A DEMO OF ADMM METHOD FOR 1D SINGAL RECOVERY

Liang-Jian Deng *

School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), China
Emai: liangjian.deng@uestc.edu.cn
Address: A-401, Innovation Center, UESTC

ABSTRACT

In this manuscript, we will give the details of how to implement ADMM method for 1D signal recovery.

1. INTRODUCTION

Problem (degraded signal formulation):

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ is the observed degraded signal, $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ is the true *sparse* signal (no noise, no downsampling), $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a degraded matrix/operator (downsampling), and $\boldsymbol{n} \in \mathbb{R}^{m\times 1}$ is the added *sparse* noise. **Our Goal** is to recover $\boldsymbol{x}$ from the given model (1).

Optimization model for (1):

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1 + \lambda\|\boldsymbol{x}\|_1, \tag{2}$$

To solve (2), we use ADMM algorithm to do it. The following is the detailed steps of ADMM.

1. Using two new variables $\boldsymbol{u}, \boldsymbol{v}$ to substitute $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{x}$, respectively, i.e., $\boldsymbol{u} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{v} = \boldsymbol{x}$, then get the following constrained minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{v}} \ \|\boldsymbol{u}\|_1 + \lambda\|\boldsymbol{v}\|_1, \\ &\text{s.t.,} \ \ \boldsymbol{u} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}, \ \ \boldsymbol{v} = \boldsymbol{x} \end{aligned} \tag{3}$$

2. For the constrained problem (3), we may get Augmented Lagrange Equation as follows,

$$\begin{aligned} \mathcal{L}(\boldsymbol{u},\boldsymbol{v},\boldsymbol{x},\boldsymbol{b_1},\boldsymbol{b_2}) =& \|\boldsymbol{u}\|_1 + \frac{\beta_1}{2}\|\boldsymbol{u} - (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) + \boldsymbol{b}_1\|_2^2 \\ & \lambda\|\boldsymbol{v}\|_1 + \frac{\beta_2}{2}\|\boldsymbol{v} - \boldsymbol{x} + \boldsymbol{b}_2\|_2^2, \end{aligned} \tag{4}$$

3. Now, we turn to minimize the problem (4) with multiple unknown variables, see, e.g., $\boldsymbol{u},\boldsymbol{v},\boldsymbol{x},\boldsymbol{b_1},\boldsymbol{b_2}$,

$$\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{x},\boldsymbol{b_1},\boldsymbol{b_2}} \mathcal{L}(\boldsymbol{u},\boldsymbol{v},\boldsymbol{x},\boldsymbol{b_1},\boldsymbol{b_2}) \tag{5}$$

How to compute this minimization problem with multiple unknown variables?
We use alternative direction method of multipliers (ADMM), which to compute one variable problem by fixing other variables for each time! (Details of ADMM will be illustrated in Sect. 1.1)

1.1. Details of ADMM

For the minimization of (5), we compute one variable problem by fixing other variables for each time!

Initialization: $\boldsymbol{y}$ and $\boldsymbol{A}$ known; set vectors $\boldsymbol{x}^{(0)} = \boldsymbol{b}_1^{(0)} = \boldsymbol{b}_1^{(0)} = \boldsymbol{0}$; given numbers λ, β_1 and β_2; $k = 0$

1. **$\boldsymbol{u}$-subproblem:** First, to find the $\boldsymbol{u}$-involved minimization problem from (5), i.e.,

$$\min_{\boldsymbol{u}} \ \|\boldsymbol{u}\|_1 + \frac{\beta_1}{2}\|\boldsymbol{u} - (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) + \boldsymbol{b}_1\|_2^2, \tag{6}$$

which has the following closed-form solution on $(k+1)$ iteration:

$$\boldsymbol{u}^{(k+1)} = \text{Shrink}\left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}_1^{(k)}, \frac{1}{\beta_1}\right), \tag{7}$$

where $\text{Shrink}(a,b) = \text{sign}(a).*\max(|a| - b, 0)$.

2. **$\boldsymbol{v}$-subproblem:** to find the $\boldsymbol{v}$-involved minimization problem from (5), i.e.,

$$\min_{\boldsymbol{v}} \ \lambda\|\boldsymbol{v}\|_1 + \frac{\beta_2}{2}\|\boldsymbol{v} - \boldsymbol{x} - \boldsymbol{b}_2\|_2^2, \tag{8}$$

which has the following closed-form solution:

$$\boldsymbol{v}^{(k+1)} = \text{Shrink}\left(\boldsymbol{x}^{(k)} - \boldsymbol{b}_2^{(k)}, \frac{\lambda}{\beta_2}\right). \tag{9}$$

3. **$\boldsymbol{x}$-subproblem:** to find the $\boldsymbol{x}$-involved minimization problem from (5), i.e.,

$$\min_{\boldsymbol{x}} \ \frac{\beta_1}{2}\|\boldsymbol{u} - (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) + \boldsymbol{b}_1\|_2^2 + \frac{\beta_2}{2}\|\boldsymbol{v} - \boldsymbol{x} + \boldsymbol{b}_2\|_2^2, \tag{10}$$

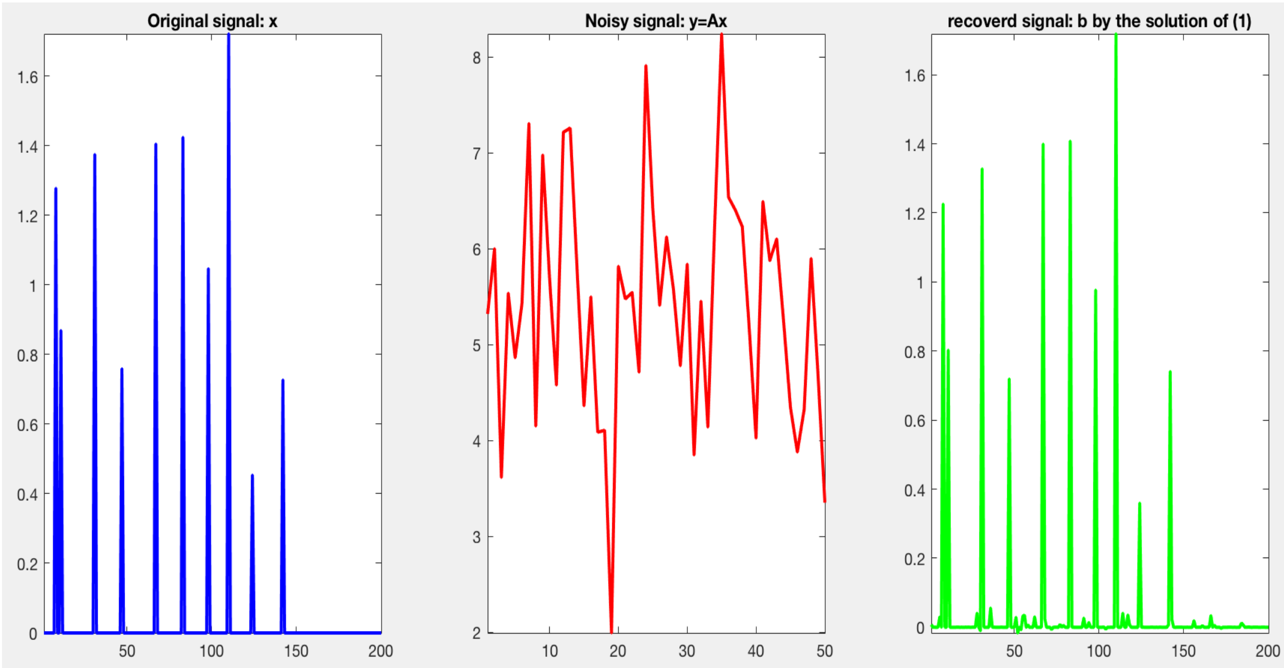

Fig. 1. **Check if your results follow this figure? If yes, your code is right!** (Left) Original signal; (Middle) Noisy signal; (Right) Recoverd signal by model (2) and ADMM algorithm.

which has the following closed-form solution by least squares method:

$$\boldsymbol{x}^{(k+1)} = (\beta_1 \boldsymbol{A}^T \boldsymbol{A} + \beta_2 \boldsymbol{I})^{-1} \boldsymbol{r}^{(k)} \tag{11}$$

where $\boldsymbol{r}^{(k)} = \beta_2(\boldsymbol{v}^{(k+1)} + \boldsymbol{b}_2^{(k)}) - \beta_1 \boldsymbol{A}^T(\boldsymbol{u}^{(k+1)} - \boldsymbol{y} + \boldsymbol{b}_1^{(k)})$, and $\boldsymbol{I}$ is an identity matrix.

4. **Update $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$** :

$$\begin{aligned} \boldsymbol{b}_1^{(k+1)} &= \boldsymbol{b}_1^{(k)} + 1.618 * \left(\boldsymbol{u}^{(k+1)} - (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(k+1)})\right) \\ \boldsymbol{b}_2^{(k+1)} &= \boldsymbol{b}_2^{(k)} + 1.618 * (\boldsymbol{v}^{(k+1)} - \boldsymbol{x}^{(k+1)}) \end{aligned} \tag{12}$$

5. Go back to step 1 until reaching the pre-defined tolerance.

Summarize the above algorithm to get the following Algorithm 1 for solve (2).

Algorithm 1: The ADMM algorithm for the model (2)

Input: $\boldsymbol{y}$, $\boldsymbol{A}$; λ, β_1 and β_2; MaxIte $= 500$; $tol = 10^{-8}$
Output: Recovered signal $\tilde{\boldsymbol{x}}$
Initialize: $\boldsymbol{x}^{(0)} = \boldsymbol{b}_1^{(0)} = \boldsymbol{b}_1^{(0)} = \boldsymbol{0}$; ReErr $= 1$, $k = 0$
While ReErr $> tol$ && $k <$ MaxIter
  1) Compute $\boldsymbol{u}^{(k+1)}$ by (7)
  2) Compute $\boldsymbol{v}^{(k+1)}$ by (9)
  3) Compute $\boldsymbol{x}^{(k+1)}$ by (11)
  4) Update $\boldsymbol{b}_1^{(k+1)}$ and $\boldsymbol{b}_2^{(k+1)}$ by (12)
  5) Calculate ReErr $= \frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2}{\|\boldsymbol{x}^{(k+1)}\|_2}$
  6) $k = k + 1$
end
  7) Output the final recovered signal $\boldsymbol{x}$ as the final result $\tilde{\boldsymbol{x}}$.

1.2. Please coding Algorithm 1 and getting the corresponding solution for 1D signal recovery

Setting the simulation experiment by the following (given in the code), then please coding ADMM by Algorithm 1 to get the results

1. clear all; close all;
2. rng(0); % for reproducibility
3. m = 50; % num samples
4. n = 200; % num variables, note that n ¿ m
5. A = rand(m, n);
6. x = zeros(n, 1);
7. nz = 10; % 10 non-zeros variables
8. nz_idx = randperm(n);
9. x(nz_idx(1:nz)) = 2 * rand(nz, 1);
10. y = A*x;
11. y = y + 0.1 * rand(m, 1); % add some noise
12. ······ (coding Algorithm 1 by yourself from here)

1.3. Q: How to extend 1D signal recovery to 2D image recovery?