

Coding Project - Compressed Sensing

Wonnetz Phanthavong

November 17, 2020

1 Introduction and Overview

Compressed Sensing involves reconstructing a sparse signal $x \in \mathbb{R}^n$. This takes the form $b = Ax$, where $b \in \mathbb{R}^n$ is our sampled vector and $A \in \mathbb{R}^{m \times n}$ is our predefined sampling matrix. Note, A can be populated with either Gaussian or Bernoulli entries, the key to A is to have some randomness built into it [1]. In this case we shall use Gaussian entries to populate our sampling matrix A . It has been shown that we can solve the Compressed Sensing problem through

$$\min_x \{\|x\|_0\} \quad \text{subject to } Ax = b$$

However, this is a NP hard problem, which means that this problem is difficult to compute. Thus, we shall instead opt to solve the below problem, which still has a high probability of reconstructing x [2].

$$\min_x \{\|x\|_1 + \frac{\lambda}{2\tau} \|Ax - b\|_2^2\} \quad (1)$$

A natural question here is, "Why not use the L2 norm $\|\cdot\|_2$ "? Observe that for our vector x , if we were to use the L2 norm to minimize the problem, then there is a high probability to recover a non-sparse solution [1]. The L1 norm, $\|\cdot\|_1$, allows us to reconstruct a solution that is more likely to be sparse. Intuitively, one can observe the geometry of the L1 norm and L2 norm to see this difference. Supposing that we are working in \mathbb{R}^2 and we have sampled a vector. Minimizing the L1 norm gives us a rhombus that grows to intersect the vector on either the x-axis or y-axis, where as minimizing the L2 norm gives us a circle that grows to intersect the vector at some point that is a combination of both x and y. Hence, minimizing the L1 norm gives us a sparse vector, where as minimizing the L2 norm gives us a non-sparse vector.

For notation sake, sometimes we will use the following, let $\phi(x) = \|x\|_1$ and $f(x) = \frac{\lambda}{2} \|Ax - b\|_2^2$. Then we will have,

$$\min_x \{\phi(x) + f(x)\}$$

Note, λ is a user chosen parameter and $\tau \in (0, \frac{2}{L})$, where L is the Lipschitz constant of ∇f , i.e. $L = \lambda \|A^T A\|_2$. Here we shall use $\lambda = 2$ and $\tau = \frac{1}{L}$. First we shall observe what happens when we attempt to solve (1) through the Proximal Gradient Descent and then Tseng's Accelerated Proximal Gradient Method.

2 Regular Proximal Operator Method

First we shall describe how to compute the Proximal Operator on the Gradient Descent step, shown below,

$$prox_{\tau\phi} := argmin_{x \in \mathbb{R}^n} \{\phi(x) + \frac{1}{2} \|x - z\|^2\}$$

To figure this out, we shall first work out the Proximal Operator of the Absolute Value function $|\cdot|$, below

$$prox_{\tau|\cdot|}(x) = argmin_{x \in \mathbb{R}^n} \{|x| + \frac{1}{2\tau} |x - z|^2\}$$

As we are working with absolute values, naturally, let us consider two different cases. The case when $x \geq 0$ and the case when $x \leq 0$.

Case 1: $x \geq 0$

Observe that this reduces the Proximal Operator to

$$\text{prox}_{\tau|\cdot|}(x) = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ x + \frac{1}{2\tau}(x - z)^2 \right\}$$

Through some Calculus 1, we find that the minimizer for the above will be given as

$$x = z - \tau$$

Recall that we restricted our x to be $x \geq 0$, hence we have one more condition.

Hence, $x \geq 0 \implies z - \tau \geq 0 \implies z \geq \tau$, then for $z < \tau$ we shall then set $x = 0$.

Hence, the closed form formula will be given as

$$\text{prox}_{\tau|\cdot|}(x) = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ x + \frac{1}{2\tau}(x - z)^2 \right\} = \begin{cases} x = z - \tau & \text{for } z \geq \tau \\ x = 0 & \text{for } z < \tau \end{cases}$$

Case 2: $x \leq 0$

Similar to how we handled Case 1, we shall arrive at

$$\text{prox}_{\tau|\cdot|}(x) = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ -x + \frac{1}{2\tau}(x - z)^2 \right\} = \begin{cases} x = z + \tau & \text{for } z \leq -\tau \\ x = 0 & \text{for } z > -\tau \end{cases}$$

Putting Case 1 and Case 2 together gives us

$$\text{prox}_{\tau|\cdot|}(x) = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ |x| + \frac{1}{2\tau}|x - z|^2 \right\} = \begin{cases} x = z - \tau & \text{for } z \geq \tau \\ x = z + \tau & \text{for } z \leq -\tau \\ x = 0 & \text{for } -\tau < z < \tau \end{cases}$$

Now that we have figured out the Proximal Operator on the Absolute Value function, let us see how this relates to the main problem at hand.

Observe that for,

$$\text{prox}_{\tau\phi} := \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \|x\|_1 + \frac{\lambda}{2}\|x - z\|_2^2 \right\}$$

we can view this as minimizing each vector component for $\|x\|_1$ and $\frac{\lambda}{2}\|x - z\|_2^2$. Which means that we can perform the Proximal Operator on the Absolute Value function from $i \rightarrow n$ for $|x_k^{(i)}|$ and $\frac{\lambda}{2}(x_k^{(i)} - z_k^{(i)})^2$ where k represents the iteration number and i represents the i^{th} component of the vectors.

3 Accelerated Proximal Gradient Method

Now we shall figure out how to implement the Accelerated Proximal Gradient Method [4]. To reiterate the problem, the nonsmooth convex optimization problem that we are working with is given as

$$\min_x \{f(x) + \phi(x)\}$$

In light of Nesterov extending his third method with this Bregman function, we shall choose the same one. Let $D(x, y) = \frac{1}{2}\|x - y\|_2^2$ be our Bregman function [3].

The key step here involves figuring out how to implement

$$z_{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \{l_f(x; y_k) + \theta_k LD(x, z_k)\}$$

Where,

$$l_f(x; y) := f(y) + \langle \nabla f(y), x - y \rangle + \phi(x)$$

$$\theta_0 = 1 \text{ and } \theta_k = \frac{2}{k+2}$$

$$D(x, y) = \frac{1}{2} \|x - y\|_2^2$$

Note, $f(x)$, $\phi(x)$, L are chosen as in **Section 2**.

Plugging everything in, we have

$$z_{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \frac{\lambda}{2} \|Ay_k - b\|_2^2 + \langle \nabla f(y_k), x - y_k \rangle + \|x\|_1 + \theta_k \frac{L}{2} \|x - z_k\|_2^2 \right\}$$

Let us focus primarily on the 1-Dimensional case of the above, given below

$$\operatorname{argmin}_{x \in \mathbb{R}} \left\{ \frac{\lambda}{2} ((Ay_k)^{(i)} - b^{(i)})^2 + \langle \nabla f(y_k)^{(i)}, x - y_k^{(i)} \rangle + |x| + \theta_k \frac{L}{2} (x - z_k^{(i)})^2 \right\}$$

Familiarly, the above looks similar to the Proximal Operator case!

Hence, by observing the cases when $x \leq 0$ and $x \geq 0$, we find the closed form of the 1-Dimensional case to be given below

$$\operatorname{argmin}_{x \in \mathbb{R}} \left\{ \frac{\lambda}{2} ((Ay_k)^{(i)} - b^{(i)})^2 + \langle \nabla f(y_k)^{(i)}, x - y_k^{(i)} \rangle + |x| + \theta_k \frac{L}{2} (x - z_k^{(i)})^2 \right\} = \begin{cases} x^{(i)} = z^{(i)} - \frac{1 + \nabla f(y_k)^{(i)}}{\theta_k L} & \text{for } z \geq \frac{1 + \nabla f(y_k)^{(i)}}{\theta_k L} \\ x^{(i)} = z^{(i)} + \frac{1 - \nabla f(y_k)^{(i)}}{\theta_k L} & \text{for } z \leq \frac{\nabla f(y_k)^{(i)} - 1}{\theta_k L} \\ x^{(i)} = 0 & \text{for } \frac{1 + \nabla f(y_k)^{(i)}}{\theta_k L} < z < \frac{\nabla f(y_k)^{(i)} - 1}{\theta_k L} \end{cases}$$

Finally, for the case of $x \in \mathbb{R}^n$, we have

$$\operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \frac{\lambda}{2} \|Ay_k - b\|_2^2 + \langle \nabla f(y_k), x - y_k \rangle + \|x\|_1 + \theta_k \frac{L}{2} \|x - z_k\|_2^2 \right\} = \begin{cases} x = z - \frac{1 + \nabla f(y_k)}{\theta_k L} & \text{for } z \geq \frac{1 + \nabla f(y_k)}{\theta_k L} \\ x = z + \frac{1 - \nabla f(y_k)}{\theta_k L} & \text{for } z \leq \frac{\nabla f(y_k) - 1}{\theta_k L} \\ x = 0 & \text{for } \frac{1 + \nabla f(y_k)}{\theta_k L} < z < \frac{\nabla f(y_k) - 1}{\theta_k L} \end{cases}$$

4 Summary and Conclusions

Now let us look at the results of both of these algorithms! Both algorithms converge to the unique solution for **(1)**, \hat{x} . Note, the solution is unique, due to the convexity of **(1)**. This can be seen from graphing the 1-Dimensional case of **(1)**. Unsurprisingly, we see that the Accelerated Proximal Gradient Method converges faster than the Regular Proximal Gradient Method. For a randomly generated $x_0, z_0 \in \mathbb{R}^n$ with normally distributed values, we will see the following graphs,

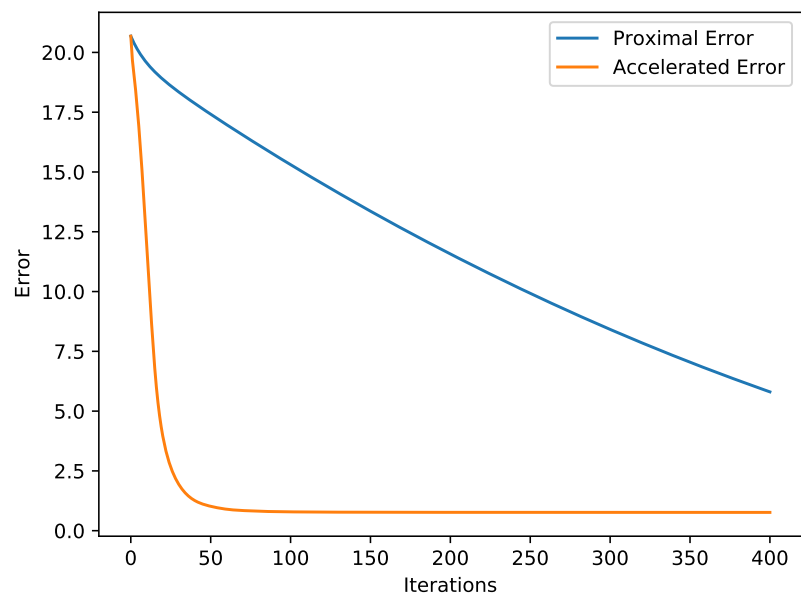


Figure 1: Error between Proximal Method and the Accelerated Proximal Method

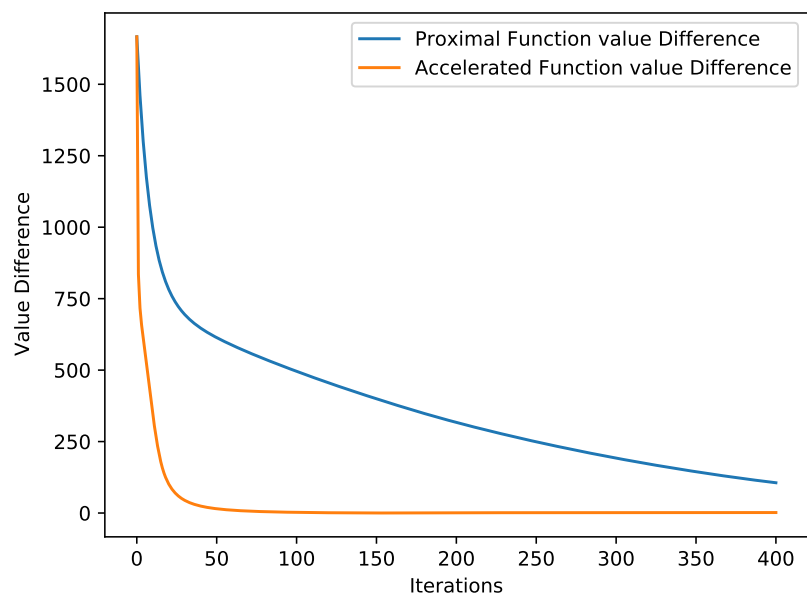


Figure 2: Difference in Function Values for the Proximal Method and the Accelerated Proximal Method

5 Final Remarks

1. I found that in the plot of $f(x_k) - f(\hat{x})$, for the Proximal Gradient Method, it did not have a convergence rate of $O(\frac{1}{k})$, but rather that it converged at a rate of $O(\frac{1}{\sqrt{k}})$.
2. I decided to withhold on coding Section 3 for now. Perhaps I will come back to it during Thanksgiving Break or after Finals. I left off with trying to figure out how to create either an Orthonormal Haar matrix or an Orthonormal Fourier matrix to transform a non-sparse vector to a sparse vector. I opted to construct these matrices instead of using Python's library, because the Pywt package and Fast Fourier Transform function returned a vector that had half the length I desired. I was not sure how to proceed with this, so I'll look into it at a later date.
3. Lastly, there are a couple formatting nuances that I wanted to rework with my Python code. The aesthetics of the graphs could be improved and the code itself is structured a bit odd. I see that Nathan had a main environment that he ran his python code and he kept his algorithms in separate python files. In the future I will follow this technique. That way is a bit neater and I think it's standard.

References

- [1] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.
- [2] Emmanuel J Candès, Justin Romberg, and Terence Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on information theory* 52.2 (2006), pp. 489–509.
- [3] Yurii Nesterov. "Dual extrapolation and its applications to solving variational inequalities and related problems". In: *Mathematical Programming* 109.2-3 (2007), pp. 319–344.
- [4] Paul Tseng. "On accelerated proximal gradient methods for convex-concave optimization". In: *submitted to SIAM Journal on Optimization* 2.3 (2008).