Algorithms III

Pontus Giselsson

Today's lecture

- coordinate descent
- coordinate gradient descent
- preconditioning
- envelope methods
- the error bound framework
- algorithm selection

Coordinate descent

- we want to minimize f which is proper closed and convex
- in coordinate descent, we optimize over one variable at a time
- consider

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

- algorithm is

$$\begin{aligned}
x_1^{k+1} &\in \operatorname*{argmin}_{x_1} f(x_1, x_2^k, x_3^k, \ldots, x_n^k) \\
x_2^{k+1} &\in \operatorname*{argmin}_{x_2} f(x_1^{k+1}, x_2, x_3^k, \ldots, x_n^k) \\
x_3^{k+1} &\in \operatorname*{argmin}_{x_3} f(x_1^{k+1}, x_2^{k+1}, x_3, \ldots, x_n^k) \\
&\vdots \\
x_n^{k+1} &\in \operatorname*{argmin}_{x_n} f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \ldots, x_n)
\end{aligned}$$

- also called coordinate minimization algorithm

Solves problem? – differentiable case

- assume f differentiable, and stationary point found
- are we guaranteed to have found solution?

Solves problem? – differentiable case

- assume f differentiable, and stationary point found
- are we guaranteed to have found solution?
- yes, $0 = \frac{\partial f}{\partial x_i}(x)$ for all i, that is

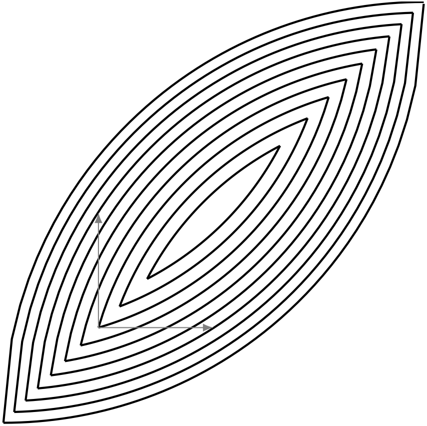
$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right) = 0$$

Solves problem? – nondifferentiable case

- what if f not differentiable?

Solves problem? – nondifferentiable case

- what if f not differentiable?
- no, consider e.g., $f(x,y) = |x-y| + \frac{1}{2}(\|x\|^2 + \|y\|^2)$



Solves problem? – separable case

- consider

$$\text{minimize } f(x) = g(x) + h(x)$$

- assume that g is convex and differentiable
- assume that $h(x) = \sum_{i=1}^{n} h_i(x_i)$ is PCC and non-differentiable
- will a stationary point of algorithm solve problem?

Solves problem? – separable case

- yes: let $\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_n^k)$
- first prove that x_i optimizes ith update if:

$$\langle \nabla_i g(\mathbf{x}_i^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i) \geq 0, \qquad \forall y_i \in \mathbb{R}$$

- proof: let
$\mathbf{y}_i^k = \mathbf{x}_i^k + e_i(y_i - x_i) = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y_i, x_{i+1}^k, \ldots, x_n^k)$, then

$$g(\mathbf{y}_i^k) - g(\mathbf{x}_i^k) \geq \langle \nabla g(\mathbf{x}_i^k), \mathbf{y}_i^k - \mathbf{x}_i^k \rangle = \langle \nabla g_i(\mathbf{x}_i^k), y_i - x_i \rangle$$

- therefore, if condition holds, we have for all $\mathbf{y}_i^k$

$$\begin{aligned} f(\mathbf{y}_i^k) - f(\mathbf{x}_i^k) &= g(\mathbf{y}_i^k) - g(\mathbf{x}_i^k) + h_i(y_i) - h_i(x_i) \\ &\geq \langle \nabla_i g(\mathbf{x}_i^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i) \geq 0 \end{aligned}$$

- that is, $f(\mathbf{x}_i^k)$ has lowest value along ith coordinate

Solves problem? – separable case

- assume that $\mathbf{x}_1^k = \mathbf{x}_j^k$ for all $j = 2, \ldots, n$
- that is, assume that we have reached a stationary point
- then, for any y and all $\mathbf{x}_j^k$ (since they are equal), we have

$$
\begin{aligned}
f(y) - f(\mathbf{x}_j) &= g(y) - g(\mathbf{x}_j^k) + \sum_{i=1}^{n} [h_i(y_i) - h_i(x_i)] \\
&\geq \langle \nabla g(\mathbf{x}_j^k), y - \mathbf{x}_j^k \rangle + \sum_{i=1}^{n} [h_i(y_i) - h_i(x_i)] \\
&= \sum_{i=1}^{n} \underbrace{\langle \nabla_i g(\mathbf{x}_j^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i)}_{\geq 0} \geq 0
\end{aligned}
$$

- that is, $\mathbf{x}_j$ optimizes f

How about convergence?

- strong convergence guarantees are somewhat scarce
- we know that function value is nonincreasing, i.e.,

$$f(x_{i+1}^{k+1}) \leq f(x_i^{k+1})$$

  assume level set $\{x \mid f(x) \leq f(x^0)\}$ closed bounded

  $\Rightarrow$ subsequence converges
- 2-block coordinate descent:
  - sublinear convergence for smooth functions
  - linear convergence under additional strong convexity
- linear convergence under error bound property for $F = f + g$ with
  - f smooth with component-wise strong convexity
  - $g = \iota_C$ with $C = I_1 \times \cdots \times I_n$ and I_i being intervals

Comments

- order of updates can be changed
- methods where coordinate to update is chosen on random exist
- can update on groups of coordinates instead of individual

Coordinate gradient descent

- in (proximal) coordinate gradient descent, we solve problem

$$\text{minimize } f(x) + g(x)$$

- assume $g(x) = \sum_{i=1}^{p} g_i(x_i)$ block-separable and convex
- assume f block-smooth: let

$$\begin{aligned}\mathbf{x}_i &= (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_p) \\ \mathbf{y}_i &= (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_p)\end{aligned}$$

- then f is block-smooth if it satisfies

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{y}_i - \mathbf{x}_i \rangle + \tfrac{L_i}{2} \|\mathbf{y}_i - \mathbf{x}_i\|^2$$

  for some L_i, all $\mathbf{x}_i, \mathbf{y}_i$ and $i = \{1, \ldots, p\}$
- equivalent condition

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla_i f(\mathbf{x}_i), y_i - x_i \rangle + \tfrac{L_i}{2} \|y_i - x_i\|^2$$

Coordinate gradient descent

- the algorithm looks like (assuming cyclic updates in 2,3)
  1. choose block-coordinate $i_k = i \in \{1, 2, \ldots, n\}$ to update
  2. let $\mathbf{x}_i = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_p^k)$
  3. let $\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_p^k)$
  4. compute

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\{f(\mathbf{x}_i^k) + \langle \nabla f(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \tfrac{L_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 + g(\mathbf{x}_i)\}$$

- f is approximated by block-smoothness upper bound

Simplification of main step

- recall

$$\mathbf{x}_i = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_p^k)$$
$$\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_p^k$$

- the main step can be written as

$$\begin{aligned}
x_i^{k+1} &= \operatorname*{argmin}_{x_i} \{ f(\mathbf{x}_i^k) + \langle \nabla f(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \tfrac{L_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|^2 + g(\mathbf{x}_i) \} \\
&= \operatorname*{argmin}_{x_i} \{ \langle \nabla_i f(\mathbf{x}_i^k), x_i - x_i^k \rangle + \tfrac{L_i}{2} \|x_i - x_i^k\|^2 + g_i(x_i) \} \\
&= \operatorname*{argmin}_{x_i} \{ \tfrac{L_i}{2} \|x_i - x_i^k + \tfrac{1}{L_i} \nabla_i f(\mathbf{x}_i^k)\|^2 + g_i(x_i) \} \\
&= \mathsf{prox}_{\frac{1}{L_i} g_i}(x_i^k - \tfrac{1}{L_i} \nabla_i f(\mathbf{x}_i^k))
\end{aligned}$$

- we take coordinate-wise forward-backward steps

Convergence

- convergence if block-coordinate to update is randomly chosen
- can use probabilities proportional to L_i^γ for $\gamma \in [0, 1]$
  - $\gamma = 0$ implies uniform distribution
  - $\gamma = 1$ implies coordinates with high L_i are chosen more often
- the convergence is in expectation or "with high probability"
- some results on deterministic schemes also exist:
  - cyclic order with $g \equiv 0$ (smooth case)
  - cyclic order with $g = \iota_C$ with $C = C_1 \times \cdots \times C_p$
- if p updates (one sweep) same cost as one gradient computation

  $\Rightarrow$ coordinate gradient descent faster than gradient descent

Smoothness with different metric

- let's use a different metric in f block-smoothness
- let $W = \text{diag}(W_1, \ldots, W_p)$ with $W_i \succ 0$ and

$$\mathbf{x}_i = (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_p)$$
$$\mathbf{y}_i = (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_p)$$

- f is block-smooth with metric W if it satisfies

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{y}_i - \mathbf{x}_i \rangle + \tfrac{1}{2} \|\mathbf{y}_i - \mathbf{x}_i\|_W^2$$

  for all $\mathbf{x}_i, \mathbf{y}_i$ and $i = \{1, \ldots, p\}$
- equivalent to the above:

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla_i f(\mathbf{x}_i), y_i - x_i \rangle + \tfrac{1}{2} \|y_i - x_i\|_{W_i}^2$$

- previous block-smoothness obtained by $W = \text{diag}(L_1 I, \ldots, L_p I)$

Generalized method with different metric

- the algorithm looks like (assuming cyclic updates in 2,3)
    1. choose block-coordinate $i_k = i \in \{1, 2, \ldots, n\}$ to update
    2. let $\mathbf{x}_i = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_n^k)$
    3. let $\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_n^k)$
    4. compute

$$x_i^{k+1} = \operatorname*{argmin}_{x_i}\{f(\mathbf{x}_i^k) + \langle \nabla f(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \tfrac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_W^2 + g(\mathbf{x}_i)\}$$

- f again approximated by block-smoothness upper bound

Simplification of main step

- recall

$$\mathbf{x}_i = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \dots, x_p^k)$$
$$\mathbf{x}_i^k = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \dots, x_p^k$$

- the main step can be written as

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}}\{f(\mathbf{x}_i^k) + \langle \nabla f(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \tfrac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_W^2 + g(\mathbf{x}_i)\} \\ &= \underset{x_i}{\operatorname{argmin}}\{\langle \nabla_i f(\mathbf{x}_i^k), x_i - x_i^k \rangle + \tfrac{1}{2}\|x_i - x_i^k\|_{W_i}^2 + g_i(x_i)\} \\ &= \underset{x_i}{\operatorname{argmin}}\{\tfrac{1}{2}\|x_i - x_i^k + W_i^{-1}\nabla_i f(\mathbf{x}_i^k)\|_{W_i}^2 + g_i(x_i)\} \end{aligned}$$

- we take skewed coordinate-wise forward-backward steps
- (can use skewed metric also in normal forward-backward splitting)

Quadratic case

- consider the quadratic case

$$\text{minimize } f(x) + g(x)$$

- the function $f(x) = \frac{1}{2}x^T Hx + q^T x$ where

$$H = \begin{bmatrix} H_{11} & \cdots & H_{1p} \\ \vdots & \ddots & \vdots \\ H_{p1} & \cdots & H_{pp} \end{bmatrix}, \qquad q = \begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}$$

- assume that H positive semi-definite and H_{ii} positive definite
- assume g is block-separable (as before)

Assumption on H

- the assumption that H_{ii} positive definite not very conservative
- consider the case with blocks of size 1
- requirement is that all diagonal elements of H are positive
- consider a rank 1 matrix $H = hh^T$
  - $H_{ii} = 0$ if and only if row and column i all zeros
- to require H to be positive definite, it must have full rank

Algorithm

- apply generalized coordinate gradient descent with $W_i = H_{ii}$
- the algorithm becomes (assuming cyclic updates in 2,3)
  1. choose block-coordinate $i_k = i \in \{1, 2, \ldots, n\}$ to update
  2. let $\mathbf{x}_i = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_n^k)$
  3. let $\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_n^k)$
  4. compute

$$x_i^{k+1} = \operatorname*{argmin}_{x_i} \{ \tfrac{1}{2} \| x_i - x_i^k + H_{ii}^{-1} \nabla_i f(\mathbf{x}_i^k) \|_{H_{ii}}^2 + g_i(x_i) \}$$

Simplification of main step

- recall that $f(x) = \frac{1}{2}x^T Hx + q^T x$,

$$\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_n^k)$$

and let

$$U^T = [0, \ldots, 0, I, 0, \ldots, 0]$$

- then

$$\nabla_i f(\mathbf{x}_i^k) = U^T(H\mathbf{x}_i^k + q) = H_{ii}x_i^k + \sum_{j<i} H_{ij}x_j^{k+1} + \sum_{j>i} H_{ij}x^k + q_i$$

- inserting this into main step gives:

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}}\{\tfrac{1}{2}\|x_i - x_i^k + H_{ii}^{-1}\nabla_i f(\mathbf{x}_i^k)\|_{H_{ii}}^2 + g_i(x_i)\} \\ &= \underset{x_i}{\operatorname{argmin}}\{\tfrac{1}{2}\|x_i + H_{ii}^{-1}(\sum_{j<i} H_{ij}x_j^{k+1} + \sum_{j>i} H_{ij}x_j^k + q_i)\|_{H_{ii}}^2 + g_i(x_i)\} \\ &= \underset{x_i}{\operatorname{argmin}}\{\tfrac{1}{2}x_i^T H_{ii}x_i + x_i^T(\sum_{j<i} H_{ij}x_j^{k+1} + \sum_{j>i} H_{ij}x_j^k + q_i) + g_i(x_i)\} \end{aligned}$$

Compare to coordinate descent step

- let $F = f + g$
- the ith update in the coordinate descent method is

$$x_i^{k+1} = \operatorname*{argmin}_{x_i} F(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_p^k)$$

- since g block-separable, this can be written as

$$\begin{aligned} x_i^{k+1} &= \operatorname*{argmin}_{x_i} \{ f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_p^k) + g_i(x_i) \} \\ &= \operatorname*{argmin}_{x_i} \{ \tfrac{1}{2} x_i^T H_{ii} x_i + x_i^T (\sum_{j<i} H_{ij} x_j^{k+1} + \sum_{j>i} H_{ij} x_j^k) + q_i^T x_i + g_i(x_i) \} \end{aligned}$$

- last step holds since

$$f(x) = \tfrac{1}{2} x^T H x + q^T x = \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p x_i^T H_{ij} x_j + \sum_{i=1}^p q_i^T x_i$$

Algorithms identical

- in this setting iterates are identical
- coordinate descent special case of coordinate gradient descent
- to my knowledge, first time this link has been provided
- why are iterates identical? since skewed block-smoothness

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{y}_i - \mathbf{x}_i \rangle + \tfrac{1}{2} \|\mathbf{y}_i - \mathbf{x}_i\|_W^2$$

holds with equality for $W = \text{diag}(H_{11}, \ldots, H_{pp})$ and

$$\begin{aligned} \mathbf{x}_i &= (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_p) \\ \mathbf{y}_i &= (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_p) \end{aligned}$$

Implications

- convergence for coordinate gradient descent more mature
- can use this to get new convergence *rates* of coordinate descent

Preconditioning

- consider solving the problem using a first-order method

$$\text{minimize } f(x) + g(x)$$

- change of variables $x = Tq$ (T invertible) gives problem

$$\text{minimize } f(Tq) + g(Tq) =: f_T(q) + g_T(q)$$

- optimal $x^\star$ to original problem is $x^\star = Tq^\star$
- with appropriate T, performance may be significantly improved
- (Newton's method is invariant to change of variables)

Preconditioning in FB splitting

- solve $\min_q\{f_T(q)+g_T(q)\}$ using forward-backward splitting:

$$
\begin{aligned}
q^{k+1} &= \mathsf{prox}_{\gamma g_T}(\mathrm{Id}-\gamma\nabla f_T)q^k \\
&= \operatorname*{argmin}_q\{g_T(q)+\tfrac{1}{2\gamma}\|q-q^k+\gamma\nabla f_T q^k\|^2\} \\
&= \operatorname*{argmin}_q\{g(Tq)+\tfrac{1}{2\gamma}\|q-q^k+\gamma T^T\nabla f(Tq^k)\|^2\} \\
&= \operatorname*{argmin}_q\{g(Tq)+\langle T^T\nabla f(Tq^k),q-q^k\rangle+\tfrac{1}{2\gamma}\|q-q^k\|^2\} \\
&= \operatorname*{argmin}_q\{g(Tq)+\langle \nabla f(Tq^k),Tq-Tq^k\rangle+\tfrac{1}{2\gamma}\|q-q^k\|^2\} \\
&= T^{-1}\operatorname*{argmin}_x\{g(x)+\langle \nabla f(x^k),x-x^k\rangle+\tfrac{1}{2\gamma}\|x-x^k\|^2_{T^{-2}}\}
\end{aligned}
$$

- we assumed that $Tq^k=x^k$, therefore, iteration is equivalent to

$$
x^{k+1}=\operatorname*{argmin}_x\{g(x)+f(x^k)+\langle \nabla f(x^k),x-x^k\rangle+\tfrac{1}{2\gamma}\|x-x^k\|^2_{T^{-2}}\}
$$

- standard FB splitting obtained by letting $T=I$
- that is, we have a different quadratic approximation to f
- if approximation better, probably faster convergence!

Linear convergence

- look at problem with properties that guarantee linear convergence
- if f is σ-strongly convex and β-smooth then
  - DR converges as $\frac{\sqrt{\beta/\sigma}-1}{\sqrt{\beta/\sigma}+1}$ if optimal parameters used
  - FB converges as $\frac{\beta/\sigma-1}{\beta/\sigma+1}$ if optimal parameters used
- both rates get better with decreasing β/σ
- most first-order methods perform better with decreasing β/σ

  $\Rightarrow$ choose preconditioner T such that β/σ decreased

Example

- consider the problem with $f(x) = \frac{1}{2}x^T Hx + h^T x$ where $H \succ 0$
- then f is $\lambda_{\max}(H)$-smooth and $\lambda_{\min}(H)$-strongly convex
- that is $\beta(f)/\sigma(f) = \lambda_{\max}(H)/\lambda_{\min}(H) =: \kappa(H)$
- what is $\beta(f_T)$ and $\sigma(f_T)$?
- we have $f_T(q) = f(Tq) = \frac{1}{2}q^T THTq + h^T Tx$
- therefore $\beta(f_T) = \lambda_{\max}(T^T HT)$ and $\sigma(f_T) = \lambda_{\min}(T^T HT)$

Example

- we have $\beta(f_T) = \lambda_{\max}(T^T H T)$ and $\sigma(f_T) = \lambda_{\min}(T^T H T)$
- so the rates depend on $\lambda_{\max}(T^T H T)/\lambda_{\min}(T^T H T)$
- we want to choose T such that this ratio is minimized
- by letting $T = H^{-1/2}$, we get $\lambda_{\max}(T^T H T)/\lambda_{\min}(T^T H T) = 1$
- are there any drawbacks with this choice?

Full variable transformations

- consider, e.g., forward backward splitting on

$$\text{minimize } f_T(q) + g_T(q)$$

  with $f_T(q) = \frac{1}{2} q^T T^T H T q + h^T T q$
- this is given by

$$q^{k+1} = \text{prox}_{\gamma g_T}(\mathrm{Id} - \gamma \nabla f_T) q^k = \text{prox}_{\gamma g_T}(\mathrm{Id} - \gamma (T^T H T q^k + T^T h)) q^k$$

- forward step $\mathrm{Id} - \gamma(T^T H T q^k + T^T h)$ a bit more expensive
- backward step

$$\begin{aligned} \text{prox}_{\gamma g_T}(z) &= \underset{q}{\operatorname{argmin}} \{ g_T(q) + \tfrac{1}{2\gamma} \|q - z\|^2 \} \\ &= \underset{q}{\operatorname{argmin}} \{ g(Tq) + \tfrac{1}{2\gamma} \|q - z\|^2 \} \end{aligned}$$

  might be much more expensive (if g separable)
- however, if T diagonal $\Rightarrow \approx$ unchanged computational cost!

Computing T

- in the quadratic case, we want to solve

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(T^T H T)/\lambda_{\min}(T^T H T) \\ \text{subject to} & T \text{ diagonal} \end{array}$$

- this can be posed as (convex) semi-definite program

  $\Rightarrow$ optimal diagonal preconditioning can be achieved
- drawback: computationally very expensive to find optimal T
  also limited to fairly small-scale problems
- works in applications (MPC) where T can be computed offline

Heuristic methods to compute T

- the objective is to find diagonal T such that ratio decreased

$$\lambda_{\max}(T^T HT)/\lambda_{\min}(T^T HT)$$

- finding T should be much faster than solving $\min_x\{f(x) + g(x)\}$
- cheap heuristics: 1-norm, 2-norm, and ∞-norm equilibration

Equilibration

- let $s = 1$, $s = 2$ or $s = \infty$
- in symmetric s-norm equilibration, the objective is to find T s.t.:

$$\|[T^T HT]_{i\cdot}\| = 1 \qquad \text{for all } i$$

- we want the resulting matrix to have equal s-norm in every row
- since matrix symmetric, also all columns have same s-norm

∞-norm equilibration

- in ∞-norm equilibration ∞-norm of each row should be the same
- we have $\|x\|_\infty = \max_i |x_i|$
- therefore set the absolute value of largest element in each row to 1
- this is obtained by letting $T_{ii} = 1/\sqrt{H_{ii}}$
- this is also called Jacobi scaling

1-norm equilibration

- recall $\|x\|_1 = \sum_i |x_i|$ and that T diagonal
- let $\bar{H} = \text{abs}(H)$ and $t = \text{diag}(T)$
- assume $T > 0$, then 1-norm of row i is given by

$$\|[T^T HT]_i\|_1 = \sum_{j=1}^{n} |T_{ii} H_{ij} T_{jj}| = T_{ii} \sum_{j=1}^{n} \bar{H}_{ij} T_{jj} = T_{ii} \bar{H}_{i,\cdot} t$$

- therefore, 1-norm equilibration is obtained by solving

$$T\bar{H}t = \mathbf{1} \quad \Leftrightarrow \quad \bar{H}t - T^{-1}\mathbf{1} = 0$$

- l.h.s. is gradient of function (recall $t > 0$)

$$\tfrac{1}{2} t^T \bar{H} t - \log(t)$$

- change of variables $t \to e^t$ makes it convex
- solved readily by coordinate descent or Sinkhorn-Knopp algorithm

2-norm equilibration

- in 2-norm equilibration we want all rows to have equal 2-norm
- the squared 2-norm of row i is given by

$$\|[T^T H T]_i\|_2^2 = \sum_{j=1}^{n} (T_{ii} H_{ij} T_{jj})^2 = T_{ii}^2 \sum_{j=1}^{n} \bar{H}_{ij}^2 T_{jj}^2 = T_{ii}^2 \bar{H}_{i,\cdot}^2 t^2$$

  where square on vectors are element-wise
- let $S = T^2$, and $\hat{H} = \bar{H}^2$ (element-wise)
- then problem is same as in 1-norm case

$$S\hat{H}s = \mathbf{1} \quad \Leftrightarrow \quad \hat{H}s - S^{-1}\mathbf{1} = 0$$

- solve using same techniques

If not linear convergence?

- consider again a quadratic problem

$$\text{minimize } f(x) + g(x)$$

  with $f(x) = \frac{1}{2}x^T Hx + h^T x$ and g separable
- if H not positive definite, we do not get linear convergence
- we can use heuristic to choose T by optimizing

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(T^T HT)/\lambda_{\min>0}(T^T HT) \\ \text{subject to} & T \text{ diagonal} \end{array}$$

  where $\lambda_{\min>0}$ is smallest non-zero eigenvalue
- can be posed a (convex) semi-definite program
- can use equilibration heuristics to approximate solution

Dual algorithms

- consider the problem

$$\text{minimize } f(x) + g(Lx)$$

- such problems are often solved via the dual

$$\text{minimize } f^*(-L^*\mu) + g^*(\mu) =: d(\mu) + g^*(\mu)$$

- linear convergence rate of DR and FB depends on $\beta(d)/\sigma(d)$

Precondition dual problem

- we perform a linear change of variables for dual problem $\mu = T\nu$

$$\text{minimize } f(-L^*\nu) + g^*(T\nu) = d_T(\mu) + g_T^*(\nu)$$

- this is dual problem to

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & TLx = Ty \end{array}$$

- or to the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g_{T^{-1}}(z) \\ \text{subject to} & TLx = z \end{array}$$

where $g_{T^{-1}}(z) = g(T^{-1}z)$

Quadratic problems

- consider again a quadratic problem

$$\text{minimize } f(x) + g(Lx)$$

  with $f(x) = \frac{1}{2}x^T Hx + h^T x$, H positive definite, and g separable

- the conjugate of f is

$$f^*(\lambda) = \frac{1}{2}(\lambda - h)^T H^{-1}(\lambda - h)$$

- therefore

$$d(\mu) = f^*(-L^T\mu) = \frac{1}{2}(L^T\mu + h)^T H^{-1}(L^T\mu + h)$$

  i.e., a quadratic with Hessian $LH^{-1}L^T$

- linear convergence of many algorithms if $\lambda_{\min}(LH^{-1}L^T) > 0$

Selecting preconditioner

- we select preconditioner to minimize condition number of Hessian

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(T^T L H^{-1} L^T T)/\lambda_{\min}(T^T L H^{-1} L^T T) \\ \text{subject to} & T \text{ diagonal} \end{array}$$

- again, we select T diagonal, if not, iteration complexity increased

If not linear convergence?

- consider problems of the form

$$\text{minimize } f_1(x) + f_2(x) + g(Lx)$$

- let $f = f_1 + f_2$ and assume $f_1(x) = \frac{1}{2}x^T Hx + h^T x$
- assume f_2 does not have curvature (indicator function, piece-wise linear)
- precondition as if $f_2 \equiv 0$
- then the dual to precondition is

$$\begin{aligned} d_1(\mu) &= (f_1^* \circ -L^*)(\mu) \\ &= \begin{cases} \frac{1}{2}(L^T\mu + h)^T H^\dagger (L^T\mu + h) & \text{if } (L^T\mu + q) \in \mathcal{R}(H) \\ \infty & \text{else} \end{cases} \end{aligned}$$

where $H^\dagger$ is the pseudo-inverse

Heuristic preconditioning

- the dual to be preconditioned is

$$
\begin{aligned}
d_1(\mu) &= (f_1^* \circ -L^*)(\mu) \\
&= \begin{cases} \frac{1}{2}(L^T\mu + h)^T H^\dagger (L^T\mu + h) & \text{if } (L^T\mu + q) \in \mathcal{R}(H) \\ \infty & \text{else} \end{cases}
\end{aligned}
$$

- precondition by change of variables to get $d_T = (d \circ T)$
- make d_T as well conditioned as possible, i.e., select T as:

$$
\begin{array}{ll}
\text{minimize} & \lambda_{\max}(TLH^\dagger L^T T)/\lambda_{\min>0}(TLH^\dagger L^T T) \\
\text{subject to} & T \text{ diagonal}
\end{array}
$$

- reduces to linearly convergent preconditioner if H invertible
- can be posed as (convex) semi-definite program
- can use equilibration methods to find "good enough" T

Envelope methods

- we have considered methods for nonsmooth problems
- what if we can formulate equivalent smooth problem?
- here equivalent means the have same set of solutions
- then could use smooth optimization to solve nonsmooth problems

Moreau envelope

- recall the Moreau envelope

$$ {}^{\gamma}f(z) = \min_{x}\{f(x) + \tfrac{1}{2\gamma}\|x - z\|^2\} $$

- the gradient of ${}^{\gamma}f$ is

$$ \begin{aligned} \nabla {}^{\gamma}f(z) &= \gamma^{-1}(\mathrm{Id} - \operatorname*{argmin}_{x}\{f(x) + \tfrac{1}{2\gamma}\|x - z\|^2\}) \\ &= \gamma^{-1}(\mathrm{Id} - \mathsf{prox}_{\gamma f})z \end{aligned} $$

- the gradient is γ^{-1}-Lipschitz continuous
  ($\mathsf{prox}_{\gamma f}$ firmly nonexpansive $\Leftrightarrow$ $\mathrm{Id} - \mathsf{prox}_{\gamma f}$ firmly nonexpansive)

Gradient method

- the gradient method with $t = \gamma$ to minimize the Moreau envelope:

$$\begin{aligned} z^{k+1} &= z^k - t\nabla^\gamma f(z^k) \\ &= z^k - t\gamma^{-1}(z^k - \mathsf{prox}_{\gamma f}(z^k)) \\ &= \mathsf{prox}_{\gamma f}(z^k) \end{aligned}$$

- it is the proximal point algorithm on f
- minimize a nonsmooth f by gradient method on smooth function
- can use any method for smooth optimization to solve problem

Forward-backward splitting for quadratic problem

- assume that $f(x) = \frac{1}{2}x^T Hx + h^T x$ with H positive semi-definite
- assume also that g is proper closed and convex
- we want to solve

$$\text{minimize } f(x) + g(x)$$

- forward-backward splitting applied to this problem is

$$x^{k+1} = \text{prox}_{\gamma g}(I - \gamma\nabla f)x^k = \text{prox}_{\gamma g}((I - \gamma H)x^k - \gamma h)$$

- let $L_\gamma = (I - \gamma H)$, then FB algorithm can be written as

$$x^{k+1} = \text{prox}_{\gamma g}(L_\gamma x^k - \gamma h)$$

- further introduce $h_\gamma = \gamma g + \frac{1}{2}\|\cdot\|^2$, then $\text{prox}_{\gamma g} = \nabla h_\gamma^*$, i.e.,:

$$x^{k+1} = \nabla h_\gamma^*(L_\gamma x^k - \gamma h)$$

Forward-backward envelope

- assume γ such that $L_\gamma = (I - \gamma H)$ invertible
- consider the function, called the *forward-backward envelope*

$$F_\gamma^{\mathrm{FB}}(x) = \tfrac{1}{2}\|x\|_{L_\gamma}^2 - (h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h)$$

- the gradient of F_γ is given by (since $L_\gamma = L_\gamma^*$)

$$\nabla F_\gamma^{\mathrm{FB}}(x) = L_\gamma x - L_\gamma \nabla h_\gamma^*(L_\gamma x - \gamma h)$$

- consider the skewed gradient method on F_γ^{FB}:

$$\begin{aligned} x^{k+1} &= x^k - L_\gamma^{-1}\nabla F_\gamma^{\mathrm{FB}}(x^k) \\ &= x^k - L_\gamma^{-1}(L_\gamma x^k - L_\gamma \nabla h_\gamma^*(L_\gamma x^k - \gamma h)) \\ &= x^k - x^k + \nabla h_\gamma^*(L_\gamma x^k - \gamma h) \\ &= \nabla h_\gamma^*(L_\gamma x^k - \gamma h) \\ &= \mathrm{prox}_{\gamma g}((I - \gamma H)x^k - \gamma h) \\ &= \mathrm{prox}_{\gamma g}(I - \gamma \nabla f)x^k \end{aligned}$$

- it is the proximal gradient method

FB envelope stationary points

- stationary points of FB envelope satisfy

$$0 = \nabla F_\gamma^{\mathrm{FB}}(x) = L_\gamma x - L_\gamma \nabla h_\gamma^*(L_\gamma x - \gamma h)$$

- since L_γ assumed invertible, this is equivalent to

$$x = \nabla h_\gamma^*(L_\gamma x - \gamma h) = \mathsf{prox}_{\gamma g}(I - \gamma \nabla f)x$$

- set of critical points of envelope agrees with minimizers of $f + g$

FB envelope convexity

- let $\gamma \in (0, \frac{1}{\beta}) = (0, \frac{1}{\lambda_{\max}(H)})$
- then $L_\gamma = (I - \gamma H)$ is invertible and F_γ^{FB} is convex
- proof:
  F_γ^{FB} is convex $\Leftrightarrow$ $(h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h)$ is 1-smooth w.r.t. $\|\cdot\|_{L_\gamma}$
- we know h_γ is 1-strongly convex for all γ
- therefore h_γ^* is 1-smooth, i.e.,

$$
\begin{aligned}
&(h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h) = h_\gamma^*(L_\gamma y - \gamma h) \\
&\leq h_\gamma^*(L_\gamma x - \gamma h) + \langle \nabla h_\gamma^*(L_\gamma x - \gamma h), L_\gamma y - L_\gamma x\rangle + \tfrac{1}{2}\|L_\gamma(x-y)\|^2 \\
&= (h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h) + \langle L_\gamma \nabla h_\gamma^*(L_\gamma x - \gamma h), y - x\rangle + \tfrac{1}{2}\|L_\gamma(x-y)\|^2 \\
&= (h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h) + \langle \nabla(h_\gamma^* \circ L)(x - L_\gamma^{-1}\gamma h), y - x\rangle + \tfrac{1}{2}\|x-y\|_{L_\gamma^2}^2
\end{aligned}
$$

- that is $(h_\gamma^* \circ L_\gamma)(x - L_\gamma^{-1}\gamma h)$ is 1-smooth w.r.t. $\|\cdot\|_{L_\gamma^2}$

FB envelope convexity

- recall $L_\gamma = (I - \gamma H)$
- let $L_\gamma = U \Sigma U^T$, where Σ diagonal with singular values
- since $\gamma \in (0, \frac{1}{\lambda_{\max}(H)})$, then $0 \prec L_\gamma \prec I$ and $\sigma_i \in (0, 1)$
- therefore

$$
\begin{aligned}
x^T L_\gamma^2 x = x^T U \Sigma^2 U^T x = v^T \Sigma^2 v = \sum_{i=1}^n \sigma_i^2 v_i^2 \leq \sum_{i=1}^n \sigma_i v_i^2 \\
\leq x^T L_\gamma x
\end{aligned}
$$

- that is, for $\gamma \in (0, \frac{1}{\lambda_{\max}(H)})$ we have $\|x - y\|_{L_\gamma^2} \leq \|x - y\|_{L_\gamma}$
- therefore our function is 1-smooth w.r.t. the L_γ-norm
- hence F_γ^{FB} is convex (and also 1-smooth w.r.t. $\|\cdot\|_{L_\gamma}$)

Consequence

- for quadratic f, the FB method with $\gamma \in (0, \frac{1}{\beta})$ is gradient method to envelope
- we have shown that stationary points coincide with optimizers to $\min_x\{f(x) + g(x)\}$
- we have shown that envelope convex in this case
- then stationary points are minimizers of envelope
- i.e., equivalent to minimize smooth envelope and to minimize composite problem
- can use any smooth method to solve problem

FB envelope and DR envelope

- similar envelope function can be created for DR splitting
- envelope function properties:
  - take gradient step on envelope function to get back algorithm
  - stationary points to envelopes coincide with fixed-points to operators
- in quadratic case, envelopes convex
- then, can solve nonsmooth problems using smooth methods
- can, e.g, incorporate second order information (quasi-Newton)

  $\Rightarrow$ might improve (asymptotic) convergence
- caveat: envelope often not twice continuously differentiable (however, twice continuously differentiable almost everywhere)

The error bound property

- assume that $T : \mathbb{R}^n \to \mathbb{R}^n$ is α-averaged
- that is, assume that $T = (1-\alpha)\mathrm{Id} + \alpha R$ for nonexpansive R
- assume that for all $x \in \mathbb{R}^n$, the following holds

$$\mathrm{dist}_{\mathsf{fix}R}(x) \leq \tau \|x - Rx\|$$

  for some $\tau \in (0, \infty)$
- iterate the operator as $x^{k+1} = Tx^k$
- then we get linear convergence in distance to fixed-point set

Linear convergence, proof

- proof: an α-averaged operator satisfies

$$\|Tx - Ty\|^2 \leq \|x - y\|^2 - \tfrac{1-\alpha}{\alpha}\|(\mathrm{Id} - T)x - (\mathrm{Id} - T)y)\|^2$$

- recall the error bound property

$$\begin{aligned}\mathrm{dist}_{\mathsf{fix}R}(x) &\leq \tau\|x - Rx\| \\ &= \tau\|x - (1-\alpha^{-1})x - \alpha^{-1}Tx\| = \tau\alpha^{-1}\|x - Tx\|\end{aligned}$$

- let $x = x^k$ and $y = x^\star$ where $x^\star \in \mathsf{fix}T$ is closest point to x^k

$$\begin{aligned}\mathrm{dist}^2_{\mathsf{fix}T}(x^{k+1}) &\leq \|x^{k+1} - x^\star\|^2 \\ &\leq \|x^k - x^\star\|^2 - \tfrac{1-\alpha}{\alpha}\|x^k - Tx^k\|^2 \\ &\leq \mathrm{dist}^2_{\mathsf{fix}T}(x^k) - \tfrac{\alpha(1-\alpha)}{\tau^2}\mathrm{dist}^2_{\mathsf{fix}T}(x^k) \\ &= (1 - \tfrac{\alpha(1-\alpha)}{\tau^2})\mathrm{dist}^2_{\mathsf{fix}T}(x^k)\end{aligned}$$

(recall $Tx^\star = x^\star$ and $Tx^k = x^{k+1}$)

- that is, linear convergence with rate $\sqrt{1 - \frac{\alpha(1-\alpha)}{\tau^2}}$

Questions

- what problems and algorithms satisfy error bound property?
- can we quantify τ for those
- can it be used to show linear convergence for
  - coordinate descent methods with operators (with cyclic updates)?
  - FB splitting method with less restrictive assumptions?
  - DR splitting method with less restrictive assumptions?

Problem splitting

- to choose splitting is to formulate optimization problem on form

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Lx = y \end{array}$$

- main splitting rule:

  "choose f, g, and L to get as cheap iterates as possible"
- if, e.g, g separable, we would like to exploit this in algorithm
- if many iterations, try different splitting or different algorithm

Algorithm selection for large-scale problems

- consider the following list of algorithms
  - coordinate gradient descent
  - coordinate descent
  - (stochastic) subgradient method
  - forward-backward splitting (and accelerated variants)
  - linearized ADMM
  - Douglas-Rachford splitting
  - ADMM
  - three-splitting method
  - envelope methods with second order information
  - active set methods (sometimes applicable, not covered here)
  - interior-point methods (not covered here)
- iteration complexity grows downwards in the list
- typically, number of iterations grows upwards in the list
  $\Rightarrow$ trade-off
- for large-scale problems
  - start with (feasible) method with cheapest iteration cost
  - if too many iterations, then traverse down the list