

Report for Homework 1

Runyu Zhang

November 4, 2018

1 Problem Description

Use the total variation model to deblur and denoise images.

$$\hat{u} = \arg \min_u \lambda \int |\nabla u| dx + \frac{1}{2} \int (\mathcal{A}u - f)^2 dx$$

where f is the image to be deblurred and denoised. \mathcal{A} is the convolution operator with kernel k :

$$\mathcal{A}u = k * u.$$

Here we choose the kernel to be gaussian. In implementation k is generated by the matlab command `k = fspecial('gaussian', [s,s], sigma)`. Here s and σ are predetermined parameters for the model.

2 Scheme Design

2.1 Formulation of the Optimization Algorithm

We use ADMM to solve this optimization problem:

$$\min_u \lambda \|Wu\|_1 + \frac{1}{2} \|Au - f\|_2^2,$$

where A and W are the discretized convolution and gradient operator respectively.

The problem can be reformulated as:

$$\begin{aligned} \min_{u,d} \lambda \|d\|_1 + \frac{1}{2} \|Au - f\|_2^2 \\ \text{s.t } d = Wu. \end{aligned}$$

Write out the augment Lagrangian as follow:

$$\mathcal{L}(u, d, p) = \lambda \|d\|_1 + \frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu - d \rangle + \frac{\mu}{2} \|Wu - d\|_2^2$$

Then the ADMM scheme can be derived by solving the following problem using coordinate descent:

$$\min_{u,d} \max_p \mathcal{L}(u, d, p).$$

i.e.:

$$\begin{aligned} d_k &= \arg \min_d \mathcal{L}(u_{k-1}, d, p_{k-1}) \\ &= \text{shrink}(Wu_{k-1} + \frac{1}{\mu}p_{k-1}, \frac{\lambda}{\mu}) \\ u_k &= \arg \min_u \mathcal{L}(u, d_{k-1}, p_{k-1}) \\ &= (A^T A + \mu W^T W)^{-1} (A^T f + \mu W^T d - W^T p) \\ p_k &= p_{k-1} + \mu(Wu - d) \end{aligned}$$

2.2 Boundary Condition for the Convolution Operator

The updating of u_k is the step that takes the most computational time, since it requires solving a linear equation system. However, if the boundary condition for both the convolution operator and the gradient operator is periodic, then this step can be accelerated via Fast Fourier Transform.

The updating of u_k can be written as:

$$\begin{aligned} u_k &= \arg \min_u \mathcal{L}(u, d_{k-1}, p_{k-1}) \\ &= \arg \min_u \lambda \|d\|_1 + \frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu - d \rangle + \frac{\mu}{2} \|Wu - d\|_2^2 \\ &= \arg \min_u \frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu \rangle + \frac{\mu}{2} \|Wu - d\|_2^2 \\ &= \arg \min_u \frac{1}{2} \|Au - f\|_2^2 + \frac{\mu}{2} \|Wu - (d - \frac{1}{\mu}p)\|_2^2 \end{aligned}$$

According to Parseval Equation, we have

$$\begin{aligned} u_k &= \arg \min_u \frac{1}{2} \|\mathcal{F}(A) \cdot \mathcal{F}(u) - \mathcal{F}(f)\|_2^2 + \frac{\mu}{2} \|\mathcal{F}(W) \cdot \mathcal{F}(u) - \mathcal{F}(d - \frac{1}{\mu}p)\|_2^2 \\ &= \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(A)}\mathcal{F}(f) + \mu \overline{\mathcal{F}(W)}\mathcal{F}(d - \frac{1}{\mu}p)}{\overline{\mathcal{F}(A)}\mathcal{F}(A) + \mu \overline{\mathcal{F}(W)}\mathcal{F}(W)} \right) \end{aligned}$$

Here, \mathcal{F} denotes Fourier Transform, the multiplication and division are element-wise multiplication and division.

2.3 Derivation of the Transpose Operator

If the boundary condition for the convolution operator is not periodic, then we need to solve the linear equation system:

$$(A^T A + \mu W^T W)u = A^T f + \mu W^T (d - \frac{1}{\mu} p)$$

Here we use conjugate gradient to solve the following equation. Since the gradient operator can also be viewed as a special convolution operator, we only need to find the operator that corresponds to the transpose of convolution. For simplicity, we consider 1-d convolution here, but the derivation of 2-d convolution is almost identical.

By the definition of transpose, we have:

$$\begin{aligned} \langle A^T y, x \rangle &= \langle y, Ax \rangle = \langle y, k * x \rangle \\ &= \sum_{i=0}^n y_i \left(\sum_{j=0}^i k_{i-j} x_j \right) \\ &= \sum_{j=0}^n x_j \sum_{i=j}^n k_{i-j} y_i \end{aligned}$$

Thus we know that:

$$(A^T y)_j = \sum_{i=j}^n k_{i-j} y_i$$

Thus we have:

$$(A^T y)_{n-j} = \sum_{i=n-j}^n k_{i-n+j} y_i$$

By doing the index substitution: $i' = n - i$, we have:

$$(A^T y)_{n-j} = \sum_{i'=0}^j k_{j-i'} y_{n-i'}$$

Thus, if we define a new vector y' so that: $y'_i = y_{n-i}$, we will have:

$$(A^T y)_{n-j} = \sum_{i=0}^j k_{j-i} y'_i$$

Thus the transpose operator can be calculated as:

Algorithm 1 Transpose Operator

Input: The convolution kernel k , the vector y

Output: The vector $A^T y$, which is the transpose of the convolution acting on the vector y

- 1: Reverse the vector y : $y = y(end : -1 : 1)$;
 - 2: convolve the vector y with k : $y = k * y$;
 - 3: Reverse the vector again: $y = y(end : -1 : 1)$;
 - 4: Output y .
-

2.4 Duality of the Optimization Problem

Since we use ADMM to solve the optimization problem, it might be helpful to write out the duality of the primal problem. The duality gap can also be used as a stopping criterion for the algorithm.

The primal problem is:

$$\begin{aligned} \min_{u,d} \quad & \lambda \|d\|_1 + \frac{1}{2} \|Au - f\|_2^2 \\ \text{s.t.} \quad & d = Wu. \end{aligned}$$

The Lagrangian of this problem is:

$$\begin{aligned} L(u, d, p) &= \frac{1}{2} \|Au - f\|_2^2 + \lambda \|d\|_1 + \langle p, Wu - d \rangle \\ &= (\lambda \|d\|_1 - \langle p, d \rangle) + \left(\frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu \rangle \right) \end{aligned}$$

The dual problem is $\inf_{u,d} (\lambda \|d\|_1 - \langle p, d \rangle) + \left(\frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu \rangle \right)$. In order for the first term $\inf_d \lambda \|d\|_1 - \langle p, d \rangle$ not to be $-\infty$, we must and the constraint :

$$\lambda \mathbf{1} \geq p \geq -\lambda \mathbf{1}.$$

For the second term: $\inf_u \left(\frac{1}{2} \|Au - f\|_2^2 + \langle p, Wu \rangle \right)$, we have:

$$u = (A^T A)^{-1} (A^T f - W^T p)$$

Thus we have:

$$\inf_u = -\frac{1}{2} (A^T f - W^T p)^T (A^T A)^{-1} (A^T f - W^T p) + \frac{1}{2} f^T f.$$

Then the dual problem is

$$\begin{aligned} \max_p \quad & -\frac{1}{2} (A^T f - W^T p)^T (A^T A)^{-1} (A^T f - W^T p) + \frac{1}{2} f^T f \\ \text{s.t.} \quad & \lambda \mathbf{1} \geq p \geq -\lambda \mathbf{1}. \end{aligned}$$

The duality gap can be used as a stopping criterion, but strangely, for this problem, **the duality gap does not seem to converge to zero for $\lambda \leq 10$!** But still, the deblurring and denoising results are acceptable. I still haven't figured out what is the problem up till now.

3 Numerical Results

For more detailed experimental results, see Appendix and 'results.pdf'

3.1 Periodic Boundary Condition

Table 1: Experimental Results for Periodic Boundary Condition

| Image | Kamiya | Shizuwo | Classic | Koala |
|----------------|---------|---------|---------|---------|
| recovered PSNR | 28.3532 | 25.4685 | 24.3043 | 29.8847 |
| blurred PSNR | 23.4197 | 22.8125 | 23.2668 | 27.9442 |
| recovered SSIM | 0.9315 | 0.9258 | 0.9227 | 0.9127 |
| blurred SSIM | 0.8434 | 0.853 | 0.9046 | 0.8725 |

For an image of size 716*500*3, the running time is 14.5 s

3.2 Non-Periodic Boundary Condition

Table 2: Experimental Results for Non-Periodic Boundary Condition

| Image | Kamiya | Shizuwo | Classic | Koala |
|----------------|---------|---------|---------|---------|
| recovered PSNR | 28.0124 | 25.4396 | 24.293 | 29.957 |
| blurred PSNR | 22.5695 | 22.2524 | 22.7652 | 27.6617 |
| recovered SSIM | 0.9128 | 0.9132 | 0.9222 | 0.9132 |
| blurred SSIM | 0.8193 | 0.8355 | 0.8979 | 0.8708 |

For an image of size 716*500*3, the running time is 139 s

3.3 Some Observation

- The larger λ is, the more robust to noise is the algorithm.
- Smaller λ seems to lead to better deblurring results, but may have ringing effects caused by noise.

- The choice of μ cannot be too large, or the recovered image will still be blurred. (See 'results.pdf')
- We don't need to wait until the algorithm converges to the optimal points. Results gotten by twenty iterations does not seem to have much difference between results gotten by a hundred iterations.

4 Appendix

Figure 1: **Experimental Results for Periodic Boundary Condition:** The picture on the left is the blurred image, on the right the recovered

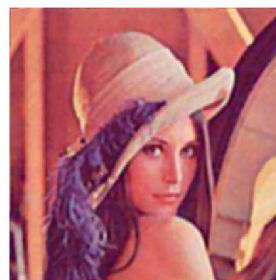
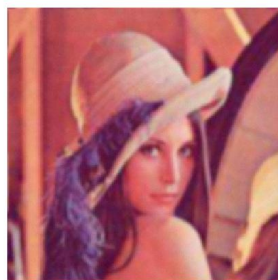


Figure 2: **Experimental Results for Non- Periodic Boundary Condition:** The picture on the left is the blurred image, on the right the recovered

