Advanced Optimization

(10-801: CMU)

Lecture 19

Parallel proximal; Incremental gradient

26 Mar, 2014

Suvrit Sra

Douglas-Rachford

$$\boxed{\min \quad f(x) + h(x)}$$

$$z \leftarrow \frac{1}{2}(I + R_f R_h)z$$

Douglas-Rachford

$$\min \quad f(x) + h(x)$$

$$z \leftarrow \tfrac{1}{2}(I + R_f R_h)z$$

Reflection operator

$$R_f := 2\operatorname{prox}_f - I.$$

Douglas-Rachford

$$\min \quad f(x) + h(x)$$

$$z \leftarrow \tfrac{1}{2}(I + R_f R_h) z$$

Reflection operator

$$R_f := 2\operatorname{prox}_f - I.$$

Observe: $R_f = -R_{f^*}$ (another justification of "reflection")

Douglas-Rachford

$$\min \quad f(x) + h(x)$$

$$z \leftarrow \tfrac{1}{2}(I + R_f R_h)z$$

Reflection operator

$$R_f := 2\operatorname{prox}_f - I.$$

Observe: $R_f = -R_{f^*}$ (another justification of "reflection")

$$\operatorname{prox}_f + \operatorname{prox}_{f^*} \quad = \quad I$$

Douglas-Rachford

$$\min \quad f(x) + h(x)$$

$$z \leftarrow \tfrac{1}{2}(I + R_f R_h)z$$

Reflection operator

$$R_f := 2\operatorname{prox}_f - I.$$

Observe: $R_f = -R_{f^*}$ (another justification of "reflection")

$$\begin{aligned} \operatorname{prox}_f + \operatorname{prox}_{f^*} &= I \\ 2\operatorname{prox}_f &= 2I - 2\operatorname{prox}_{f^*} \end{aligned}$$

Douglas-Rachford

$$\min \quad f(x) + h(x)$$

$$z \leftarrow \tfrac{1}{2}(I + R_f R_h) z$$

Reflection operator

$$R_f := 2\,\mathrm{prox}_f - I.$$

Observe: $R_f = -R_{f^*}$ (another justification of "reflection")

$$\begin{aligned}
\mathrm{prox}_f + \mathrm{prox}_{f^*} &= I \\
2\,\mathrm{prox}_f &= 2I - 2\,\mathrm{prox}_{f^*} \\
2\,\mathrm{prox}_f - I &= I - 2\,\mathrm{prox}_{f^*} \\
R_f &= -R_{f^*}
\end{aligned}$$

Douglas-Rachford – open problem

$$\min f(x) + g(x) + h(x)$$

Douglas-Rachford – open problem

$$\min f(x) + g(x) + h(x)$$

$$z \leftarrow \frac{1}{2}(I + R_f R_g R_h) z$$

Douglas-Rachford – open problem

$$\min f(x) + g(x) + h(x)$$

$$\begin{aligned} 0 &\in \partial f(x) + \partial g(x) + \partial h(x) \\ 3x &\in (I + \partial f)(x) + (I + \partial g)(x) + (I + \partial h)(x) \\ 3x &\in (I + \partial f)(x) + z + w \end{aligned}$$

now what?

Douglas-Rachford – open problem

$$\min f(x) + g(x) + h(x)$$

Partial solution (Borwein, Tam (2013))

$$\begin{aligned} T_{hf} &:= \tfrac{1}{2}(I + R_f R_h) \\ T_{[fgh]} &:= T_{hf} T_{gh} T_{fg} \\ z &\leftarrow T_{[fgh]} z \end{aligned}$$

Douglas-Rachford – open problem

$$\min f(x) + g(x) + h(x)$$

Partial solution (Borwein, Tam (2013))

$$T_{hf} := \frac{1}{2}(I + R_f R_h)$$
$$T_{[fgh]} := T_{hf} T_{gh} T_{fg}$$
$$z \leftarrow T_{[fgh]} z$$

- Works for more than 3 functions too!
- For two functions $T_{[fg]} = T_{gf} T_{fg}$
- Does not coincide with usual DR.
- Finding "correct" generalization an open problem

Parallel proximal methods

Optimizing separable objective functions

$$f(x) := \tfrac{1}{2}\|x-y\|_2^2 + \sum_i f_i(x)$$

$$f(x) := \sum_i f_i(x)$$

Parallel proximal methods

Optimizing separable objective functions

$$f(x) := \tfrac{1}{2}\|x - y\|_2^2 + \sum\nolimits_i f_i(x)$$
$$f(x) := \sum\nolimits_i f_i(x)$$

Let us consider

$$\min \quad f(x) = \sum\nolimits_{i=1}^m f_i(x), \qquad x \in \mathbb{R}^n.$$

Product space technique

- Original problem over $\mathcal{H} = \mathbb{R}^n$

Product space technique

- Original problem over $\mathcal{H} = \mathbb{R}^n$
- Suppose we have $\sum_{i=1}^{m} f_i(x)$

Product space technique

- Original problem over $\mathcal{H} = \mathbb{R}^n$
- Suppose we have $\sum_{i=1}^{m} f_i(x)$
- Introduce new variables $(x_1, \ldots, x_m)$

Product space technique

- Original problem over $\mathcal{H} = \mathbb{R}^n$
- Suppose we have $\sum_{i=1}^m f_i(x)$
- Introduce new variables $(x_1, \ldots, x_m)$
- Now problem is over domain $\mathcal{H}^m := \mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}$ (m-times)

Product space technique

- Original problem over $\mathcal{H} = \mathbb{R}^n$
- Suppose we have $\sum_{i=1}^m f_i(x)$
- Introduce new variables $(x_1, \ldots, x_m)$
- Now problem is over domain $\mathcal{H}^m := \mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}$ (m-times)
- New constraint: $x_1 = x_2 = \ldots = x_m$

$$\begin{aligned} \min_{(x_1,\ldots,x_m)} \quad & \sum\nolimits_i f_i(x_i) \\ \text{s.t.} \quad & x_1 = x_2 = \cdots = x_m. \end{aligned}$$

Technique due to: G. Pierra (1976)

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$
- $\mathrm{prox}_f(\boldsymbol{y}) = (\mathrm{prox}_{f_1}(y_1), \ldots, \mathrm{prox}_{f_m}(y_m))$

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$
- $\mathrm{prox}_f(\boldsymbol{y}) = (\mathrm{prox}_{f_1}(y_1), \ldots, \mathrm{prox}_{f_m}(y_m))$
- $\mathrm{prox}_{\mathbb{I}_{\mathcal{B}}} \equiv \Pi_{\mathcal{B}}(\boldsymbol{y})$ can be solved as follows:

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$
- $\operatorname{prox}_f(\boldsymbol{y}) = (\operatorname{prox}_{f_1}(y_1), \ldots, \operatorname{prox}_{f_m}(y_m))$
- $\operatorname{prox}_{\mathbb{I}_{\mathcal{B}}} \equiv \Pi_{\mathcal{B}}(\boldsymbol{y})$ can be solved as follows:

$$\begin{aligned} \min\nolimits_{\boldsymbol{z} \in \mathcal{B}} \quad & \tfrac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \\ \min\nolimits_{x \in \mathcal{H}} \quad & \textstyle\sum_i \tfrac{1}{2}\|x - y_i\|_2^2 \\ \Longrightarrow \quad & x = \tfrac{1}{m}\textstyle\sum_i y_i \end{aligned}$$

Product space technique

Two block problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{B}}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathcal{H}^m$ and $\mathcal{B} = \{\boldsymbol{z} \in \mathcal{H}^m \mid \boldsymbol{z} = (x, x, \ldots, x)\}$

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$
- $\mathrm{prox}_f(\boldsymbol{y}) = (\mathrm{prox}_{f_1}(y_1), \ldots, \mathrm{prox}_{f_m}(y_m))$
- $\mathrm{prox}_{\mathbb{I}_{\mathcal{B}}} \equiv \Pi_{\mathcal{B}}(\boldsymbol{y})$ can be solved as follows:

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{B}} \quad & \tfrac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \\ \min_{x\in\mathcal{H}} \quad & \textstyle\sum_i \tfrac{1}{2}\|x - y_i\|_2^2 \\ \Longrightarrow \quad & x = \tfrac{1}{m}\textstyle\sum_i y_i \end{aligned}$$

Exercise: Work out the details of DR using the product space idea

This technique commonly exploited in ADMM too

Alternative: two block proximity

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

Alternative: two block proximity

$$\min_x \ \frac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

Usually $\mathrm{prox}_{f+h} \neq \mathrm{prox}_f \circ \mathrm{prox}_h$

Alternative: two block proximity

$$\min_x \ \frac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

Usually $\mathrm{prox}_{f+h} \neq \mathrm{prox}_f \circ \mathrm{prox}_h$

Proximal-Dykstra method

1. Let $x_0 = y$; $u_0 = 0$, $z_0 = 0$
2. k-th iteration ($k \geq 0$)

Alternative: two block proximity

$$\min_x \ \frac{1}{2}\|x-y\|_2^2 + f(x) + h(x)$$

Usually $\mathrm{prox}_{f+h} \neq \mathrm{prox}_f \circ \mathrm{prox}_h$

Proximal-Dykstra method

1. Let $x_0 = y$; $u_0 = 0$, $z_0 = 0$
2. k-th iteration ($k \geq 0$)
   - $w_k = \mathrm{prox}_f(x_k + u_k)$
   - $u_{k+1} = x_k + u_k - w_k$

Alternative: two block proximity

$$\min_x \ \frac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

Usually $\operatorname{prox}_{f+h} \neq \operatorname{prox}_f \circ \operatorname{prox}_h$

Proximal-Dykstra method

1. Let $x_0 = y$; $u_0 = 0$, $z_0 = 0$
2. k-th iteration ($k \geq 0$)
   - $w_k = \operatorname{prox}_f(x_k + u_k)$
   - $u_{k+1} = x_k + u_k - w_k$
   - $x_{k+1} = \operatorname{prox}_h(w_k + z_k)$
   - $z_{k+1} = w_k + z_k - x_{k+1}$

Alternative: two block proximity

$$\min_x \; \tfrac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

Usually $\mathrm{prox}_{f+h} \neq \mathrm{prox}_f \circ \mathrm{prox}_h$

Proximal-Dykstra method

1. Let $x_0 = y$; $u_0 = 0$, $z_0 = 0$
2. k-th iteration ($k \geq 0$)
   - $w_k = \mathrm{prox}_f(x_k + u_k)$
   - $u_{k+1} = x_k + u_k - w_k$
   - $x_{k+1} = \mathrm{prox}_h(w_k + z_k)$
   - $z_{k+1} = w_k + z_k - x_{k+1}$

Why does it work?

Alternative: two block proximity

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

Usually $\mathrm{prox}_{f+h} \neq \mathrm{prox}_f \circ \mathrm{prox}_h$

Proximal-Dykstra method

1. Let $x_0 = y$; $u_0 = 0$, $z_0 = 0$
2. k-th iteration ($k \geq 0$)
   - $w_k = \mathrm{prox}_f(x_k + u_k)$
   - $u_{k+1} = x_k + u_k - w_k$
   - $x_{k+1} = \mathrm{prox}_h(w_k + z_k)$
   - $z_{k+1} = w_k + z_k - x_{k+1}$

Why does it work?

Exercise: Use the product-space technique to extend this to a *parallel prox-Dykstra* method for $m \geq 3$ functions.

Combettes, Pesquet (2010); Bauschke, Combettes (2012)

Proximal-Dykstra – some insight

$$\min_x \ \frac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

Proximal-Dykstra – some insight

$$\min_x \ \frac{1}{2}\|x - y\|_2^2 + f(x) + h(x)$$

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

Proximal-Dykstra – some insight

$$\min_x \ \frac{1}{2}\|x-y\|_2^2 + f(x) + h(x)$$

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

- Let's derive the dual from L:

$$g(\nu,\mu) \quad := \quad \inf_{x,z,w} L(x,z,\nu,\mu)$$

Proximal-Dykstra – some insight

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

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

- Let's derive the dual from L:

$$\begin{aligned} g(\nu,\mu) \quad &:= \quad \inf_{x,z,w} L(x,z,\nu,\mu) \\ x-y+\nu+\mu=0 \quad &\Longrightarrow \quad x = y-\nu-\mu \end{aligned}$$

Proximal-Dykstra – some insight

$$\min_x \; \tfrac{1}{2}\|x-y\|_2^2 + f(x) + h(x)$$

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

- Let's derive the dual from L:

$$\begin{aligned} g(\nu,\mu) \quad &:= \quad \inf_{x,z,w} L(x,z,\nu,\mu) \\ x - y + \nu + \mu = 0 \quad &\Longrightarrow \quad x = y - \nu - \mu \\ \inf_z f(z) - \nu^T z \quad &= \quad -f^*(\nu), \qquad \text{(similarly get } -h^*(\mu)) \end{aligned}$$

Proximal-Dykstra – some insight

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

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

- Let's derive the dual from L:

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

Equivalent dual problem

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

Proximal-Dykstra – key insight

Dual problem

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

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

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

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

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

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

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

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

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

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

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

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

Proximal-Dykstra – key insight

Dual problem

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

Solve this dual via Block-Coordinate Descent!

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

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

Similarly, $\mu_{k+1} = y - \nu_{k+1} - \operatorname{prox}_h(y - \nu_{k+1})$

Proximal-Dykstra – key insight

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

Proximal-Dykstra – key insight

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

$$\nu_{k+1} = y - \mu_k - \operatorname{prox}_f(y - \mu_k)$$
$$\mu_{k+1} = y - \nu_{k+1} - \operatorname{prox}_h(y - \nu_{k+1})$$

Proximal-Dykstra – key insight

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

$$\nu_{k+1} = y - \mu_k - \mathrm{prox}_f(y - \mu_k)$$
$$\mu_{k+1} = y - \nu_{k+1} - \mathrm{prox}_h(y - \nu_{k+1})$$

Now use Lagrangian stationarity condition

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

to rewrite BCD using primal and dual variables.

Proximal-Dykstra – key insight

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

$$\nu_{k+1} = y - \mu_k - \mathrm{prox}_f(y - \mu_k)$$
$$\mu_{k+1} = y - \nu_{k+1} - \mathrm{prox}_h(y - \nu_{k+1})$$

Now use Lagrangian stationarity condition

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

to rewrite BCD using primal and dual variables.

BCD

$$\nu_{k+1} = \mathrm{argmin}_\nu \; G(\nu, \mu_k),$$
$$\mu_{k+1} = \mathrm{argmin}_\mu \; G(\nu_{k+1}, \mu).$$

Proximal-Dykstra – key insight

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

$$\nu_{k+1} = y - \mu_k - \mathrm{prox}_f(y - \mu_k)$$
$$\mu_{k+1} = y - \nu_{k+1} - \mathrm{prox}_h(y - \nu_{k+1})$$

Now use Lagrangian stationarity condition

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

to rewrite BCD using primal and dual variables.

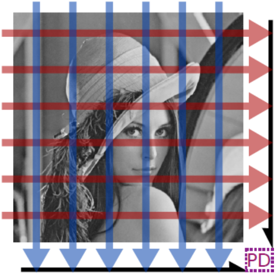
Prox-Dykstra

$$w_k \leftarrow \mathrm{prox}_f(x_k + \nu_k)$$
$$\nu_{k+1} \leftarrow x_k + \nu_k - w_k$$
$$x_{k+1} \leftarrow \mathrm{prox}_h(w_k + \mu_k)$$
$$\mu_{k+1} \leftarrow \mu_k + w_k - x_{k+1}$$

Example practical use

Anisotropic 2D-TV Proximity operator

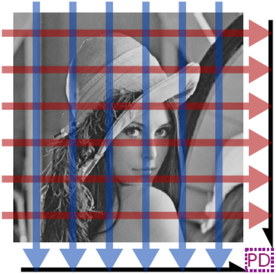
$$\min_X \quad \tfrac{1}{2}\|X - Y\|_{\mathsf{F}}^2 + \sum_{ij} w_{ij}^c |x_{i,j+1} - x_{ij}| + \sum_{ij} w_{ij}^r |x_{i+1,j} - x_{ij}|$$



Example practical use

Anisotropic 2D-TV Proximity operator

$$\min_{X} \quad \tfrac{1}{2}\|X - Y\|_{\mathsf{F}}^2 + \sum\nolimits_{ij} w_{ij}^c |x_{i,j+1} - x_{ij}| + \sum\nolimits_{ij} w_{ij}^r |x_{i+1,j} - x_{ij}|$$



- Amenable to prox-Dykstra
- Used in (Barbero, Sra, ICML 2011).
- The subproblem:

  $\min \frac{1}{2}\|a - b\|_2^2 + \sum_i w_i |a_i - a_{i+1}|$

  itself has been subject of over 15 papers!
- I still use it now and then

Incremental first-order methods

Separable objectives

$$\min \quad f(x) = \sum_i^m f_i(x) + \lambda r(x)$$

Separable objectives

$$\min \quad f(x) = \sum_i^m f_i(x) + \lambda r(x)$$

Gradient / subgradient methods

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \qquad \lambda = 0, \\ x_{k+1} &= x_k - \alpha_k g(x_k), \qquad g(x_k) \in \partial f(x_k) + \lambda \partial r(x_k) \\ x_{k+1} &= \text{prox}_{\alpha_k r}(x_k - \alpha_k \nabla f(x_k)) \end{aligned}$$

Product-space based methods

$$\min F(x_1, \ldots, x_m) + \mathbb{I}_{\mathcal{B}}(x_1, \ldots, x_m)$$

$$(x_{1,k+1}, \ldots, x_{m,k+1}) \leftarrow \text{prox}_F(y_{1,k}, \ldots, y_{m,k})$$

Separable objectives

$$\min \quad f(x) = \sum_i^m f_i(x) + \lambda r(x)$$

Gradient / subgradient methods

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \qquad \lambda = 0, \\ x_{k+1} &= x_k - \alpha_k g(x_k), \qquad g(x_k) \in \partial f(x_k) + \lambda \partial r(x_k) \\ x_{k+1} &= \mathrm{prox}_{\alpha_k r}(x_k - \alpha_k \nabla f(x_k)) \end{aligned}$$

Product-space based methods

$$\min F(x_1, \ldots, x_m) + \mathbb{I}_{\mathcal{B}}(x_1, \ldots, x_m)$$

$$(x_{1,k+1}, \ldots, x_{m,k+1}) \leftarrow \mathrm{prox}_F(y_{1,k}, \ldots, y_{m,k})$$

How much computation does one iteration take?

Incremental gradient methods

What if at iteration k, we randomly pick an integer $i(k) \in \{1, 2, \ldots, m\}$?

Incremental gradient methods

What if at iteration k, we randomly pick an integer $i(k) \in \{1, 2, \ldots, m\}$?

And instead just perform the update?

$$x_{k+1} = x_k - \alpha_k \nabla f_{i(k)}(x_k)$$

Incremental gradient methods

What if at iteration k, we randomly pick an integer $i(k) \in \{1, 2, \ldots, m\}$?

And instead just perform the update?

$$x_{k+1} = x_k - \alpha_k \nabla f_{i(k)}(x_k)$$

- The update requires only gradient for $f_{i(k)}$
- One iteration now m times faster than with $\nabla f(x)$

Incremental gradient methods

What if at iteration k, we randomly pick an integer $i(k) \in \{1, 2, \ldots, m\}$?

And instead just perform the update?

$$x_{k+1} = x_k - \alpha_k \nabla f_{i(k)}(x_k)$$

- The update requires only gradient for $f_{i(k)}$
- One iteration now m times faster than with $\nabla f(x)$

But does this make sense?

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly
- ♠ For large m many $f_i(x)$ may have similar minimizers;

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly
- ♠ For large m many $f_i(x)$ may have similar minimizers; using the f_i individually we could take advantage, and greatly speed up.

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly
- ♠ For large m many $f_i(x)$ may have similar minimizers; using the f_i individually we could take advantage, and greatly speed up.
- ♠ Incremental methods usually effective far from the eventual limit (solution) — become very slow close to the solution.

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly
- ♠ For large m many $f_i(x)$ may have similar minimizers; using the f_i individually we could take advantage, and greatly speed up.
- ♠ Incremental methods usually effective far from the eventual limit (solution) — become very slow close to the solution.
- ♠ Several open questions related to convergence and rate of convergence (for both convex, nonconvex)

Incremental gradient methods

- ♠ Old idea; has been used extensively as *backpropagation* in neural networks, Widrow-Hoff least mean squares, gradient methods with errors, stochastic gradient, etc.
- ♠ Can "stream" through data — go through components one by one, say *cyclically* instead of randomly
- ♠ For large m many $f_i(x)$ may have similar minimizers; using the f_i individually we could take advantage, and greatly speed up.
- ♠ Incremental methods usually effective far from the eventual limit (solution) — become very slow close to the solution.
- ♠ Several open questions related to convergence and rate of convergence (for both convex, nonconvex)
- ♠ Usually randomization greatly simplifies convergence analysis

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2} \sum_{i=1}^{m} (a_i x - b_i)^2$$

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2}\sum_{i=1}^{m}(a_i x - b_i)^2$$

- Solving $f'(x) = 0$ we obtain

$$x^* = \frac{\sum_i a_i b_i}{\sum_i a_i^2}$$

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2} \sum_{i=1}^{m} (a_i x - b_i)^2$$

- Solving $f'(x) = 0$ we obtain

$$x^* = \frac{\sum_i a_i b_i}{\sum_i a_i^2}$$

- Minimum of a single $f_i(x) = \frac{1}{2}(a_i x - b_i)^2$ is $x_i^* = b_i / a_i$

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \tfrac{1}{2} \sum_{i=1}^{m} (a_i x - b_i)^2$$

- Solving $f'(x) = 0$ we obtain

$$x^* = \frac{\sum_i a_i b_i}{\sum_i a_i^2}$$

- Minimum of a single $f_i(x) = \frac{1}{2}(a_i x - b_i)^2$ is $x_i^* = b_i/a_i$
- Notice now that

$$x^* \in [\min_i x_i^*, \max_i x_i^*] =: R$$

(Use: $\sum_i a_i b_i = \sum_i a_i^2 (b_i/a_i)$)

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2}\sum_{i=1}^{m}(a_i x - b_i)^2$$

- Notice: $x^* \in [\min_i x_i^*, \max_i x_i^*] =: R$

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \tfrac{1}{2}\sum_{i=1}^{m}(a_i x - b_i)^2$$

- Notice: $x^* \in [\min_i x_i^*, \max_i x_i^*] =: R$
- If we have a scalar x that lies outside R?
- We see that

$$\nabla f_i(x) = a_i(a_i x - b_i)$$
$$\nabla f(x) = \sum_i a_i(a_i x - b_i)$$

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2}\sum_{i=1}^{m}(a_i x - b_i)^2$$

- Notice: $x^* \in [\min_i x_i^*, \max_i x_i^*] =: R$
- If we have a scalar x that lies outside R?
- We see that

$$\nabla f_i(x) = a_i(a_i x - b_i)$$
$$\nabla f(x) = \sum_i a_i(a_i x - b_i)$$

- $\nabla f_i(x)$ has **same sign** as $\nabla f(x)$. So using $\nabla f_i(x)$ **instead** of $\nabla f(x)$ also ensures progress.

Example (Bertsekas)

- Assume all variables involved are **scalars**.

$$\min \quad f(x) = \frac{1}{2}\sum_{i=1}^{m}(a_i x - b_i)^2$$

- Notice: $x^* \in [\min_i x_i^*, \max_i x_i^*] =: R$
- If we have a scalar x that lies outside R?
- We see that

$$\nabla f_i(x) = a_i(a_i x - b_i)$$
$$\nabla f(x) = \sum_i a_i(a_i x - b_i)$$

- $\nabla f_i(x)$ has **same sign** as $\nabla f(x)$. So using $\nabla f_i(x)$ **instead** of $\nabla f(x)$ also ensures progress.
- But once inside region R, **no guarantee** that incremental method will make progress towards optimum.

Incremental proximal method

$$\min \quad f(x) = \sum_i f_i(x)$$

What if the f_i are nonsmooth?

Incremental proximal method

$$\boxed{\min \quad f(x) = \sum_i f_i(x)}$$

What if the f_i are nonsmooth?

$$\cancel{x_{k+1} = \operatorname{prox}_{\alpha_k f}(x_k)}$$

Incremental proximal method

$$\min \quad f(x) = \sum_i f_i(x)$$

What if the f_i are nonsmooth?

$$\cancel{x_{k+1} = \mathrm{prox}_{\alpha_k f}(x_k)}$$

$$x_{k+1} = \mathrm{prox}_{\alpha_k f_{i(k)}}(x_k)$$

$$x_{k+1} = \mathrm{argmin}\left(\tfrac{1}{2}\|x - x_k\|_2^2 + \alpha_k f_{i(k)}(x)\right)$$

$i(k) \in \{1, 2, \ldots, m\}$ picked uniformly at random.

Incremental proximal method

$$\min \quad f(x) = \sum_i f_i(x)$$

What if the f_i are nonsmooth?

$$\xcancel{x_{k+1} = \operatorname{prox}_{\alpha_k f}(x_k)}$$

$$x_{k+1} = \operatorname{prox}_{\alpha_k f_{i(k)}}(x_k)$$

$$x_{k+1} = \operatorname{argmin}\left(\tfrac{1}{2}\|x - x_k\|_2^2 + \alpha_k f_{i(k)}(x)\right)$$

$i(k) \in \{1, 2, \ldots, m\}$ picked uniformly at random.

Convergence rate analysis?

Example

Example

Fermat-Weber problem
(historically the first **facility-location** problem)

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

Example

Fermat-Weber problem
(historically the first **facility-location** problem)

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

- Assuming $\|\cdot\| = \|\cdot\|_2$
- Also assume no a_i is an optimum
- [Weiszfeld; '37] Let $T := x \mapsto \left(\sum_i \frac{w_i a_i}{\|x-a_i\|}\right) / \left(\sum_i \frac{w_i}{\|x-a_i\|}\right)$
- Assuming T is well-defined, $T^k(x_0) \to \mathrm{argmin}$
- [Kuhn; 73] completed the proof

Example

Fermat-Weber problem
(historically the first **facility-location** problem)

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

- Assuming $\|\cdot\| = \|\cdot\|_2$
- Also assume no a_i is an optimum
- [Weiszfeld; '37] Let $T := x \mapsto \left(\sum_i \frac{w_i a_i}{\|x-a_i\|}\right) / \left(\sum_i \frac{w_i}{\|x-a_i\|}\right)$
- Assuming T is well-defined, $T^k(x_0) \to \mathrm{argmin}$
- [Kuhn; 73] completed the proof
- What if $\|\cdot\| = \|\cdot\|_p$?
- 100s of papers discuss the Fermat-Weber problem

Incremental proximal method

Fermat-Weber problem

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

Incremental proximal method

Fermat-Weber problem

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

Now, $f_i(x) := w_i \|x - a_i\|_2$.

$$x_{k+1} = \mathrm{prox}_{\alpha_k f_{i(k)}}(x_k)$$

$$x_{k+1} = \mathrm{argmin}\left(\tfrac{1}{2}\|x - x_k\|_2^2 + \alpha_k w_{i(k)} \|x - a_{i(k)}\|_2\right)$$

$i(k) \in \{1, 2, \ldots, m\}$ picked uniformly at random.

Incremental proximal method

Fermat-Weber problem

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

Now, $f_i(x) := w_i \|x - a_i\|_2$.

$$x_{k+1} = \text{prox}_{\alpha_k f_{i(k)}}(x_k)$$

$$x_{k+1} = \text{argmin}\left(\tfrac{1}{2}\|x - x_k\|_2^2 + \alpha_k w_{i(k)} \|x - a_{i(k)}\|_2\right)$$

$i(k) \in \{1, 2, \ldots, m\}$ picked uniformly at random.

Exercise: Obtain closed form solution to x_{k+1}

Incremental proximal method

Fermat-Weber problem

$$\min_x \quad \sum_i w_i \|x - a_i\|$$

Now, $f_i(x) := w_i \|x - a_i\|_2$.

$$x_{k+1} = \text{prox}_{\alpha_k f_{i(k)}}(x_k)$$

$$x_{k+1} = \text{argmin}\left(\tfrac{1}{2}\|x - x_k\|_2^2 + \alpha_k w_{i(k)} \|x - a_{i(k)}\|_2\right)$$

$i(k) \in \{1, 2, \ldots, m\}$ picked uniformly at random.

Exercise: Obtain closed form solution to x_{k+1}

Rate of convergence? Most likely, sublinear? Can we somehow get linear convergence?

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$x_{k+1} \quad = \quad \operatorname{prox}_{\eta_k r}\bigl(x_k - \eta_k \sum_{i=1}^{m} \nabla f_i(z_i)\bigr), \quad k = 0, 1, \ldots,$$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$\begin{aligned} x_{k+1} \quad &= \quad \operatorname{prox}_{\eta_k r}\big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\big), \quad k = 0, 1, \ldots, \\ & \quad\quad z_1 = x_k \\ & \quad\quad z_{i+1} = z_i - \eta_k \nabla f_i(z_i), \quad i = 1, \ldots, m-1. \end{aligned}$$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$\begin{aligned} x_{k+1} \quad &= \quad \operatorname{prox}_{\eta_k r}\Big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\Big), \quad k = 0, 1, \ldots, \\ & \quad\quad z_1 = x_k \\ & \quad\quad z_{i+1} = z_i - \eta_k \nabla f_i(z_i), \quad i = 1, \ldots, m-1. \end{aligned}$$

We can choose $\eta_k = 1/L$, where L is Lipschitz constant of $\nabla f(x)$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$\begin{aligned} x_{k+1} &= \operatorname{prox}_{\eta_k r}\big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\big), \quad k = 0, 1, \ldots, \\ & z_1 = x_k \\ & z_{i+1} = z_i - \eta_k \nabla f_i(z_i), \quad i = 1, \ldots, m-1. \end{aligned}$$

We can choose $\eta_k = 1/L$, where L is Lipschitz constant of $\nabla f(x)$
Might be easier to analyze

$$x_{k+1} = \operatorname{prox}_{\eta_k r}\big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\big), \quad k = 0, 1, \ldots,$$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$\begin{aligned} x_{k+1} \quad &= \quad \mathrm{prox}_{\eta_k r}\big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\big), \quad k = 0, 1, \ldots, \\ &\quad\; z_1 = x_k \\ &\quad\; z_{i+1} = z_i - \eta_k \nabla f_i(z_i), \quad i = 1, \ldots, m-1. \end{aligned}$$

We can choose $\eta_k = 1/L$, where L is Lipschitz constant of $\nabla f(x)$
Might be easier to analyze

$$\begin{aligned} x_{k+1} \quad &= \quad \mathrm{prox}_{\eta_k r}\big(x_k - \eta_k \sum_{i=1}^m \nabla f_i(z_i)\big), \quad k = 0, 1, \ldots, \\ &\quad\; z_1 = x_k \\ &\quad\; z_{i+1} = \mathrm{prox}_{\eta_k r}(z_i - \eta_k \nabla f_i(z_i)), \quad i = 1, \ldots, m-1. \end{aligned}$$

Incremental proximal-gradients

$$\min \quad \sum_i f_i(x) + r(x).$$

$$\begin{aligned} x_{k+1} \quad &= \quad \operatorname{prox}_{\eta_k r}\Big(x_k - \eta_k \sum\nolimits_{i=1}^{m} \nabla f_i(z_i)\Big), \quad k = 0, 1, \ldots, \\ & \quad\; z_1 = x_k \\ & \quad\; z_{i+1} = z_i - \eta_k \nabla f_i(z_i), \quad i = 1, \ldots, m-1. \end{aligned}$$

We can choose $\eta_k = 1/L$, where L is Lipschitz constant of $\nabla f(x)$
Might be easier to analyze

$$\begin{aligned} x_{k+1} \quad &= \quad \operatorname{prox}_{\eta_k r}\Big(x_k - \eta_k \sum\nolimits_{i=1}^{m} \nabla f_i(z_i)\Big), \quad k = 0, 1, \ldots, \\ & \quad\; z_1 = x_k \\ & \quad\; z_{i+1} = \operatorname{prox}_{\eta_k r}(z_i - \eta_k \nabla f_i(z_i)), \quad i = 1, \ldots, m-1. \end{aligned}$$

Moreover, analysis easier if we go through the f_i randomly
(so-called stochastic)

Incremental methods: deterministic

$$\min \quad \left(f(x) = \sum_i f_i(x)\right) + r(x)$$

Gradient with error

$$\nabla f_{i(k)}(x) = \nabla f(x) + e$$

$$x_{k+1} = \mathrm{prox}_{\alpha r}[x_k - \alpha_k(\nabla f(x_k) + e_k)]$$

Incremental methods: deterministic

$$\min \quad \left(f(x) = \sum_i f_i(x)\right) + r(x)$$

Gradient with error

$$\nabla f_{i(k)}(x) = \nabla f(x) + e$$

$$x_{k+1} = \text{prox}_{\alpha r}[x_k - \alpha_k(\nabla f(x_k) + e_k)]$$

So if in the limit error $\alpha_k e_k$ disappears, we should be ok!

Incremental gradient methods

Incremental gradient methods may be viewed as

Gradient methods with error in gradient computation

Incremental gradient methods

Incremental gradient methods may be viewed as

Gradient methods with error in gradient computation

- If we can control this error, we can control convergence

Incremental gradient methods

Incremental gradient methods may be viewed as

Gradient methods with error in gradient computation

- If we can control this error, we can control convergence
- Error makes even smooth case more like nonsmooth case

Incremental gradient methods

Incremental gradient methods may be viewed as

Gradient methods with error in gradient computation

- If we can control this error, we can control convergence
- Error makes even smooth case more like nonsmooth case
- So, convergence crucially depends on stepsize α_k

Incremental gradient methods

Incremental gradient methods may be viewed as

Gradient methods with error in gradient computation

- If we can control this error, we can control convergence
- Error makes even smooth case more like nonsmooth case
- So, convergence crucially depends on stepsize α_k

Some stepsize choices

- ♠ $\alpha_k = c$, a small enough constant
- ♠ $\alpha_k \to 0$, $\sum_k \alpha_k = \infty$ (diminishing scalar)
- ♠ Constant for some iterations, diminish, again constant, repeat
- ♠ $\alpha_k = \min(c, a/(b+k))$, where $a, b, c > 0$ (user chosen).

Incremental gradient – summary

♠ Usually much faster (large m) when *far* from convergence

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence
- ♠ Usually slow, sublinear rate of convergence

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence
- ♠ Usually slow, sublinear rate of convergence
- ♠ If f_i strongly convex, linear rate available (SAG, SVRG)

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence
- ♠ Usually slow, sublinear rate of convergence
- ♠ If f_i strongly convex, linear rate available (SAG, SVRG)
- ♠ Idea extends to subgradient, and proximal setups

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence
- ♠ Usually slow, sublinear rate of convergence
- ♠ If f_i strongly convex, linear rate available (SAG, SVRG)
- ♠ Idea extends to subgradient, and proximal setups
- ♠ Some extensions also apply to nonconvex problems

Incremental gradient – summary

- ♠ Usually much faster (large m) when *far* from convergence
- ♠ Slow progress near optimum (because α_k often too small)
- ♠ Constant step $\alpha_k = \alpha$, doesn't always yield convergence
- ♠ Diminishing step $\alpha_k = O(1/k)$ leads to convergence
- ♠ Usually slow, sublinear rate of convergence
- ♠ If f_i strongly convex, linear rate available (SAG, SVRG)
- ♠ Idea extends to subgradient, and proximal setups
- ♠ Some extensions also apply to nonconvex problems
- ♠ Some extend to parallel and distributed computation

References

- ♠ *EE227A slides, S. Sra*
- ♠ *Introductory Lectures on Convex Optimization, Yu. Nesterov*
- ♠ *Proximal splitting methods, Combettes & Pesquet*