

Dual/Lagrangian Methods for Constrained Optimization

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

<http://www.stanford.edu/~yyye>

Chapter 14

The Lagrangian Function and Method

We consider

$$f^* := \min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in X. \quad (1)$$

Recall that the **Lagrangian** function:

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{h}(\mathbf{x}).$$

and the **dual function**:

$$\phi(\mathbf{y}) = \min_{\mathbf{x} \in X} L(\mathbf{x}, \mathbf{y}); \quad (2)$$

and the **dual problem**

$$(f^* \geq) \phi^* := \max_{\mathbf{y}} \phi(\mathbf{y}). \quad (3)$$

In many cases, one can find \mathbf{y}^* of dual problem (3), a **unconstrained** optimization problem; then go ahead to find \mathbf{x}^* using (2).

The Local Duality Theorem

Suppose \mathbf{x}^* is a local minimizer, and consider the **localized problem**

$$f(\mathbf{x}^*) := \min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in X, \|\mathbf{x} - \mathbf{x}^*\|^2 \leq \epsilon. \quad (4)$$

Then, the **localized Lagrangian function**:

$$L_{\mathbf{x}^*}(\mathbf{x}, \mathbf{y}, \mu(\leq 0)) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{h}(\mathbf{x}) - \mu(\|\mathbf{x} - \mathbf{x}^*\|^2 - \epsilon).$$

and the localized dual function:

$$\phi_{\mathbf{x}^*}(\mathbf{y}, \mu) = \min_{\mathbf{x} \in X, \|\mathbf{x} - \mathbf{x}^*\|^2 \leq \epsilon} L_{\mathbf{x}^*}(\mathbf{x}, \mathbf{y}, \mu); \quad (5)$$

and the **localized dual problem**

$$\max_{\mathbf{y}, \mu \leq 0} \phi(\mathbf{y}, \mu \leq 0). \quad (6)$$

Under certain **constraint qualification and local convexity conditions**, we must have

$f(\mathbf{x}^*) = \phi(\mathbf{y}^*, \mu^* = 0)$ where the localization constraint becomes **inactive**.

The gradient and Hessian of ϕ

Let $\mathbf{x}(\mathbf{y})$ be a minimizer of (2). Then

$$\phi(\mathbf{y}) = f(\mathbf{x}(\mathbf{y})) - \mathbf{y}^T \mathbf{h}(\mathbf{x}(\mathbf{y}))$$

Thus,

$$\begin{aligned} \nabla \phi(\mathbf{y}) &= \nabla f(\mathbf{x}(\mathbf{y}))^T \nabla \mathbf{x}(\mathbf{y}) - \mathbf{y}^T \nabla \mathbf{h}(\mathbf{x}(\mathbf{y})) \nabla \mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y})) \\ &= (\nabla f(\mathbf{x}(\mathbf{y}))^T - \mathbf{y}^T \nabla \mathbf{h}(\mathbf{x}(\mathbf{y}))) \nabla \mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y})) \\ &= -\mathbf{h}(\mathbf{x}(\mathbf{y})). \end{aligned}$$

Similarly, we can derive

$$\nabla^2 \phi(\mathbf{y}) = -\nabla \mathbf{h}(\mathbf{x}(\mathbf{y})) \left(\nabla_{\mathbf{x}}^2 L(\mathbf{x}(\mathbf{y}), \mathbf{y}) \right)^{-1} \nabla \mathbf{h}(\mathbf{x}(\mathbf{y}))^T,$$

where $\nabla_{\mathbf{x}}^2 L(\mathbf{x}(\mathbf{y}), \mathbf{y})$ is the Hessian of the Lagrangian function that is assumed to be positive definite at any (local) minimizer.

The Toy Example

$$\text{minimize} \quad (x_1 - 1)^2 + (x_2 - 1)^2$$

$$\text{subject to} \quad x_1 + 2x_2 - 1 = 0, \quad 2x_1 + x_2 - 1 = 0.$$

$$L(\mathbf{x}, \mathbf{y}) = (x_1 - 1)^2 + (x_2 - 1)^2 - y_1(x_1 + 2x_2 - 1) - y_2(2x_1 + x_2 - 1).$$

$$x_1 = 0.5y_1 + y_2 + 1, \quad x_2 = y_1 + 0.5y_2 + 1.$$

$$\phi(\mathbf{y}) = -1.25y_1^2 - 1.25y_2^2 - 2y_1y_2 - 2y_1 - 2y_2.$$

$$\nabla\phi(\mathbf{y}) = \begin{pmatrix} 2.5y_1 + 2y_2 + 2 \\ 2y_1 + 2.5y_2 + 2 \end{pmatrix},$$

$$\nabla^2\phi(\mathbf{y}) = - \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}^T = - \begin{pmatrix} 2.5 & 2 \\ 2 & 2.5 \end{pmatrix}$$

The Augmented Lagrangian Function

In both theory and practice, we actually consider an **augmented** Lagrangian function (ALF)

$$L_a(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{h}(\mathbf{x}) + \frac{\beta}{2} \|\mathbf{h}(\mathbf{x})\|^2,$$

which corresponds to an **equivalent problem** of (1):

$$f^* := \min_{\mathbf{x} \in X} f(\mathbf{x}) + \frac{\beta}{2} \|\mathbf{h}(\mathbf{x})\|^2 \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in X.$$

Note that, although at feasibility the additional square term in objective is **redundant**, it helps to improve strict convexity of the Lagrangian function.

The Augmented Lagrangian Dual

Now the **dual function**:

$$\phi_a(\mathbf{y}) = \min_{\mathbf{x} \in X} L_a(\mathbf{x}, \mathbf{y}); \quad (7)$$

and the **dual problem**

$$(f^* \geq) \phi_a^* := \max \phi_a(\mathbf{y}). \quad (8)$$

Note that the dual function satisfies $\frac{1}{\beta}$ -**Lipschitz** condition (see Chapter 14 of L&Y).

For the **convex optimization** case, $\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$, we have

$$\nabla^2 L_a(\mathbf{x}, \mathbf{y}) = \nabla^2 f(\mathbf{x}) + \beta(A^T A).$$

The Augmented Lagrangian Method

The **augmented Lagrangian method** (ALM) is:

Start from any $(\mathbf{x}^0 \in X, \mathbf{y}^0)$, we compute a new iterate pair

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x} \in X} L_a(\mathbf{x}, \mathbf{y}^k), \text{ and } \mathbf{y}^{k+1} = \mathbf{y}^k - \beta \mathbf{h}(\mathbf{x}^{k+1}).$$

The calculation of \mathbf{x} is used to compute the gradient vector of $\phi_a(\mathbf{y})$, which is a steepest **ascent** direction.

The method converges just like the SDM, because the dual function satisfies $\frac{1}{\beta}$ -**Lipschitz** condition.

Other SDM strategies may be adapted to update \mathbf{y} (the BB, ASDM, Conjugate, Quasi-Newton ...).

Analysis of the Augmented Lagrangian Method

Consider the convex optimization case $\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$. Since \mathbf{x}^{k+1} makes KKT condition:

$$\begin{aligned} \mathbf{0} &= \nabla f(\mathbf{x}^{k+1}) - A^T \mathbf{y}^k + \beta A^T (A\mathbf{x}^{k+1} - \mathbf{b}) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T (\mathbf{y}^k - \beta(A\mathbf{x}^{k+1} - \mathbf{b})) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T \mathbf{y}^{k+1}, \end{aligned}$$

we only need to be concerned about whether or not $\|A\mathbf{x}^k - \mathbf{b}\|$ converges to zero and how fast it converges. First, from the convexity of $f(\mathbf{x})$, we have

$$\begin{aligned} \mathbf{0} &\leq (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k))^T (\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (-A^T \mathbf{y}^{k+1} + A^T \mathbf{y}^k)^T (\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (\mathbf{y}^{k+1} - \mathbf{y}^k)^T (A\mathbf{x}^{k+1} - A\mathbf{x}^k) \\ &= -\beta(A\mathbf{x}^{k+1} - \mathbf{b})(A\mathbf{x}^{k+1} - \mathbf{b} - (A\mathbf{x}^k - \mathbf{b})), \end{aligned}$$

which implies that $\|A\mathbf{x}^{k+1} - \mathbf{b}\| \leq \|A\mathbf{x}^k - \mathbf{b}\|$, that is, the error is **non-increasing**.

Again, from the convexity, we have

$$\begin{aligned}
 0 &\leq (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*))^T (\mathbf{x}^{k+1} - \mathbf{x}^*) \\
 &= (A^T \mathbf{y}^{k+1} - A^T \mathbf{y}^*)^T (\mathbf{x}^{k+1} - \mathbf{x}^*) \\
 &= (\mathbf{y}^{k+1} - \mathbf{y}^*)^T (A\mathbf{x}^{k+1} - A\mathbf{x}^*) = (\mathbf{y}^{k+1} - \mathbf{y}^*)^T (A\mathbf{x}^{k+1} - \mathbf{b}) \\
 &= \frac{1}{\beta} (\mathbf{y}^{k+1} - \mathbf{y}^*)^T (\mathbf{y}^k - \mathbf{y}^{k+1}).
 \end{aligned}$$

Thus, from the positivity of the cross product, we have

$$\begin{aligned}
 \|\mathbf{y}^k - \mathbf{y}^*\|^2 &= \|\mathbf{y}^k - \mathbf{y}^{k+1} + \mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
 &\geq \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
 &= \beta \|A\mathbf{x}^{k+1} - \mathbf{b}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2.
 \end{aligned}$$

Sum up from 0 to k of the inequality we have

$$\begin{aligned}
 \|\mathbf{y}^0 - \mathbf{y}^*\|^2 &\geq \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 + \beta \sum_{l=0}^k \|A\mathbf{x}^{l+1} - \mathbf{b}\|^2 \\
 &\geq \beta \sum_{l=0}^k \|A\mathbf{x}^{l+1} - \mathbf{b}\|^2 \\
 &\geq (k+1)\beta \|A\mathbf{x}^{k+1} - \mathbf{b}\|^2.
 \end{aligned}$$

The Alternating Direction Method with Multipliers

For the ADMM method, we consider **structured problem**

$$\min f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \quad \text{s.t.} \quad A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b}.$$

Consider

$$L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) - \mathbf{y}^T (A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}) + \frac{\beta}{2} \|A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}\|^2.$$

Then, for any given $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{y}^k)$, we compute a new iterate

$$\begin{aligned} \mathbf{x}_1^{k+1} &= \arg \min_{\mathbf{x}_1} L(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{y}^k), \\ \mathbf{x}_2^{k+1} &= \arg \min_{\mathbf{x}_2} L(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{y}^k), \\ \mathbf{y}^{k+1} &= \mathbf{y}^k - \beta (A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} - \mathbf{b}). \end{aligned}$$

Again, we can prove that the iterates converge with the same speed.

The ADMM method resembles the **Block Coordinate Descent (BCD)** Method ...

The ADMM with Three Blocks

What about ADMM for

$$\min f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + f_3(\mathbf{x}_3) \quad \text{s.t.} \quad A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 = \mathbf{b},$$

where the Lagrangian function

$$\begin{aligned} L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}) = & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + f_3(\mathbf{x}_3) - \mathbf{y}^T (A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 - \mathbf{b}) \\ & + \frac{\beta}{2} \|A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 - \mathbf{b}\|^2. \end{aligned}$$

Then, for any given $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k, \mathbf{y}^k)$, we compute a new iterate

$$\begin{aligned} \mathbf{x}_1^{k+1} &= \arg \min_{\mathbf{x}_1} L(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{x}_3^k, \mathbf{y}^k), \\ \mathbf{x}_2^{k+1} &= \arg \min_{\mathbf{x}_2} L(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{x}_3^k, \mathbf{y}^k), \\ \mathbf{x}_3^{k+1} &= \arg \min_{\mathbf{x}_3} L(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3, \mathbf{y}^k), \\ \mathbf{y}^{k+1} &= \mathbf{y}^k - \beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}). \end{aligned}$$

Does it Converge?

Not easy to analyze the convergence: the operator theory for the ADMM cannot be directly extended to the ADMM with three blocks, since the proof for two blocks **breaks down** for three blocks.

Existing results for convergence:

- **Strong convexity**; plus carefully select β in a specific range.
- Other restricted conditions on the problem, and take a **sufficiently smaller** step-size factor $1 > \gamma > 0$ in dual update

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}).$$

- Various post **correction steps**, which are costly.

But, these did not answer the open question whether or not the **direct extension** of multi-block ADMM converges under the original simple convexity assumption.

The Direct Extension does Not Work

Theorem 1 *There existing an example where the direct extension of ADMM of three blocks is not necessarily convergent for any choice of β . Moreover, for any randomly generated initial point, ADMM diverges with probability one.*

The problem with **unique solution** $\mathbf{x}^* = \mathbf{0}$:

$$\min \quad 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 \quad \text{s.t.} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0},$$

Does the smaller step-size ($1 > \gamma > 0$) dual update work? Answer: it remains divergent when solving

$$\min \quad 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 \quad \text{s.t.} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 + \gamma \\ 1 & 1 + \gamma & 1 + \gamma \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0},$$

The Algorithmic Mapping is Not Contracting

The ADMM with $\beta = 1$ is a **linear matrix mapping**

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 0 & 0 & 0 & 0 \\ 5 & 7 & 9 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -4 & -5 & 1 & 1 & 1 \\ 0 & 0 & -7 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^k \\ \mathbf{y}^k \end{pmatrix}.$$

which can be reduced to

$$\begin{pmatrix} x_2^{k+1} \\ x_3^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = M \begin{pmatrix} x_2^k \\ x_3^k \\ \mathbf{y}^k \end{pmatrix},$$

where

$$M = \frac{1}{162} \begin{pmatrix} 144 & -9 & -9 & -9 & 18 \\ 8 & 157 & -5 & 13 & -8 \\ 64 & 122 & 122 & -58 & -64 \\ 56 & -35 & -35 & 91 & -56 \\ -88 & -26 & -26 & -62 & 88 \end{pmatrix}.$$

But the **spectral radius** of the matrix, $\rho(M) = 1.0087 > 1$, which implies that the mapping is not a **contraction**.

Multi-Block Problems and ADMM

In general, consider a convex optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in R^N} \quad & f_1(\mathbf{x}_1) + \dots + f_n(\mathbf{x}_n), \\ \text{s.t.} \quad & A\mathbf{x} := A_1\mathbf{x}_1 + \dots + A_n\mathbf{x}_n = \mathbf{b}, \\ & \mathbf{x}_i \in \mathcal{X}_i \subset R^{d_i}, i = 1, \dots, n. \end{aligned} \tag{9}$$

$$L(\mathbf{x}_1, \dots, \mathbf{x}_n; \mathbf{y}) = \sum_i f_i(x_i) - \mathbf{y}^T \left(\sum_i A_i \mathbf{x}_i - \mathbf{b} \right) + \frac{\beta}{2} \left\| \sum_i A_i \mathbf{x}_i - \mathbf{b} \right\|^2$$

The direct **Cyclic Extension** Multi-block ADMM:

$$\begin{aligned} \mathbf{x}_1 &\longleftarrow \arg \min_{\mathbf{x}_1 \in \mathcal{X}_1} L(\mathbf{x}_1, \dots, \mathbf{x}_n; \mathbf{y}), \\ &\vdots \\ \mathbf{x}_n &\longleftarrow \arg \min_{\mathbf{x}_n \in \mathcal{X}_n} L(\mathbf{x}_1, \dots, \mathbf{x}_n; \mathbf{y}), \\ \mathbf{y} &\longleftarrow \mathbf{y} - \beta(A\mathbf{x} - \mathbf{b}), \end{aligned}$$

Randomly Permuted ADMM

Random-Permuted ADMM (RP-ADMM): in each round, draw a random permutation $\sigma = (\sigma(1), \dots, \sigma(n))$ of $\{1, \dots, n\}$, and use the

Update Order : $\mathbf{x}_{\sigma(1)} \rightarrow \mathbf{x}_{\sigma(2)} \rightarrow \dots \rightarrow \mathbf{x}_{\sigma(n)} \rightarrow \mathbf{y}$.

- This is equivalent to a random **sample without replacement** so it costs nothing.
- Interpretation: Force “**absolute fairness**” among blocks.
- Simulation Test Result on solving linear equations: always converges!

Any theory behind the success?

We produced a **positive result** for ADMM on solving the system of linear equations.

Random Permuted ADMM for Linear Systems

Consider solving a **nonsingular square system** of linear equations ($f_i = 0, \forall i$).

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^N} \quad & 0, \\ \text{s.t.} \quad & A_1 \mathbf{x}_1 + \cdots + A_n \mathbf{x}_n = \mathbf{b}, \end{aligned}$$

RP-ADMM generates \mathbf{z}^k , an r.v., depending on

$$\boldsymbol{\xi}_k = (\sigma_1, \dots, \sigma_k), \quad \mathbf{z}^i = M_{\sigma_i} \mathbf{z}^{i-1}, \quad i = 1, \dots, k,$$

where σ_i is the random permutation at i -th round.

Denote the **expected iterate** $\boldsymbol{\phi}^k := E_{\boldsymbol{\xi}_k}(\mathbf{z}^k)$

Theorem 2 *The expected output converges to the unique solution of the linear system equations any integer $N \geq 1$.*

Remark: Expected convergence \neq convergence, but is a strong evidence for convergence for solving most problems, e.g., when iterates are bounded.

The Average Mapping is a Contraction

- The update equation of RP-ADMM is

$$\mathbf{z}^{k+1} = M_{\sigma} \mathbf{z}^k,$$

where $M_{\sigma} \in \mathbb{R}^{2N \times 2N}$ depend on σ .

- Define the expected update matrix as

$$M = E_{\sigma}(M_{\sigma}) = \frac{1}{n!} \sum_{\sigma} M_{\sigma}.$$

Theorem 3 The spectral radius of M , $\rho(M)$, is strictly less than 1 for any integer $N \geq 1$.

Remark: For A in the divergence example, $\rho(M_{\sigma}) > 1$ for any σ

- Averaging Helps, a lot.

RP-ADMM for Linear Constrained Convex QP

In general, consider a convex quadratic optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^N} \quad & \mathbf{c}_1^T \mathbf{x}_1 + \dots + \mathbf{c}_n^T \mathbf{x}_n + \frac{1}{2} \mathbf{x}^T Q \mathbf{x}, \\ \text{s.t.} \quad & A \mathbf{x} := A_1 \mathbf{x}_1 + \dots + A_n \mathbf{x}_n = \mathbf{b}. \end{aligned} \tag{10}$$

Theorem 4 *Under some technical assumptions, the expected output of randomly permuted ADMM converges to the solution of the original problem for any integer $N \geq 1$.*

Direct Application of ADMM to Dual Linear Programming I

Consider the dual LP

$$\begin{aligned} &\text{maximize}_{(\mathbf{y}, \mathbf{s})} && \mathbf{b}^T \mathbf{y} \\ &\text{s.t.} && A^T \mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{s} \geq \mathbf{0}. \end{aligned}$$

The augmented Lagrangian function would be

$$L_a(\mathbf{y}, \mathbf{s}, \mathbf{x}) = -\mathbf{b}^T \mathbf{y} - \mathbf{x}^T (A^T \mathbf{y} + \mathbf{s} - \mathbf{c}) + \frac{\beta}{2} \|A^T \mathbf{y} + \mathbf{s} - \mathbf{c}\|^2,$$

where β is a positive parameter, and \mathbf{x} is the multiplier vector.

Direct Application of ADMM to Dual Linear Programming II

The ADMM for the dual is straightforward: starting from any \mathbf{y}^0 , $\mathbf{s}^0 \geq \mathbf{0}$, and multiplier \mathbf{x}^0 ,

- Update variable \mathbf{y} :

$$\mathbf{y}^{k+1} = \arg \min_{\mathbf{y}} L(\mathbf{y}, \mathbf{s}^k, \mathbf{x}^k);$$

- Update slack variable \mathbf{s} :

$$\mathbf{s}^{k+1} = \arg \min_{\mathbf{s} \geq \mathbf{0}} L(\mathbf{y}^{k+1}, \mathbf{s}, \mathbf{x}^k);$$

- Update multipliers \mathbf{x} :

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta(A^T \mathbf{y}^{k+1} + \mathbf{s}^{k+1} - \mathbf{c}).$$

Note that the updates of \mathbf{y} is a **least-squares problem** with constant matrix, and the update of \mathbf{s} has a **simple close form**. (Also note that \mathbf{x} would be non-positive at the end, since we changed maximization to minimization of the dual.)

To split \mathbf{y} into **multi blocks** and update cyclically in random order?

Direct Application of ADMM to Dual SDP I

$$\begin{aligned}
 & \text{maximize}_{(\mathbf{y}, S)} && \mathbf{b}^T \mathbf{y} \\
 & \text{s.t.} && \mathcal{A}^T \mathbf{y} + S = C, \quad (X) \\
 & && S \succeq \mathbf{0};
 \end{aligned} \tag{11}$$

The augmented Lagrangian function for problem for problem (11) is

$$L_a(\mathbf{y}, S, X) = -\mathbf{b}^T \mathbf{y} - X \bullet (\mathcal{A}^T \mathbf{y} + S - C) + \frac{\beta}{2} \|\mathcal{A}^T \mathbf{y} + S - C\|_f^2. \tag{12}$$

The ADMM cycle is straightforward: starting from any $(\mathbf{y}^0, S^0 \succeq \mathbf{0})$, and multiplier X^0 , do the iterative update:

- Update variable block $\mathbf{y} \in R^m$: $\mathbf{y}^{k+1} = \arg \min_{\mathbf{y}} L^d(\mathbf{y}, S^k, X^k)$;
- Update variable block $S \in S^n$: $S^{k+1} = \arg \min_{S \succeq \mathbf{0}} L^d(\mathbf{y}^{k+1}, S, X^k)$;
- Update multiplier block $X \in S^n$: $X^{k+1} = X^k - \beta(\mathcal{A}^T \mathbf{y}^{k+1} + S^{k+1} - C)$.

Direct Application of ADMM to Dual SDP II

- Update variable block $\mathbf{y} \in \mathbb{R}^m$: it is a linear least-squares problem or convex quadratic optimization problem, where its Hessian $\beta \mathcal{A}\mathcal{A}^T \in \mathbb{S}^m$ is fixed and only linear coefficient vector is changing, where

$$(\mathcal{A}\mathcal{A}^T)_{ij} = A_i \bullet A_j.$$

- Update variable block $S \in \mathbb{S}^n$: We face another quadratic optimization problem

$$\min_S \frac{\beta}{2} \|S - C^k\|_f^2 - X^k \bullet S, \quad \text{s.t. } S \succeq \mathbf{0}.$$

This is still special quadratic minimization: if no PSD constraint, $S = C^k + \frac{1}{\beta} X^k$. Then, the minimizer with PSD constraint is

$$S^{k+1} = \max\{0, S = C^k + \frac{1}{\beta} X^k\},$$

that is, replace all negative eigenvalues of $S = C^k + \frac{1}{\beta} X^k (= \sum_j \lambda_j \mathbf{s}_j \mathbf{s}_j^T)$ by zeros.

Does the update-order of \mathbf{y} and S makes a difference? Or in a random-order?

Extensions and Research Directions (Suggested Project #5?)

- Non-square system of linear equations – “yes”
- Non-separable convex quadratic minimization with linear equality constraints – “yes”
- Convergence w.h.p.??
- Generalize to inequality systems or convex optimization at large??
- Generalize to non-convex optimization??
- ADMM where, in every iteration, each block are randomly assembled without replacement??