# ADMM: Classical Derivation

12:11 PM

## Dual problem

$$\min.\ f(x)$$
$$\text{s.t. } Ax=b \quad \ldots (1)$$

$$L(x,y) = f(x) + y^T(Ax-b)$$

$$g(y) = \inf_x L(x,y), \quad x^*(y) = \operatorname*{argmin}_x L(x,y)$$
$$= L(x^*(y), y)$$

dual problem: $\max.\ g(y)$

$\Downarrow$

argmax $y^*$

$\therefore$ optimal solution, $x^* = x^*(y^*) = \operatorname*{argmin}_x L(x, y^*)$

Dual ascent:

$$y^{k+1} = y^k + \alpha^k \underbrace{\tilde{\nabla} g(y^k)}$$

$$\tilde{\nabla} g(y^k) = A\tilde{x} - b, \quad \tilde{x} = \operatorname*{argmin}_x L(x, y^k)$$

$$\tilde{x} = \operatorname*{argmin}_x L(x, y^k)$$

$$y^{k+1} = y^k + \alpha^k (A\tilde{x} - b)$$

$\downarrow$ to keep track introduce iteration counter $x^{k+1}$

$$x^{k+1} = \operatorname*{argmin}_x L(x, y^k)$$
$$y^{k+1} = y^k + \alpha^k (Ax^{k+1} - b)$$

Works under lot of strong assumptions.

Dual Decomposition.

$f(x) = f_1(x_1) + \dots + f_N(x_N)$ /* each convex in $x_i$ */

$$\min \; f(x) = \sum_{i=1}^{n} f_i(x_i)$$
$$\text{s.t } Ax = b \Leftrightarrow \sum A_i x_i = b$$

$$L(x,y) = \sum f_i(x_i) + y^T\left(\sum A_i x_i - b\right)$$
$$= \sum f_i(x_i) + \sum y^T A_i x_i - y^T b$$
$$= \sum_i \left(f_i(x_i) + y^T A_i x_i\right) - y^T b$$

$$g(y) = \inf_x L(x,y)$$
$$= \inf_x \left( \sum_i \left(f_i(x_i) + y^T A_i x_i\right)\right) - y^T b$$
$$= -y^T b + \underbrace{\inf_x \sum_i f_i(x_i) + y^T A_i x_i}_{\sum_i \inf_{x_i} \left(f_i(x_i) + y^T A_i x_i\right)}$$

$$\downarrow$$

$$\tilde{x}_i = \operatorname*{argmin}_{x_i} f_i(x_i) + y^T A_i x_i$$

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_N \end{bmatrix} = \operatorname*{argmin}_x L(x, y^k)$$ /* parallel step */

$$y^{k+1} = y^k + \alpha^k \left(\sum A_i \tilde{x}_i - b\right)$$ /* consensus or coordination between agents */
$$= y^k - \alpha^k b + \sum_i \alpha^k A_i \tilde{x}_i$$

* solves large scale problem
* work with lot of assumptions.
* slow.

⊛ Method of multipliers:

• robustifies dual ascent.

Problem (1) is written as:

$$\min \ f(x) + \frac{\rho}{2}\|Ax-b\|^2$$
$$\text{s.t.} \quad Ax=b$$

augmented Lagrangian:

$$L_\rho(x,y) = f(x) + y^T(Ax-b) + \frac{\rho}{2}\|Ax-b\|^2$$

$$\left\| \begin{bmatrix} a_1^Tx-b_1 \\ \vdots \\ a_m^Tx-b_m \end{bmatrix} \right\|^2 = \sum_i (a_i^Tx-b_i)^2$$

now there is an x in every term, so when $f(x)=\sum f_i(x_i)$, the splitting would not work any more.

$$x^{k+1} = \operatorname*{argmin}_x L_\rho(x,y^k)$$
$$y^{k+1} = y^k + \rho(Ax^{k+1}-b) \tag{2}$$

specific dual update step length $\rho$.

KKT condition: at $x^*, y^*$

• Primal Feasibility: $Ax^*-b=0$

• Dual feasibility: no inequality constraints so not needed

• Vanishing gradient of the Lagrangian: $\nabla f(x^*) + \rho(Ax^*-b) + A^Ty^* = 0$

$$\nabla f(x^*) + A^Ty^* = 0$$

$x^{k+1}$: $\nabla_x L_\rho(x,y^k)=0$

$$\Leftrightarrow \nabla_x \left(f(x) + y^{kT}(Ax-b) + \frac{\rho}{2}\|Ax-b\|^2\right) = 0$$

$$\Leftrightarrow \nabla f(x^{k+1}) + \underbrace{A^Ty^k + \rho A^T(Ax^{k+1}-b)}_{A^Ty^{k+1}} = 0$$

/* $y^{k+1} = y^k + \rho(Ax^{k+1}-b)$

$\Leftrightarrow A^Ty^{k+1} = A^Ty^k + \rho A^T(Ax^{k+1}-b)$ */

$$\Leftrightarrow \nabla f(x^{k+1}) + A^Ty^{k+1} = 0$$

$x^{k+1}$: $\nabla f(x^{k+1}) + A^Ty^{k+1} = 0$

$$y^{k+1} = y^k + \rho(Ax^{k+1} - b)$$

advantage: converges under much more relaxed conditions

dis " : quadratic penalty destroys splitting.

* Alternating direction method of multipliers.

ADMM problem form:

$$\min.\ f(x) + g(z)$$
$$\text{s.t.}\quad Ax + Bz = c$$
$$\Downarrow$$
$$\min.\ \underbrace{f(x)+g(z)}_{\tilde{f}(\tilde{x})} + \frac{\rho}{2}\|Ax+Bz-c\|^2 = \tilde{f}(\tilde{x}) + \frac{\rho}{2}\|\tilde{A}\tilde{x} - c\|^2$$
$$\text{s.t.}\quad Ax+Bz=c. \Leftrightarrow [A\ B]\underbrace{\begin{bmatrix} x \\ z \end{bmatrix}}_{\tilde{x}} = \underbrace{[A\ B]}_{\tilde{A}}\tilde{x} = c$$

$$L_\rho(x,z,y) = f(x)+g(z)+\frac{\rho}{2}\|Ax+Bz-c\|^2$$
$$L_\rho(\tilde{x},y) = \tilde{f}(\tilde{x}) + \frac{\rho}{2}\|\tilde{A}\tilde{x} - c\|^2$$

From (2):

$$\tilde{x}^{k+1} = \underset{x}{\text{argmin}}\ L_\rho(\tilde{x}, y^k)$$
$$y^{k+1} = y^k + \rho(Ax^{k+1} - b)$$

$$\tilde{x}^{k+1} = \begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix} : 0 = \nabla_{\tilde{x}} L_\rho(\tilde{x}, y^k) = \begin{bmatrix} \frac{d}{dx_1} L_\rho(\tilde{x}, y^k) \\ \vdots \\ \frac{d}{dx_N} L_\rho(\tilde{x}, y^k) \\ \frac{d}{dz_1} L_\rho(\tilde{x}, y^k) \\ \vdots \\ \frac{d}{dz_N} L_\rho(\tilde{x}, y^k) \end{bmatrix} = \begin{bmatrix} \nabla_x\ f(x)+g(z)+\frac{\rho}{2}\|Ax+Bz-c\|^2 \\ \nabla_z\ f(x)+g(z)+\frac{\rho}{2}\|Ax+Bz-c\|^2 \end{bmatrix}$$

$$\rightarrow\quad x^{k+1}:\ \nabla_x L_\rho(x,z,y^k) = 0$$
$$z^{k+1}:\ \nabla_z L_\rho(x,z,y^k) = 0$$
$$y^{k+1} = y^k + \rho(\tilde{A}\tilde{x}^{k+1} - b)$$

$\Downarrow$ why not use the latest values of $x$ and $z$?
lets do it

$x^{k+1}$: $\nabla_x L_\rho(x, z^k, y^k) = 0 \leftrightarrow x^{k+1} = \underset{x}{\operatorname{argmin}}\ L_\rho(x, z^k, y^k)$

$z^{k+1}$: $\nabla_z L_\rho(x^{k+1}, z, y^k) = 0 \leftrightarrow z^{k+1} = \underset{z}{\operatorname{argmin}}\ L_\rho(x^{k+1}, z, y^k)$

$y^{k+1} = y^k + (Ax^{k+1} + Bz^{k+1} - c)$