MS&E 310 Course Project IV: ADMM for Linear Programming

Yinyu Ye

Updated 11/6, 2017

This project is to explore ADMM methods for solving linear programs. Consider solving the linear program

$$\begin{aligned} \text{minimize}_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b}, \quad (\mathbf{y}) \\ & \mathbf{x} \geq \mathbf{0}; \end{aligned} \tag{1}$$

or its dual

$$\begin{aligned} \text{maximize}_{\mathbf{y},\mathbf{s}} \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad (\mathbf{x}) \\ & \mathbf{s} \geq \mathbf{0}; \end{aligned} \tag{2}$$

The augmented Lagrangian function for the primal would be

$$L^p(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T(A\mathbf{x} - \mathbf{b}) + \frac{\beta}{2}\|A\mathbf{x} - \mathbf{b}\|^2, \tag{3}$$

where $\mathbf{y}$ is the Lagrange multiplier vector of equality constraints and β is a positive parameter; and the one for the dual would be

$$L^d(\mathbf{y}, \mathbf{s}, \mathbf{x}) = -\mathbf{b}^T\mathbf{y} - \mathbf{x}^T(A^T\mathbf{y} + \mathbf{s} - \mathbf{c}) + \frac{\beta}{2}\|A^T\mathbf{y} + \mathbf{s} - \mathbf{c}\|^2, \tag{4}$$

where $\mathbf{x}$ is the Lagrange multiplier vector of equality constraints of the dual.

1 ADMM for the Primal

The **Augmented Lagrangian Method (ALM)** for the primal would be: starting from any $\mathbf{x}^0 \geq 0$ and $\mathbf{y}^0$, do the iterative update:

- Update variable $\mathbf{x}$:

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

- Update multiplier $\mathbf{y}$:

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

However, the computation of new $\mathbf{x}$ is still too much work – it is a quadratic minimization over the nonnegative cone.

We now reformulate the LP problem as

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & \mathbf{c}^T\mathbf{x}_1 \\ \text{s.t.} & A\mathbf{x}_1 = \mathbf{b}, \quad (\mathbf{y}) \\ & \mathbf{x}_1 - \mathbf{x}_2 = \mathbf{0}; \quad (\mathbf{s}) \\ & \mathbf{x}_2 \geq \mathbf{0}, \end{array} \tag{5}$$

and consider the split augmented Lagrangian function:

$$L^p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = \mathbf{c}^T\mathbf{x}_1 - \mathbf{y}^T(A\mathbf{x}_1 - \mathbf{b}) - \mathbf{s}^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\beta}{2}\left(\|A\mathbf{x}_1 - \mathbf{b}\|^2 + \|\mathbf{x}_1 - \mathbf{x}_2\|^2\right). \tag{6}$$

Then the **Alternating Direction Method with Multipliers (ADMM)** would be: starting from any $\mathbf{x}_1^0$, $\mathbf{x}_2^0 \geq \mathbf{0}$, and multiplier $(\mathbf{y}^0, \mathbf{s}^0)$, do the iterative update:

- Update variable $\mathbf{x}_1$:

$$\mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1} L^p(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{y}^k);$$

- Update variable $\mathbf{x}_2$:

$$\mathbf{x}_2^{k+1} = \arg\min_{\mathbf{x}_2 \geq \mathbf{0}} L^p(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{y}^k);$$

- Update multipliers $\mathbf{y}$ and $\mathbf{s}$:

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \beta(A\mathbf{x}_1^{k+1} - \mathbf{b}) \quad \text{and} \quad \mathbf{s}^{k+1} = \mathbf{s}^k - \beta(\mathbf{x}_1^{k+1} - \mathbf{x}_2^{k+1}).$$

You may now find out that the updates of $\mathbf{x}_1$ and $\mathbf{x}_2$ become much easier! The update of $\mathbf{x}_1$ is a unconstrained quadratic minimization; and the update of $\mathbf{x}_2$, although still QP over the nonnegative cone, has a simple close form.

Question 1: Write out the explicit formula for updating of $\mathbf{x}_1$ and $\mathbf{x}_2$. Implement the Primal ADMM in your favorite language or platform, and try it on some LP problems. How does it perform?

Let $A' = (AA^T)^{-1/2}A$ and $\mathbf{b}' = (AA^T)^{-1/2}\mathbf{b}$, and consider

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}_1,\mathbf{x}_2} & \mathbf{c}^T\mathbf{x}_1 \\ \text{s.t.} & A'\mathbf{x}_1 = \mathbf{b}' \\ & \mathbf{x}_1 - \mathbf{x}_2 = \mathbf{0}; \\ & \mathbf{x}_2 \geq \mathbf{0}, \end{array} \tag{7}$$

This problem is equivalent to the original problem but the constraint matrix is preconditioned. Apply the Primal ADMM and try it on the preconditioned LP formulation (7), and compare its performance with that on solving (5).

2 ADMM for the Dual

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

- Update variable $\mathbf{y}$:
$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} L^d(\mathbf{y}, \mathbf{s}^k, \mathbf{x}^k);$$
- Update slack variable $\mathbf{s}$:
$$\mathbf{s}^{k+1} = \arg\min_{\mathbf{s}\geq\mathbf{0}} L^d(\mathbf{y}^{k+1}, \mathbf{s}, \mathbf{x}^k);$$
- 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 since we changed maximization to minimization of the dual.

Question 2: Write out the explicit formula for updating of $\mathbf{y}$ and $\mathbf{s}$. Implement the Dual ADMM in your favorite language or platform, and try it on some LP problems. How does it perform?

Again, try your implementation on solving the dual of the preconditioned formulation (7).

3 Interior-Point ADMM

Now solving the linear program with the logarithmic barrier function

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}} & \mathbf{c}^T\mathbf{x} - \mu \sum_j \ln(x_j) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} > \mathbf{0}; \end{array} \tag{8}$$

or its dual

$$\begin{array}{ll} \text{maximize}_{\mathbf{y},\mathbf{s}} & \mathbf{b}^T\mathbf{y} + \mu \sum_j \ln(s_j) \\ \text{s.t.} & A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \ \mathbf{s} > \mathbf{0} \\ & \mathbf{x} \geq \mathbf{0}; \end{array} \tag{9}$$

where μ is a fixed positive constant.

The primal augmented Lagrangian function would be

$$L^p_\mu(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T\mathbf{x} - \mu \sum_j \ln(x_j) - \mathbf{y}^T(A\mathbf{x} - \mathbf{b}) + \frac{\beta}{2}\|A\mathbf{x} - \mathbf{b}\|^2; \tag{10}$$

and primal augmented Lagrangian function would be

$$L^d_\mu(\mathbf{y}, \mathbf{s}, \mathbf{x}) = -\mathbf{b}^T\mathbf{y} - \mu \sum_j \ln(s_j) - \mathbf{x}^T(A^T\mathbf{y} + \mathbf{s} - \mathbf{c}) + \frac{\beta}{2}\|A^T\mathbf{y} + \mathbf{s} - \mathbf{c}\|^2, \tag{11}$$

Question 3: Apply ADMM for barrier-primal (8) and barrier-dual (9). Again, you may split $\mathbf{x}$ in the primal to $\mathbf{x}_1$ and $\mathbf{x}_2$ to simplify the update. How do they perform? Again try your implementation on solving the preconditioned formulation (7) with barrier.

Now, we gradually reduced μ as an outer iteration. That is, we start some $\mu = \mu^0$ and apply the ADMM to compute an approximate optimizer, with its multiplier, for barrier-primal (8) or barrier-dual (9). Now set $\mu = \mu^1 = \gamma\mu^0$ where $0 < \gamma < 1$. Then we use the approximate optimizer and multiplier as the initial point to start ADMM for barrier-primal (8) or barrier-dual (9) with the new μ.

Question 4: Implement the Outer-Iteration process described above, and try different β and γ to see how it performs.

4 Multi-Block ADMM

Question 5: What about to further split variables $\mathbf{x}$ in the primal and/or $\mathbf{y}$ in the dual, and apply the fixed order or random permutation order in each update cycle.

More precisely, consider solving the dual and matrix $A = [A_1;\ A_2]$, $\mathbf{b} = [\mathbf{b}_1; \mathbf{b}_2]$, and $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2]$, then the augmented Lagrangian function

$$L^d(\mathbf{y}_1, \mathbf{y}_2, \mathbf{s}, \mathbf{x}) = -\mathbf{b}_1^T\mathbf{y}_1 - \mathbf{b}_2^T\mathbf{y}_2 - \mathbf{x}^T(A_1^T\mathbf{y}_1 + A_2^T\mathbf{y}_2 + \mathbf{s} - \mathbf{c}) + \frac{\beta}{2}\|A_1^T\mathbf{y}_1 + A_2^T\mathbf{y}_2 + \mathbf{s} - \mathbf{c}\|^2. \tag{12}$$

Starting from any $\mathbf{y}_1^0$, $\mathbf{y}_2^0$, $\mathbf{s}^0 \geq \mathbf{0}$, and multiplier $\mathbf{x}^0$, do the iterative update:

- Update variable $\mathbf{y}_1$:
$$\mathbf{y}_1^{k+1} = \arg\min_{\mathbf{y}_1} L^d(\mathbf{y}_1, \mathbf{y}_2^k, \mathbf{s}^k, \mathbf{x}^k);$$
- Update variable $\mathbf{y}_2$:
$$\mathbf{y}_2^{k+1} = \arg\min_{\mathbf{y}_2} L^d(\mathbf{y}_1^{k+1}, \mathbf{y}_2, \mathbf{s}^k, \mathbf{x}^k);$$

- Update slack variable $\mathbf{s}$:

$$\mathbf{s}^{k+1} = \arg\min_{\mathbf{s}\geq\mathbf{0}} L^d(\mathbf{y}_1^{k+1}, \mathbf{y}_2^{k+1}, \mathbf{s}, \mathbf{x}^k);$$

- Update multipliers $\mathbf{x}$:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta(A_1^T\mathbf{y}_1^{k+1} + A_2^T\mathbf{y}_2^{k+1} + \mathbf{s}^{k+1} - \mathbf{c}).$$

Note that the least-squares problem for each individual block $\mathbf{y}_i$ involves a smaller matrix $(A_iA_i^T)$.

One can also consider to reformulate the dual as

$$\begin{array}{lll} \text{maximize}_{\mathbf{y},\mathbf{s},\mathbf{u}_1,\mathbf{u}_2} & \mathbf{b}^T\mathbf{y} & \\ \text{s.t.} & A_1^T\mathbf{y}_1 - \mathbf{u}_1 = \mathbf{0}, & (\mathbf{v}_1) \\ & A_2^T\mathbf{y}_2 - \mathbf{u}_2 = \mathbf{0}, & (\mathbf{v}_2) \\ & \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} = \mathbf{c}, & (\mathbf{x}) \\ & \mathbf{s} \geq \mathbf{0}; & \end{array} \tag{13}$$

with the multiplier $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{x}$ for the three sets of the equality constraints. The augmented Lagrangian function becomes

$$\begin{aligned} L^d(\mathbf{y}_1, \mathbf{y}_2, \mathbf{u}_1, \mathbf{u}_2, \mathbf{s}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{x}) &= -\mathbf{b}_1^T\mathbf{y}_1 - \mathbf{b}_2^T\mathbf{y}_2 - \mathbf{v}_1^T(A_1^T\mathbf{y}_1 - \mathbf{u}_1) - \mathbf{v}_2^T(A_2^T\mathbf{y}_2 - \mathbf{u}_2) - \mathbf{x}^T(\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} - \mathbf{c}) \\ &\quad + \frac{\beta}{2}\left(\|A_1^T\mathbf{y}_1 - \mathbf{u}_1\|^2 + \|A_2^T\mathbf{y}_2 - \mathbf{u}_2\|^2 + \|\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} - \mathbf{c}\|^2\right). \end{aligned} \tag{14}$$

Note that $\mathbf{y}_i$, $i = 1, 2$, and $\mathbf{s} \geq \mathbf{0}$ can be independently and in parallel, and $\mathbf{u}_i$, $i = 1, 2$, can be updated jointly with a close form(?). This is essentially a two-block ADMM and guaranteed to be convergent.

References

[1] Chen, He, Ye, and Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Mathematical Programming, 155 (1-2), 2016, 57-79.

[2] D. Davis and W. Yin. Convergence rate analysis of several splitting schemes. http://www.math.ucla.edu/ wotaoyin/papers/convergence_rate_splitting.html

[3] D. Luenberger and Y. Ye. Linear and Nonlinear Programming. http://web.stanford.edu/class/msande310/310trialtext.pdf

[4] Sun, Luo, and Ye. On the expected convergence of randomly permuted ADMM. 2015, https://arxiv.org/abs/1503.06387

[5] Yinyu Ye. 310 Lecture Notes 14 and 15. http://web.stanford.edu/class/msande310/handout.shtml