Self Equivalence of the Alternating Direction Method of Multipliers

Wotao Yin (UCLA Math)

M. Yan and W. Yin, UCLA CAM 14-59, 2014.

Advanced Workshop at Shanghai U.

Brief history of ADMM

- Peaceman-Rachford Splitting (PRS) and Douglas-Rachford Splitting (DRS) appeared in 1950s
- For 30-40 years, they are used to solve PDEs and find $x \in C_1 \cap C_2$
- Gabay[1] established equivalence between ADM[2] and DRS on the dual
- Eckstein[3] shows DRS is equivalent to DRS on the dual, for a special case
- Eckstein and Fukushima[4] shows ADM is equivalent to ADM on the dual, for a special case $\mathbf{A}\mathbf{A}^T = I$. This result is rarely mention in the literature.

[1] Gabay. Applications of the method of multipliers to variational inequalities, 1983.

[2] ADM or ADMM = alternating direction method of multipliers, appeared in 60s and formalized in 80s

[3] Eckstien. Splitting methods for monotone operators with applications to parallel optimization, PhD thesis, 89'.

[4] Eckstein and Fukushima. Some reformulations and applications of the alternating direction, 1994.

- Recently, ADM or ADMM revived and rediscovered as Split Bregman[5]
- Recent popularity starts in the imaging (total variation), compressed sensing (ℓ_1), and parallel and distributed computing
- Many new applications are found in statistical and machine learning, matrix completion, finance, control, and decentralized optimization
- On the other hand, primal-dual algorithms become popular

[5] T.Goldstein and S.Osher. The split Bregman method for L1-regularized problems, 2009.

Overall features of ADM

- **easy to implement**
- **convergence requires very few conditions**
- **(nearly) state-of-the-art performance**
- **very versatile:**
  - as simple as alternating projection
  - can handle two or more objective terms and constraints
  - give rise to parallel and distributed algorithms for problems with big data

Original problem and its ADM

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \;\; & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{subject to } & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{b} \end{aligned} \tag{P1}$$

Algorithm 1 ADM on (P1)

initialize $\mathbf{x}_1^0$, $\mathbf{z}_1^0$, $\lambda > 0$

for $k = 0, 1, \cdots$ **do**

$$\mathbf{y}_1^{k+1} = \arg\min_{\mathbf{y}} \, g(\mathbf{y}) + (2\lambda)^{-1} \|\mathbf{A}\mathbf{x}_1^k + \mathbf{B}\mathbf{y} - \mathbf{b} + \lambda \mathbf{z}_1^k\|_2^2$$

$$\mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}} \, f(\mathbf{x}) + (2\lambda)^{-1} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}_1^{k+1} - \mathbf{b} + \lambda \mathbf{z}_1^k\|_2^2$$

$$\mathbf{z}_1^{k+1} = \mathbf{z}_1^k + \lambda^{-1}(\mathbf{A}\mathbf{x}_1^{k+1} + \mathbf{B}\mathbf{y}_1^{k+1} - \mathbf{b})$$

end for

- Similar to the augmented Lagrangian method
- Deal with $(f, \mathbf{A})$ and $(g, \mathbf{B})$ only one at a time

Master-slave problem and its ADM

Define slave problems:

$$F(\mathbf{s}) := \min_{\mathbf{x}} f(\mathbf{x}) + \iota_{\{\mathbf{x}:\mathbf{A}\mathbf{x}=\mathbf{s}\}}(\mathbf{x}), \tag{1a}$$

$$G(\mathbf{t}) := \min_{\mathbf{y}} g(\mathbf{y}) + \iota_{\{\mathbf{y}:\mathbf{B}\mathbf{y}=\mathbf{b}-\mathbf{t}\}}(\mathbf{y}). \tag{1b}$$

Master formulation:

$$\begin{aligned} \underset{\mathbf{s},\mathbf{t}}{\text{minimize}} \quad & F(\mathbf{s}) + G(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{s} - \mathbf{t} = \mathbf{0}. \end{aligned} \tag{P2}$$

(P2) is equivalent to (P1): they have the same solutions $\mathbf{x}^*, \mathbf{y}^*$ and objective.

Algorithm 2 ADM on (P2)

initialize $\mathbf{s}^0$, $\mathbf{z}_2^0$, $\lambda > 0$

for $k = 0, 1, \cdots$ **do**

$\quad \mathbf{t}^{k+1} = \arg\min_{\mathbf{t}} G(\mathbf{t}) + (2\lambda)^{-1}\|\mathbf{s}^k - \mathbf{t} + \lambda \mathbf{z}_2^k\|_2^2$

$\quad \mathbf{s}^{k+1} = \arg\min_{\mathbf{s}} F(\mathbf{s}) + (2\lambda)^{-1}\|\mathbf{s} - \mathbf{t}^{\mathbf{k}+1} + \lambda \mathbf{z}_2\|_2^2$

$\quad \mathbf{z}_2^{k+1} = \mathbf{z}_2^k + \lambda^{-1}(\mathbf{s}^{k+1} - \mathbf{t}^{k+1})$

end for

Lagrange dual problems of (P1) and (P2)

ADM is applied to **both primal and dual** problems.

Dual ADM examples: YALL1 package[6], ADM for ℓ_1-ℓ_1 model[7], traffic equilibrium problem[8]

Lagrange dual of (P1) (where * means convex conjugate or adjoint):

$$\underset{\mathbf{v}}{\text{minimize}} \quad f^*(-\mathbf{A}^*\mathbf{v}) + g^*(-\mathbf{B}^*\mathbf{v}) + \langle \mathbf{v}, \mathbf{b} \rangle.$$

Apply variable splitting gives the ADM-ready reformulation:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & f^*(-\mathbf{A}^*\mathbf{u}) + (g^*(-\mathbf{B}^*\mathbf{v}) + \langle \mathbf{v}, \mathbf{b} \rangle) \\ \text{subject to} \quad & \mathbf{u} - \mathbf{v} = \mathbf{0}. \end{aligned} \tag{D1}$$

[6] J.Yang and Y.Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, 2011.

[7] Y.Xiao, H.Zhu, S.-Y. Wu. Primal and dual alternating direction algorithms for l1-l1-norm minimization problems in compressive sensing, 2013.

[8] Primal: Fukushima'96; dual: Gabay'83.

Similarly, the ADM-ready formulation of (P2)'s Lagrange dual:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & F^*(-\mathbf{u}) + G^*(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{u} - \mathbf{v} = \mathbf{0}. \end{aligned} \tag{D2}$$

Problems (D1) and (D2) are equivalent through the identities:

$$\begin{aligned} F^*(-\mathbf{u}) &= f^*(-\mathbf{A}^*\mathbf{u}) \\ G^*(\mathbf{v}) &= g^*(-\mathbf{B}^*\mathbf{v}) + \langle \mathbf{v}, \mathbf{b} \rangle. \end{aligned}$$

ADM on the Lagrange dual

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & F^*(-\mathbf{u}) + G^*(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{u} - \mathbf{v} = \mathbf{0}. \end{aligned} \tag{D2}$$

Algorithm 3 ADM on (D1)/(D2)

initialize $\mathbf{u}^0$, $\mathbf{z}_3^0$, $\lambda > 0$

for $k = 0, 1, \cdots$ **do**

$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \ G^*(\mathbf{v}) + \frac{\lambda}{2}\|\mathbf{u}^k - \mathbf{v} + \lambda^{-1}\mathbf{z}_3^k\|_2^2$

$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \ F^*(-\mathbf{u}) + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{v}^{\mathbf{k}+1} + \lambda^{-1}\mathbf{z}_3\|_2^2$

$\mathbf{z}_3^{k+1} = \mathbf{z}_3^k + \lambda(\mathbf{u}^{k+1} - \mathbf{v}^{k+1})$

end for

Self-equivalence theorem

Theorem

$$\boxed{\text{ADM on (P1)}} \Longleftrightarrow \boxed{\text{ADM on (P2)}} \Longleftrightarrow \boxed{\text{ADM on (D1)/(D2)}}$$

Suppose Algorithms 1-3 initialize $\mathbf{A}\mathbf{x}_1^0 = \mathbf{s}^0 = \mathbf{z}_3^0$ *and* $\mathbf{z}_1^0 = \mathbf{z}_2^0 = \mathbf{u}^0$ *and use the same* λ*. Then, from the iterates of any algorithm, the iterates of the others can be explicitly recovered.*

Proof.

The equivalence between ADMs on (P1) and (P2) follows from definitions. The equivalence between ADMs on (P2) and (D1)/(D2) follows from algebraic manipulation and the property: $x \in \partial f(y) \Longleftrightarrow y \in \partial f^*(x)$ for proper, closed, convex function f. □

Remarks

- ADM essentially applies only to the master problem

$$\underset{\mathbf{s},\mathbf{t}}{\text{minimize}}\, F(\mathbf{s}) + G(\mathbf{t}) \quad \text{subject to} \;\; \mathbf{s} - \mathbf{t} = \mathbf{0},$$

  which is an exchange problem and a zero-sum convex game.
  $(f, \mathbf{A})$ and $(g, \mathbf{B})$ are only dealt in the subproblems, not by ADM.
  The often-seen ADM, Algorithm 1, has obscured this fact.

- ADMs on (P2) and (D2) have the primal-dual mapping:

$$\mathbf{u}^k = \mathbf{z}_2^k, \qquad \mathbf{z}_3^k = \mathbf{s}^k.$$

  The later updated variable, $\mathbf{u}$ or $\mathbf{s}$, is the dual variable in the dual ADM.

- Penalty parameter λ in the primal ADM becomes λ^{-1} in the dual ADM. It balances primal-dual updates.

- The perfect symmetry between primal and dual ADMs suggest that ADM is a primal-dual algorithm to a saddle-point formulation (come later ...)

ADM with the swapped $\mathbf{x}/\mathbf{y}$-update order

Algorithm 4 "The other ADM" on (P1)

initialize $\mathbf{x}_1^0$, $\mathbf{z}_1^0$, $\lambda > 0$
for $k = 0, 1, \cdots$ **do**
  $\mathbf{x}_4^{k+1} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + (2\lambda)^{-1}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}_4^k - \mathbf{b} + \lambda\mathbf{z}_1^k\|_2^2$
  $\mathbf{y}_4^{k+1} = \arg\min_{\mathbf{y}} g(\mathbf{y}) + (2\lambda)^{-1}\|\mathbf{A}\mathbf{x}_4^{k+1} + \mathbf{B}\mathbf{y} - \mathbf{b} + \lambda\mathbf{z}_1^k\|_2^2$
  $\mathbf{z}_4^{k+1} = \mathbf{z}_1^k + \lambda^{-1}(\mathbf{A}\mathbf{x}_4^{k+1} + \mathbf{B}\mathbf{y}_4^{k+1} - \mathbf{b})$
end for

The only difference between Algorithms 1 (ADM) and 4 (the other ADM):

- Algorithm 1 updates $\mathbf{y}$, then $\mathbf{x}$
- Algorithm 4 updates $\mathbf{x}$, then $\mathbf{y}$

In general, they produce different iterates, but there are exceptions.

Affine proximal mapping

Definition

A mapping T is affine if, for any $\mathbf{r}_1$ and $\mathbf{r}_2$,

$$T\left(\frac{1}{2}\mathbf{r}_1 + \frac{1}{2}\mathbf{r}_2\right) = \frac{1}{2}T\mathbf{r}_1 + \frac{1}{2}T\mathbf{r}_2.$$

Proposition

Let G be a proper, closed, convex function. The following statements are equivalent:

1. $\mathbf{prox}_{G(\cdot)}$ *is affine;*
2. $\mathbf{prox}_{\lambda G(\cdot)}$ *is affine for $\lambda > 0$;*
3. $a\mathbf{prox}_{G(\cdot)} \circ b\mathbf{I} + c\mathbf{I}$ *is affine for any scalars a, b and c;*
4. $\mathbf{prox}_{G^*(\cdot)}$ *is affine;*
5. G *is* ***convex quadratic*** *(or, affine or constant) and has an* ***affine domain*** *(either $\mathcal{G}$ or the intersection of hyperplanes in $\mathcal{G}$).*

If function g obeys Part 5, then G defined in (1b) satisfies Part 5, too.

Order-swapping equivalence

Theorem

1. *Assume* $\mathbf{prox}_G$ *is affine. Given the iterates of "the other ADM", if* $\mathbf{z}_4^0 \in \partial G(\mathbf{b} - \mathbf{B}\mathbf{y}_4^0)$*, then the iterates of ADM can be recovered as*

$$\mathbf{x}_1^k = \mathbf{x}_4^{k+1}, \qquad \mathbf{z}_1^k = \mathbf{z}_4^k + \lambda^{-1}(\mathbf{A}\mathbf{x}_4^{k+1} + \mathbf{B}\mathbf{y}_4^k - \mathbf{b}).$$

2. *Assume* $\mathbf{prox}_F$ *is affine. Given the iterates of ADM, if* $-\mathbf{z}_1^0 \in \partial G(\mathbf{A}\mathbf{x}_1^0)$ *, then the iterates of "the other ADM" can be recovered as*

$$\mathbf{y}_4^k = \mathbf{y}_1^{k+1}, \qquad \mathbf{z}_4^k = \mathbf{z}_1^k + \lambda^{-1}(\mathbf{A}\mathbf{x}_1^{k+1} + \mathbf{B}\mathbf{y}_1^{k+1} - \mathbf{b}).$$

Proof. Part 1 is based on algebraic manipulations, where a key step needs:

$$\mathbf{prox}_{\lambda G}(2\mathbf{r}_1 - \mathbf{r}_2) = 2\mathbf{prox}_{\lambda G}\mathbf{r}_1 - \mathbf{prox}_{\lambda G}\mathbf{r}_2,$$

which is equivalent to $\mathbf{prox}_G$ being affine.

Same on $\mathbf{prox}_{\lambda F}$ for Part 2.

Remark: Condition $\mathbf{z}_4^0 \in \partial G(\mathbf{b} - \mathbf{B}\mathbf{y}_4^0)$ can be removed by adding 1 to all the iterates of $\mathbf{x}_4, \mathbf{y}_4, \mathbf{z}_4$ since $\mathbf{z}_4^1 \in \partial G(\mathbf{b} - \mathbf{B}\mathbf{y}_4^1)$ always holds.

Same for Part 2.

Saddle-point formulation and its algorithm

The original problem (P1) is equivalent to

$$\min_{\mathbf{y}} \max_{\mathbf{u}} \; g(\mathbf{y}) + \langle \mathbf{u}, \mathbf{B}\mathbf{y} - \mathbf{b} \rangle - f^*(-\mathbf{A}^*\mathbf{u}).$$

Algorithm 5 Primal-dual saddle-point algorithm

initialize $\mathbf{u}^0$, $\mathbf{u}^{-1}$, $\mathbf{y}^0$, $\lambda > 0$

for $k = 0, 1, \cdots$ **do**

$\bar{\mathbf{u}}^k = 2\mathbf{u}^k - \mathbf{u}^{k-1}$

$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} \; g(\mathbf{y}) + (2\lambda)^{-1}\|\mathbf{B}\mathbf{y} - \mathbf{B}\mathbf{y}^k + \lambda\bar{\mathbf{u}}^k\|_2^2$

$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \; f^*(-\mathbf{A}^*\mathbf{u}) + \lambda/2\|\mathbf{u} - \mathbf{u}^k - \lambda^{-1}(\mathbf{B}\mathbf{y}^{k+1} - \mathbf{b})\|_2^2$

end for

- If $\mathbf{B} = \mathbf{I}$, then it is equivalent to the primal-dual algorithm [9]; the paper also noted the equivalence between it and ADM.

[9] Chambolle and Pock.

ADM equivalence to the primal-dual algorithm

Theorem

Suppose in Algorithms 1 and 5, the initial iterates satisfy $\mathbf{z}_1^0 = \mathbf{u}^0$ *and* $\mathbf{A}\mathbf{x}^0 = \lambda(\mathbf{u}^0 - \mathbf{u}^{-1}) + \mathbf{b} - \mathbf{B}\mathbf{y}^0$*. Then Algorithms 1 and 5 are equivalent by*

$$\mathbf{A}\mathbf{x}^k = \lambda(\mathbf{u}^k - \mathbf{u}^{k-1}) + \mathbf{b} - \mathbf{B}\mathbf{y}^k, \qquad \mathbf{z}_1^k = \mathbf{u}^k,$$

for $k \geq 0$.

Application: total variation image processing

- Rudin-Osher-Fatemi model:

$$\underset{x \in BV(\Omega)}{\text{minimize}} \int_{\Omega} |Dx| + \frac{\alpha}{2}\|x - b\|_2^2,$$

  which recovers a piece-wise constant (thus noise-free) image from a noisy observation b.

- Discretization: $\|\nabla \mathbf{x}\|_{2,1} = \sum_{ij} |(\nabla \mathbf{x})_{ij}|$, where $|\cdot|$ is 2-norm.
- ADM-ready form:

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \ \|\mathbf{y}\|_{2,1} + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{b}\|_2^2, \quad \text{subject to } \mathbf{y} - \nabla \mathbf{x} = \mathbf{0}.$$

- Chambolle's dual form:

$$\underset{\mathbf{v},\mathbf{u}}{\text{minimize}} \ \frac{1}{2\alpha}\|\mathsf{div}\ \mathbf{u} + \alpha \mathbf{b}\|_2^2 + \iota_{\{\|\cdot\|_{2,\infty} \le 1\}}(\mathbf{v}), \quad \text{subject to } \mathbf{u} - \mathbf{v} = \mathbf{0},$$

  where $\|\mathbf{v}\|_{2,\infty} = \max_{ij} |(\mathbf{v})_{ij}|$.

Equivalent algorithms

1. Algorithm 1 (primal ADM) is

$$\begin{aligned}
\mathbf{x}_1^{k+1} =& \arg\min_{\mathbf{x}} \frac{\alpha}{2}\|\mathbf{x}-\mathbf{b}\|_2^2 + (2\lambda)^{-1}\|\nabla\mathbf{x} - \mathbf{y}_1^k + \lambda\mathbf{z}_1^k\|_2^2, \\
\mathbf{y}_1^{k+1} =& \arg\min_{\mathbf{y}} \|\mathbf{y}\|_{2,1} + (2\lambda)^{-1}\|\nabla\mathbf{x}_1^{k+1} - \mathbf{y} + \lambda\mathbf{z}_1^k\|_2^2, \\
\mathbf{z}_1^{k+1} =& \mathbf{z}_1^k + \lambda^{-1}(\nabla\mathbf{x}_1^{k+1} - \mathbf{y}_1^{k+1}).
\end{aligned}$$

2. Algorithm 3 (dual ADM) is

$$\begin{aligned}
\mathbf{u}_2^{k+1} =& \arg\min_{\mathbf{u}} \frac{1}{2\alpha}\|\mathsf{div}\ \mathbf{u} + \alpha\mathbf{b}\|_2^2 + \frac{\lambda}{2}\|\mathbf{v}_2^k - \mathbf{u} + \lambda^{-1}\mathbf{z}_2^k\|_2^2, \\
\mathbf{v}_2^{k+1} =& \arg\min_{\mathbf{v}} \iota_{\{\|\cdot\|_{2,\infty}\le 1\}}(\mathbf{v}) + \frac{\lambda}{2}\|\mathbf{v} - \mathbf{u}_2^{k+1} + \lambda^{-1}\mathbf{z}_2^k\|_2^2, \\
\mathbf{z}_2^{k+1} =& \mathbf{z}_2^k + \lambda(\mathbf{v}_2^{k+1} - \mathbf{u}_2^{k+1}).
\end{aligned}$$

3. Algorithm 5 (primal-dual) is

$$
\begin{aligned}
\bar{\mathbf{v}}_3^k =& 2\mathbf{v}_3^k - \mathbf{v}_3^{k-1} \\
\mathbf{x}_3^{k+1} =& \arg\min_{\mathbf{x}} \frac{\alpha}{2}\|\mathbf{x} - \mathbf{b}\|_2^2 + \langle \bar{\mathbf{v}}_3^k, \nabla \mathbf{x} \rangle + (2\lambda)^{-1}\|\nabla \mathbf{x} - \nabla \mathbf{x}_3^k\|_2^2, \\
\mathbf{v}_3^{k+1} =& \arg\min_{\mathbf{v}} \iota_{\{\mathbf{v}:\|\mathbf{v}\|_{2,\infty} \le 1\}} - \langle \mathbf{v}, \nabla \mathbf{x}_3^{k+1} \rangle + \frac{\lambda}{2}\|\mathbf{v} - \mathbf{v}^k\|_2^2.
\end{aligned}
$$

4. Algorithm 4 (primal ADM with update order swapped) is

$$
\begin{aligned}
\mathbf{y}_4^{k+1} =& \arg\min_{\mathbf{y}} \|\mathbf{y}\|_{2,1} + (2\lambda)^{-1}\|\nabla \mathbf{x}_4^k - \mathbf{y} + \lambda \mathbf{z}_4^k\|_2^2, \\
\mathbf{x}_4^{k+1} =& \arg\min_{\mathbf{x}} \frac{\alpha}{2}\|\mathbf{x} - \mathbf{b}\|_2^2 + (2\lambda)^{-1}\|\nabla \mathbf{x} - \mathbf{y}_4^{k+1} + \lambda \mathbf{z}_4^k\|_2^2, \\
\mathbf{z}_4^{k+1} =& \mathbf{z}_4^k + \lambda^{-1}(\nabla \mathbf{x}_4^{k+1} - \mathbf{y}_4^{k+1}).
\end{aligned}
$$

Corollary

Let $\mathbf{x}_4^0 = \mathbf{b} + \alpha^{-1}\text{div } \mathbf{z}_4^0$. *If initialize* $\mathbf{y}_1^0 = -\mathbf{z}_2^0 = \nabla \mathbf{x}_3^0 - \lambda(\mathbf{v}_3^0 - \mathbf{v}_3^{-1}) = \mathbf{y}_4^1$ *and* $\mathbf{z}_1^0 = \mathbf{v}_2^0 = \mathbf{v}_3^0 = \mathbf{z}_4^0 + \lambda^{-1}(\nabla \mathbf{x}_4^0 - \mathbf{y}_4^1)$. *Then for* $k \geq 1$, *we have the following equivalence between the iterations of the four algorithms:*

$$\begin{aligned} \mathbf{y}_1^k &= -\mathbf{z}_2^k &&= \nabla \mathbf{x}_3^k - \lambda(\mathbf{v}_3^k - \mathbf{v}_3^{k-1}) &&= \mathbf{y}_4^{k+1}, \\ \mathbf{z}_1^k &= \mathbf{v}_2^k &&= \mathbf{v}_3^k &&= \mathbf{z}_4^k + \lambda^{-1}(\nabla \mathbf{x}_4^k - \mathbf{y}_4^{k+1}). \end{aligned}$$

Conclusions

Concluding remarks:

- ADM is a *primal-dual algorithm* that is *self-dual*, though seemingly a variant of the *augmented Lagrangian method*.
- When one of function is quadratic, the update order can be swapped.
- This work bridges the studies of ADM and primal-dual algorithms.

Open questions: The equivalence and improved understanding for

- Variants of ADM.
- Multiple-block extension of ADM.

Also, apply the extensions of ADM to primal-dual algorithms in a parallel way.