Modern Optimization Techniques

2. Unconstrained Optimization / 2.6. Coordinate Descent

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

Mon. 28.10.	(0)	0. Overview
		1. Theory
Mon. 4.11.	(1)	1. Convex Sets and Functions
		2. Unconstrained Optimization
Mon. 11.11.	(2)	2.1 Gradient Descent
Mon. 18.11.	(3)	2.2 Stochastic Gradient Descent
Mon. 25.11.	(4)	2.3 Newton's Method
Mon. 2.12.	(5)	2.4 Quasi-Newton Methods
Mon. 19.12.	(6)	2.5 Subgradient Methods
Mon. 16.12.	(7)	2.6 Coordinate Descent
	—	— *Christmas Break* —
		3. Equality Constrained Optimization
Mon. 6.1.	(8)	3.1 Duality
Mon. 13.1.	(9)	3.2 Methods
		4. Inequality Constrained Optimization
Mon. 20.1.	(10)	4.1 Primal Methods
Mon. 27.1.	(11)	4.2 Barrier and Penalty Methods
Mon. 3.2.	(12)	4.3 Cutting Plane Methods

Outline

1. Idea and Optimality
2. Coordinate Descent Algorithm
3. Convergence

Outline

Coordinate Descent

- Gradient Descent and Stochastic Gradient Descent:
  - update **all** variables simultaneously.
  - use the gradient to do so.
    (first order methods)
- **Coordinate Descent**:
  - update **one** variable at a time.
  - use an **analytic solver** to do so
    (**derivative-free method**)
    - if not possible: use one-dimensional gradient steps
      (first order method; often slow)

Coordinate Descent

We start with an initial guess $\mathbf{x}^{(0)}$

For $k = 1, 2, 3, \ldots$:

Coordinate Descent

We start with an initial guess $\mathbf{x}^{(0)}$

For $k = 1, 2, 3, \ldots$:

$$\mathbf{x}_1^{(k)} \leftarrow \arg\min_{x_1} f(x_1, \mathbf{x}_2^{(k-1)}, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

Coordinate Descent

We start with an initial guess $\mathbf{x}^{(0)}$

For $k = 1, 2, 3, \ldots$:

$$\mathbf{x}_1^{(k)} \leftarrow \arg\min_{x_1} f(x_1, \mathbf{x}_2^{(k-1)}, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$
$$\mathbf{x}_2^{(k)} \leftarrow \arg\min_{x_2} f(\mathbf{x}_1^{(k)}, x_2, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

Coordinate Descent

We start with an initial guess $\mathbf{x}^{(0)}$

For $k = 1, 2, 3, \ldots$:

$$\mathbf{x}_1^{(k)} \leftarrow \arg\min_{x_1} f(x_1, \mathbf{x}_2^{(k-1)}, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

$$\mathbf{x}_2^{(k)} \leftarrow \arg\min_{x_2} f(\mathbf{x}_1^{(k)}, x_2, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

$$\mathbf{x}_3^{(k)} \leftarrow \arg\min_{x_3} f(\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, x_3, \ldots, \mathbf{x}_n^{(k-1)})$$

Coordinate Descent

We start with an initial guess $\mathbf{x}^{(0)}$

For $k = 1, 2, 3, \ldots$:

$$\mathbf{x}_1^{(k)} \leftarrow \arg\min_{x_1} f(x_1, \mathbf{x}_2^{(k-1)}, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

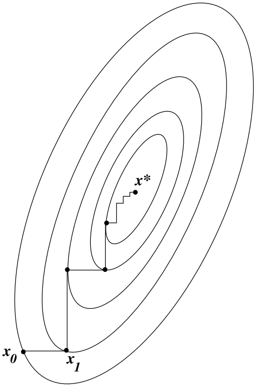
$$\mathbf{x}_2^{(k)} \leftarrow \arg\min_{x_2} f(\mathbf{x}_1^{(k)}, x_2, \mathbf{x}_3^{(k-1)}, \ldots, \mathbf{x}_n^{(k-1)})$$

$$\mathbf{x}_3^{(k)} \leftarrow \arg\min_{x_3} f(\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, x_3, \ldots, \mathbf{x}_n^{(k-1)})$$

$$\vdots$$

$$\mathbf{x}_n^{(k)} \leftarrow \arg\min_{x_n} f(\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \mathbf{x}_3^{(k)}, \ldots, x_n)$$

Coordinate Descent Algorithm



[Nocedal and Wright, 2006, p.249]

Does Coordinate-wise Minimization Lead to Global Minima?

Question: If a point $\mathbf{x}$ is minimal along each axis, is it a global minimum?

$$f(\mathbf{x} + t\,\mathbf{e}^{(\mathbf{n})}) \geq f(\mathbf{x}) \quad \forall t \in \mathbb{R}, \forall n \in \{1, \ldots, N\}$$
$$\stackrel{?}{\Longrightarrow} f(\mathbf{x}) = \min_{\mathbf{y}} f(\mathbf{y})$$

where:

- $\mathbf{e}^{(\mathbf{n})} \in \mathbb{R}^N$ is the n-th unit vector with $\mathbf{e}_n^{(n)} := 1$ and $\mathbf{e}_m^{(n)} := 0$ for $m \neq n$.

Does Coordinate-wise Minimization Lead to Global Minima?

If f is **convex** and **differentiable**: yes.

Proof: $g^{(i)}(\mu) := f(\mathbf{x} + \mu\,\mathbf{e^{(i)}})$ are convex and differentiable.
$\mu = 0$ is their minimum, thus

$$0 = \frac{\partial g^{(i)}}{\partial \mu}(\mu) = \frac{\partial f}{\partial \mathbf{x}_i}(\mathbf{x})$$

And then

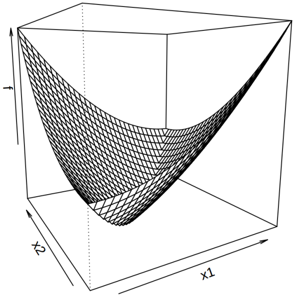
$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_1}, \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_n} \right)^T = \mathbf{0}$$

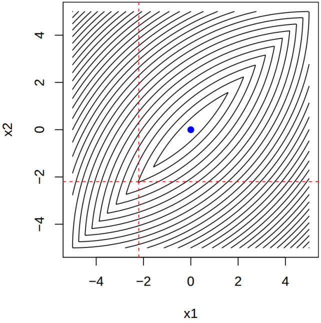
and thus x is the global optimum.

Does Coordinate-wise Minimization Lead to Global Minima?

If f is **convex**, but **not differentiable**: in general, no.

Counter example:





https://www.cs.cmu.edu/~ggordon/10725-F12/slides/25-coord-desc.pdf

Does Coordinate-wise Minimization Lead to Global Minima?

If f is a sum of a differentiable convex and a separable convex function: yes.

$$f(\mathbf{x}) = g(\mathbf{x}) + \sum_{n=1}^{N} h_n(\mathbf{x}_n)$$

with

i. g is differentiable and convex

ii. all h_n are convex
(but each h_n depends only on a single $\mathbf{x}_n$)

Does Coordinate-wise Minimization Lead to Global Minima? / Proof

Proof: For any $\mathbf{y} \in \operatorname{dom} f$:

$$f(\mathbf{y}) - f(\mathbf{x}) \underset{i.}{\geq} \nabla g(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \sum_{n=1}^{N}(h_n(\mathbf{y}_n) - h_n(\mathbf{x}_n))$$

$$= \sum_{n=1}^{N} \underbrace{\nabla_n g(\mathbf{x})(\mathbf{y}_n - \mathbf{x}_n) + h_n(\mathbf{y}_n) - h_n(\mathbf{x}_n)}_{\geq 0} \geq 0$$

where $\geq$ is argued as follows: as x is minimal along axis n,

$\leadsto$ $f(X_n; X_{-n})$ is convex and has minimum at $X_n = x_n$

$\leadsto$ 0 is a subgradient of $f(X_n; X_{-n})$ at x_n :

$$0 \in \partial f(X_n; X_{-n}) = \nabla g(x) + \partial h_n(x_n)$$

$\leadsto$ $\exists s \in \partial h_n(x_n) : \nabla g(x) + s = 0$

$$\nabla_n g(x)(y_n - x_n) + h_n(y_n) - h_n(x_n)$$

$$\underset{ii.\text{ sub grad}}{\geq} \nabla g(x)(y_n - x_n) + s(y_n - x_n) = 0$$

Outline

One-dimensional Subproblems

Let $f : X \rightarrow \mathbb{R}, X \subseteq \mathbb{R}^N$ be a function, $x \in \operatorname{dom} f$ a point.
***n*-th one-dimensional subproblem at x:**

$$
\begin{aligned}
g_n^{(x)} &: T_n^{(x)} \rightarrow \mathbb{R} \\
g_n^{(x)}(t) &:= f(x + t\, e^{(n)}) \\
T_n^{(x)} &:= \{t \in \mathbb{R} \mid x + t\, e^{(n)} \in \operatorname{dom} f\}
\end{aligned}
$$

***n*-th one-dimensional subproblem solver at x:**

$$
h_n(x) := \arg\min_{t \in T_n^{(x)}} g_n^{(x)}(t)
$$

One-dimensional Subproblems / Example

Solving a linear system of equations / least squares / linear regression:

$$f(x) := x^T A x - b^T x, \quad A \in \mathbb{R}^{N\times N} \text{ sym. pos.def.}, b \in \mathbb{R}^N$$

$$g_n^{(x)}(t) = f(x_n = t; x_{-n}), \quad T_n^{(x)} = \mathbb{R}$$

$$f(x_n; x_{-n}) = A_{n,n} x_n^2 + (2(x_{-n}^T A_{-n,.})_n - b_n) x_n + x_{-n}^T A_{-n,-n} x_{-n} - b_{-n}^T x_{-n}$$

$$= A_{n,n} x_n^2 + (2A_{n,-n} x_{-n} - b_n) x_n + x_{-n}^T A_{-n,-n} x_{-n} - b_{-n}^T x_{-n}$$

analytic minimum:

$$f'(x_n; x_{-n}) = 2A_{n,n} x_n + 2A_{n,-n} x_{-n} - b_n \stackrel{!}{=} 0$$

$$\rightsquigarrow \quad x_n = \frac{b_n - 2A_{n,-n} x_{-n}}{2A_{n,n}}$$

i.e., $h_n(x) := \dfrac{b_n - 2A_{n,-n} x_{-n}}{2A_{n,n}}$

Coordinate Descent / Random Coordinate

min−cd($f, h, x^{(0)}, K, \epsilon$):
  for $k := 1, \ldots, K$:
    draw $n^{(k)} \sim \text{unif}(\{1, \ldots, N\})$
    $x_m^{(k)} := x_m^{(k-1)}$ for $m \in \{1, \ldots, N\}, m \neq n^{(k)}$
    $x_{n^{(k)}}^{(k)} := h_{n^{(k)}}(x^{(k-1)})$
    if $k \geq N$ and $||x^{(k)} - x^{(k-N)}|| \leq \epsilon$
      return $x^{(k)}$
  return "not converged"

where

- h solvers for one-dimensional subproblems $g_n^{(x)}$.
- ϵ convergence threshold step in an epoche

Coordinate Descent / Cyclic Epochs

$\text{min-cd}(f, h, x^{(0)}, K, \epsilon)$:
  for $k := 1, \ldots, K$:
    $x_1^{(k)} := h_1(x_1^{(k-1)}, \ldots, x_N^{(k-1)})$
    $x_2^{(k)} := h_2(x_1^{(k)}, x_2^{(k-1)}, \ldots, x_N^{(k-1)})$
    $x_3^{(k)} := h_3(x_1^{(k)}, x_2^{(k)}, x_3^{(k-1)}, \ldots, x_N^{(k-1)})$
    $\vdots$
    $x_N^{(k)} := h_N(x_1^{(k)}, \ldots, x_{N-1}^{(k)}, x_N^{(k-1)})$
    if $||x^{(k)} - x^{(k-1)}|| \leq \epsilon$
      return $x^{(k)}$
  return "not converged"

where

- h solvers for one-dimensional subproblems $g_n^{(x)}$.
- ϵ convergence threshold step in an epoche

Coordinate Descent / Cyclic Epochs

min−cd$(f, h, x^{(0)}, K, \epsilon)$:
    $x^{(0,N)} := x^{(0)}$
    for $k := 1, \ldots, K$:
        $x^{(k,0)} := x^{(k-1,N)}$
        for $n := 1, \ldots, N$:
            $x_m^{(k,n)} := x_m^{(k,n-1)}$ for $m \in \{1, \ldots, N\}, m \neq n$
            $x_n^{(k,n)} := h_n(x^{(k,n-1)})$
        if $||x^{(k,N)} - x^{(k-1,N)}|| \leq \epsilon$
            return $x^{(k,N)}$
    return "not converged"

where

- h solvers for one-dimensional subproblems $g_n^{(x)}$.
- ϵ convergence threshold step in an epoche

Coordinate Descent / 1-dim Gradient Steps

$\text{min}-\text{cd}(f, \nabla f, x^{(0)}, K, \epsilon)$:
  for $k := 1, \ldots, K$:
    draw $n^{(k)} \sim \text{unif}(\{1, \ldots, N\})$
    $x^{(k)} := x^{(k-1)} - \mu^{(k)} (\nabla f(x^{(k-1)}))_{n^{(k)}} e^{(n^{(k)})}$
    if $k \geq N$ and $||x^{(k)} - x^{(k-N)}|| \leq \epsilon$
      return $x^{(k)}$
  return "not converged"

where

- ∇f gradients of objective function.
- $\mu \in (\mathbb{R}^+)^*$ step length schedule (or controller)
- ϵ convergence threshold step in an epoche

Coordinate Descent - Considerations

- The order in which we cycle through the coordinates is arbitrary.
  - e.g., cyclic
  - better should be randomized.
- We may update blocks of coordinates at a time instead of only one (**block coordinate descent**)
- No need to adjust a step-size!
  - if we have exact solvers h for the 1-dim. subproblems.
- Does not work in general with non-differentiable functions

Outline

Coordinate Lipschitz Constant

- standard Lipschitz constant L:

$$||\nabla f(x) - \nabla f(y)|| \leq L||x-y|| \qquad \forall x,y \in \text{dom } f$$

or equivalently

$$||\nabla f(x+d) - \nabla f(x)|| \leq L||d|| \qquad \forall x \in \text{dom } f, d \in \mathbb{R}^N : x+d \in \text{dom } f$$

- **component Lipschitz constant** L_n: $(n \in \{1,\ldots,N\})$

$$||\nabla f(x+t\, e^{(n)}) - \nabla f(x)|| \leq |t|\, L_n \qquad \forall x \in \text{dom } f, t \in \mathbb{R} : x + t\, e^{(n)} \in \text{dom } f$$

- **coordinate Lipschitz constant** $L_{\max}$:

$$L_{\max} := \max_{n \in \{1,\ldots,N\}} L_n$$

Note: In the following $||\cdot||$ denotes the L2-norm.

Lipschitz Continuous Functions / Bounded Derivative

Lemma

A differentiable function $f : X \to \mathbb{R}, X \subseteq \mathbb{R}$ is Lipschitz-continuous iff its derivative is bounded:

$$|f(x) - f(y)| \leq L|x - y| \quad \forall x, y \in \text{dom}\, f \quad \iff \quad |f'(x)| \leq L \quad \forall x \in \text{dom}\, f$$

Proof: "$\Rightarrow$":

$$\left|\frac{f(x) - f(x+t)}{|t|}\right| = \frac{|f(x) - f(x+t)|}{|t|} \leq \frac{|t|L}{|t|} \leq L \quad |t \to 0$$

$$|f'(x)| \leq L$$

"$\Leftarrow$":

$$f(x) \underset{Taylor}{=} f(y) + f'(\xi)(x-y) \quad \textit{for a } \xi \in [x, y]$$

$$\leq f(y) + L(x-y)$$

$$|f(x) - f(y)| \leq L|x-y|$$

Theorem (convergence of coordinate descent)

If

i. $f : X \to \mathbb{R}$, $X \subseteq \mathbb{R}^N$ *is convex and differentiable,*

ii. ∇f *is uniformly Lipschitz-continuous:*
$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||, \quad L \in \mathbb{R}_0^+$

iii. *the sublevel set of* $x^{(0)}$ *is bounded:*
$\max_{x^*: f(x^*)=\min_x f(x)} \max_{x \in S_f(x^{(0)})} ||x - x^*|| \leq R, \quad R \in \mathbb{R}_0^+$

iv. *constant steplength* $\mu^{(k)} := 1/L_{\max}$ *is used,*

then coordinate descent converges and

$$\mathbb{E}(f(x^{(k)})) - f(x^*) \leq \frac{2NL_{\max}R^2}{k}$$

Convergence / Proof (1/3)

$$
\begin{aligned}
f(x^{\text{next}}) &= f(x - \mu(\nabla f(x))_n e^{(n)}) \\
&\underset{\text{Taylor}}{=} f(x) - \nabla f(x)^T \mu(\nabla f(x))_n e^{(n)} \\
&\quad + \frac{1}{2}(\mu(\nabla f(x))_n e^{(n)})^T \underbrace{\nabla^2 f(x - \xi\mu(\nabla f(x))_n e^{(n)})}_{\underset{\text{ii. bound.deriv.}}{\leq} L_n} \mu(\nabla f(x))_n e^{(n)} \\
&\leq f(x) - \nabla f(x))_n^2 \mu + \frac{1}{2}\mu^2(\nabla f(x))_n^2 L_n \\
&= f(x) - (\nabla f(x))_n^2 \mu(1 - \frac{1\mu L_n}{2}) \\
&\leq f(x) - (\nabla f(x))_n^2 \mu(1 - \frac{1\mu L_{\max}}{2}) \\
&\underset{iv.}{=} f(x) - \frac{(\nabla f(x))_n^2}{2L_{\max}}
\end{aligned}
$$

Convergence / Proof (2/3)

$$f(x^{\text{next}}) \leq f(x) - \frac{(\nabla f(x))_n^2}{2L_{\max}} \quad |\mathbb{E}_n(\ldots)$$

$$\mathbb{E}_n(f(x^{\text{next}})) \leq \mathbb{E}_n(f(x) - \frac{(\nabla f(x))_n^2}{2L_{\max}})$$

$$= f(x) - \frac{1}{N}\sum_{n=1}^{N} \frac{(\nabla f(x))_n^2}{2L_{\max}}) = f(x) - \frac{1}{2NL_{\max}}||\nabla f(x)||^2 \quad (1)$$

$$\mathbb{E}_{n^{(0:K)}} := \mathbb{E}_{n^{(0)},n^{(1)},\ldots,n^K} \quad \text{expectation over } \textbf{all } n^{(k)}$$

$$\phi^{(k)} := \mathbb{E}_{n^{(0:K)}}(f(x^{(k)})) - f(x^*)$$

$$f(x) - f(x^*) \underset{i.}{\leq} \nabla f(x)^T(x - x^*) \leq ||\nabla f(x)|| \, ||x - x^*|| \underset{iii.}{\leq} R||\nabla f(x)||$$

$$\mathbb{E}_{n^{(0:K)}}(||\nabla f(x^{(k)})||) \geq \frac{1}{R}\phi^{(k)} \quad (2)$$

Convergence / Proof (3/3)

$$\mathbb{E}_{n^{(k)}}(f(x^{(k+1)})) \underset{(1)}{\leq} f(x^{(k)}) - \frac{1}{2NL_{\max}}||\nabla f(x^{(k)})||^2 \quad |\mathbb{E}_{n^{(0:K)}}$$

$$\begin{aligned}
\phi^{(k+1)} &\leq \phi^{(k)} - \frac{1}{2NL_{\max}}\mathbb{E}_{n^{(0:K)}}(||\nabla f(x^{(k)})||^2) \\
&\underset{\text{Jensen's Ineq.}}{\leq} \phi^{(k)} - \frac{1}{2NL_{\max}}(\mathbb{E}_{n^{(0:K)}}(||\nabla f(x^{(k)})||))^2 \\
&\underset{(2)}{\leq} \phi^{(k)} - \frac{1}{2NL_{\max}R^2}(\phi^{(k)})^2 \qquad (3)
\end{aligned}$$

$$\frac{1}{\phi^{(k+1)}} - \frac{1}{\phi^{(k)}} = \frac{\phi^{(k)} - \phi^{(k+1)}}{\phi^{(k)}\phi^{(k+1)}} \geq \frac{\phi^{(k)} - \phi^{(k+1)}}{(\phi^{(k)})^2} \underset{(3)}{\geq} \frac{1}{2NL_{\max}R^2}$$

$$\frac{1}{\phi^{(k+1)}} \underset{\text{rec}}{\geq} \frac{1}{\phi^{(0)}} + \frac{k+1}{2NL_{\max}R^2} \geq \frac{k+1}{2NL_{\max}R^2}$$

Summary

- **Coordinate descent** minimizes one-dimensional subproblems for **a single variable x_n** at a time.
  - in cyclic or random order

- Coordinate descent can be fast if the **one-variable subproblems can be solved analytically**.

- If an x^* **minimizes all single-variable problems**, then it also is a minimizer for the all-variable problem
  - if f is convex and differentiable, or
  - if $f = g + h$ is the sum of
    - a convex and differentiable g and
    - a convex, possibly **non-differentiable**, but and **separable** $h(x_1, \ldots, x_N) = h_1(x_1) + h_2(x_2) + \cdots + h_N(x_N)$

- **Convergence** of coordinate descent can be proven for mild conditions
  - e.g., f differentiable, ∇f Lipschitz, sublevel set of $x^{(0)}$ bounded, and constant steplength $1/L_{\max}$.

Further Readings

- The coordinate descent method is not covered by Boyd and Vandenberghe [2004]
- Coordinate descent:
  - very briefly [Nocedal and Wright, 2006, ch. 9.3]
  - A brief, but dense survey: Wright [2015]
- Convergence proof: Wright [2015]

References

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Science+ Business Media, 2006.

Stephen J. Wright. Coordinate Descent Algorithms. *Math. Program.*, 151(1):3–34, June 2015. ISSN 0025-5610. doi: 10.1007/s10107-015-0892-3.

Outline

4. Examples

Coordinate Descent - Example

For $\mathbf{x} \in \mathbb{R}^2$

$$\min_{\mathbf{x}} \quad (a_1 x_1 - a_2 x_2 + a_3)^2$$

Algorithm:

- Initialize $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}$
- Repeat until convergence:
  - $\mathbf{x}_1^{(k)} \leftarrow \arg\min_{x_1} (a_1 x_1 - a_2 \mathbf{x}_2^{(k-1)} + a_3)^2$
  - $\mathbf{x}_2^{(k)} \leftarrow \arg\min_{x_2} (a_1 \mathbf{x}_1^{(k)} - a_2 x_2 + a_3)^2$

Coordinate Descent - Example

$$\arg\min_{x_1} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_1} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Coordinate Descent - Example

$$\arg\min_{x_1} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_1}(a_1 x_1 - a_2 x_2 + a_3)^2 = 2(a_1 x_1 - a_2 x_2 + a_3)a_1$$

Coordinate Descent - Example

$$\arg\min_{x_1} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_1}(a_1 x_1 - a_2 x_2 + a_3)^2 = 2(a_1 x_1 - a_2 x_2 + a_3)a_1$$

$$x_1 = \frac{a_2 x_2 - a_3}{a_1}$$

Coordinate Descent - Example

$$\arg\min_{x_2}(a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_2}(a_1 x_1 - a_2 x_2 + a_3)^2$$

Coordinate Descent - Example

$$\arg\min_{x_2} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_2}(a_1 x_1 - a_2 x_2 + a_3)^2 = -2(a_1 x_1 - a_2 x_2 + a_3) a_2$$

Coordinate Descent - Example

$$\arg\min_{x_2} (a_1 x_1 - a_2 x_2 + a_3)^2$$

Find a closed form solution:

$$0 \stackrel{!}{=} \frac{d}{dx_2}(a_1 x_1 - a_2 x_2 + a_3)^2 = -2(a_1 x_1 - a_2 x_2 + a_3)a_2$$

$$x_2 = \frac{a_1 x_1 + a_3}{a_2}$$

Coordinate Descent - Example

For $\mathbf{x} \in \mathbb{R}^2$

$$\min_{\mathbf{x}} \quad (a_1 x_1 - a_2 x_2 + a_3)^2$$

Algorithm:

- Initialize $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}$
- Repeat until convergence:
  - $\mathbf{x}_1^{(k)} \leftarrow \frac{a_2 \mathbf{x}_2^{(k-1)} - a_3}{a_1}$
  - $\mathbf{x}_2^{(k)} \leftarrow \frac{a_1 \mathbf{x}_1^{(k)} + a_3}{a_2}$

Coordinate Descent - Example

For $\mathbf{x} \in \mathbb{R}^2$, $a_1 = 0.1$, $a_2 = 2$, $a_3 = 1$

$$\min_{\mathbf{x}} \quad (0.1x_1 - 2x_2 + 1)^2$$

$$\mathbf{x}_1^{(k)} \leftarrow \frac{2\mathbf{x}_2^{(k-1)} - 1}{0.1} \quad \mathbf{x}_2^{(k)} \leftarrow \frac{0.1\mathbf{x}_1^{(k)} + 1}{2}$$

Coordinate Descent - Example

For $\mathbf{x} \in \mathbb{R}^2$, $a_1 = 0.1$, $a_2 = 2$, $a_3 = 1$

$$\min_{\mathbf{x}} \quad (0.1x_1 - 2x_2 + 1)^2$$

$$\mathbf{x}_1^{(k)} \leftarrow \frac{2\mathbf{x}_2^{(k-1)} - 1}{0.1} \quad \mathbf{x}_2^{(k)} \leftarrow \frac{0.1\mathbf{x}_1^{(k)} + 1}{2}$$

Start with $\mathbf{x}_1^{(0)} = 1, \mathbf{x}_2^{(0)} = 2$

- $\mathbf{x}_1^{(1)} \leftarrow \frac{2 \cdot 2 - 1}{0.1} = 30$
- $\mathbf{x}_2^{(1)} \leftarrow \frac{0.1 \cdot 30 + 1}{2} = 2$

Coordinate Descent - Example

For $\mathbf{x} \in \mathbb{R}^2$, $a_1 = 0.1$, $a_2 = 2$, $a_3 = 1$

$$\min_{\mathbf{x}} \quad (0.1x_1 - 2x_2 + 1)^2$$

$$\mathbf{x}_1^{(k)} \leftarrow \frac{2\mathbf{x}_2^{(k-1)} - 1}{0.1} \quad \mathbf{x}_2^{(k)} \leftarrow \frac{0.1\mathbf{x}_1^{(k)} + 1}{2}$$

Start with $\mathbf{x}_1^{(0)} = 1, \mathbf{x}_2^{(0)} = 2$

- $\mathbf{x}_1^{(1)} \leftarrow \frac{2 \cdot 2 - 1}{0.1} = 30$
- $\mathbf{x}_2^{(1)} \leftarrow \frac{0.1 \cdot 30 + 1}{2} = 2$

$\mathbf{x}_1^{(1)} = 30, \mathbf{x}_2^{(1)} = 2$

- $\mathbf{x}_1^{(2)} \leftarrow \frac{2 \cdot 2 - 1}{0.1} = 30$
- $\mathbf{x}_2^{(2)} \leftarrow \frac{0.1 \cdot 30 + 1}{2} = 2$

Coordinate Descent for Linear Regression

For the problem

$$\text{minimize} \quad \sum_{i=1}^{m}(y_i - \mathbf{x}^T \mathbf{a_i})^2 = \sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{n} x_j a_{ij}\right)^2$$

We can compute the update rule for a specific x_k:

$$\frac{\partial f(\mathbf{x})}{\partial x_k} \stackrel{!}{=} 0$$

$$\frac{\partial f(\mathbf{x})}{\partial x_k} = 2 \cdot \sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij}\right)$$

Coordinate Descent for Linear Regression

$$2 \cdot \sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij} \right) = 0$$

$$\sum_{i=1}^{m} a_{ik} \cdot y_i - \sum_{i=1}^{m} a_{ik} \cdot \sum_{j=1}^{n} x_j a_{ij} = 0$$

$$\sum_{i=1}^{m} a_{ik} \cdot y_i - \sum_{i=1}^{m} a_{ik} \cdot \left(x_k a_{ik} + \sum_{j=1,\, j \neq k}^{n} x_j a_{ij} \right) = 0$$

$$\sum_{i=1}^{m} a_{ik} \cdot y_i - \sum_{i=1}^{m} a_{ik} \cdot x_k a_{ik} - \sum_{i=1}^{m} a_{ik} \cdot \sum_{j=1,\, j \neq k}^{n} x_j a_{ij} = 0$$

$$\sum_{i=1}^{m} a_{ik} \cdot y_i - x_k \sum_{i=1}^{m} a_{ik}^2 - \sum_{i=1}^{m} a_{ik} \cdot \sum_{j=1,\, j \neq k}^{n} x_j a_{ij} = 0$$

Coordinate Descent for Linear Regression

$$\sum_{i=1}^{m} a_{ik} \cdot y_i - x_k \sum_{i=1}^{m} a_{ik}^2 - \sum_{i=1}^{m} a_{ik} \cdot \sum_{j=1,\, j\neq k}^{n} x_j a_{ij} = 0$$

$$x_k \cdot \sum_{i=1}^{m} a_{ik}^2 = \sum_{i=1}^{m} a_{ik} \cdot y_i - \sum_{i=1}^{m} a_{ik} \cdot \sum_{j=1,\, j\neq k}^{n} x_j a_{ij}$$

$$x_k = \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1,\, j\neq k}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2}$$

Linear Regression Coordinate Descent - Simple Algorithm

procedure LINEAR REGRESSION-CD
  input: f
  Get initial point $\mathbf{x}^{(0)}$
  repeat
    for $k \in 1, \ldots n$ **do**
      $$x_k \leftarrow \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1,\, j \neq k}^{n} x_j a_{ij}\right)}{\sum_{i=1}^{m} a_{ik}^2}$$
    end for
  until convergence
  return $\mathbf{x}$, $f(\mathbf{x})$
end procedure

Coordinate Descent for Linear Regression

For each parameter we have the following update rule:

$$x_k = \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1,\, j \neq k}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2}$$

One Coordinate descent epoch has a cost of $O(m \cdot n^2)$!

Can we do it faster?

CD for Linear Regression - Smart Update

For each parameter we have the following update rule:

$$x_k = \frac{\sum_{i=1}^m a_{ik} \cdot \left(y_i - \sum_{j=1,\ j \neq k}^n x_j a_{ij} \right)}{\sum_{i=1}^m a_{ik}^2}$$

We can rewrite:

$$\begin{aligned}
\sum_{i=1}^m \left(y_i - \sum_{j=1,\ j\neq k}^n x_j a_{ij} \right) &= \sum_{i=1}^m \left(y_i - \sum_{j=1}^n x_j a_{ij} + x_k a_{ik} \right) \\
&= \sum_{i=1}^m \left(y_i - \sum_{j=1}^n x_j a_{ij} \right) + \sum_{i=1}^m x_k a_{ik} \\
&= \sum_{i=1}^m \left(y_i - \sum_{j=1}^n x_j a_{ij} \right) + x_k \sum_{i=1}^m a_{ik}
\end{aligned}$$

CD for Linear Regression - Smart Update

From which we have:

$$x_k = \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1,\ j\neq k}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2}$$

$$= \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij} + \mathbf{x}_k^{old} a_{ik} \right)}{\sum_{i=1}^{m} a_{ik}^2}$$

$$= \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2} + \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(\mathbf{x}_k^{old} a_{ik} \right)}{\sum_{i=1}^{m} a_{ik}^2}$$

$$= \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2} + \frac{\mathbf{x}_k^{old} \cdot \sum_{i=1}^{m} a_{ik}^2}{\sum_{i=1}^{m} a_{ik}^2}$$

$$= \frac{\sum_{i=1}^{m} a_{ik} \cdot \left(y_i - \sum_{j=1}^{n} x_j a_{ij} \right)}{\sum_{i=1}^{m} a_{ik}^2} + \mathbf{x}_k^{old}$$

CD for Linear Regression - Smart Update

Now we have:

$$x_k = \frac{\sum_{i=1}^m a_{ik} \cdot \left(y_i - \sum_{j=1}^n x_j a_{ij} \right)}{\sum_{i=1}^m a_{ik}^2} + \mathbf{x}_k^{old}$$

So we can define our residual vector $\mathbf{r} \in \mathbb{R}^m$ such that

$$r_i = y_i - \sum_{j=1}^n x_j a_{ij}$$

CD for Linear Regression - Smart Update

After each update of x_k, we can maintain r_i instead of recomputing it given the old value $\mathbf{x}_k^{old}$:

$$\begin{aligned} r_i^{new} &= y_i - \left(\sum_{j=1}^{n} x_j a_{ij} - \mathbf{x}_k^{old} a_{ik} + x_k a_{ik} \right) \\ &= r_i^{old} + (\mathbf{x}_k^{old} - x_k) a_{ik} \end{aligned}$$

CD for Linear Regression - Smart Update

Now our algorithm looks like:

1. Initialize $\mathbf{x}$
2. Compute $r_i = y_i - \sum_{j=1}^{n} x_j a_{ij}$
3. While Not Converged
   3.1 For each $k = 1, \ldots, n$
      3.1.1 $\mathbf{x}_k^{old} \leftarrow x_k$
      3.1.2 $x_k \leftarrow \frac{\sum_{i=1}^{m} a_{ik} \cdot (r_i)}{\sum_{i=1}^{m} a_{ik}^2} + \mathbf{x}_k^{old}$
      3.1.3 For all i $r_i \leftarrow r_i + (\mathbf{x}_k^{old} - x_k) a_{ik}$

This algorithm is now $O(m \cdot n)$!

Linear Regression Coordinate Descent Algorithm

procedure LINEAR REGRESSION-CD
 input: f
 Get initial point $\mathbf{x}^{(0)}$
 $\mathbf{r} \leftarrow \mathbf{y} - A\mathbf{x}^{(0)}$
 repeat
  for $k \in 1, \ldots n$ **do**
   $\mathbf{x}_k^{old} \leftarrow x_k$
   $x_k \leftarrow \frac{\sum_{i=1}^m a_{ik} \cdot r_i}{\sum_{i=1}^m a_{ik}^2} + \mathbf{x}_k^{old}$
   for $i \in 1, \ldots m$ **do**
    $r_i \leftarrow r_i + (\mathbf{x}_k^{old} - x_k) a_{ik}$
   end for
  end for
 until convergence
 return $\mathbf{x}$, $f(\mathbf{x})$
end procedure

Real World Dataset: Body Fat prediction

We want to estimate the percentage of body fat based on various attributes:

- Age (years)
- Weight (lbs)
- Height (inches)
- Neck circumference (cm)
- Chest circumference (cm)
- Abdomen 2 circumference (cm)
- Hip circumference (cm)
- Thigh circumference (cm)

Real World Dataset: Body Fat prediction

The data is represented it as:

$$A_{m,n} = \begin{pmatrix} 1 & a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 1 & a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

with $m = 252$, $n = 14$

We can model the percentage of body fat y is a linear combination of the body measurements with parameters $\mathbf{x}$:

$$\hat{y}_i = \mathbf{x}^T \mathbf{a_i} = x_0 1 + x_1 a_{i,1} + x_2 a_{i,2} + \ldots + x_n a_{i,n}$$

Coordinate Descent - Body fat dataset

Year Prediction Data Set

- Least Squares Problem
- Prediction of the release year of a song from audio features
- 90 features
- Experiments done on a subset of 1000 instances of the data

Coordinate Descent - Year Prediction