Solving Nonnegative Least Squares by ADMM

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

manshun.ang@umons.ac.be   Homepage: angms.science

First draft : July 9, 2019
Last update : February 19, 2020

NNLS and ADMM

Nonn**e**gative **L**east **S**quares (NNLS) : given $\mathbf{Q} \in \mathbb{R}^{m \times n}$, $\mathbf{p} \in \mathbb{R}^{m}$, find $\mathbf{x} \in \mathbb{R}^{n}_{+}$ by solving

$$(\mathcal{P}) \;:\; \underset{\mathbf{x} \geq 0}{\operatorname{argmin}} \; f(\mathbf{x}) := \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2.$$

Note. This problem is convex.

Alternating **D**irection **M**ethod of **M**ultipliers (**ADMM**) can be used to solve the convex problem

$$\min_{\mathbf{x},\mathbf{y}} \; f_1(\mathbf{x}) + f_2(\mathbf{y}) \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c} \tag{1}$$

where f_1, f_2 are both convex.

Question : is it possible to fit NNLS into model (1)? If yes, then how?

Fitting NNLS problem into ADMM framework

We fit NNLS into the model (1) as follows

- Let f_1 be the objective function of the NNLS.
- Let f_2 be the indicator function of the nonnegative orthant.
- Let $\mathbf{A} = \mathbf{I}$ (identity matrix).
- Let $\mathbf{B} = -\mathbf{I}$.
- Let $\mathbf{c} = \mathbf{0}$ (zero vector).

Then we have

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{Qx} - \mathbf{p}\|_2^2 + i_{\mathcal{C}}(\mathbf{y}) \text{ s.t. } \mathbf{x} = \mathbf{y}.$$

The non-negativity constraint in NNLS is represented by $i_{\mathcal{C}}$ with $\mathcal{C} = \mathbb{R}^n_+$:

$$i_{\mathcal{C}}(\mathbf{y}) = \begin{cases} 0 & \mathbf{y} \in \mathbb{R}^n_+ \\ \infty & \mathbf{y} \notin \mathbb{R}^n_+ \end{cases}$$

Although the indicator function is only applied on $\mathbf{y}$, the vector $\mathbf{y}$ is linked with $\mathbf{x}$ by the constraint $\mathbf{x} = \mathbf{y}$.

The way ADMM solves the problem

For the general problem

$$\min_{\mathbf{x},\mathbf{y}} \; f_1(\mathbf{x}) + f_2(\mathbf{y}) \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c},$$

ADMM = minimize over $\mathbf{x}$ and $\mathbf{y}$ separately (variable splitting) + update the dual variable (Lagrangian multiplier).

Why ADMM uses Lagrangian multiplier : it is a standard approach to deal with optimization problem with equality constraint.

To use ADMM in NNLS, think this way : the optimization variable $\mathbf{x}$ and the auxiliary variable $\mathbf{y}$ are the "same" variable (which is reflected by the equality $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}$ with $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = -\mathbf{I}$ and $\mathbf{c} = \mathbf{0}$). The objective function is applied on the first variable $\mathbf{x}$ while the constraint is applied on the second variable $\mathbf{y}$. ADMM splits the update of $\mathbf{x}$ and update of $\mathbf{y}$ separately, which is exactly a kind of coordinate descent. But what makes ADMM different from coordinate descent is that it has a dual variable.

Also, what makes ADMM different from Method of Lagrangian multiplier is that $\mathbf{x}$ and $\mathbf{y}$ are updated separately in ADMM. In method of Lagrangian multiplier, $\mathbf{x}$ and $\mathbf{y}$ are updated jointly.

Augmented Lagrangian function

For the general problem

$$\min_{\mathbf{x},\mathbf{y}} \; f_1(\mathbf{x}) + f_2(\mathbf{y}) \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}.$$

Augmented Lagrangian function

$$L(\mathbf{x}, \mathbf{y}, \lambda) = f_1(\mathbf{x}) + f_2(\mathbf{y}) + \langle \lambda, \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c} \rangle + \frac{\tau}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}\|^2,$$

where λ is the Lagrangian multiplier, $\tau > 0$ is a constant (the penalty parameter for the constraint).

Let $\mu = \tau^{-1}\lambda$, the scaled dual form of the Augmented Lagrangian is

$$\begin{aligned} L_\tau(\mathbf{x}, \mathbf{y}, \lambda) &= f_1(\mathbf{x}) + f_2(\mathbf{y}) + \frac{\tau}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c} + \mu\|^2 \\ &= L(\mathbf{x}, \mathbf{y}, \lambda) + \text{const.} \end{aligned}$$

ADMM algorithm

For the general problem

$$\min_{\mathbf{x},\mathbf{y}} \; f_1(\mathbf{x}) + f_2(\mathbf{y}) \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}.$$

Augmented Lagrangian (in scaled dual form)

$$L_\tau(\mathbf{x}, \mathbf{y}, \mu) = f_1(\mathbf{x}) + f_2(\mathbf{y}) + \frac{\tau}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c} + \mu\|^2$$

With starting variables $(\mathbf{x}_0, \mathbf{y}_0, \mu_0)$, ADMM loop the 3 steps

1. $\mathbf{x}_k = \arg\min_{\mathbf{x}} L_\tau(\mathbf{x}, \mathbf{y}_{k-1}, \mu_{k-1})$: minimize L w.r.t. $\mathbf{x}$
2. $\mathbf{y}_k = \arg\min_{\mathbf{y}} L_\tau(\mathbf{x}_{k-1}, \mathbf{y}, \mu_{k-1})$: minimize L w.r.t. $\mathbf{y}$
3. $\mu_k = \mu_k + (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{y}_k - \mathbf{c})$: update μ (dual variable)

ADMM algorithm for the NNLS problem

NNLS problem in ADMM form

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + i_{\mathcal{C}}(\mathbf{y}) \text{ s.t. } \mathbf{x} = \mathbf{y}.$$

The Augmented Lagrangian

$$L_\tau(\mathbf{x}, \mathbf{y}, \mu) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + i_{\mathcal{C}}(\mathbf{y}) + \frac{\tau}{2}\|\mathbf{x} - \mathbf{y} + \mu\|^2.$$

Start with $(\mathbf{x}_0, \mathbf{y}_0, \mu_0)$, ADMM algorithm loop through the 3 steps

1. $\mathbf{x}_k = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + \frac{\tau}{2}\|\mathbf{x} - \mathbf{y}_{k-1} + \mu_{k-1}\|_2^2$
2. $\mathbf{y}_k = \arg\min_{\mathbf{y}} i_{\mathcal{C}}(\mathbf{y}) + \frac{\tau}{2}\|\mathbf{x}_k - \mathbf{y} + \mu_{k-1}\|_2^2$
3. $\mu_k = \mu_k + \mathbf{x}_k - \mathbf{y}_k$

The sub-problems in step 1 and step 2 are unconstrained optimization problem with close form solution.

Close form update

Step 1 Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + \frac{\tau}{2}\|\mathbf{x} - \mathbf{y} + \mu\|^2$, then

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{Q}^\top(\mathbf{Q}\mathbf{x} - \mathbf{p}) + \tau(\mathbf{x} - \mathbf{y} + \mu)$$

Set $\nabla f(\mathbf{x}) = 0$ gives $\mathbf{x} = (\mathbf{Q}^\top\mathbf{Q} + \tau\mathbf{I})^{-1}\Big(\mathbf{Q}^\top\mathbf{p} + \tau(\mathbf{y} - \mu)\Big)$.

Note : $\mathbf{Q}^\top\mathbf{Q} + \tau\mathbf{I}$ is always positive definite for $\tau > 0$, thus it is invertible.

Step2 Let $F(\mathbf{y}) = i_{\mathcal{C}}(\mathbf{y}) + \underbrace{\frac{\tau}{2}\|\mathbf{x} - \mathbf{y} + \mu\|^2}_{f(\mathbf{y})}$, then

$$\nabla_{\mathbf{y}} f(\mathbf{y}) = -\tau(\mathbf{x} - \mathbf{y} + \mu)$$

Set $\nabla_y f(\mathbf{y}) = 0$ we have

$$\mathbf{y} = \mathbf{x} + \mu.$$

For the non-smooth part $i_{\mathcal{C}}(\mathbf{y})$, using proximal operator yields nonnegative projection :

$$\mathbf{y} = [\mathbf{x} + \mu]_+ .$$

Solving NNLS by ADMM algorithm

$$(\mathcal{P}) \;:\; \underset{\mathbf{x}\geq 0}{\operatorname{argmin}}\; f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2.$$

Algorithm 1: ADMM for NNLS

Result: A solution $\mathbf{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^n_+$, $\lambda_0 \in \mathbb{R}^n$, $\tau > 0$, $\mu_0 = \tau^{-1}\lambda_0$, $k = 1$

while *stopping condition is not met* **do**

$\quad \mathbf{x}_k = (\mathbf{Q}^\top\mathbf{Q} + \tau\mathbf{I})^{-1}\Big(\mathbf{Q}^\top\mathbf{p} + \tau(\mathbf{y}_{k-1} - \mu_{k-1})\Big)$

$\quad \mathbf{y}_k = [\mathbf{x}_k + \mu_{k-1}]_+$

$\quad \mu_k = \mu_k + \mathbf{x}_k - \mathbf{y}_k$

$\quad k = k + 1$

end

Implementation issues : constant terms should be pre-computed outside the loop.

Advantage of ADMM : flexibility

The advantage of ADMM is flexibility, which comes from step 2 :

$$\mathbf{y}_k = \arg\min_{\mathbf{y}} \; i_{\mathcal{C}}(\mathbf{y}) + \frac{\tau}{2}\|\mathbf{x}_k - \mathbf{y} + \mu_{k-1}\|^2$$

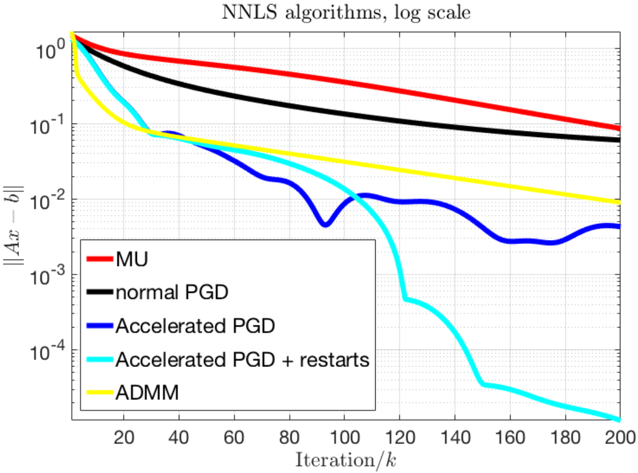
It is $i_{\mathcal{C}}(\mathbf{y})$ encode the constraint in the original optimization problem.

By replacing the set $\mathcal{C}$ to other constraint, the same algorithm framework works for other kinds of constrained least square problems :

- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2$ s.t. $\|\mathbf{x}\|_2 \leq \epsilon$
- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2$ s.t. $\|\mathbf{x}\|_1 \leq \epsilon$
- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2$ s.t. $\|\mathbf{x}\|_0 \leq \epsilon$
- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + \nu\|\mathbf{x}\|_2$
- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + \nu\|\mathbf{x}\|_1$
- $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{Q}\mathbf{x} - \mathbf{p}\|_2^2 + \nu\|\mathbf{x}\|_0$

Drawback of ADMM : slow

ADMM is famous for being slow. Furthermore, in general, the update of $\mathbf{x}$ is as expensive as solving the problem itself.



To improve it : *accelerated-ADMM*.

Last page - summary

Summary :

- ADMM algorithm for NNLS

Not discussed :

- Convergence of ADMM
- Accelerated ADMM algorithm
- Non-convex ADMM

End of document