Alternating Direction Method of Multiplier (ADMM) for Signal Processing : Notes

Tarmizi Adam

October 10, 2021

Abstract

In this note we introduce the reader to Total variation denoising and deblurring using Augmented lagrangian method (ALM) and the alternating direction method of multipliers (ADMM).

Contents

1 Introduction

The computation of the Rudin-Osher-Fatemi (ROF) model for image denoising/deconvolution is difficult due to the non-differentiability of the ℓ_1-norm in the objective function. Some popular methods to solve the ROF model are:

1. **Dual method**: Based on dual formulation of the problem.
2. **Split Bregman**: Uses function splitting and Bregman iteration to deal with the constrained problem.
3. **Augmented Lagrangian**: Also uses a splitting scheme and have some similarities to the *split Bregman* method.

For deblurring, we are interested in the following mathematical model of deblurring

$$\mathbf{f} = \mathbf{K}\mathbf{u} + \eta, \tag{1}$$

where $\mathbf{f} \in \mathbb{R}^n$ is the noisy and blurred image, $\mathbf{K} \in \mathbb{R}^{m\times n}$ is a linear operator, η is additive noise and $\mathbf{u}$ is the true image to be estimated. In the case of denoising, $\mathbf{K} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. For pure deblurring (also deconvolution), (1) takes the following model

$$\mathbf{f} = \mathbf{K}\mathbf{u}. \tag{2}$$

Even for the pure deblurring case, the problem is still difficult because $\mathbf{K}$ is ill-conditioned (K is highly sensitive to small perturbations). A direct[1] solution would be highly oscillatory. However, we can still solve (1) and (2) using regularization. To do so, we introduce a regularization function $R(\mathbf{u})$

$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}}\, R(\mathbf{u}) \\ &\text{s.t.} \|\mathbf{f} - \mathbf{K}\mathbf{u}\|^2 = \sigma^2, \end{aligned} \tag{3}$$

where σ is the noise variance. Problem (3) is equivalent to the following problem

$$\underset{\mathbf{u}}{\text{minimize}}\, F(\mathbf{u}) := \frac{\lambda}{2}\|\mathbf{f} - \mathbf{K}\mathbf{u}\|^2 + R(\mathbf{u}). \tag{4}$$

There are many choices of the regularization function $R(\mathbf{u})$. However in this note, the regularization function (also known as regularizer or penalty function) that we are interested is the total variation regularizer/penalty that is defined as

$$\int_\Omega |\nabla \mathbf{u}|, \tag{5}$$

which is the sum of the gradient of the true image in the entire image space Ω. Thus, problem (4) becomes

$$\underset{\mathbf{u}}{\text{minimize}}\, F_{ROF}(\mathbf{u}) := \frac{\lambda}{2}\|\mathbf{f} - \mathbf{K}\mathbf{u}\|^2 + \int_\Omega |\nabla \mathbf{u}|. \tag{6}$$

When $\mathbf{K} = \mathbf{I}$, the ROF model for denoising (TV denoising) is defined as

$$\underset{\mathbf{u}}{\text{minimize}}\, F_{ROF}(\mathbf{u}) := \frac{\lambda}{2}\|\mathbf{f} - \mathbf{u}\|^2 + \int_\Omega |\nabla \mathbf{u}|. \tag{7}$$

[1] Directly finding the inverse

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{f}$$

would be useless.

2 The Augmented Lagrangian Method

The ROF denoising model is difficult to solve because of the non-differentiability of the ROF regularization term[2]. However, there are various algorithms that can be used to solve the ROF model. One such algorithm is the Augmented Lagrangian method (ALM). The ALM is used to solve the following problem

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{q}}{\text{minimize}}\ F_{ROF}\left(\mathbf{u}\right) := \frac{\lambda}{2}\|\mathbf{f}-\mathbf{K}\mathbf{u}\|^2 + \int_\Omega |\mathbf{q}| \\ &\text{s.t. } \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \partial_x \mathbf{u} \\ \partial_y \mathbf{u} \end{bmatrix} = \nabla \mathbf{u}, \end{aligned} \tag{8}$$

where we have introduced an intermediate/auxiliary variable q for the regularizer. The augmented Lagrangian function of problem (8) is

$$\min_{\mathbf{u},\mathbf{q}} \max_{\mu} \mathcal{L}_{ROF}\left(\mathbf{u},\mathbf{q},\mu\right) = \frac{\lambda}{2}\|\mathbf{f}-\mathbf{K}\mathbf{u}\|^2 + \int_\Omega |\mathbf{q}| + \int_\Omega \mu\cdot\left(\mathbf{q}-\nabla\mathbf{u}\right) + \frac{r}{2}\int_\Omega |\mathbf{q}-\nabla\mathbf{u}|^2, \tag{9}$$

where the parameters $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and r are the Lagrange multipliers and a positive constant (the regularization parameter associate with the term $|q-\nabla\mathbf{u}|^2$). To solve problem (8) we use alternating direction method of multipliers (ADMM) and separate it into two sub-problems i.e.

u sub-problem:
The **u** sub-problem has the following objective function

$$\underset{\mathbf{u}}{\operatorname{argmin}}\ \frac{\lambda}{2}|\mathbf{K}\mathbf{u}-\mathbf{f}|^2 + \int_\Omega \mu^k\cdot\nabla\mathbf{u} + \frac{r}{2}\int_\Omega |\mathbf{q}-\nabla\mathbf{u}|^2. \tag{10}$$

q sub-problem:
The **q** sub-problem has the following objective function

$$\underset{\mathbf{q}}{\operatorname{argmin}}\ \int_\Omega |\mathbf{q}| + \int_\Omega \mu\cdot\mathbf{q} + \frac{r}{2}|\mathbf{q}-\nabla\mathbf{u}|^2. \tag{11}$$

The reasoning of using ADMM to decouple and solve problem (8) will be discussed in subsequent sections.

3 Alternating direction method of multipliers (ADMM)

For easy discussion of numerical algorithms, we formulate the above problems using matrix and vectors. For most applications involving ADMM, we wish to solve the following problem

$$\min_{\mathbf{u}} f\left(\mathbf{u}\right) + g\left(\mathbf{D}\mathbf{u}\right). \tag{12}$$

[2] The absolute value function $|\cdot|$, is non-differentiable at zero.

Where $\mathbf{D} \in \mathbb{R}^{m\times n}$, $\mathbf{u} \in \mathbb{R}^n$ and $f(\cdot)$, $g(\cdot)$ are convex functions on $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. By introducing an auxiliary variable $\mathbf{v}$, problem (12) can be re-written in the form

$$\begin{aligned}&\min_{\mathbf{u},\mathbf{v}} f(\mathbf{u}) + g(\mathbf{v})\\ &\text{s.t. } \mathbf{v} = \mathbf{Du}\end{aligned} \tag{13}$$

By the classical ALM[3], the above problem is solved by the following iteration[4]

$$(\mathbf{u}_{k+1},\, \mathbf{v}_{k+1}) \in \operatorname*{argmin}_{\mathbf{u},\mathbf{v}} \left\{ f(\mathbf{u}) + g(\mathbf{v}) - \mu_k^T (\mathbf{v} - \mathbf{Du}) + \frac{r}{2}\, ||\mathbf{v} - \mathbf{Du}||_2^2 \right\} \tag{14}$$

$$\mu_{k+1} = \mu_k - r(\mathbf{v}_{k+1} - \mathbf{Du}_{k+1}).$$

However, solving (14) is hard due to the strong coupling of variables $\mathbf{u}$ and $\mathbf{v}$ in the term $\frac{r}{2}\,||\mathbf{v} - \mathbf{Du}||_2^2$. This strong coupling can be resolved (de-coupled) by applying ADMM which results in alternating minimization of $\mathbf{u}$ and $\mathbf{v}$. By using ADMM we have the following iterative process

$$\mathbf{u}_{k+1} \in \operatorname*{argmin}_{\mathbf{u}} \left\{ f(\mathbf{u}) + g(\mathbf{v}_k) - \mu_k^T (\mathbf{v}_k - \mathbf{Du}) + \frac{r}{2}\, ||\mathbf{v}_k - \mathbf{Du}||_2^2 \right\} \tag{15}$$

$$\mathbf{v}_{k+1} \in \operatorname*{argmin}_{\mathbf{v}} \left\{ f(\mathbf{u}_{k+1}) + g(\mathbf{v}) - \mu_k^T (\mathbf{v} - \mathbf{Du}_{k+1}) + \frac{r}{2}\, ||\mathbf{v} - \mathbf{Du}_{k+1}||_2^2 \right\} \tag{16}$$

$$\mu_{k+1} = \mu_k - r(\mathbf{v}_{k+1} - \mathbf{Du}_{k+1}). \tag{17}$$

Notice that by using ADMM we have split problem (14) of the classical ALM into two sub-problems. This splitting has decoupled the variable $\mathbf{u}$ and $\mathbf{v}$ in the term $\frac{r}{2}\,||\mathbf{v} - \mathbf{Du}||_2^2$. That is to say, when minimizing (15), variable $\mathbf{v}$ is treated as constant values ($\mathbf{v}$ is the vector at the current iteration k). While minimizing (16), variable $\mathbf{u}$ is treated a constant. Terms $g(\mathbf{v}_k)$ and $f(\mathbf{u}_{k+1})$ are also treated as constant values and could be dropped off (because they boil down to values evaluated by the function). Thus, the implication is that we can now minimize only quadratic perturbations of $g(\cdot)$ and $f(\cdot)$ i.e.

$$\mathbf{u}_{k+1} \in \operatorname*{argmin}_{\mathbf{u}} \left\{ f(\mathbf{u}) - \mu_k^T (\mathbf{v}_k - \mathbf{Du}) + \frac{r}{2}\, ||\mathbf{v}_k - \mathbf{Du}||_2^2 \right\}$$

$$\mathbf{v}_{k+1} \in \operatorname*{argmin}_{\mathbf{v}} \left\{ g(\mathbf{v}) - \mu_k^T (\mathbf{v} - \mathbf{Du}_{k+1}) + \frac{r}{2}\, ||\mathbf{v} - \mathbf{Du}_{k+1}||_2^2 \right\}$$

4 ADMM for total variation denoising

In this section, we derive the ADMM algorithm to aid us denoise a noisy signal. For basic and the simplicity of discussion, we will first discuss 1D signal denoising in this section. Comments to extend the algorithm to 2D (images) signal denoising will be provided at the end of this section.

We can model a noisy observation (signal) by the following model

$$\mathbf{f} = \mathbf{u} + \mathbf{n}, \tag{18}$$

[3]Here, the "classical ALM" means that we do not use the ADMM to split the problem into sub-problems of $\mathbf{u}$ and $\mathbf{v}$.

[4]k denotes the k^{th} iteration count.

where $\mathbf{f}$, $\mathbf{u}$ and $\mathbf{n}$ are the observed noisy signal, original clean signal and i.i.d Gaussian noise of size $N \times 1$ respectively. The main objective in signal denoising is to estimate the original clean signal $\mathbf{u}$ from its noisy observation. To derive an ADMM algorithm for this purpose, we formulate the denoising problem as the following optimization problem

$$\min_{\mathbf{u}} \frac{\lambda}{2} ||\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{D}\mathbf{u}||_1 , \tag{19}$$

where $\mathbf{D}$ is the $N \times N$ first order difference matrix.

4.1 Problem formulation

Using variable splitting as discussed in the previous section, we have the following

$$\begin{aligned} &\min_{\mathbf{u}} \frac{\lambda}{2} ||\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 , \\ &\text{s.t } \mathbf{v} = \mathbf{D}\mathbf{u}. \end{aligned} \tag{20}$$

Define the Augmented Lagrangian function of the constrained problem

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mu) = \frac{\lambda}{2} ||\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 - \mu^T (\mathbf{v} - \mathbf{D}\mathbf{u}) + \frac{\rho}{2} ||\mathbf{v} - \mathbf{D}\mathbf{u}||_2^2 , \tag{21}$$

with μ being the Lagrange multipliers and $\rho > 0$ the regularization parameter related to the constraint (also called the ADMM parameter). In its basic form of the ADMM, this parameter ρ is fixed to a certain value. However, this parameter can be updated at each iteration by some rule in order to speed up convergence of the algorithm. This will be discussed later.

The minimization of the Augmented Lagrangian function $\mathcal{L}(\cdot)$ (finding a saddle point of $\mathcal{L}(\cdot)$) presents us with two subproblems to be minimized alternately i.e. the $\mathbf{u}$ subproblem and the $\mathbf{v}$ subproblem. The algorithm runs and solves the two subproblems in each iteration until convergence is achieved.

The $\mathbf{u}$ subproblem is defined as

$$\mathbf{u}_{k+1} = \underset{\mathbf{u}}{\text{argmin}} \frac{\lambda}{2} ||\mathbf{u} - \mathbf{f}||_2^2 - \mu_k^T (\mathbf{v}_k - \mathbf{D}\mathbf{u}) + \frac{\rho}{2} ||\mathbf{v}_k - \mathbf{D}\mathbf{u}||_2^2 , \tag{22}$$

and the $\mathbf{v}$ subproblem is

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\text{argmin}} \ ||\mathbf{v}||_1 - \mu_k^T (\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}) + \frac{\rho}{2} ||\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}||_2^2 . \tag{23}$$

In the next subsection, we will discuss in more detail on how to minimize both subproblems.

4.2 Solving the sub-problems

For the $\mathbf{u}$ subproblem, if we expand, differentiate with respect to $\mathbf{u}$ and equate to zero we have the following

$$\begin{aligned} &\lambda\mathbf{u} - \lambda\mathbf{f} + \mathbf{D}^\top\mu - \rho\mathbf{D}^\top\mathbf{v} + \rho\mathbf{D}^\top\mathbf{D}\mathbf{u} = 0 \\ &\left(\rho\mathbf{D}^\top\mathbf{D} + \lambda\right)\mathbf{u} = \lambda\mathbf{f} + \rho\mathbf{D}^\top\left(\mathbf{v} - \frac{\mu}{\rho}\right) \end{aligned} \tag{24}$$

In solving for $\mathbf{u}$ in (24) we need to solve the equation

$$\mathbf{u} = \left(\rho \mathbf{D}^\top \mathbf{D} + \lambda\right)^{-1} \left[\lambda \mathbf{f} + \rho \mathbf{D}^\top \left(\mathbf{v} - \frac{\mu}{\rho}\right)\right], \tag{25}$$

in other words, we have to find the inverse of the matrix $\left(\rho \mathbf{D}^\top \mathbf{D} + \lambda\right)$ to solve for $\mathbf{u}$. In some cases, it is not practical to find the inverse because $\mathbf{D}$ is very huge. Another way to solve for $\mathbf{u}$ is by solving a linear system of equation without computing the inverse. This can be done by the help of efficients algorithms such as the Conjugate Gradient Method (CGM). Recall,

$$\left(\rho \mathbf{D}^\top \mathbf{D} + \lambda\right) \mathbf{u} = \lambda \mathbf{f} + \rho \mathbf{D}^\top \left(\mathbf{v} - \frac{\mu}{\rho}\right) \tag{26}$$

This is in a form of a linear system of equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ i.e.

$$\underbrace{\left(\rho \mathbf{D}^\top \mathbf{D} + \lambda\right)}_{\mathbf{A}} \underbrace{\mathbf{u}}_{\mathbf{x}} = \underbrace{\lambda \mathbf{f} + \rho \mathbf{D}^\top \left(\mathbf{v} - \frac{\mu}{\rho}\right)}_{\mathbf{b}}. \tag{27}$$

Solving the above linear system in each iteration, we obtain $\mathbf{u}$ i.e. we solve the $\mathbf{u}$ subproblem (22).

The $\mathbf{v}$ subproblem (23) can be written as[5],

$$\mathbf{v}_{k+1} = \operatorname*{argmin}_{\mathbf{v}} \frac{\rho}{2} \left|\left|\mathbf{v} - \left(\mathbf{D}\mathbf{u}_{k+1} + \frac{1}{\rho}\mu_k\right)\right|\right|_2^2 + ||\mathbf{v}||_1 . \tag{28}$$

If we let

$$\mathbf{x} = \mathbf{D}\mathbf{u}_{k+1} + \frac{1}{\rho}\mu_k,$$

we have

$$\mathbf{v}_{k+1} = \operatorname*{argmin}_{\mathbf{v}} \frac{\rho}{2} ||\mathbf{v} - \mathbf{x}||_2^2 + ||\mathbf{v}||_1 ,$$

which is a proximal operator[6] and can be solved using the 1D shrinkage/soft thresholding[7]. Finally we update the Lagrange multipliers as

$$\mu_{k+1} = \mu_k + (\mathbf{D}\mathbf{u}_{k+1} - \mathbf{v}_{k+1}) \tag{29}$$

To further understand the ADMM for 1D TV denoising, we present a code listing in appendix F. It is highly recommended that the reader trace each line of code in the code listing given and relate them to the equations discussed in this section.

5 ADMM for total variation deblurring

We wish to minimize the following problem

$$\min_{\mathbf{u}} F(\mathbf{u}) := \frac{\lambda}{2} ||\mathbf{K}\mathbf{u} - \mathbf{f}||^2 + R(\mathbf{u}). \tag{30}$$

[5] Convince yourself by expanding equation (28) above.
[6] Also called the denoising operator. Refer to the Image deblurring section and Appendix.
[7] Refer to the TV image deblurring section for details

For the ROF model, the regularity term $R(\mathbf{u})$, can be either anisotropic or isotropic. For Anisotropic

$$R_{ATV}(\mathbf{u}) := \sum_i \left|[\mathbf{D}_v\mathbf{u}]_i + [\mathbf{D}_h\mathbf{u}]_i\right|, \tag{31}$$

and for isotropic

$$R_{ATV}(\mathbf{u}) := \sum_i \sqrt{\left|[\mathbf{D}_v\mathbf{u}]_i^2 + [\mathbf{D}_h\mathbf{u}]_i^2\right|}. \tag{32}$$

Define $\mathbf{D}$ as

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_v \\ \mathbf{D}_h \end{bmatrix}, \tag{33}$$

where $\mathbf{D}_v$ and $\mathbf{D}_h$ are the first order difference operator in the vertical and horizontal directions of the image. The matrix $\mathbf{D}$ is the global first-order difference operator. We can now write both the anisotropic and isotropic total variation more compactly as

$$R_{ATV}(\mathbf{u}) = ||\mathbf{D}\mathbf{u}||_1, \tag{34}$$

and

$$R_{ITV}(\mathbf{u}) = \sqrt{[\mathbf{D}\mathbf{u}]^2}. \tag{35}$$

We now treat the anisotropic total variation. The anisotropic TV has the following objective function

$$\min_{\mathbf{u}} \frac{\lambda}{2} ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{D}\mathbf{u}||_1 \tag{36}$$

By introducing intermediate variable $\mathbf{v}$ we can write problem (36) as a constrained problem

$$\begin{aligned} &\min_{\mathbf{u}} \frac{\lambda}{2} ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 \\ &\text{s.t. } \mathbf{v} = \mathbf{D}\mathbf{u}, \end{aligned} \tag{37}$$

This method of introducing an intermediate variable is called *variable splitting*. This splitting is done because the sum of the functions of the optimization problem may very different in nature. By variable splitting, we will be able to perform optimization of the two functions separately using algorithms that are appropriate to each function.

5.1 Problem formulation

To solve the constrained optimization problem of (37) first, we define the augmented Lagrangian function of (37)

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mu) = \frac{\lambda}{2} ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 - \mu^T(\mathbf{v} - \mathbf{D}\mathbf{u}) + \frac{r}{2} ||\mathbf{v} - \mathbf{D}\mathbf{u}||_2^2, \tag{38}$$

where μ is the Lagrange multiplier for the constraint $\mathbf{v} = \mathbf{D}\mathbf{u}$ and $r > 0$ is the regularization parameter associated with the term $||\mathbf{v} - \mathbf{D}\mathbf{u}||_2^2$. Basically, the objective of the augmented Lagrangian method is to find a saddle point of $\mathcal{L}(\cdot)$. By this, we also solve the original constrained problem of (37).

As discussed in section 2, we can split the problem into two sub-problems $\mathbf{u}$ and $\mathbf{v}$. For the $\mathbf{u}$ sub-problem we have

$$\mathbf{u}_{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{\lambda}{2} ||\mathbf{Ku} - \mathbf{f}||_2^2 - \mu_k^T (\mathbf{v}_k - \mathbf{Du}) + \frac{r}{2} ||\mathbf{v}_k - \mathbf{Du}||_2^2 \tag{39}$$

The $\mathbf{v}$ sub-problem is

$$\begin{aligned} \mathbf{v}_{k+1} &= \underset{\mathbf{v}}{\operatorname{argmin}} \; ||\mathbf{v}||_1 - \mu_k^T (\mathbf{v} - \mathbf{Du}_{k+1}) + \frac{r}{2} ||\mathbf{v} - \mathbf{Du}_{k+1}||_2^2 . \\ &= \underset{\mathbf{v}}{\operatorname{argmin}} \; ||\mathbf{v}||_1 + \frac{r}{2} \left|\left| \mathbf{v} - \left(\mathbf{Du}_{k+1} + \frac{1}{r}\mu_k \right) \right|\right|_2^2 \end{aligned} \tag{40}$$

5.2 Solving the sub-problems

Having the problem formulated, we now discuss how to solve the problems. The $\mathbf{u}$ sub-problem is a least squares problem. This problem requires us to solve a system of linear equation by solving a corresponding normal equation,

$$\left(\lambda \mathbf{K}^T \mathbf{K} + r\mathbf{D}^T \mathbf{D}\right) \mathbf{u} = \lambda \mathbf{K}^T \mathbf{f} + r\mathbf{D}^T \mathbf{v} - \mathbf{D}^T \mu. \tag{41}$$

Under periodic boundary conditions for $\mathbf{u}$ the matrices $\mathbf{K}^T\mathbf{K}$ and $\mathbf{D}^T\mathbf{D}$ has the structure of block circulant with circulant blocks (BCCB) and thus can be diagonalized by 2D discrete Fourier transforms (FFT). Thus, solving for $\mathbf{u}$ can be efficiently solved using 2D FFT and 2D IFFT as

$$\mathbf{u} = \mathcal{F}^{-1} \left[\frac{\mathcal{F}\left[\lambda \mathbf{K}^T \mathbf{f} + r\mathbf{D}^T \mathbf{v} - \mathbf{D}^T \mu\right]}{\lambda \left|\mathcal{F}[\mathbf{K}]\right|^2 + r\left(\left|\mathcal{F}[\mathbf{D}_v]\right|^2 + \left|\mathcal{F}[\mathbf{D}_h]\right|^2\right)} \right], \tag{42}$$

where $\mathcal{F}$ denotes the 2D-FFT operator. Matrices $\mathcal{F}[\mathbf{D}_v]$, $\mathcal{F}[\mathbf{D}_h]$ and $\mathcal{F}[\mathbf{K}]$ can be pre-calculated and stored.

The $\mathbf{v}$ sub-problem (40) can be solved exactly using a shrinkage/proximal operator. This is because for this sub-problem, we have a non-differential term $||\mathbf{v}||_1$ thus, cannot be solved as the $\mathbf{u}$ sub-problem. Let

$$\mathbf{x} = \mathbf{Du} + \frac{1}{r}\mu. \tag{43}$$

Then, the shrinkage/proximal operator is given by

$$\mathbf{v} = \max\left\{ |\mathbf{x}| - \frac{1}{r}, 0 \right\} \cdot \operatorname{sign}(\mathbf{x}). \tag{44}$$

The same is used to obtain the update for $\mathbf{v}_h$. The shrinkage/proximal operator have a close relation with the concept of sub-differentials.

If the anisotropic TV regularization function is used, the $\mathbf{v}$ sub-problem is obtained by (44). Another choice of preference is the isotropic TV (refer to equation (35).) The isotropic TV usually yields better results than the anisotropic TV. In isotropic case, the $\mathbf{v}$ sub-problem is obtained by the following

$$\mathbf{v} = \max\left\{ ||\mathbf{x}|| - \frac{1}{r}, 0 \right\} \cdot \frac{\mathbf{x}}{||\mathbf{x}||} \tag{45}$$

5.3 Updating the Lagrange Multipliers

Finally, the Lagrange multipliers μ can be updated as follows

$$\mu_{k+1} = \mu_k - r\left(\mathbf{v}_{k+1} - \mathbf{D}\mathbf{u}_{k+1}\right). \tag{46}$$

In summary, we iteratively solve (39), (40) and (46).

Algorithm 1. Augmented Lagrangian Method for Total Variation (ALMTV)

Input $\mathbf{f}$, $\mathbf{K}$ and parameter $\lambda > 0$
Initialize $\mathbf{u}_0 = \mathbf{f}$, $k = 0$, $\mathbf{v}_0 = \mathbf{Df}$, $\mu = \mathbf{0}$.
Compute the matrices $\mathcal{F}(\mathbf{D}_h)$, $\mathcal{F}(\mathbf{D}_v)$ and $\mathcal{F}(\mathbf{K})$
1. **while** not converged
2. Solve the **u**-subproblem (39) using (41)
3. Solve the **v**-subproblems (40) using (44)
4. Update Lagrange multipliers using (46)
5. Check for Convergence:
6. **if** converge
7. break
8. **end if**
9. $k = k + 1$
10. **end while**

5.4 Denoising in the presence of impulse noise

In this section, we discuss the use of ADMM for total variation denoising when the noise is impulsive as opposed to additive Gaussian noise. Two main impulse noise are the salt and pepper (SnP) and the random valued impulse noise. Setting up the ADMM framework for impulsive noise is also similar as discussed before. However, instead of using the ℓ_2-norm data fidelity, it has been observed that the ℓ_1-norm fidelity is more suitable for impulse noise removal. Using the ℓ_1-norm as the data fidelity has been the standard way for impulse noise denoising.

Let us start by giving the main objective function to be minimized for the impulse noise restoration model. The objective function is of the following,

$$F(\mathbf{u}) = \lambda \left|\left|\mathbf{Ku} - \mathbf{f}\right|\right|_1 + \left|\left|\nabla \mathbf{u}\right|\right|_1 + \mathcal{I}(\mathbf{u}) \tag{47}$$

Notice the first term of the right hand side of the above equation. This term which is the data fidelity term is now defined in the ℓ_1-norm case as opposed to the ℓ_2-norm for additive Gaussian noise.

The constrained form is

$$\begin{aligned} &G(\mathbf{u}) = \lambda \left|\left|\mathbf{r}\right|\right|_1 + \left|\left|\mathbf{v}\right|\right|_1 + \mathcal{I}(\mathbf{z}) \\ &\text{s.t } \mathbf{v} = \nabla \mathbf{u},\ \mathbf{r} = \mathbf{Ku} - \mathbf{f},\ \mathbf{z} = \mathbf{u} \end{aligned} \tag{48}$$

By this constrained form, the astute reader should be able to see that minimizing the constrained form will amount to solve two ℓ_1-norm proximal operator (soft thresholding) as the ADMM subproblems. Proceeding with the augmented Lagrangian function,

$$\mathcal{L}_{\mathcal{A}}\left(\mathbf{u},\, \mathbf{v},\, \mathbf{r},\, \mathbf{z},\, \mu_1,\, \mu_2,\, \mu_3\right) = \lambda \left|\left|\mathbf{r}\right|\right|_1 + \left|\left|\mathbf{v}\right|\right|_1 + \mathcal{I}\left(\mathbf{z}\right) + \frac{\rho_1}{2}\left|\left|\mathbf{r} - \left(\mathbf{K}\mathbf{u} - \mathbf{f}\right)\right|\right|_2^2 - \mu_1^\top\left(\mathbf{r} - \left(\mathbf{K}\mathbf{u} - \mathbf{f}\right)\right)$$
$$+ \frac{\rho_2}{2}\left|\left|\mathbf{v} - \nabla\mathbf{u}\right|\right|_2^2 - \mu_2^\top\left(\mathbf{v} - \nabla\mathbf{u}\right) + \frac{\rho_3}{2}\left|\left|\mathbf{z} - \mathbf{u}\right|\right|_2^2 - \mu_3^\top\left(\mathbf{z} - \mathbf{u}\right). \tag{49}$$

From the augmented Lagrangian function, we now have to solve four subproblems. The **u**-subproblem has the form

$$\left(\mathbf{K}^\top\mathbf{K} + \frac{\rho_2}{\rho_1}\nabla^\top\nabla + \frac{\rho_3}{\rho_1}\right)\mathbf{u} = \mathbf{K}^\top\left(\mathbf{r} + \mathbf{f} - \frac{\mu_1}{\rho_1}\right) + \frac{1}{\rho_1}\nabla^\top\left(\rho_2\mathbf{v} - \mu_2\right) + \frac{\rho_3}{\rho_1}\mathbf{z} - \frac{\mu_3}{\rho_1}, \tag{50}$$

that is, we have to solve a linear system to obtain $\mathbf{u}_{k+1}$.

The **v**-subproblem has the following form

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ \frac{1}{2}\left|\left|\mathbf{v} - \left(\nabla\mathbf{u} + \frac{\mu_2}{\rho_2}\right)\right|\right|_2^2 + \frac{1}{\rho_2}\left|\left|\mathbf{v}\right|\right|_1. \tag{51}$$

Next, the **r**-subproblem has the following structure

$$\mathbf{r}_{k+1} = \underset{\mathbf{r}}{\operatorname{argmin}}\ \frac{1}{2}\left|\left|\mathbf{r} - \left(\mathbf{K}\mathbf{u} - \mathbf{f} + \frac{\mu_1}{\rho_1}\right)\right|\right|_2^2 + \frac{\lambda}{\rho_1}\left|\left|\mathbf{r}\right|\right|_1. \tag{52}$$

Both the **v** and **r** sub-problems are ℓ_1 regularization problems and can be solved in closed form via the soft thresholding or shrinkage operator as

$$\mathbf{v}_{k+1} = \operatorname{sign}\left(\nabla\mathbf{u} + \frac{\mu_2}{\rho_2}\right) \odot \max\left(\left|\nabla\mathbf{u} + \frac{\mu_2}{\rho_2}\right| - \frac{1}{\rho_2}\right), \tag{53}$$

and

$$\mathbf{r}_{k+1} = \operatorname{sign}\left(\mathbf{K}\mathbf{u} - \mathbf{f} + \frac{\mu_1}{\rho_1}\right) \odot \max\left(\left|\mathbf{K}\mathbf{u} - \mathbf{f} + \frac{\mu_1}{\rho_1}\right| - \frac{\lambda}{\rho_1}\right), \tag{54}$$

respectively.

Finally, the **z**-subproblem is minimized by the following

$$\mathbf{z}_{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}}\ \frac{\rho_3}{2}\left|\left|\mathbf{z} - \left(\mathbf{u} + \frac{\mu_3}{\rho_3}\right)\right|\right|_2^2 + \mathcal{I}\left(\mathbf{z}\right), \tag{55}$$

where the closed form solution is the projection operator of the following

$$\mathbf{z}_{k+1} = \min\left(255,\, \max\left(\mathbf{u} + \frac{\mu_3}{\rho_3},\, 0\right)\right). \tag{56}$$

The update of the Lagrange multipliers goes as follows,

$$\begin{aligned}\mu_2^{k+1} &= \mu_2^k - \rho_2\left(\mathbf{v} - \nabla\mathbf{u}\right),\\ \mu_3^{k+1} &= \mu_3^k - \rho_3\left(\mathbf{z} - \mathbf{u}\right),\\ \mu_1^{k+1} &= \mu_1^k - \rho_1\left(\mathbf{r} - \mathbf{q}\right),\end{aligned}$$

where $\mathbf{q} = \mathbf{K}\mathbf{u} - \mathbf{f}$.

5.4.1 The ℓ_0-TV for impulse noise

The state of the art TV denoising and deblurring affected with impulse noise is the ℓ_0-TV proposed by Yuan and Ghanem. The method is to used the ℓ_0-norm as the data fidelity and use a formulation called mathematical program for equally constrained (MPEC) formulation to solve the ℓ_0 subproblem. Let us define the problem setting and how to formulate the MPEC for ℓ_0-TV. As usuall, we have our observation $\mathbf{b}$ form the following degradation model,

$$\mathbf{b} = \mathbf{K}\mathbf{u} + \mathbf{n}, \tag{57}$$

with $\mathbf{K}$, $\mathbf{n}$ and $\mathbf{u}$ is the linear operator (blur PSF), additive noise and original image respectively. We wish to minimize the following problem

$$\min_{0 \leq \mathbf{u} \leq} \; ||\mathbf{o} \odot (\mathbf{K}\mathbf{u} - \mathbf{b})||_0 + \lambda \, ||\nabla \mathbf{u}||_1 \tag{58}$$

Lemma 1. *For any given* $\mathbf{w} \in \mathbb{R}^n$*, it holds that*

$$||\mathbf{w}||_0 = \min_{0 \leq \mathbf{u} \leq}$$

6 Frame Based Image Deblurring

This section, we show how to perform wavelet image deblurring using Parseval frames of the form

$$\underset{\mathbf{u}}{\text{minimize}} \; F(\mathbf{u}) = \frac{\lambda}{2} \, ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{W}\mathbf{u}||_1 \,, \tag{59}$$

where the matrix $\mathbf{W}$ is an analysis matrix of a Parseval frame. This problem is also referred as the *analysis* based regularization. The analysis matrix satisfy

$$\mathbf{W}^\top \mathbf{W} = \mathbf{I}. \tag{60}$$

Let us reformulate the analysis based regularization problem above as constrained optimization problem of the form

$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}} \; F(\mathbf{u}) = \frac{\lambda}{2} \, ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 \,, \\ &\text{s.t. } \mathbf{v} = \mathbf{W}\mathbf{u}. \end{aligned} \tag{61}$$

The augmented Lagrangian of the above constrained optimization problem would be

$$\mathcal{L}_{\mathcal{A}}(\mathbf{u}, \mathbf{v}, \mu) = \frac{\lambda}{2} \, ||\mathbf{K}\mathbf{u} - \mathbf{f}||_2^2 + ||\mathbf{v}||_1 - \mu^\top (\mathbf{v} - \mathbf{W}\mathbf{u}) + \frac{\rho}{2} \, ||\mathbf{v} - \mathbf{W}\mathbf{u}||_2^2 \,, \tag{62}$$

and the subproblems would be

$$\begin{aligned} \mathbf{v}^{k+1} &= \underset{\mathbf{v}}{\text{argmin}} \; \frac{\rho}{2} \, ||\mathbf{v} - \mathbf{W}\mathbf{u}||_2^2 - \mu^\top (\mathbf{v} - \mathbf{W}\mathbf{u}) + ||\mathbf{v}||_1 \,, \\ &= \frac{\rho}{2} \left|\left| \mathbf{v} - \left(\mathbf{W}\mathbf{u} + \frac{\mu}{\beta} \right) \right|\right|_2^2 + ||\mathbf{v}||_1 \,, \end{aligned} \tag{63}$$

and

$$\mathbf{u}^{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}} \ \frac{\lambda}{2} \left|\left|\mathbf{K}\mathbf{u} - \mathbf{f}\right|\right|_2^2 - \mu^\top \left(\mathbf{v} - \mathbf{W}\mathbf{u}\right) + \frac{\rho}{2} \left|\left|\mathbf{v} - \mathbf{W}\mathbf{u}\right|\right|_2^2 . \tag{64}$$

The subproblem (63) is solved using soft-thresholding while (64) is a linear system (least squares problem) of the form

$$\left(\mathbf{K}^\top\mathbf{K} + \frac{\rho}{\lambda}\mathbf{W}^\top\mathbf{W}\right)\mathbf{u} = \mathbf{K}^\top\mathbf{f} + \frac{1}{\lambda}\mathbf{W}^\top\left(\mathbf{v} - \mu\right), \tag{65}$$

where $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$. As discussed earlier in this notes, this linear system can also be solved via 2D fast Fourier transform.

7 Low Rank Matrix Recovery

In the previous sections we have seen TV denoising which could be seen as a sparse signal recovery problem. There, the sparsity of the signal/image was assumed to be in the gradient domain. Along a similar line, compressed sensing (CS) problems i.e., recovering a sparse signal at a lower rate than the classical Nyquist sampling rate also rely on sparsity of the signal (or sparse in some transform domain/basis). This sparsity concept is also extended in low rank matrix completion problems.

Low rank matrix recovery has been a very active line of research in the past several years due to the increasing amount of data. Low rank recovery of matrices are concerned with recovering low dimensional structures from high dimensional data. Low rank structure is an extension of sparsity of the singular values of a matrix. This means that the values of the singular values of a matrix is sparse. Some problems that fit under low rank matrix recovery problems are

1. Robust principal component analysis (RPCA). This problem is a form of *matrix separation problem*.
2. Low rank matrix completion. As the name suggest, the task is to complete the missing entries of a matrix subject to the completion producing a low rank matrix.
3. Low rank matrix factorization.
4. Non-negative matrix factorization (NNMF).

7.1 Preliminaries

Before we start, we introduce some mathematical preliminaries for the reader to understand the problems of low rank matrix recovery. Let $\mathbf{X}_\Omega \in \mathbb{R}^{m\times n}$ be a matrix with missing entries. The subset Ω is the subset of complete entries of index pairs $(i\,,j)$. The subscript $(\cdot)_\Omega$ denotes the projection on the known entries. The $(i\,,j)$ entries of $\mathbf{X}_\Omega$ is denoted $\left[\mathbf{X}_\Omega\right]_{i\,,j}$ and written as

$$\left[\mathbf{X}_\Omega\right]_{i\,,j} = \begin{cases} \mathbf{X}_{i\,,j}, & \text{if } (i\,,j) \in \Omega \\ 0, & \text{Otherwise} \end{cases}$$

The vector stacking (column-by-column) of all entries of $\mathbf{X}_\Omega$ is denoted as $\mathbf{x} \in \mathbb{R}^{|\Omega|}$ where $|\Omega|$ stands for the number of elements in Ω i.e., the cardinality of Ω or number of observed entries. To further understand the notations, we provide several simple examples

Example 7.1. Suppose the matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 4 \\ 6 & 2 & 6 \\ 3 & 4 & 7 \end{bmatrix},$$

if only four elements are observed, $\Omega = \{(1, 1), (3, 1), (2, 2), (2, 3)\}$ we have,

$$\mathbf{X}_\Omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 6 \\ 3 & 0 & 0 \end{bmatrix}.$$

The vector stacking of $\mathbf{X}$ into $\mathbf{x}_\Omega$ is,

$$\mathbf{x}_\Omega = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 6 \end{bmatrix}, \quad \mathbf{x}_\Omega \in \mathbb{R}^{|\Omega|}$$

7.2 Low rank matrix completion via ADMM

Low rank matrices frequently arises in science and engineering. Therefore, one of the problem of interest is low rank matrix completion (LRMC). The aim is to find the missing or unknown entries of a low-rank matrix from incomplete and possibly noisy observations. Originally, LRMC problem is posed as a rank minimization problem of the form

$$\begin{aligned} &\min_{\mathbf{M}} \operatorname{rank}(\mathbf{M}) \\ &\text{s.t. } \mathbf{M}_\Omega = \mathbf{X}_\Omega. \end{aligned} \tag{66}$$

Unfortunately, the problem is NP-hard due to the nonconvexity of the rank function. Therefore, convex relaxations of problem (66) are minimized instead,

$$\begin{aligned} &\min_{\mathbf{M}} \ ||\mathbf{M}||_* \\ &\text{s.t. } \mathbf{M}_\Omega = \mathbf{X}_\Omega. \end{aligned} \tag{67}$$

7.3 Noisy low rank matrix completion

In the noisy setting where the observed low rank matrix is corrupted by i.i.d additive Gaussian noise, we have the following forward model,

$$\begin{aligned} \mathbf{Y}_\Omega &= \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{N}, \\ &= \mathbf{X}_\Omega + \mathbf{N}, \end{aligned} \tag{68}$$

where, $\mathbf{Y}_\Omega \in \mathbb{R}^{m\times n}$ is the observed matrix, $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the low rank matrix to be estimated and $\mathbf{N}^{m\times n}$ is additive noise. Note that, some of the entries of the observed low rank matrix $\mathbf{Y}_\Omega$ are missing and noisy. To estimate the low rank $\mathbf{X}$ we minimize the following optimization problem

$$\underset{\mathbf{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} \ F(\mathbf{X}) = \frac{1}{2} ||\mathbf{Y}_\Omega - \mathbf{X}_\Omega||_2^2 + \lambda ||\mathbf{X}||_*, \tag{69}$$

where $||\cdot||_*$ denotes the nuclear norm[8]. The nuclear norm denotes the sum of the singular values of the matrix. For example, by assuming $\mathbf{X}$ has r positive singular values than

$$||\mathbf{X}||_* = \sum_{i=1}^{r} \sigma_i. \tag{70}$$

To apply ADMM for problem (69), we can re-write the problem as a constrained optimization problem as follows

$$\begin{aligned} &\underset{\mathbf{X}\in\mathbb{R}^{m\times n},\, \mathbf{M}\in\mathbb{R}^{m\times n}}{\text{minimize}} \; F(\mathbf{X}) = \frac{1}{2}||\mathbf{M}_\Omega||_2^2 + \lambda ||\mathbf{X}||_*, \\ &\text{s.t. } \mathbf{M}_\Omega = \mathbf{Y}_\Omega - \mathbf{X}_\Omega. \end{aligned} \tag{71}$$

The augmented Lagrangian function of problem (71) is written as

$$\mathcal{L}_{\mathcal{A}}(\mathbf{X},\, \mathbf{M},\, \mu) = \frac{1}{2}||\mathbf{M}_\Omega||_2^2 + \lambda ||\mathbf{X}||_* - \mu^\top (\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega) + \frac{\beta}{2}||\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega||_2^2. \tag{72}$$

Minimizing the above Lagrangian function results in a two block alternating minimization problem. We minimize $\mathbf{X}$ while holding other variables constant then, we minimize $\mathbf{M}$ and finally, we update the dual variables (Lagrange multipliers) μ. Minimization with respect to X is as follows

$$\begin{aligned} \mathbf{X}_{k+1} &= \underset{\mathbf{X}}{\text{argmin}} \frac{\beta}{2}||\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega||_2^2 - \mu^\top (\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega) + \lambda ||\mathbf{X}||_*. \\ &= \underset{\mathbf{X}}{\text{argmin}} \frac{\beta}{2}\left|\left|\mathbf{X} - \left(\mathbf{M}_\Omega + \mathbf{X}_\Omega - \mathbf{Y}_\Omega - \frac{\mu}{\beta}\right)\right|\right|_2^2 + \lambda ||\mathbf{X}||_*. \end{aligned} \tag{73}$$

Minimization of problem[9] (73) can be seen as a proximal operator. The problem has a closed form solution in terms of singular value thresholding (SVT).

The second minimization is to find $\mathbf{M}$. Minimizing with respect to $\mathbf{M}$ reads

$$\mathbf{M}_{k+1} = \underset{\mathbf{M}}{\text{argmin}} \frac{\beta}{2}||\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega||_2^2 - \mu^\top (\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega) + \frac{1}{2}||\mathbf{M}_\Omega||_2^2. \tag{74}$$

This minimization is quadratic and the minimizer can be found by differentiating the right hand side of (74) with respect to $\mathbf{M}$ i.e., $\frac{\partial}{\partial \mathbf{M}}$ and equating to zero. By doing so, we obtain the following closed form solution

$$\mathbf{M}_{k+1} = \frac{\beta(\mathbf{Y}_\Omega - \mathbf{X}_\Omega) + \mu}{\beta + 1}. \tag{75}$$

Finally, we update the Lagrange multiplier as

$$\mu_{k+1} = \mu_k - \beta(\mathbf{M}_\Omega - \mathbf{Y}_\Omega + \mathbf{X}_\Omega). \tag{76}$$

[8]It is also known as the Ky Fan norm or the trace norm.

[9]See appendix for derivations.

7.4 Robust principal component analysis via ADMM

Another problem that is appealing for two block ADMM is decomposition of a given matrix into its corresponding low rank matrix and sparse matrix. This problem is called robust principal component analysis (RPCA). Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, we wish to decompose it into its corresponding low rank matrix $\mathbf{L}$ and its sparse matrix $\mathbf{S}$. Specifically, $\mathbf{M} = \mathbf{L} + \mathbf{S}$. The RPCA problem is concerned with the following minimization problem

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{S}} \; ||\mathbf{L}||_* + \lambda \, ||\mathbf{S}||_1 \\ &\text{s.t. } \mathbf{L} + \mathbf{S} = \mathbf{M}. \end{aligned}$$

where $||\mathbf{L}||_*$ is the nuclear norm of the matrix (sum of its singular values). Defining the augmented Lagrangian,

$$\mathcal{L}_A \left(\mathbf{L} \, , \mathbf{S} \, \mu \right) = ||\mathbf{L}||_* + \lambda \, ||\mathbf{S}||_1 - \mu^\top \left(\mathbf{L} + \mathbf{S} - \mathbf{M} \right) + \frac{\rho}{2} \, ||\mathbf{L} + \mathbf{S} - \mathbf{M}||_2^2 \, , \tag{77}$$

As discussed previously, we are presented with two subproblems. The $\mathbf{L}$ subproblem is

$$\begin{aligned} \mathbf{L}_{k+1} &= \arg\min_{\mathbf{L}} \frac{\rho}{2} \, ||\mathbf{L} + \mathbf{S}_k - \mathbf{M}||_2^2 - \mu_k^\top \left(\mathbf{L} + \mathbf{S}_k - \mathbf{M} \right) + ||\mathbf{L}||_* \\ &= \frac{\rho}{2} \left|\left| \mathbf{L} - \left(-\mathbf{S}_k + \mathbf{M} + \frac{\mu_k}{\rho} \right) \right|\right|_2^2 + ||\mathbf{L}||_* \, . \end{aligned} \tag{78}$$

Problem (78) can be solved in a closed form using singular value thresholding.

The second subproblem is solving for the sparse matrix $\mathbf{S}$. Specifically we have

$$\begin{aligned} \mathbf{S}_{k+1} &= \operatorname*{argmin}_{\mathbf{S}} \frac{\rho}{2} \, ||\mathbf{L}_{k+1} + \mathbf{S} - \mathbf{M}||_2^2 - \mu_k^\top \left(\mathbf{L}_{k+1} + \mathbf{S} - \mathbf{M} \right) + \lambda \, ||\mathbf{S}||_1 \\ &= \operatorname*{argmin}_{\mathbf{S}} \frac{\rho}{2} \left|\left| \mathbf{S} - \left(-\mathbf{L}_{k+1} + \mathbf{M} + \frac{\mu_k}{\rho} \right) \right|\right|_2^2 + \lambda \, ||\mathbf{S}||_1 \, . \end{aligned} \tag{79}$$

Problem (79) is solved by soft thresholding. Finally, the Lagrange multipliers are update as follows,

$$\mu_{k+1} = \mu_k - \rho \left(\mathbf{L}_k + \mathbf{S}_k - \mathbf{M} \right) \tag{80}$$

It is worth mentioning that the RPCA problem solved via ADMM exhibit closed form solutions for both subproblems i.e. both subproblems are solved in closed form by solving their respective proximal operators. Specifically, by singular value thresholding and soft thresholding. Unlike the denoising and deblurring problems via ADMM, there is no linear system of equations that should be solved in the RPCA problem. In Figure 1, we show the result of image inpainting using RPCA.

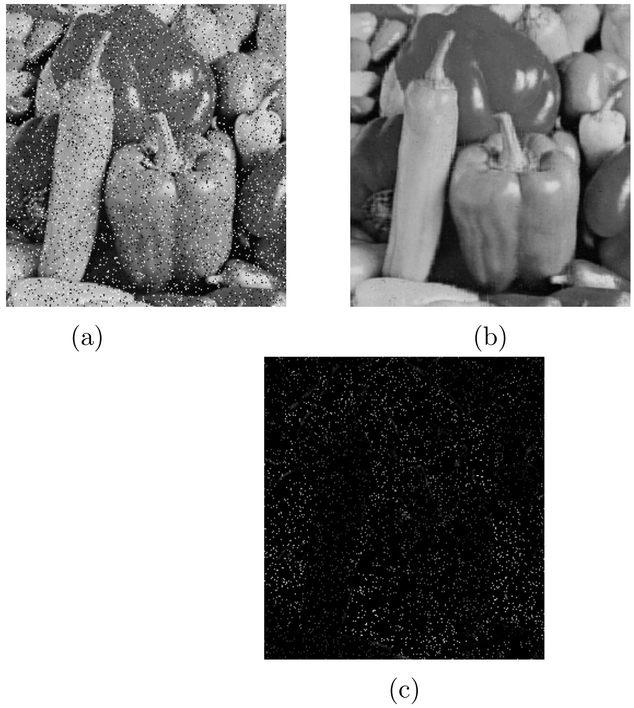

Figure 1: (a) Corrupted missing pixel (impulse noise), (b) restored low rank and (c) sparse component.

8 The Alternating Minimization Algorithm (AMA)

An almost similar algorithm to the ADMM, is the one proposed by Tseng [1] is worth mentioning. This algorithm, called alternating minimization algorithm (AMA) differs with the ADMM in that, instead of just minimizing the augmented Lagrangian function in ADMM, AMA minimizes both the Lagrangian function and the augmented Lagrangian function. The AMA algorithm has been used for image restoration for example in [2, 3]. To demonstrate the difference between AMA and ADMM, in the following subsections, we formulate 1D total variation denoising using AMA.

To start with, consider again the problem of minimizing two convex functionals $f(\cdot)$ and $g(\cdot)$ in (13)

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v}} f(\mathbf{u}) + g(\mathbf{v}) \\ &\text{s.t. } \mathbf{v} = \mathbf{D}\mathbf{u}. \end{aligned} \tag{81}$$

The Lagrangian function and the augmented Lagrangian function for the above are as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{u},\mathbf{v},\mu) &= f(\mathbf{u}) + g(\mathbf{v}) + \mu^T(\mathbf{v} - \mathbf{D}\mathbf{u}), \\ \mathcal{L}_a(\mathbf{u},\mathbf{v},\mu) &= f(\mathbf{u}) + g(\mathbf{v}) + \mu^T(\mathbf{v} - \mathbf{D}\mathbf{u}) + \frac{\rho}{2}||\mathbf{v} - \mathbf{D}\mathbf{u}||_2^2. \end{aligned}$$

In order to solve (81), in each k^{th} iteration we solve the Lagrangian and the augmented Lagrangian as follows:

$$\begin{cases} \mathbf{u}_{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}}\ \mathcal{L}\left(\mathbf{u}, \mathbf{v}_k, \mu_k\right) \\ \mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ \mathcal{L}_a\left(\mathbf{u}_{k+1}, \mathbf{v}, \mu_k\right) \\ \mu_{k+1} = \mu_k + \left(\mathbf{v}_{k+1} - \mathbf{D}\mathbf{u}_{k+1}\right) \end{cases}$$

As can be seen, we obtain the current estimate $\mathbf{u}_{k+1}$, by solving the Lagrangian function. This is different from ADMM where we obtain the current estimate of $\mathbf{u}$ by the augmented Lagrangian function. This is the only difference between ADMM and AMA. For the estimation of $\mathbf{v}_{k+1}$ both AMA and ADMM uses the augmented Lagrangian function.

Even though there is only one fundamental difference between AMA and ADMM, as we will see, for the problem of 1D signal denoising, we can omit the heavy computational cost of solving a linear system at each iteration which we had to do when using ADMM. Now that we know the difference between AMA and ADMM, we will apply AMA for 1D TV denoising and compare the results with ADMM.

9 AMA for total variation denoising

9.1 Problem formulation

For convenience, we restate back the objective function for total variation denoising (20) here

$$\begin{aligned} &\min_{\mathbf{u}} \frac{\lambda}{2} \left|\left|\mathbf{u} - \mathbf{f}\right|\right|_2^2 + \left|\left|\mathbf{v}\right|\right|_1, \\ &\text{s.t } \mathbf{v} = \mathbf{D}\mathbf{u}. \end{aligned} \tag{82}$$

Define the Lagrangian function

$$\mathcal{L}\left(\mathbf{u},\, \mathbf{v},\, \mu\right) = \frac{\lambda}{2} \left|\left|\mathbf{u} - \mathbf{f}\right|\right|_2^2 + \left|\left|\mathbf{v}\right|\right|_1 + \mu^\top \left(\mathbf{v} - \mathbf{d}\mathbf{u}\right), \tag{83}$$

and the augmented Lagrangian function

$$\mathcal{L}_a\left(\mathbf{u}, \mathbf{v}, \mu\right) = \frac{\lambda}{2} \left|\left|\mathbf{u} - \mathbf{f}\right|\right|_2^2 + \left|\left|\mathbf{v}\right|\right|_1 + \mu^\top \left(\mathbf{v} - \mathbf{D}\mathbf{u}\right) + \frac{\rho}{2} \left|\left|\mathbf{v} - \mathbf{D}\mathbf{u}\right|\right|_2^2. \tag{84}$$

As indicated, AMA also produces two subproblems[10] $\mathbf{u}$ and $\mathbf{v}$. However, $\mathbf{u}$ subproblem update is obtained from the Lagrangian function while $\mathbf{v}$ subproblem update is obtained from the augmented Lagrangian function. We now investigate the two subproblems.

The $\mathbf{u}$ subproblem is defined as

$$\mathbf{u}_{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{\lambda}{2} \left|\left|\mathbf{u} - \mathbf{f}\right|\right|_2^2 + \mu_k^\top \left(\mathbf{v}_k - \mathbf{D}\mathbf{u}\right), \tag{85}$$

and the $\mathbf{v}$ subproblem is

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}} \frac{\rho}{2} \left|\left|\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}\right|\right|_2^2 + \mu_k^\top \left(\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}\right) + \left|\left|\mathbf{v}\right|\right|_1 \tag{86}$$

The next subsection deals with the two subproblems in more detail.

[10] Apart from the Lagrange multiplier update.

9.2 Solving the subproblems

To minimize[11] the $\mathbf{u}$ subproblem, we expand (85) as follows

$$
\begin{aligned}
&= \frac{\lambda}{2}\left[\mathbf{u}^\top\mathbf{u} - 2\mathbf{u}^\top\mathbf{f} + \mathbf{f}^\top\mathbf{f}\right] + \mu^\top\mathbf{v} - \mu^\top\mathbf{D}\mathbf{u} \\
&= \frac{\lambda}{2}\mathbf{u}^\top\mathbf{u} - \lambda\mathbf{u}^\top\mathbf{f} + \frac{\lambda}{2}\mathbf{f}^\top\mathbf{f} + \mu^\top\mathbf{v} - \mu^\top\mathbf{D}\mathbf{u}. \qquad (87) \\
& \qquad (88)
\end{aligned}
$$

Differentiating (87) with respect to $\mathbf{u}$ and equating to zero i.e. $\frac{\partial}{\partial\mathbf{u}} = 0$ we arrive at

$$\lambda\mathbf{u} - \lambda\mathbf{f} - \mathbf{D}^\top\mu = 0,$$

rearranging the terms we obtain

$$\mathbf{u} = \mathbf{f} + \frac{1}{\lambda}\mathbf{D}^\top\mu.$$

Therefore, in each iteration count k, we solve

$$\mathbf{u}_{k+1} = \mathbf{f} + \frac{1}{\lambda}\mathbf{D}^\top\mu. \qquad (89)$$

Closely note that unlike ADMM, we do not need to find an inverse matrix as in (25). This imply that we do not need to solve any linear system of equations as in the update for $\mathbf{u}$ in ADMM. This can be seen as an advantage of AMA compared to ADMM in dealing with 1D signal TV denoising. AMA has cut the computational costs of solving a linear system in each iteration.

Similar to ADMM, the $\mathbf{v}$ can be written as

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ \frac{\rho}{2}\left|\left|\mathbf{v} - \left(\mathbf{D}\mathbf{u}_{k+1} - \frac{1}{\rho}\mu_k\right)\right|\right|_2^2 + ||\mathbf{v}||_1. \qquad (90)$$

with

$$\mathbf{x} = \mathbf{D}\mathbf{u}_{k+1} - \frac{1}{\rho}\mu_k,$$

we have

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ \frac{\rho}{2}||\mathbf{v} - \mathbf{x}||_2^2 + ||\mathbf{v}||_1,$$

Note that the signs are different in the above equation. This is due to the different in the positive and negative signs in the Lagrangian and augmented Lagrangian function used in AMA. Take a close look and compare the signs of the Lagrangian functions of AMA and ADMM. Finally we update the Lagrangian multiplier as follows

$$\mu_{k+1} = \mu_k + (\mathbf{v}_{k+1} - \mathbf{D}\mathbf{u}_{k+1}) \qquad (91)$$

As can bee seen, due to the AMA not having to solve a linear system in each iteration, it is expected that the CPU time for AMA would be faster than of ADMM for 1D signal denoising. To better understand the implementation of AMA a MATLAB code is provided in appendix G. In Figure 2, we give a simple comparison between the TV denoising using AMA and ADMM.

[11] Finding a saddle point of the Lagrangian function.

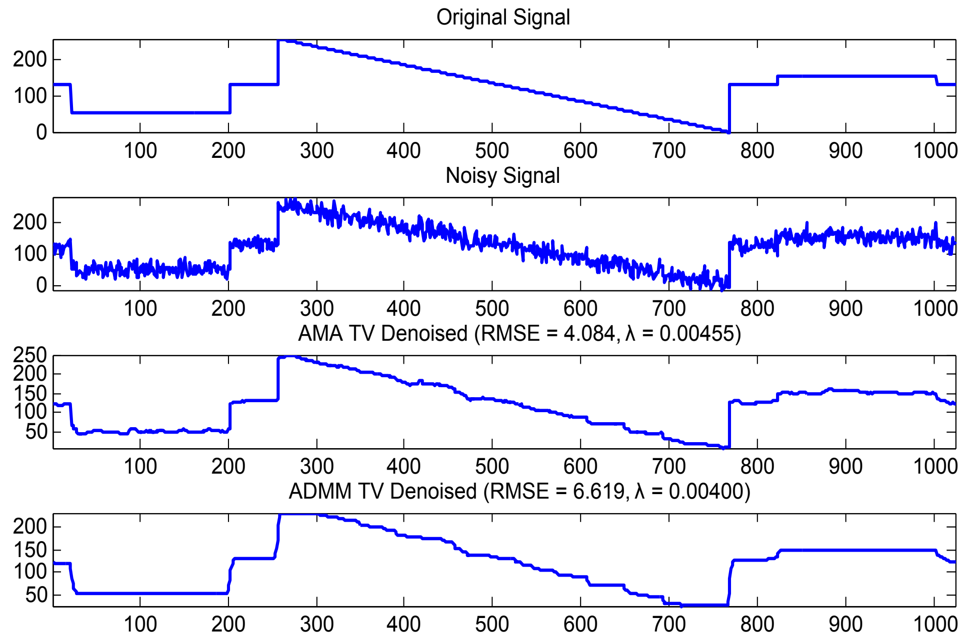

Figure 2: Comparison between AMA and ADMM for 1D signal denoising. The regularization λ, plays an important role to denoise the signal. Here, we have manually set the regularization parameter. An optimal regularization paramter is an open research subject.

Appendix

A Derivation of normal equation and Some comments on the proximal operator

A.1 Normal equation derivation

The normal equation (41) is obtained as follows. For ease of derivation, we drop k and expand (39)

$$\frac{\lambda}{2}\left[(\mathbf{Ku}-\mathbf{u})^T(\mathbf{Ku}-\mathbf{u})\right]-\mu^T\mathbf{v}+\mu^T\mathbf{Du}+\frac{r}{2}\left[(\mathbf{v}-\mathbf{Du})^T(\mathbf{v}-\mathbf{Du})\right]$$
$$=\frac{\lambda}{2}\left[\mathbf{K}^T\mathbf{Ku}^T\mathbf{u}-\mathbf{K}^T\mathbf{u}^T\mathbf{f}-\mathbf{f}^T\mathbf{Ku}+\mathbf{f}^T\mathbf{f}\right]-\mu^T\mathbf{v}+\mu^T\mathbf{Du}+$$
$$\frac{r}{2}\left[\mathbf{v}^T\mathbf{v}-\mathbf{v}^T\mathbf{Du}-\mathbf{D}^T\mathbf{u}^T\mathbf{v}+\mathbf{D}^T\mathbf{u}^T\mathbf{Du}\right]$$
$$=\frac{\lambda}{2}\left[\mathbf{K}^T\mathbf{Ku}^T\mathbf{u}-2\mathbf{K}^T\mathbf{u}^T\mathbf{f}+\mathbf{f}^T\mathbf{f}\right]-\mu^T\mathbf{v}+\mu^T\mathbf{Du}+\frac{r}{2}\left[\mathbf{v}^T\mathbf{v}-2\mathbf{v}^T\mathbf{Du}+\mathbf{D}^T\mathbf{u}^T\mathbf{Du}\right]$$
$$=\frac{\lambda}{2}\mathbf{K}^T\mathbf{Ku}^T\mathbf{u}-\lambda\mathbf{K}^T\mathbf{u}^T\mathbf{f}+\frac{\lambda}{2}\mathbf{f}^T\mathbf{f}-\mu^T\mathbf{v}+\mu^T\mathbf{Du}+\frac{r}{2}\mathbf{v}^T\mathbf{v}-r\mathbf{v}^T\mathbf{Du}+\frac{r}{2}\mathbf{D}^T\mathbf{u}^T\mathbf{Du},$$

Differentiation with respect to $\mathbf{u}$

$$\frac{\partial}{\partial\mathbf{u}}\left(\frac{\lambda}{2}\mathbf{K}^T\mathbf{Ku}^T\mathbf{u}-\lambda\mathbf{K}^T\mathbf{u}^T\mathbf{f}+\frac{\lambda}{2}\mathbf{f}^T\mathbf{f}-\mu^T\mathbf{v}+\mu^T\mathbf{Du}+\frac{r}{2}\mathbf{v}^T\mathbf{v}-r\mathbf{v}^T\mathbf{Du}+\frac{r}{2}\mathbf{D}^T\mathbf{u}^T\mathbf{Du}\right)$$
$$=\lambda\mathbf{K}^T\mathbf{Ku}-\lambda\mathbf{K}^T\mathbf{f}+\mathbf{D}^T\mu-r\mathbf{D}^T\mathbf{v}+r\mathbf{D}^T\mathbf{Du}.$$

Setting the derivative to zero

$$\lambda\mathbf{K}^T\mathbf{K}\mathbf{u} - \lambda\mathbf{K}^T\mathbf{f} + \mathbf{D}^T\mu - r\mathbf{D}^T\mathbf{v} + r\mathbf{D}^T\mathbf{D}\mathbf{u} = 0$$
$$\lambda\mathbf{K}^T\mathbf{K}\mathbf{u} + r\mathbf{D}^T\mathbf{D}\mathbf{u} = \lambda\mathbf{K}^T\mathbf{f} + r\mathbf{D}^T\mathbf{v} - \mathbf{D}^T\mu$$
$$\left(\lambda\mathbf{K}^T\mathbf{K}\mathbf{u} + r\mathbf{D}^T\mathbf{D}\right)\mathbf{u} = \lambda\mathbf{K}^T\mathbf{f} + r\mathbf{D}^T\mathbf{v} - \mathbf{D}^T\mu,$$

where the derivation is completed.

A.2 Comments on the proximal operator

Recall that in solving the $\mathbf{v}$ sub-problem, we set $\mathbf{x}_v = \mathbf{D}_v\mathbf{u} + \frac{1}{r}\mu_v$ and made use of the proximal operator. The reader might ask, where did this $\mathbf{x}_v$ came from ?. We restate the $\mathbf{v}$ sub-problem here

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ ||\mathbf{v}||_1 - \mu_k^T\left(\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}\right) + \frac{r}{2}\,||\mathbf{v} - \mathbf{D}\mathbf{u}_{k+1}||_2^2\,,$$

as before, we drop the index k for ease of derivation. Now, expanding the right hand side of the $\mathbf{v}$ subproblem we get

$$= ||\mathbf{v}||_1 - \mu^\top\mathbf{v} + \mu^\top\mathbf{D}\mathbf{u} + \frac{r}{2}\left[\mathbf{v}^\top\mathbf{v} - 2\mathbf{v}^\top\mathbf{D}\mathbf{u} + \mathbf{u}^\top\mathbf{D}^\top\mathbf{D}\mathbf{u}\right]. \tag{92}$$

Differentiating with respect to $\mathbf{v}$ we obtain the following

$$\frac{\partial}{\partial\mathbf{v}}\left(||\mathbf{v}||_1\right) - \mu + r\mathbf{v} - r\mathbf{D}\mathbf{u}.$$

Because the non-differentiable nature of the ℓ_1 norm, we leave out the term $||\mathbf{v}||_1$. Next, be equating the zero

$$\begin{aligned} -\mu + r\mathbf{v} - r\mathbf{D}\mathbf{u} &= 0 \\ r\mathbf{v} &= r\mathbf{D}\mathbf{u} + \mu \\ \mathbf{v} &= \frac{\mu}{r} + \mathbf{D}\mathbf{u}. \end{aligned} \tag{93}$$

Putting this into the $\mathbf{v}$ subproblem (28)

$$\mathbf{v}_{k+1} = \underset{\mathbf{v}}{\operatorname{argmin}}\ \frac{\rho}{2}\left|\left|\mathbf{v} - \left(\mathbf{D}\mathbf{u} + \frac{\mu}{\rho}\right)\right|\right|_2^2 + ||\mathbf{v}||_1\,, \tag{94}$$

with $r = \rho$.

B Remarks on ALM/ADMM

In the literature, the alternating direction method or alternating direction method of multipliers (ADM/ADMM) is a variant of the augmented Lagrangian method (ALM). ADM/ADMM is a splitting version of the ALM where the objective function is split into two sub-problems as discussed in this note. Next, the sub-problems are solved alternately in each iteration. Therefore, in this note we have used the ADM/ADMM method to solve the augmented Lagrangian function.

C ADMM for the nonconvex ℓ_p norm

We can use the ADMM algorithm to perform denoising or deblurring for the nonconvex ℓ_p norm, $||\mathbf{Dx}||_p^p$. Consider for example the problem

$$\begin{aligned} \arg\min_{\mathbf{x}} \frac{1}{2} ||\mathbf{y} - \mathbf{Hx}||_2^2 + \mu ||\mathbf{v}||_p^p \\ \text{s.t. } \mathbf{v} = \mathbf{Dx}, \end{aligned} \tag{95}$$

and its associated augmented Lagrangian function

$$\mathcal{L}(\mathbf{x}, \mathbf{v}, \gamma) = \frac{1}{2} ||\mathbf{y} - \mathbf{Hx}||_2^2 + \mu ||\mathbf{v}||_p^p - \gamma^\top (\mathbf{v} - \mathbf{Dx}) + \frac{\rho}{2} ||\mathbf{v} - \mathbf{Dx}||_2^2 . \tag{96}$$

We obtain two sub-problems and minimize it alternately,

$$\begin{aligned} \mathbf{x}_{k+1} &= \min_{\mathbf{x}} \frac{1}{2} ||\mathbf{y} - \mathbf{Hx}||_2^2 - \gamma_k^\top (\mathbf{v}_k - \mathbf{Dx}) + \frac{\rho}{2} ||\mathbf{v}_k - \mathbf{Dx}||_2^2 , \\ \mathbf{v}_{k+1} &= \min_{\mathbf{v}} \mu ||\mathbf{v}||_p^p - \gamma_k^\top \left(\mathbf{v} - \mathbf{Dx}^{k+1} \right) + \frac{\rho}{2} \left|\left| \mathbf{v} - \mathbf{Dx}^{k+1} \right|\right|_2^2 . \end{aligned}$$

The $\mathbf{x}$ sub-problem as already mentioned, is a least squares problem and can be solved as the $\mathbf{u}$ sub-problem in subsection 5.2. The main challenge now, is to solve the $\mathbf{v}$ sub-problem. This is because we now have a nonconvex ℓ_p regularizer instead of a convex ℓ_1 regularizer. Several techniques exist to solve problems involving nonconvex regularizers. In this note, we will solve the $\mathbf{v}$ sub-problem using the Iteratively Re-weighted L1 (IRL1) algorithm.

C.1 Iteratively Re-weighted L1 algorithm

Note that the $\mathbf{v}$ sub-problem is a denoising step (recall the proximal operator). Suppose $\mathbf{u} = \mathbf{Dx}^{k+1} + \frac{\gamma_k}{\rho}$, then we can re-write the $\mathbf{v}$ sub-problem as

$$\begin{aligned} \mathbf{v}_{k+1} &= \min_{\mathbf{v}} \mu ||\mathbf{v}||_p^p + \frac{\rho}{2} \left|\left| \mathbf{v} - \left(\mathbf{Dx}^{k+1} + \frac{\gamma_k}{\rho} \right) \right|\right|_2^2 . \\ &= \min_{\mathbf{v}} \mu ||\mathbf{v}||_p^p + \frac{\rho}{2} ||\mathbf{v} - \mathbf{u}||_2^2 . \end{aligned} \tag{97}$$

The IRL1 algorithm used to solve (97) does so by approximating the problem with the following approximation

$$\mathbf{v}_{k+1} = \min_{\mathbf{v}} \frac{\rho}{2} ||\mathbf{v} - \mathbf{u}||_2^2 + \sum_i \mu p \left(\left| v_i^k \right| + \epsilon \right)^{p-1} |v_i| . \tag{98}$$

If we define the weight $\mathbf{w}$ as

$$\mathbf{w}_i = \frac{\mu p}{\left(\left| v_i^k \right| + \epsilon \right)^{1-p}}, \tag{99}$$

then, we can re-write the $\mathbf{v}$ sub-problem as

$$\mathbf{v}_{k+1} = \min_{\mathbf{v}} \frac{\rho}{2} ||\mathbf{v} - \mathbf{u}||_2^2 + \sum_i \mathbf{w}_i |v_i| . \tag{100}$$

Table 1: ADMM for minimization problem (101)

Initialize $\mathbf{x}_1^0$, $\mathbf{x}_2^0$, μ^0 and $\rho > 0$
while not converged

1. $\mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1} f(\mathbf{x}_1) + \frac{\rho}{2}\left|\left|\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2^k - \mathbf{d} + \frac{\mu^k}{\rho}\right|\right|_2^2$
2. $\mathbf{x}_2^{k+1} = \arg\min_{\mathbf{x}_2} g(\mathbf{x}_2) + \frac{\rho}{2}\left|\left|\mathbf{A}\mathbf{x}_1^{k+1} + \mathbf{B}\mathbf{x}_2 - \mathbf{d} + \frac{\mu^k}{\rho}\right|\right|_2^2$
3. $\mu^{k+1} = \mu^k + \rho\left(\mathbf{A}\mathbf{x}_1^{k+1} + \mathbf{B}\mathbf{x}_2^{k+1} - \mathbf{d}\right)$
4. $k = k + 1$

end while

In each iteration of the ADMM algorithm, the weight $\mathbf{w}_i$ is updated (re-weighting process) until the minimum solution is achieved.

D ADMM and relations to nonlinear block Gauss-Siedel method

A normal two block ADMM solves the following problem

$$\begin{aligned} &\min_{\mathbf{x}_1,\mathbf{x}_2} \quad f(\mathbf{x}_1) + g(\mathbf{x}_2) \\ &\text{s.t} \quad \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 = \mathbf{d} \\ &\mathbf{x}_i \in \mathcal{X}_i,\ i = 1, 2 \end{aligned} \tag{101}$$

where $f(\cdot), g(\cdot) : \mathcal{X}_i \to \mathbb{R}$ are closed convex functions, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{l\times m_i}$ are linear transforms, $\mathcal{X}_i \in \mathbb{R}^{m_i}$ are nonempty closed convex sets, and $\mathbf{d} \in \mathbb{R}^l$ is a given vector. The augmented Lagrangian function for the problem (101) is

$$\begin{aligned} \mathcal{L}_A(\mathbf{x}_1, \mathbf{x}_2; \rho) = f(\mathbf{x}_1) + g(\mathbf{x}_2) + \mu^\top(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 - \mathbf{d}) \\ + \frac{\rho}{2}||\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 - \mathbf{d}||_2^2 \end{aligned} \tag{102}$$

where $\mu \in \mathbb{R}^l$ is the Lagrange multiplier and $\rho > 0$ is a penalty parameter. The objective is to find the saddle point of $\mathcal{L}$ by alternatively minimizing $\mathcal{L}$ with respect to $\mathbf{x}_1$, $\mathbf{x}_2$ and μ. The ADMM algorithm to solve problem (101) is presented as Algorithm 1.

Note that steps 1 and 2 in Table 1 exhibit a nonlinear Gauss-Siedel type updating for variables $\mathbf{x}_1$ and $\mathbf{x}_2$. That is, the most recent value of the variables are used in the update.

E ADMM for image restoration code

Here, we list the MATLAB code to perform ADMM for both Deblurring and Denoising. The filename "ALMTV" stands for Augmented Lagrangian Method Total Variation however, the scheme used is ADMM.

Deblurring code: ALMTV.m

```
function out = ALMTV(g,Img,H,lam,rho,Nit,tol)
%%========== ALMTV FOR DEBLURRING !!!=================================
%%
%==========Anisotropic Total variation using Augmented Lagrangian====
% ALMTV: Augmented Lagrangian Method Total Variation
% The original paper [1] Takes into account image and video restorations.
% This code is a little modification for just gray scale image restoration.
% original codes can be found here:

% http://www.mathworks.com/matlabcentral/fileexchange/
%       43600-deconvtv-fast-algorithm-for-total-variation-deconvolution

% This code does not support video restoration like [1]. Only the
% mathematical theory used in [1] to code this program.

%inputs:
%       g       : Observed blurred(and possibly noisy) image
%       Img     : Original image(clean/unblurred)
%       H       : Point spread function/Blurring kernel (A linear operator)
%       lam     : regularization parameter
%       rho     : regularization parameter of the Augmented Lagrangian form
%                 of the main problem.
%       Nit     : Number of iterations
%       tol     : Error tolerance for stopping criteria

%============== References====================
% See the following papers

% [1] Chan, Stanley H., Ramsin Khoshabeh, Kristofor B. Gibson, Philip E. Gill,
%     and Truong Q. Nguyen. "An augmented Lagrangian method for total variation
%     video restoration." Image Processing, IEEE Transactions on 20, no. 11
%     (2011): 3097-3111.

%  [2] Chan, Stanley H., Ramsin Khoshabeh, Kristofor B. Gibson, Philip E. Gill, and
%      Truong Q. Nguyen. "An augmented Lagrangian method for video restoration." In
%      Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International
%      Conference on, pp. 941-944. IEEE, 2011.

% The program solves the following core objective function

%   min_f   lam/2*||Hf - g||^2 + ||Df||_1

%%

[row,col] = size(g);
f         = g;
```

```
u1        = zeros(row,col); %Initialize intermediat variables for u subproblem
u2        = zeros(row,col); %     "       "           "       for u subproblem

y1        = zeros(row,col); %Initialize Lagrange Multipliers
y2        = zeros(row,col); %   "       Lagrange Multipliers

relError     = zeros(Nit,1); % Compute error relative to the previous iteration.
relErrorImg  = relError;     %Compute error relative to the original image
psnrGain     = relError;     % PSNR improvement every iteration
funcVal      = relError;     %Function value at each iteration

eigHtH  = abs(fft2(H,row,col)).^2; %eigen value for HtH
eigDtD  = abs(fft2([1 -1], row, col)).^2 + abs(fft2([1 -1]', row, col)).^2; % eigen value ofDtD
%eigB    = abs(fft2([0 -2 0; -2 9 -2;0 -2 0],row,col)).^2;
Htg     = imfilter(g, H, 'circular');

[D,Dt]      = defDDt(); %Declare forward finite difference operators
[Df1, Df2] = D(f);

%=================== Main algorithm starts here ======================

    for k=1:Nit

        %Solving the f subproblem
        f_old   = f;
        rhs     = lam*Htg + Dt(u1 - (1/rho)*y1, u2 - (1/rho)*y2);
        eigA    = lam*eigHtH + rho*eigDtD;
        f       = fft2(rhs)./eigA;
        f       = real(ifft2(f));

        %Solving the u subproblem
        [Df1, Df2]  = D(f);
        v1          = Df1 + (1/rho)*y1;
        v2          = Df2 + (1/rho)*y2;

        u1          = shrink(v1,1/lam);
        u2          = shrink(v2,1/lam);

        %Update y, the Lagrange multipliers
        y1          = y1 - rho*(u1 - Df1);
        y2          = y2 - rho*(u2 - Df2);

        relError(k)    = norm(f - f_old,'fro')/norm(f, 'fro');
        relErrorImg(k) = norm(Img - f,'fro')/norm(Img,'fro');
        psnrGain(k)    = psnr_fun(f,Img);

        r1          = imfilter(f, H, 'circular')-g;
        funcVal(k)  = (lam/2)*norm(r1,'fro')^2 + sum(Df1(:)+Df2(:));

        if relError(k) < tol
            break
        end
    end

%======================= Results ==============================
```

```
out.psnrf                = psnr_fun(f, Img);
out.ssimf                = ssim_index(f,Img);
out.sol                  = f;              %Deblurred image
out.relativeError        = relError(1:k);
out.relativeErrorImg     = relErrorImg(1:k);
out.psnrGain             = psnrGain(1:k)
out.functionValue        = funcVal(1:k);
end

function [D,Dt] = defDDt()
D  = @(U) ForwardDiff(U);
Dt = @(X,Y) Dive(X,Y);
end

function [Dux,Duy] = ForwardDiff(U)
 Dux = [diff(U,1,2), U(:,1,:) - U(:,end,:)];
 Duy = [diff(U,1,1); U(1,:,:) - U(end,:,:)];
end

function DtXY = Dive(X,Y)
  % Transpose of the forward finite difference operator
  % is the divergence fo the forward finite difference operator
  DtXY = [X(:,end) - X(:, 1), -diff(X,1,2)];
  DtXY = DtXY + [Y(end,:) - Y(1, :); -diff(Y,1,1)];
end

function z = shrink(x,r)
z = sign(x).*max(abs(x)-r,0);
end
```

Demo code: ALMTV_Demo

This is the demo code that calls the deblurring code "ALMTV.m".

```
% ================ Demo for grayscale image deblurring=================
% This code demostrates the use of the ALMTV (Augmented Lagrangian Method
% Total Variation) code.

clc
clear all;
close all;

Img = double(imread('mosque256.bmp')); %Your Image goes here
%I = double(Img);

%H = ones(9,9)/81;
H = fspecial('gaussian', [7 7], 5);
%H = fspecial('motion',20,45); %Try a motion blur
g = imfilter(Img,H,'circular');

BSNR = 30;
sigma = BSNR2WGNsigma(g, BSNR);
g = g +  sigma * randn(size(Img)); %Add a little noise

lam     = [200];
```

```
%lam     = [500 600 700 800 900 100000]; %Regularization parameter
%lam     = [20 40 60 80 100 120 140 160 180 200 220 240 260];
res     = cell([1 size(lam,2)]);
rho     =2; %default 2
Nit     = 400;
tol     = 1e-5;

%=============Deblurr algorithm==========
for k=1:length(lam)
    tg = tic;
    out = ALMTV(g,Img,H,lam(k),rho,Nit,tol);
    tg = toc(tg);
    res{1,k} = out;
end
%========================================


figure;
imshow(uint8(out.sol));
title('Deblurred');

figure;
imshow(uint8(g));
title('Blurred');

figure;
imshow(uint8(Img));
title('Original');
```

F ADMM code

MATLAB function to perform ADMM 1D total variation denoising.

TV Denoising code: ADMM_1D.m

```
% Created by Tarmizi Adam on 5/3/2017
%This is a simple program for 1D Signal denoising using total variation
% denoising. The iterative algorithm used in this program is the
% Alternating Direction Methods of Multiplier (ADMM) which is a splitting
% version of the classical Augmented Lagrangian Method (ALM).

% This solver solves the following minimization problem:

%     min_x  lambda/2*||u - f||^2_2 + ||Du||_1

%   Input:
%         f = The noisy observed signal (1D-signal)
%       lam = Regularization parameter. Controls the smoothness of the
%             denoised signal.
%       rho = Regularization parameter related to the constraint (ADMM)
%       Nit = Number of iterations to run the ADMM iterative method.
%
%   Output:
%        out.sol         = The denoised signal
%        out.funVal      = Plot of the convergence of our objective
%                          function
%%

f          = f(:);
u          = f;          %Initialize
N          = length(f);
mu         = zeros(N,1); % Lagrange multiplier
v          = mu; %Initialize the v sub-problem
funcVal    = zeros(Nit,1);

[D,DT,DTD] = defDDt(N);

    for k = 1:Nit
        u_old = u;

        %% u sub-problem %%
        % Solves a linear system using the conjugate gradient method.
        [u,~] = cgs(rho*DTD + lam*speye(N),lam*f+rho*DT*(v - mu/rho),1e-5,100);

        %% v sub-problem %%
        x     = D*u + mu/rho;
        v     = shrink(x, 1/rho);

       %% Update Lagrange multiplier
        mu    = mu + (D*u - v);

        r1    = u - f;
        funcVal(k) = (lam/2)*norm(r1,'fro')^2 + sum(v(:));
```

```
    end

    out.sol = u;
    out.funVal = funcVal(1:k);

end

function [D,DT,DTD] = defDDt(N)
%Create a first order difference matrix D
e = ones(N,1);
B = spdiags([e -e], [1 0], N, N);
B(N,1) =  1;

D = B;
clear B;
% Create the transpose of D
DT = D'; %Remember that DT = -D, also called the backward difference.
DTD = D'*D;
end

function z = shrink(x,r)
z = sign(x).*max(abs(x)- r,0);
end
```

G AMA code

MATLAB function to perform AMA 1D total variation denoising.

TV Denoising code: AMA_1D.m

```
function out = AMA_1D(f,lam,rho, Nit)
% Created by Tarmizi Adam on 24/5/2017
%This is a simple program for 1D Signal denoising using total variation
% denoising. The iterative algorithm used in this program is the
% Alternating Minimization Algorithm (AMA) proposed by Paul Tseng in the
% paper titled:
%             "Application of a Splitting Algorithm to Decomposition in
%             Convex Programming and Variational Inequalities"

% This solver solves the following minimization problem:

%    min_x  lambda/2*||u - f||^2_2 + ||Du||_1

%   Input:
%         f = The noisy observed signal (1D-signal)
%       lam = Regularization parameter. Controls the smoothness of the
%             denoised signal.
%       rho = Regularization parameter related to the constraint (ADMM)
%       Nit = Number of iterations to run the ADMM iterative method.
%
%   Output:
%        out.sol         = The denoised signal
```

```
%        out.funVal      = Plot of the convergence of our objective
%                          function
%
% Remarks: This code is based on the derivation in the authors ADMM notes
%          in the subsection AMA TV denoising.
%%

f          = f(:);
u          = f;          %Initialize
N          = length(f);
mu         = zeros(N,1); % Lagrange multiplier
v          = mu; %Initialize the v sub-problem
funcVal    = zeros(Nit,1);

[D,DT,DTD] = defDDt(N);

    for k = 1:Nit
        u_old = u;

        %% u sub-problem %%

        u = f + (1/lam)*DT*mu; %Main difference with ADMM. No need to solve linear system.

        %% v sub-problem %%
        x    = D*u - mu/rho;
        v    = shrink(x, 1/rho);

       %% Update Lagrange multiplier
        mu   = mu + (v - D*u);

        r1   = u - f;
        funcVal(k) = (lam/2)*norm(r1,'fro')^2 + sum(v(:));

    end

    out.sol = u;
    out.funVal = funcVal(1:k);

end

function [D,DT,DTD] = defDDt(N)
%Create a first order difference matrix D
e = ones(N,1);
B = spdiags([e -e], [1 0], N, N);
B(N,1) =  1;

D = B;
clear B;
% Create the transpose of D
DT = D'; %Remember that DT = -D, also called the backward difference.
DTD = D'*D;
end

function z = shrink(x,r)
z = sign(x).*max(abs(x)- r,0);
end
```

References/Bibliography

[1] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.

[2] H. Woo and S. Yun, "Alternating minimization algorithm for speckle reduction with a shifting technique," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1701–1714, 2012.

[3] S. Yun and H. Woo, "Linearized proximal alternating minimization algorithm for motion deblurring by nonlocal regularization," *Pattern Recognition*, vol. 44, no. 6, pp. 1312–1326, 2011.

[4] J. Eckstein and W. Yao, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," *RUTCOR Research Reports*, vol. 32, 2012.

[5] X.-C. Tai and C. Wu, "Augmented lagrangian method, dual methods and split bregman iteration for rof model," in *Scale space and variational methods in computer vision.* Springer, 2009, pp. 502–513.

[6] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.

[7] C. A. Bouman, "Model based image processing," *Purdue University*, 2013.

[8] M. Tao, J. Yang, and B. He, "Alternating direction algorithms for total variation deconvolution in image reconstruction," *Research rep., Rice University*, 2009.