$ as $f(x) = g(x) + h(x)$, proximal gradient update can be used:

**Algorithm 9** CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ until convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Updating**:
   
   $x_{i_k} = \arg\min_{x_{i_k}} g(x^{k-1}) + \left< \nabla_{i_k} g(x_{i_k}^k; x_{\neq i_k}^{k-1}), x_{i_k} - x_{i_k}^- \right> + \frac{1}{2\alpha_{i_k}^{k-1}} \|x_{i_k} - x_{i_k}^-\|^2_2 + h(x_{i_k})$
5: **end for**

What it does: minimizes the local quadratic model of $g(x)$ plus the non-differentiable term $h(x)$.
Algorithm 10 CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$
2: **for** $k = 1, 2, \ldots$ **until** convergence condition is satisfied **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, \ldots, s\}$
4: **Updating**: $x_{i_k} = \text{Update}(f, x^{k-1})$
5: **end for**

As CD an algoritmic framework, other updates can be applied (depends on the structure of $f$). For examples

- second order methods (if Hessian of $f$ is "computable")
- dual method (if dual problem is easier to solve)
- primal-dual methods (if dual problem is easier to solve)
- ADMM
- etc.

In these cases we get coordinate Newton, coordinate Dual ascent, Coordinate primal-dual algorithm, coordinate ADMM.
Algorithm 11 CD framework for solving $\min_x f(x)$

1: **INITIALIZATION**: $x^0 \in \mathbb{R}^n$

2: **for** $k = 1, 2, \ldots$ **until** convergence condition is satisfied **do**

3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, \ldots, s\}$

4: **Updating**: $x_{i_k} = \text{Update}(f, x^{k-1})$

5: **end for**

Suppose the sup-problem in line 4 cannot be solved easily: coordinate minimization, coordinate gradient descent, proximal point update and proximal gradient update mentioned before are not easily applicable to the problem $f$.

As CD is an algorithmic framework, one can consider incorporating the idea of *Majorization Minimization* here and get *inexact BCD/BSUM*. 
BSUM (Block Successive Upper Bound Minimization)

Idea: as $x_{i_k} = \text{Update}(f, x^{k-1})$ is not "friendly", so instead of working on $f$, we construct a surrogate/majorizer/upper bound of $f$, denoted as $u$.

We work on the minimization of $u$, then use $u$ to update $f$.

If $f$ is non-convex but $u$ is convex, we can say such approach is a convex relaxation: as the original non-convex $f$ is now relaxed to a convex $u$.

Price to pay by relaxation: relaxation gap $u - f$. Normally after each time $u$ is minimized, the upper bound $u$ is updated to reduce the gap.

Such "relax-update-modify" approach is carried out successively, thus the framework is called SUM: Successive Upper Bound Minimization.

As we are not working on the original function $f$ but an upper bound, thus this framework is also called inexact BCD.
BSUM (Block Successive Upper Bound Minimization)

The optimization problem:

$$\min_x f(x)$$

with $f$ not so ”user-friendly”.

---

**Algorithm 12 Inexact BCD / BSUM**

1: **INITIALZATION**: $x \in \mathbb{R}^n$
2: **for** $k = 1, 2, ...$ **until convergence condition is satisfied** **do**
3: **Indexing**: Pick the coordinate index $i_k \in \{1, 2, ..., s\}$
4: **Relax**: Construct an upper bound $u$
5: **Updating**: $x_{i_k} = \text{Update}(u, x^{k-1})$
6: **Modify**: Modify the upper bound $u(x)$ to reduce the relaxation gap
7: **end for**

There are some requirements on the construction of the upper bound $u$ to ensure convergence, which are out of the scope here.
Introduced:
- Coordinate Descent in the most fundamental form
- Block Coordinate Descent
- Some variations on BCD such as indexing and updating
- Inexact BCD/BSUM

Not discussed:
- The convergence of BCD with various indexing and updating
- How to select which indexing and updating scheme to use
- Acceleration of CD
- Application of BCD and inexact BCD

End of document