Convex Optimization

(EE227A: UC Berkeley)

Lecture 20
(Coordinate descent)

04 Apr, 2013

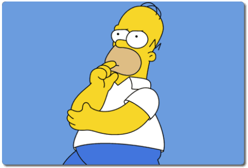
Suvrit Sra

Admin

- ♡ HW3 **due right now!**
- ♡ HW4 is out! Please ask your Qs on Piazza
- ♡ Project 4 page reports due on **4/11/2013**
- ♡ Poster presentations: 3hrs: **When in May?**

Challenge problem

$$I(p) := \sqrt{p} \int_0^\infty \left| \frac{\sin x}{x} \right|^p dx$$



$$\boxed{\textbf{Minimize } I(p) \textbf{ over } p \geq 1}$$

Coordinate descent

$$\boxed{\text{So far: } \min f(x) = \sum_i f_i(x)}$$

Coordinate descent

So far: $\min f(x) = \sum_i f_i(x)$

Since $x \in \mathbb{R}^n$, now consider

$$\min f(x) = f(x_1, x_2, \ldots, x_n)$$

Previously, we went through $f_1, \ldots, f_m$

What if we now go through $x_1, \ldots, x_n$ one by one?

Coordinate descent

- For $k = 0, 1, \ldots$

Coordinate descent

Coordinate descent

- For $k = 0, 1, \ldots$
  - Pick an index i from $\{1, \ldots, n\}$

Coordinate descent

Coordinate descent

- For $k = 0, 1, \ldots$
  - Pick an index i from $\{1, \ldots, n\}$
  - Optimize the ith coordinate

$$x_i^{k+1} \leftarrow \operatorname*{argmin}_{\xi \in \mathbb{R}} f(\underbrace{x_1^{k+1}, \ldots, x_{i-1}^{k+1}}_{\textbf{done}}, \underbrace{\xi}_{\textbf{current}}, \underbrace{x_{i+1}^k, \ldots, x_n^k}_{\textbf{todo}})$$

- Decide when/how to stop; *return* x^k

! x_i^{k+1} **overwrites** value in x_i^k (in actual implementation)

Coordinate descent

- ♣ One of the simplest optimization methods

Coordinate descent

- ♣ One of the simplest optimization methods
- ♣ Old idea: Gauss-Seidel, Jacobi methods for linear systems!

Coordinate descent

- One of the simplest optimization methods
- Old idea: Gauss-Seidel, Jacobi methods for linear systems!
- Can be "slow", but sometimes very competitive

Coordinate descent

- ♣ One of the simplest optimization methods
- ♣ Old idea: Gauss-Seidel, Jacobi methods for linear systems!
- ♣ Can be “slow”, but sometimes very competitive
- ♣ Gradient, subgradient, incremental methods also “slow”

Coordinate descent

- ♣ One of the simplest optimization methods
- ♣ Old idea: Gauss-Seidel, Jacobi methods for linear systems!
- ♣ Can be "slow", but sometimes very competitive
- ♣ Gradient, subgradient, incremental methods also "slow"
- ♣ But incremental, stochastic gradient methods are scalable

Coordinate descent

- ♣ One of the simplest optimization methods
- ♣ Old idea: Gauss-Seidel, Jacobi methods for linear systems!
- ♣ Can be “slow”, but sometimes very competitive
- ♣ Gradient, subgradient, incremental methods also “slow”
- ♣ But incremental, stochastic gradient methods are scalable
- ♣ These days renewed interest in CD for large-scale problems

Coordinate descent

- ♣ One of the simplest optimization methods
- ♣ Old idea: Gauss-Seidel, Jacobi methods for linear systems!
- ♣ Can be “slow”, but sometimes very competitive
- ♣ Gradient, subgradient, incremental methods also “slow”
- ♣ But incremental, stochastic gradient methods are scalable
- ♣ These days renewed interest in CD for large-scale problems
- ♣ Notice: in general CD is “derivative free”

Coordinate descent – which index?

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

♣ **Cyclic** order $1, 2, \ldots, n, 1, \ldots$

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

- ♣ **Cyclic** order $1, 2, \ldots, n, 1, \ldots$
- ♣ **Almost cyclic:** Each coordinate $1 \leq i \leq n$ picked at least once every B successive iterations ($B \geq n$)

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

- ♣ **Cyclic** order $1, 2, \ldots, n, 1, \ldots$
- ♣ **Almost cyclic:** Each coordinate $1 \leq i \leq n$ picked at least once every B successive iterations ($B \geq n$)
- ♣ **Double sweep**, $1, \ldots, n$ then $n-1, \ldots, 1$, repeat

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

- ♣ **Cyclic** order $1, 2, \ldots, n, 1, \ldots$
- ♣ **Almost cyclic:** Each coordinate $1 \leq i \leq n$ picked at least once every B successive iterations ($B \geq n$)
- ♣ **Double sweep**, $1, \ldots, n$ then $n-1, \ldots, 1$, repeat
- ♣ **Cylic with permutation**: random order each cycle

Coordinate descent – which index?

Gauss-Southwell: If f is differentiable, at iteration k, pick the index that minimizes $[\nabla f(x_k)]_i$

Derivative free rules:

- ♣ **Cyclic** order $1, 2, \ldots, n, 1, \ldots$
- ♣ **Almost cyclic:** Each coordinate $1 \le i \le n$ picked at least once every B successive iterations ($B \ge n$)
- ♣ **Double sweep**, $1, \ldots, n$ then $n-1, \ldots, 1$, repeat
- ♣ **Cylic with permutation**: random order each cycle
- ♣ **Random sampling**: pick random index at each iteration

Coordinate descent – Example

$$\min \|Ax - b\|_2^2$$

Coordinate descent – Example

$$\min \|Ax - b\|_2^2$$

Coordinate descent update

$$x_j \leftarrow \frac{\sum_{i=1}^m a_{ij}\left(b_i - \sum_{l \neq j} a_{il} x_l\right)}{\sum_{i=1}^m a_{ij}^2}$$

(dropped superscripts, since we overwrite)

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** 🙂

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** 🙂
- ♢ No other pesky parameters (usually) that must be tuned

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning**
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** ☺
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** 🙂
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems
- ♢ Currently quite popular; parallel versions exist

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** 🙂
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems
- ♢ Currently quite popular; parallel versions exist

Disadvantages

- ♠ Tricky if single variable optimization is hard

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning**
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems
- ♢ Currently quite popular; parallel versions exist

Disadvantages

- ♠ Tricky if single variable optimization is hard
- ♠ Convergence theory can be complicated

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning** 🙂
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems
- ♢ Currently quite popular; parallel versions exist

Disadvantages

- ♠ Tricky if single variable optimization is hard
- ♠ Convergence theory can be complicated
- ♠ Can be slower near optimum than more sophisticated methods

Coordinate descent – some remarks

Advantages

- ♢ Each iteration usually cheap (single variable optimization)
- ♢ No extra storage vectors needed
- ♢ **No stepsize tuning**
- ♢ No other pesky parameters (usually) that must be tuned
- ♢ Simple to implement (like all other methods we've seen so far)
- ♢ Works well for large-scale problems
- ♢ Currently quite popular; parallel versions exist

Disadvantages

- ♠ Tricky if single variable optimization is hard
- ♠ Convergence theory can be complicated
- ♠ Can be slower near optimum than more sophisticated methods
- ♠ Nonsmooth case more tricky

Block coordinate descent (BCD)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) := f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \\ & \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m. \end{aligned}$$

Block coordinate descent (BCD)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) := f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \\ & \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m. \end{aligned}$$

Gauss-Seidel style

$$\boxed{\boldsymbol{x}_i^{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\underbrace{\boldsymbol{x}_1^{k+1}, \ldots, \boldsymbol{x}_{i-1}^{k+1}}_{\textbf{done}}, \underbrace{\boldsymbol{\xi}}_{\textbf{current}}, \underbrace{\boldsymbol{x}_{i+1}^{k}, \ldots, \boldsymbol{x}_m^{k}}_{\textbf{todo}})}$$

Block coordinate descent (BCD)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) := f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \\ & \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m. \end{aligned}$$

Gauss-Seidel style

$$\boxed{\boldsymbol{x}_i^{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\underbrace{\boldsymbol{x}_1^{k+1}, \ldots, \boldsymbol{x}_{i-1}^{k+1}}_{\textbf{done}}, \underbrace{\boldsymbol{\xi}}_{\textbf{current}}, \underbrace{\boldsymbol{x}_{i+1}^{k}, \ldots, \boldsymbol{x}_m^{k}}_{\textbf{todo}})}$$

Jacobi style (easy to parallelize)

$$\boxed{\boldsymbol{x}_i^{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\underbrace{\boldsymbol{x}_1^{k}, \ldots, \boldsymbol{x}_{i-1}^{k}}_{\textbf{don't clobber}}, \underbrace{\boldsymbol{\xi}}_{\textbf{current}}, \underbrace{\boldsymbol{x}_{i+1}^{k}, \ldots, \boldsymbol{x}_m^{k}}_{\textbf{todo}})}$$

BCD – convergence

Theorem Let f be continuously differentiable over $\mathcal{X} := \bigtimes_{i=1}^{m} \mathcal{X}_i$. Further, assume for each block i and $x \in \mathcal{X}$, the minimum

$$\min_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i+1}, \boldsymbol{\xi}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_m)$$

is **uniquely attained**. Every limit point of the sequence $\{\boldsymbol{x}^k\}$ generated by BCD, is a stationary point of f.

BCD – convergence

Theorem Let f be continuously differentiable over $\mathcal{X} := \bigtimes_{i=1}^{m} \mathcal{X}_i$. Further, assume for each block i and $x \in \mathcal{X}$, the minimum

$$\min_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i+1}, \boldsymbol{\xi}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_m)$$

is **uniquely attained**. Every limit point of the sequence $\{\boldsymbol{x}^k\}$ generated by BCD, is a stationary point of f.

Corollary. If f is in addition convex, then every limit point of the BCD sequence $\{\boldsymbol{x}^k\}$ is a global minimum.

BCD – convergence

Theorem Let f be continuously differentiable over $\mathcal{X} := \bigtimes_{i=1}^{m} \mathcal{X}_i$. Further, assume for each block i and $x \in \mathcal{X}$, the minimum

$$\min_{\boldsymbol{\xi} \in \mathcal{X}_i} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i+1}, \boldsymbol{\xi}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_m)$$

is **uniquely attained**. Every limit point of the sequence $\{\boldsymbol{x}^k\}$ generated by BCD, is a stationary point of f.

Corollary. If f is in addition convex, then every limit point of the BCD sequence $\{\boldsymbol{x}^k\}$ is a global minimum.

- **Unique solutions** of subproblems not always possible
- Above result is only **asymptotic** (holds in the limit)
- **Warning!** BCD may cycle indefinitely without converging, if the number of blocks is > 2 and the objective is nonconvex.

BCD – convergence

Two block BCD

$$\text{minimize}\, f(\boldsymbol{x}) = f(\boldsymbol{x}_1, \boldsymbol{x}_2) \quad \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2.$$

BCD – convergence

Two block BCD

$$\text{minimize } f(\boldsymbol{x}) = f(\boldsymbol{x}_1, \boldsymbol{x}_2) \quad \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2.$$

Theorem (Grippo & Sciandrone (2000)). Let f be continuously differentiable, and the sets $\mathcal{X}_1$, $\mathcal{X}_2$ be closed and convex. Assume that the both BCD subproblems have solutions, and that the sequence $\{\boldsymbol{x}^k\}$ has limit points. Then, every limit point of $\{\boldsymbol{x}^k\}$ is stationary.

BCD – convergence

Two block BCD

$$\text{minimize } f(\boldsymbol{x}) = f(\boldsymbol{x}_1, \boldsymbol{x}_2) \quad \boldsymbol{x} \in \mathcal{X}_1 \times \mathcal{X}_2.$$

Theorem (Grippo & Sciandrone (2000)). Let f be continuously differentiable, and the sets $\mathcal{X}_1$, $\mathcal{X}_2$ be closed and convex. Assume that the both BCD subproblems have solutions, and that the sequence $\{\boldsymbol{x}^k\}$ has limit points. Then, every limit point of $\{\boldsymbol{x}^k\}$ is stationary.

- No need of **unique solutions** to subproblems
- BCD for 2 blocks is also called: **Alternating Minimization**

CD for convex problems

CD for smooth convex problems

$$\boxed{\min f(Ax) + \langle b,\, x\rangle \text{ subject to } x \geq 0}$$

- Function f is strictly convex and smooth
- Matrix $A \in \mathbb{R}^{m \times n}$ (possibly rank-deficient)

CD for smooth convex problems

$$\boxed{\min f(Ax) + \langle b,\, x\rangle \text{ subject to } x \geq 0}$$

- Function f is strictly convex and smooth
- Matrix $A \in \mathbb{R}^{m\times n}$ (possibly rank-deficient)

- Apply CD to this problem
- With some more assumptions: it works!
- Even rate of convergence analysis (asymptotic)
- Here's the theorem

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty
3. $\operatorname{dom} f$ is open, and f is strictly convex twice continuously differentiable on $\operatorname{dom} f$

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty
3. $\mathrm{dom}\, f$ is open, and f is strictly convex twice continuously differentiable on $\mathrm{dom}\, f$
4. f tends to $+\infty$ at the boundary of its effective domain

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty
3. $\operatorname{dom} f$ is open, and f is strictly convex twice continuously differentiable on $\operatorname{dom} f$
4. f tends to $+\infty$ at the boundary of its effective domain
5. The Hessian $\nabla^2 f(Ax^*) \succ 0$ for all $x^* \in \mathcal{X}^*$

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty
3. $\operatorname{dom} f$ is open, and f is strictly convex twice continuously differentiable on $\operatorname{dom} f$
4. f tends to $+\infty$ at the boundary of its effective domain
5. The Hessian $\nabla^2 f(Ax^*) \succ 0$ for all $x^* \in \mathcal{X}^*$

Theorem (Luo, Tseng (1992)). Let $\{x^k\}$ be a sequence of iterates generated by the CD method using the almost cyclic or the Gauss-Southwell rule for picking indices. Then, $\{x^k\}$ converges at least linearly to an element of $\mathcal{X}^*$.

CD – convergence theorem

Assumptions:

1. Matrix A has no entirely zero column
2. The set of optimal solutions $\mathcal{X}^*$ is nonempty
3. $\operatorname{dom} f$ is open, and f is strictly convex twice continuously differentiable on $\operatorname{dom} f$
4. f tends to $+\infty$ at the boundary of its effective domain
5. The Hessian $\nabla^2 f(Ax^*) \succ 0$ for all $x^* \in \mathcal{X}^*$

Theorem (Luo, Tseng (1992)). Let $\{x^k\}$ be a sequence of iterates generated by the CD method using the almost cyclic or the Gauss-Southwell rule for picking indices. Then, $\{x^k\}$ converges at least linearly to an element of $\mathcal{X}^*$.

Proof is intricate; see Luo & Tseng's paper on bSpace.

CD – application

Projection onto convex sets

$$\boxed{\begin{array}{ll} \min & \frac{1}{2}\|x-y\|_2^2 \\ \text{s.t.} & x \in C_1 \cap C_2 \cap \cdots \cap C_m. \end{array}}$$

CD – application

Projection onto convex sets

$$
\begin{array}{rl}
\min & \frac{1}{2}\|x-y\|_2^2 \\
\text{s.t.} & x \in C_1 \cap C_2 \cap \cdots \cap C_m.
\end{array}
$$

Solution 1: Rewrite using indicator functions

$$\min \quad \tfrac{1}{2}\|x-y\|_2^2 + \sum_{i=1}^{m} \delta_{C_i}(x).$$

CD – application

Projection onto convex sets

$$\begin{array}{rl} \min & \frac{1}{2}\|x-y\|_2^2 \\ \text{s.t.} & x \in C_1 \cap C_2 \cap \cdots \cap C_m. \end{array}$$

Solution 1: Rewrite using indicator functions

$$\min \quad \tfrac{1}{2}\|x-y\|_2^2 + \textstyle\sum_{i=1}^m \delta_{C_i}(x).$$

- Now invoke Douglas-Rachford using the product-space trick

CD – application

Projection onto convex sets

$$\boxed{\begin{array}{ll} \min & \frac{1}{2}\|x-y\|_2^2 \\ \text{s.t.} & x \in C_1 \cap C_2 \cap \cdots \cap C_m. \end{array}}$$

Solution 1: Rewrite using indicator functions

$$\min \quad \tfrac{1}{2}\|x-y\|_2^2 + \sum\nolimits_{i=1}^m \delta_{C_i}(x).$$

- Now invoke Douglas-Rachford using the product-space trick

Solution 2: Take dual of the above formulation

Convex calculus time

$$\boxed{\min \quad \tfrac{1}{2}\|x - y\|_2^2 + f(x) + h(x)}$$

Convex calculus time

$$\boxed{\min \quad \tfrac{1}{2}\|x-y\|_2^2 + f(x) + h(x)}$$

$$L(x,z,w,\nu,\mu) := \tfrac{1}{2}\|x-y\|_2^2 + f(z) + h(w) + \nu^T(x-z) + \mu^T(x-w)$$

$$\begin{aligned}
g(\nu,\mu) &\quad:=\quad \inf_{x,z,w} L(x,z,\nu,\mu) \\
x-y+\nu+\mu=0 &\quad\Longrightarrow\quad x = y-\nu-\mu \\
g(\nu,\mu) &\quad=\quad -\tfrac{1}{2}\|\nu+\mu\|_2^2 + (\nu+\mu)^T y - f^*(\nu) - h^*(\mu)
\end{aligned}$$

Convex calculus time

$$\boxed{\min \quad \tfrac{1}{2}\|x-y\|_2^2 + f(x) + h(x)}$$

$$L(x,z,w,\nu,\mu) := \tfrac{1}{2}\|x-y\|_2^2 + f(z) + h(w) + \nu^T(x-z) + \mu^T(x-w)$$

$$\begin{array}{rcl} g(\nu,\mu) & := & \inf_{x,z,w} L(x,z,\nu,\mu) \\ x - y + \nu + \mu = 0 & \Longrightarrow & x = y - \nu - \mu \\ g(\nu,\mu) & = & -\tfrac{1}{2}\|\nu+\mu\|_2^2 + (\nu+\mu)^T y - f^*(\nu) - h^*(\mu) \end{array}$$

Dual as minimization problem

$$\boxed{\min \; k(\nu,\mu) := \tfrac{1}{2}\|\nu+\mu-y\|_2^2 + f^*(\nu) + h^*(\mu)}$$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$\begin{aligned} \nu_{k+1} &= \operatorname{argmin}_{\nu}\ k(\nu, \mu_k) \\ \mu_{k+1} &= \operatorname{argmin}_{\mu}\ k(\nu_{k+1}, \mu) \end{aligned}$$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$\begin{aligned} \nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\ \mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu) \end{aligned}$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$\begin{aligned}\nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\ \mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu)\end{aligned}$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$
- $0 \in \nu_{k+1} + \mu - y + \partial h^*(\mu)$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$\begin{aligned} \nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\ \mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu) \end{aligned}$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$
- $0 \in \nu_{k+1} + \mu - y + \partial h^*(\mu)$
- $y - \mu_k \in \nu + \partial f^*(\nu) = (I + \partial f^*)(\nu)$
  $\implies \nu = \operatorname{prox}_{f^*}(y - \mu_k)$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$
\begin{aligned}
\nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\
\mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu)
\end{aligned}
$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$
- $0 \in \nu_{k+1} + \mu - y + \partial h^*(\mu)$
- $y - \mu_k \in \nu + \partial f^*(\nu) = (I + \partial f^*)(\nu)$
  $\implies \nu = \operatorname{prox}_{f^*}(y - \mu_k) \implies \nu = y - \mu_k - \operatorname{prox}_f(y - \mu_k)$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$
\begin{aligned}
\nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\
\mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu)
\end{aligned}
$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$
- $0 \in \nu_{k+1} + \mu - y + \partial h^*(\mu)$
- $y - \mu_k \in \nu + \partial f^*(\nu) = (I + \partial f^*)(\nu)$
  $\implies \nu = \operatorname{prox}_{f^*}(y - \mu_k) \implies \nu = y - \mu_k - \operatorname{prox}_f(y - \mu_k)$
- Similarly, we see that
  $\mu = y - \nu_{k+1} - \operatorname{prox}_h(y - \nu_{k+1})$

Proximal-Dykstra method as CD

Apply CD to $k(\nu, \mu)$

$$\begin{aligned} \nu_{k+1} &= \operatorname{argmin}_\nu \; k(\nu, \mu_k) \\ \mu_{k+1} &= \operatorname{argmin}_\mu \; k(\nu_{k+1}, \mu) \end{aligned}$$

- $0 \in \nu + \mu_k - y + \partial f^*(\nu)$
- $0 \in \nu_{k+1} + \mu - y + \partial h^*(\mu)$
- $y - \mu_k \in \nu + \partial f^*(\nu) = (I + \partial f^*)(\nu)$
  $\implies \nu = \operatorname{prox}_{f^*}(y - \mu_k) \implies \nu = y - \mu_k - \operatorname{prox}_f(y - \mu_k)$
- Similarly, we see that
  $\mu = y - \nu_{k+1} - \operatorname{prox}_h(y - \nu_{k+1})$

$$\boxed{\begin{aligned} \nu_{k+1} &\leftarrow y - \mu_k - \operatorname{prox}_f(y - \mu_k) \\ \mu_{k+1} &\leftarrow y - \nu_{k+1} - \operatorname{prox}_h(y - \nu_{k+1}) \end{aligned}}$$

Proximal-Dykstra as CD

- Simplify, and use Lagrangian stationarity to obtain primal

$$x = y - \nu - \mu \implies y - \mu = x + \nu$$

Proximal-Dykstra as CD

- Simplify, and use Lagrangian stationarity to obtain primal

$$x = y - \nu - \mu \implies y - \mu = x + \nu$$

- Thus, the CD iteration may be rewritten as

$$\begin{aligned} t_k &\leftarrow \mathrm{prox}_f(x_k + \nu_k) \\ \nu_{k+1} &\leftarrow x_k + \nu_k - t_k \\ x_{k+1} &\leftarrow \mathrm{prox}_h(\mu_k + t_k) \\ \mu_{k+1} &\leftarrow \mu_k + t_k - x_{k+1} \end{aligned}$$

Proximal-Dykstra as CD

- Simplify, and use Lagrangian stationarity to obtain primal

$$x = y - \nu - \mu \implies y - \mu = x + \nu$$

- Thus, the CD iteration may be rewritten as

$$\begin{aligned} t_k &\leftarrow \mathrm{prox}_f(x_k + \nu_k) \\ \nu_{k+1} &\leftarrow x_k + \nu_k - t_k \\ x_{k+1} &\leftarrow \mathrm{prox}_h(\mu_k + t_k) \\ \mu_{k+1} &\leftarrow \mu_k + t_k - x_{k+1} \end{aligned}$$

- We used: $\mathrm{prox}_h(y - \nu_{k+1}) = \mu_{k+1} - y - \nu_{k+1} = x_{k+1}$

Proximal-Dykstra as CD

- Simplify, and use Lagrangian stationarity to obtain primal

$$x = y - \nu - \mu \implies y - \mu = x + \nu$$

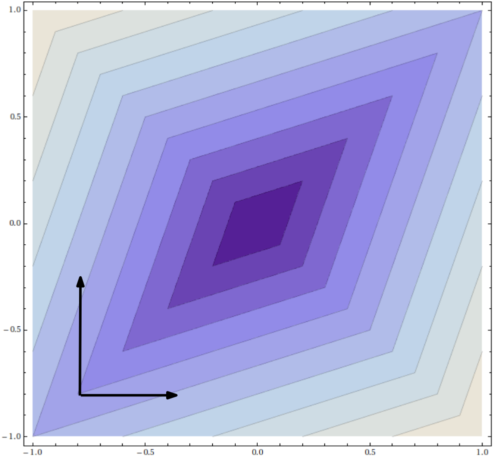
- Thus, the CD iteration may be rewritten as

$$\begin{aligned} t_k &\leftarrow \mathrm{prox}_f(x_k + \nu_k) \\ \nu_{k+1} &\leftarrow x_k + \nu_k - t_k \\ x_{k+1} &\leftarrow \mathrm{prox}_h(\mu_k + t_k) \\ \mu_{k+1} &\leftarrow \mu_k + t_k - x_{k+1} \end{aligned}$$

- We used: $\mathrm{prox}_h(y - \nu_{k+1}) = \mu_{k+1} - y - \nu_{k+1} = x_{k+1}$
- This is the proximal-Dykstra method!

CD for nonsmooth convex problems

$$\min |x_1 - x_2| + \frac{1}{2}|x_1 + x_2|$$



CD for separable nonsmoothness

- Nonsmooth part is **separable**

$$\boxed{\min_{x\in\mathbb{R}^n} \; f(x) + \sum_{i=1}^{n} r_i(x_i)}$$

CD for separable nonsmoothness

- Nonsmooth part is **separable**

$$\boxed{\min_{x \in \mathbb{R}^n} \; f(x) + \sum\nolimits_{i=1}^{n} r_i(x_i)}$$

Theorem If f is convex, continuously differentiable, each $g_i(x)$ is closed, convex, and each coordinate admits a **unique** solution. Further, assume we go through all coordinates in an essentially cyclic way. Then, the sequence $\{x^k\}$ generated by CD is bounded, and every limit point of it is optimal.

CD for separable nonsmoothness

- Nonsmooth part is **separable**

$$\boxed{\min_{x \in \mathbb{R}^n} \; f(x) + \sum\nolimits_{i=1}^{n} r_i(x_i)}$$

Theorem If f is convex, continuously differentiable, each $g_i(x)$ is closed, convex, and each coordinate admits a **unique** solution. Further, assume we go through all coordinates in an essentially cyclic way. Then, the sequence $\{x^k\}$ generated by CD is bounded, and every limit point of it is optimal.

Remark: A related result for **nonconvex** problems with separable non-smoothness (under more assumptions), can be found in: "*Convergence of Block Coordinate Descent Method for Nondifferentiable Minimization*" by P. Tseng (2001).

CD global convergence

- So far, we saw CD based on essentially cyclic rules

CD global convergence

- So far, we saw CD based on essentially cyclic rules
- It is difficult to prove **global** convergence and almost impossible to estimate **global** rate of convergence

CD global convergence

- So far, we saw CD based on essentially cyclic rules
- It is difficult to prove **global** convergence and almost impossible to estimate **global** rate of convergence
- Above results highlighted at best local (asymptotic) rates

CD global convergence

- So far, we saw CD based on essentially cyclic rules
- It is difficult to prove **global** convergence and almost impossible to estimate **global** rate of convergence
- Above results highlighted at best local (asymptotic) rates

- Consider the unconstrained problem $\min f(x)$, s.t., $x \in \mathbb{R}^n$

CD global convergence

- So far, we saw CD based on essentially cyclic rules
- It is difficult to prove **global** convergence and almost impossible to estimate **global** rate of convergence
- Above results highlighted at best local (asymptotic) rates

- Consider the unconstrained problem $\min f(x)$, s.t., $x \in \mathbb{R}^n$
- Assume f is convex, with **componentwise** Lipschitz gradients

$$|\nabla_i f(x + he_i) - \nabla_i f(x)| \leq L_i |h|, \quad x \in \mathbb{R}^n, h \in \mathbb{R}.$$

Here e_i denotes the ith canonical basis vector

CD global convergence

- So far, we saw CD based on essentially cyclic rules
- It is difficult to prove **global** convergence and almost impossible to estimate **global** rate of convergence
- Above results highlighted at best local (asymptotic) rates

- Consider the unconstrained problem $\min f(x)$, s.t., $x \in \mathbb{R}^n$
- Assume f is convex, with **componentwise** Lipschitz gradients

$$|\nabla_i f(x + he_i) - \nabla_i f(x)| \leq L_i |h|, \quad x \in \mathbb{R}^n, h \in \mathbb{R}.$$

Here e_i denotes the ith canonical basis vector

Choose $x_0 \in \mathbb{R}^n$. Let $M = \max_i L_i$; For $k \geq 0$

$$i_k = \mathrm{argmax}_{1 \leq i \leq n} |\nabla_i f(x_k)|$$
$$x_{k+1} = x_k - \tfrac{1}{M} \nabla_{i_k} f(x_k) e_{i_k}.$$

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!
- But this method is impractical

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!
- But this method is impractical
- At each step, it requires access to **full gradient**

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!
- But this method is impractical
- At each step, it requires access to **full gradient**
- With full gradient, might as well use ordinary gradient methods!

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!
- But this method is impractical
- At each step, it requires access to **full gradient**
- With full gradient, might as well use ordinary gradient methods!
- Also, if $f \in C_L^1$, it can easily happen that $M \geq L$

CD global convergence

Theorem Let $\{x^k\}$ be iterate sequence generated by above greedy CD method. Then,

$$f(x_k) - f^* \leq \frac{2nM\|x_0 - x^*\|_2^2}{k+4}, \quad k \geq 0.$$

- Looks like gradient-descent $O(1/k)$ bound for C_L^1 cvx
- Notice factor of n in the numerator!
- But this method is impractical
- At each step, it requires access to **full gradient**
- With full gradient, might as well use ordinary gradient methods!
- Also, if $f \in C_L^1$, it can easily happen that $M \geq L$
- So above rate is in general, worse than gradient methods

Randomized CD

NEXT LECTURE:

- Randomized BCD (aka Stochastic BCD)
- Parallel BCD
- Dual decomposition, ADMM, etc.