# Project Report

Fangyu Ding, 121033910177

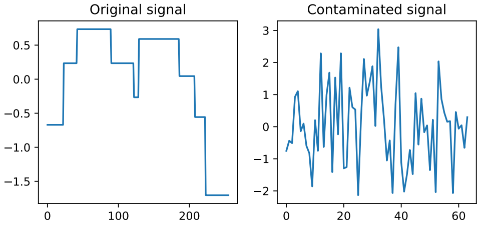
December 26, 2021

**Abstract**

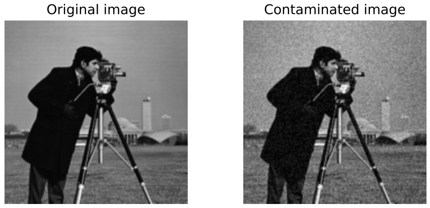
In this project, we discuss about the smoothing regularization item and its optimization for 1d signal reconstruction and 2d image denoising. We consider different forms of this smoothing regularization item and hyperparameters selection for the optimization problem. Meanwhile, a variety of optimization algorithms are introduced to solve this problem. A series of experiments on synthetic and real-world data demonstrate the effectiveness of our method and comparisons are made for various experimental settings.

# 1 Introduction and Related Work

Optimization methods are of wide usage in many scenarios, in this project we will discuss about the following two tasks: 1) 1d signal reconstruction, 2) 2d image denoising. In the first task, a square wave signal with $\kappa$ jumps is contaminated by first being conducted a dimensional reduction linear transformation, then being added with a Gaussian noise item. In the second task, a clean image is added with a Gaussian noise. We hereby solve these two problems in an optimization manner.



(a) 1d signal reconstruction problem



(b) 2d image denoising problem

The contaminating process can be described as follow:

$$b = Ax + \epsilon, \tag{1}$$

where $x$ is the ground truth signal/image, $A$ is the linear transformation matrix, $\epsilon$ is the additive white Gaussian noise, and $b$ is the contaminating result, which is our observation.

We represent the optimization object as a combination of two items: one is for data smoothing and the other is for content reconstruction, which will be denoted as $h(x)$ and $f(x)$ correspondingly in later chapters. We conduct different selection of $h(x)$'s form and hyperparameters used in the process of the optimization as well as different gradient-based optimization algorithms (Gradient Descent, Nesterov, Conjugate Gradient) to solve the problem iteratively.

# 2 Methods

## 2.1 Problem Definition

To reconstruct the contaminated 1d or 2d data, the corresponding optimization problem is defined as:

$$\min_x \Phi(x) = \mu h(x) + f(x), \tag{2}$$

where $\mu$ is the tradeoff parameter to balance the two items, $f(x)$ is the data fidelity term, typical choice of $f$ takes $f(x) = \frac{1}{2}\|Ax - b\|^2$, it is derived from the MLE estimation of Gaussian prior on noise distribution, if the noise distribution does not follow a Gaussian, this form of $f(x)$ will get poor optimization results.

And $h(x)$ is the smoothing regularization term whose form could be taken as the following two:

$$\begin{aligned} h_1(x) &= \sum_i \phi_\eta((\nabla x)_i), \\ \phi_\eta(x) &= \begin{cases} |x| - \frac{\eta}{2} & \text{if } |x| > \eta \\ \frac{x^2}{2\eta} & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

or

$$h_2(x) = \|\nabla x\|^2 \tag{4}$$

And from the derivation rule of matrix function, the gradients for items above are as follows:

$$\begin{aligned} \nabla h_1(x) &= \nabla^T \frac{\partial \phi(\nabla x)}{\partial(\nabla x)}, \\ \nabla h_2(x) &= 2\nabla^T \nabla x, \\ \nabla f(x) &= A^T(Ax - b), \end{aligned} \tag{5}$$

where

$$\frac{\partial \phi_\eta(x)}{\partial x} = \begin{cases} sign(x) & \text{if } |x| > \eta \\ \frac{x}{\eta} & \text{otherwise} \end{cases} \tag{6}$$

and $\nabla$ is the discrete gradient operator, whose matrix form is:

$$\nabla = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \\ & & & & & 0 \end{bmatrix} \tag{7}$$

which represents the fluctuation degree of the data. Tn 2d case, we consider two dimensional fluctuation degree, which can be obtained by $\nabla x$ and $\nabla x^T$, the discrete gradient operator can be easily implemented with Python Numpy ‘diff‘ function.

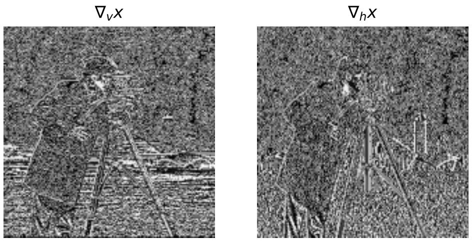


Figure 1: Fluctuation degree of image at horizontal direction and vertical direction calculated by $x\nabla^T$ and $\nabla x$

## 2.2 Lipschitz constant

We adopt three types of gradient-based optimization methods: const step-size Gradient Descent, Nesterov momentum Gradient Descent, and Conjugate Gradient Method.

In the const step-size Gradient Descent scheme, we should first determine the step-size, from the property of L-Lipschitz function, if $f \in C_L^1$:

$$f(x^*) \leq \min_y f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|y - x\|^2, \tag{8}$$

by solving the minimizer of $y$ on the right hand side of the inequation above, let

$$\frac{\partial}{\partial y}(f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|y - x\|^2) = 0, \tag{9}$$

we get

$$\begin{aligned} y - x &= -\frac{1}{L}\nabla f(x) \\ f(y) &\leq f(x) - \frac{1}{2L}\|\nabla f(x)\|^2 \end{aligned} \tag{10}$$

therefore we can choose $\frac{1}{L}$ as the step-size for the update of optimization object.

The Lipschitz constant $L$ for $h_1$, $h_2$, and $f$ can be derived with:

$$\begin{aligned} \|\nabla h_1(x) - \nabla h_1(y)\| &= \|\nabla^T \frac{\partial \phi(\nabla x)}{\partial(\nabla x)} - \nabla^T \frac{\partial \phi(\nabla y)}{\partial(\nabla y)}\| \\ &\leq \|\frac{\nabla^T \nabla(x - y)}{\eta}\| \\ &\leq \|\frac{\nabla^T \nabla}{\eta}\|\|x - y\| \\ &= \frac{4}{\eta}\|x - y\| \end{aligned} \tag{11}$$

the Lipschitz constant of $h_1(x)$ is $\frac{4}{\eta}$.

$$\begin{aligned} \|\nabla h_2(x) - \nabla h_2(y)\| &= \|2\nabla^T \nabla(x - y)\| \\ &\leq \|2\nabla^T \nabla\|\|x - y\| \\ &= 8\|x - y\| \end{aligned} \tag{12}$$

the Lipschitz constant of $h_2(x)$ is 8.

$$\begin{aligned} \|\nabla f(x) - \nabla f(y)\| &= \|A^T A(x - y)\| \\ &\leq \|A^T A\|\|x - y\| \end{aligned} \tag{13}$$

the Lipschitz constant of $f(x)$ is $\|A^T A\|$.

Therefore, if we choose $h_1(x)$ as the form of the smoothing regularization item, the Lipschitz constant of $\Phi(x)$ is $\frac{4\mu}{\eta} + \|A^T A\|$, if we choose $h_2(x)$, it is $8\mu + \|A^T A\|$.

### 2.3 Gradient Descent Algorithms

We can largely improve the const step-size Gradient Descent by introducing the idea of momentum (Nesterov) and the idea of Conjugate Gradient Method.

In the Nesterov's gradient descent scheme (Alg.1), momentum is involved at each iteration, the descent direction of previous iteration $d_{k-1}$ has effect at current iteration $k$ with an inertial parameter $a_k$. We select $a_k = \frac{k-1}{k+2}$ as described in work[SBC14], which has a good convergence performance.

```
Algorithm 1: Nesterov's Momentum Gradient Descent
Input: optimization objective $f(x)$
Initialize $x_0$, step size $\lambda = \frac{1}{L}$, $k = 0$.
while convergence criterion not meet do
    $k = k + 1$;
    $y_k = x_k + a_k \lambda d_{k-1}$;
    $x_{k+1} = y_k - \lambda \nabla f(y_k)$;
```

For the Conjugate Gradient Method (Alg.3), which overcomes the 'notched effect' and slow convergence in steepest descent and the complex computation of $\nabla^2 f$ in Newton's method, utilizing linear search to find the best step-size at each step (which could be implemented with backtracking(inexact) linear search: Alg.2), as the optimal point of objective function is not easy to achieve for functions except for quadratic ones, we adopt a restart strategy for each $n$ steps.

```
Algorithm 2: Backtracking linear search
Input: optimization objective $f(x)$, $\beta \in (0, 1)$, current state $x_k$ and current descent direction
       $d_k$.
Output: approximated optimal $\lambda$
Initialize $\lambda = 1$, $\delta \in (0, \frac{1}{2})$;
while $f(x_k + \lambda d_k) > f(x_k) + \delta \lambda \nabla f(x_k)^T d_k$ do
    $\lambda = \lambda \beta$;
Return $\lambda$;
```

```
Algorithm 3: Conjugate Gradient Method
Input: optimization objective $f(x)$, restart steps $n$.
Initialize $x_0$, step size $\lambda = \frac{1}{L}$, $k = 0$.
while convergence criterion not meet do
    $k = k + 1$;
    Find step-size $\lambda$ with Alg.2;
    $x_{k+1} = x_k - \lambda d_k$;
    $d_{k+1} = -\nabla f(x_k) + \beta_k d_k$;
    where $\beta_k$ can be defined as several forms:
    $\beta_k = \begin{cases} \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} \\ \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{d_k^T (\nabla f(x_{k+1}) - \nabla f(x_k))} \end{cases}$
    if $k \geq n$ then
        $k = 0$;
        $x_0 = x_k$;
        $d_0 = -\nabla f(x_0)$;
```

# 3 Experiments

## 3.1 Comparision on different experimental settings

For the balance parameter $\mu$ described in objective function Eq.2. The larger $\mu$ corresponds to the smoother optimization result: reconstruction signal of denoised image. With $\mu = 0.01$, a rough reconstructed signal is obtained, with $\mu = 0.1$, a smoother one is obtained, with $\mu = 0.1$, a smoother but biased signal is obtained, as a result of that, we choose $\mu = 0.1$ as a reasonable setting in following experiments of 1d signal reconstruction.

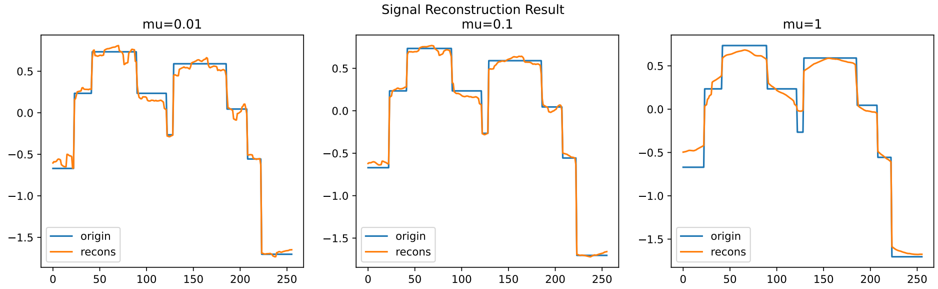


Figure 2: The reconstruction 1d signal with different setting of $\mu$.

For the selection of $h(x)$'s form and corresponding $\eta$: From the results above, we find that a smoother and more accurate reconstructed signal is obtained with $h_1(x)$ and smaller $\eta$ (i.e. when $\eta = 0.01$), the reason for which is the gradient of $h_1(x)$: $\nabla h_1(x)$ is numerically larger with smaller than bigger and $h_2$, as the gradient of $|x|$ is 1 while the gradient of $x^2$ is close to 0 when $x$ is small, according to the scenario of 1d signal reconstruction task: the signal's value is small in order of magnitude.

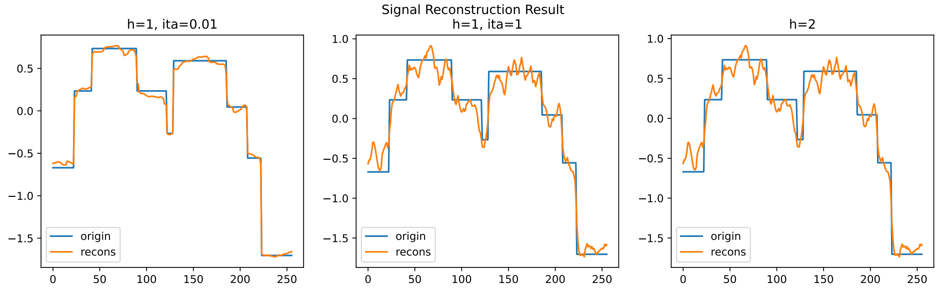


Figure 3: The reconstruction 1d signal with different setting of $h(x)$ and $\eta$.

For the image denoising task, a smoothing regularization term brings the effect of denoising and the side-effect is to blur the image. We choose $h_2(x)$ as the smoothing regulaization item, here the form of it would be:

$$h(x) = \|\nabla x\|_F^2 + \|\nabla x^T\|_F^2 \tag{14}$$

and its gradient can be derived as:

$$\begin{aligned}\nabla h(x) &= \frac{\partial}{\partial x}(\|\nabla x\|_F^2 + \|\nabla x^T\|_F^2) \\ &= \frac{\partial}{\partial x}(tr(x^T\nabla^T\nabla x) + tr(x\nabla^T\nabla x^T)) \\ &= 2\nabla^T\nabla x + 2x\nabla^T\nabla\end{aligned} \tag{15}$$

which can be effectively implemented with Python Numpy ‘diff‘.

With different settings of $\mu$, clear difference in denoising result is revealed.

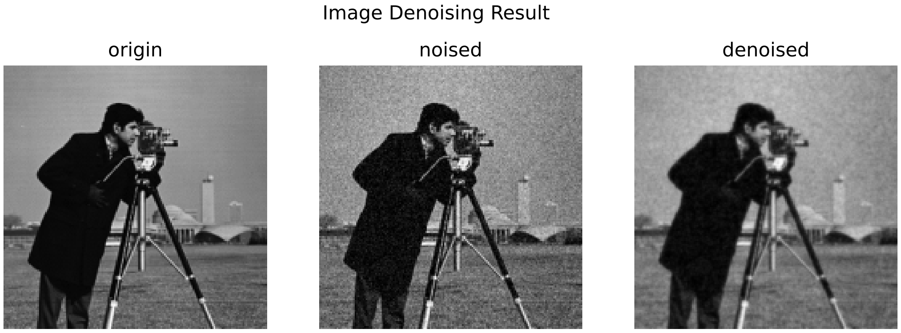


Figure 4: The denoised 2d image with smoothing regularization item as $h_2(x)$.

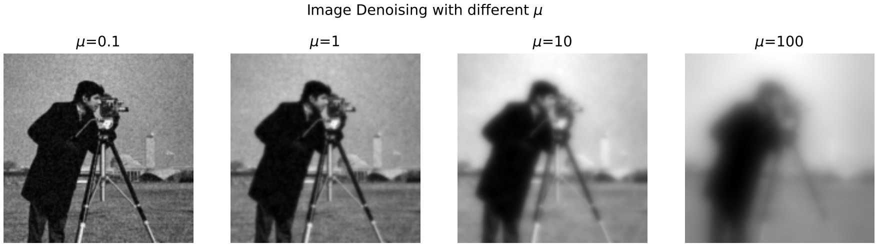


Figure 5: The denoised result with different settings of $\mu$.

## 3.2 Comparision on different optimization algorithms

We make a comparison over three different gradiend-based optimization methods: const step-size Gradient Descent, Nesterov's Momentum Gradient Descent, and Conjugate Gradient Method with different definition of $\beta$ descirbe in Alg.3.

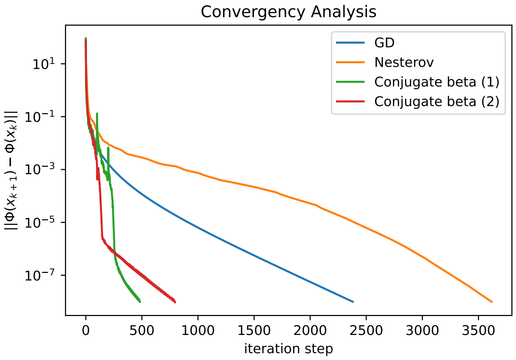


Figure 6: The convergence performance of different optimization methods on 1d signal reconstruction task.

The convergence criterion is set as $\|\Phi(x_{k+1}) - \Phi(x_k)\| < \epsilon$, where $\epsilon$=1e-8, the const step-size Gradient Descent is the slowest, with the help of the Nesterov's momentum, the convengence is accelerated, and the Conjugate Gradient Method of searching not completely according to gradient direction shows the best convergence performance on the optimization of the 1d signal reconstruction problem.

# 4 Conclusion

This project discuss about the smoothing regularization item and its optimization for 1d signal reconstruction and 2d image denoising. Different forms of this smoothing regularization item and hyperparameters setting are applied for the optimization problem. We implemened const step-size Gradient Descent, Nesterov's Momentum, and Conjugate Gradient Method to iteratively optimize this objective function. Comprehensive experiments demonstrates the effectiveness of this smoothing regularization item and the optimization process can be accelerated with the help of Nesterov's momentum and Conjugate Gradient Method.

# References

[SBC14] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Advances in neural information processing systems*, 27:2510–2518, 2014.