LASSO Problem in Pipe Burst Detection

Shenbiao Dai

May 17, 2019

Abstract

The LASSO penalizes a least squares regression by the sum of the absolute values of the coefficient. The form of this penalty encourages sparse solution. We propose weighted LASSO problem, which is adopted to reform a pipe burst model and devise efficient algorithm to figure out optimal solutions. The sparse solutions delineate the flow trend of burst pipe. After some computation and deduction, we compute closed-form solutions in two iteration methods — ADMM method and proximal gradient method, which provide theoretical basis of devising and implementing algorithms to solve the problem. Furthermore, we prove the zero duality gap between weighted LASSO problem and its dual problem and utilize the dual problem to compute similar optimal solution. Ultimately, we illustrate the burst flow represented by the optimal coefficients and the test the reliablity of the raised model.

1 Problem Statement

1.1 Background

Pipe burst problem is one class of anomaly detection problems. Smooth-sparse decomposition(SSD) is a ramification of anomaly detection. Anomaly detection is an important concept in machine learning. Usually, it is related to examination in a stable system or data mining, such as fitting model for anomal prediction, supervising stability of mechanic system, image segmentation, event detection in sensor networks, and detecting ecosystem disturbances, etc.

When SSD is utilized in image processing, an image can be regarded as a combination of a smooth part and a sparse part. Continuous trend or property in an image such as background and environment is represented by the smooth part, and discrete abnormal such as random noice or pistol destruction is represented by the sparse part. For one thing, the encouragement of sparse solutions is practical to decompose the two various trends for further research; for another, The abstruction of the sparse part is inspiring for abnormal detection. In this report, the general idea of SSD in image processing is stimulated and recapitulated to solve a specific pipe burst problem.

In some cases, SSD problems can be translated into tremendous kinds of optimization problems. For instance, the problem that we will discuss is weighted LASSO problem, which is reformed from a burst detection model raised by Ph.D student Yiwei Zhang in The University of Arizona.

1.2 Model Introduction

1.2.1 Departure Point

As a simulation of the idea in image segment, a pipe flow can also be seperated into two parts, or three part where random noise is introduced. Based on historical data, a control group of coefficients can be established to provide comparision between current flow data and historical flow data. If we separate the pipe flow data into three parts: smooth, random and sparse, the control related to sparse flow deserves attention. If the absolute value of burst flow, which is fitted by spline functions and their coefficients, exceed the threshold of a specific value, we claim that pipe burst has occured and maintenance is necessary.

1.2.2 Model Explanation

The research object is the model which is explained in detail below.

$$\min_{\beta,\beta_a} \|y - \Phi\beta - \Phi_a\beta_a\|_2^2 + \lambda_2\|\beta - \bar{\beta}\|_2^2 + \lambda_1\|\beta_a\|_1 \qquad (1)$$

- y: One-day water flow record.
- Φ, Φ_a: Basis function for flow record and additional flow record due to the burst.
- β, β_a: Corresponding coefficients of each basis function.
- $\bar{\beta}$: The mean coefficient of historical non-burst data, which derives from the solution of least square problem
$$\min_{\beta} \|\bar{y} - \Phi\beta\|_2^2 \qquad (2)$$
where $\bar{y}$ is the average of historical flow records.
- λ_1: Penalty coefficent to constrain β from too much deviation from $\bar{\beta}$.
- λ_2: Penalty coefficent to induce sparse solution.

In our research, the one-day water flow record is represented by 576 data points, which is uniformly segmented. The degree of B-spline functions is 4, which means that the basis functions are cubic. The number of interior points is 20. The matrices Φ, Φ_a are generated by the function `spcol` in MATLAB, which memories the values of different basis functions at different time points. λ_1, λ_2 are chosen to cater for demand. To satisfy actual conditions, λ_1 is significantly larger than λ_2.

2 Theoretical Analysis

In this section, we present several propositions to prepare for devising algorithms to solve the problem in Eq.1.

At first, we need to reform the problem in Eq.1 into a weighted LASSO problem.

Proposition 2.1. *The SSD problem is equivalent to a weighted LASSO problem in the form of*

$$\arg\min_{\beta_a} \mathcal{H}(\beta_a) = (z - \Phi_a\beta_a)^T H(z - \Phi_a\beta_a) + \lambda_1\|\beta_a\|_1$$

where

$$z = y - \Phi\bar{\beta}$$
$$K = \Phi^T\Phi + \lambda_2 I$$
$$H = (I - \Phi K^{-1}\Phi^T)^2 + \lambda_2\Phi(K^{-1})^2\Phi^T$$

Proof. Let $\tilde{\beta} = \beta - \bar{\beta}$. Denote the body of Eq.1 as $F(\tilde{\beta}, \beta_a)$. From first-order conditions,

$$\frac{\partial F}{\partial \tilde{\beta}} = -2\Phi^T(y - \Phi\bar{\beta}) + 2\Phi^T\Phi\tilde{\beta} + 2\Phi_a\beta_a + 2\lambda_2\tilde{\beta} = 0 \quad (3)$$

which means that

$$\arg\min_{\tilde{\beta}} F(\tilde{\beta}, \beta_a) = (\Phi^T\Phi + \lambda_2 I)^{-1}\Phi^T(y - \Phi\bar{\beta} - \Phi_a\beta_a) \quad (4)$$

The original problem was reformed into

$$\begin{aligned}
&\min_{\beta_a} \|(I - \Phi(\Phi^T\Phi + \lambda_2 I)^{-1}\Phi^T)(z - \Phi_a\beta_a)\|_2^2 \\
&+ \lambda_2\|(\Phi^T\Phi + \lambda_2 I)^{-1}\Phi^T(z - \Phi_a\beta_a)\|_2^2 + \lambda_1\|\beta_a\|_1 \\
=&\min_{\beta_a}(z - \Phi_a\beta_a)^T[(I - \Phi(\Phi^T\Phi + \lambda_2 I)^{-1}\Phi^T)^2 \\
&+ \lambda_2\Phi((\Phi^T\Phi + \lambda_2 I)^{-1})^2\Phi^T](z - \Phi_a\beta_a) + \lambda_1\|\beta_a\|_1
\end{aligned}$$

Let

$$K = \Phi^T\Phi + \lambda_2 I$$
$$H = (I - \Phi K^{-1}\Phi^T)^2 + \lambda_2\Phi(K^{-1})^2\Phi^T$$

we have proved the conclusion. □

Convexity guarantees the simplicity for solving optimization problems. We can prove that the weighted LASSO problem in Eq.2.1 is convex. Denote

$$G(\beta_a) := (z - \Phi_a\beta_a)^T H(z - \Phi_a\beta_a) \quad (5)$$

we have

Proposition 2.2. *The problem*

$$\arg\min_{\beta_a} G(\beta_a) \quad (6)$$

is convex for β_a.

Proof. Notice that in $H = (I - \Phi K^{-1}\Phi^T)^2 + \lambda_2\Phi(K^{-1})^2\Phi^T$, $I - \Phi K^{-1}\Phi^T$ and K^{-1} are symmetric matrix. Therefore, $\forall\ x \in \mathbb{R}^n$,

$$x^T H x = \|(I - \Phi K^{-1}\Phi^T)x\|_2^2 + \lambda_2\|K^{-1}\Phi^T x\|_2^2 \geq 0 \quad (7)$$

□

Generally, as a quadratic form in Eq.6, we can simply prove the Lipschitz property of the function and figure out Lipschitz constant via matrix differentiation. The Lipschitz constant is of great significance in proximal gradient algorithm.

Proposition 2.3. *$G(\cdot)$ is Lipschitz continuous, in which satisfies*

$$\|\nabla G(a) - \nabla G(b)\|_2 \leq L\|a - b\|_2 \quad (8)$$

$\forall a, b \in \mathbb{R}$ with $L = 2\|\Phi_a^T H\Phi_a\|_2$.

Proof.

$$\begin{aligned}
\|\nabla G(a) - \nabla G(b)\|_2 &= \|2\Phi_a^T H(\Phi_a a - z) - 2\Phi_a^T H(\Phi_a b - z)\|_2 \\
&= 2\|\Phi_a^T H\Phi_a(a - b)\| \\
&\leq 2\|\Phi_a^T H\Phi_a\|_2\|a - b\|_2
\end{aligned}$$

□

Lemma 2.4. *Quadratic upper bound from Lipschitz property*

$$G(y) \leq G(x) + \nabla G(x)^T(y - x) + \frac{L}{2}\|y - x\|_2^2 \quad \forall x, y \in \mathbb{R}^n \quad (9)$$

For explicity in algorithm explanation, we introduce the concept of soft-threshold operator. It is closely related to 1-norm in Eq.1. We will analyse the correlationship between them in the following proposition.

Definition 2.1. Let $S_\lambda(x)$ denote the soft-threshold operation where

$$S_\lambda(x_i) = \begin{cases} x_i - \lambda & x_i \geq \lambda \\ 0 & |x_i| \leq \lambda \\ x_i + \lambda & x_i \leq -\lambda \end{cases} \qquad i = 1, \ldots, n \quad (10)$$

In the works by Ito, Takeda and Toh[1], the order of convergence and convergence speed can be proved. Moreover, if we adopt accelerating method, the convergence speed can be promoted further.

Lemma 2.5. *If*

$$\beta_a^{k+1} = \arg\min_{\beta_a}\left\{\nabla G(\beta_a^k)^T(\beta_a - \beta_a^k) + \frac{1}{2s}\|\beta_a - \beta_a^k\|_2^2\right\} \quad (11)$$

let $s = \frac{1}{L}$ and from **Lemma 2.4**, *we guarantee the decrease of $G(\beta_a^k)$. Denote $G(\beta_a^k) \to G^*$ as $k \to \infty$, we have $G(\beta_a^{(k)}) - G^* \leq O(1/k)$. If adopting acceleration, the convergence speed can reach $O(1/k^2)$.*

Algorithm 1 Preparation of Fundamental Variables

```
Initialize β̄
m ← 22    ▷ Depend on number of interior points and spline degree
n ← 576                                   ▷ Number of time points
I_1 = eye(m)
I_2 = eye(n)
Predetermine λ_1, λ_2
K ← Φ′ * Φ + λ_2 * I_1
H = (I_2 − Φ * K^{-1} * Φ′)^2 + λ_2 * Φ * K^{-1} * Φ′
Choose y_1                    ▷ Choose one of the flow records
z ← y_1 − Φ * β̄
A_1 ← Φ′_a * H
A_2 ← A_1 * Φ_a
A_{1x} ← (A_2 − I_1/τ)^{-1}
A_{2x} = A_1 * z
```

3 Algorithm

3.1 ADMM Method

The problem in **Proposition 2.1** is equivalent to

$$\begin{aligned} &\text{minimize} && \frac{1}{2}(z-\Phi_a x)^T H(z-\Phi_a x)+\frac{1}{2}\lambda_1\|y\|_1 \\ &\text{subject to} && x=y \end{aligned} \tag{12}$$

We can write down the augmented Lagrangian function of the problem in Eq.12 using penalty $\frac{1}{\tau}$.

$$\begin{aligned} L_{\frac{1}{\tau}}(x,y,\mu) = \frac{1}{2}(z-\Phi_a x)^T H(z-\Phi_a x)+ \\ \frac{1}{2}\lambda_1\|y\|_1+\frac{1}{\tau}\mu^T(x-y)+\frac{1}{2\tau}\|x-y\|_2^2 \end{aligned} \tag{13}$$

From first-order optimality condition, ADMM method is separated into three parts in each step.[2]

(1) $x^{(k+1)} = \arg\min_x L_{\frac{1}{\tau}}(x, y^{(k)}, \mu^{(k)})$

(2) $y^{(k+1)} = \arg\min_y L_{\frac{1}{\tau}}(x^{(k+1)}, y, \mu^{(k)})$

(3) Update $\mu^{(k+1)}$ using $x^{(k+1)}$, $y^{(k+1)}$ by dual problem.

To put it into the circumstance in the original weighted LASSO problem, we have the next proposition.

Proposition 3.1. *The iterations of ADMM method to solve the weight LASSO problem are*

$$x^{(k+1)} = (\Phi_a^T H\Phi_a+\frac{1}{\tau}I)^{-1}(\Phi_a^T Hz+\frac{1}{\tau}(y^{(k)}-\mu^{(k)})) \tag{14}$$

$$y^{(k+1)} = S_{\frac{\lambda_1\tau}{2}}(x^{(k+1)}+\mu^{(k)}) \tag{15}$$

$$\mu^{(k+1)} = \mu^{(k)}+(x^{(k+1)}-y^{(k+1)}) \tag{16}$$

Proof.

(1) From first-order optimality conditions to optimize $L_{\frac{1}{\tau}}$ by x,

$$\begin{aligned} \frac{\partial L_{\frac{1}{\tau}}}{\partial x}(x^*, y^{(k)}, \mu^{(k)}) &= \Phi_a^T H(\Phi_a x^* - z)+\frac{1}{\tau}\mu^{(k)} \\ &\quad +\frac{1}{\tau}(x^*-y^{(k)}) = 0 \\ \Rightarrow (\Phi_a^T H\Phi_a+\frac{1}{\tau}I)x^* &= \Phi_a^T Hz-\frac{1}{\tau}(\mu^{(k)}-y^{(k)}) \\ \Rightarrow x^{(k+1)} &= x^* \\ &= (\Phi_a^T H\Phi_a+\frac{1}{\tau}I)^{-1}(\Phi_a^T Hz+\frac{1}{\tau}(y^{(k)}-\mu^{(k)})) \end{aligned}$$

(2) From first-order optimality conditions to optimize $L_{\frac{1}{\tau}}$ by y,

$$\begin{aligned} &\frac{\partial L_{\frac{1}{\tau}}}{\partial y}(x^{(k+1)}, y^*, \mu^{(k)}) \\ &= \frac{1}{2}\lambda_1\text{sgn}(y^*)-\frac{1}{\tau}\mu^{(k)}-\frac{1}{\tau}(x^{(k+1)}-y^*) = 0 \\ &\Rightarrow y^* = \frac{1}{\tau}(\mu^{(k)}+x^{(k+1)})-\frac{1}{2}\lambda_1\text{sgn}(y^*) \end{aligned}$$

$$y^* = \begin{cases} \frac{1}{\tau}\mu^{(k)}+x^{(k+1)}-\frac{1}{2}\lambda_1 & \mu^{(k)}+x^{(k+1)} > \frac{1}{2}\lambda_1\tau \\ \frac{1}{\tau}\mu^{(k)}+x^{(k+1)}+\frac{1}{2}\lambda_1 & \mu^{(k)}+x^{(k+1)} < -\frac{1}{2}\lambda_1\tau \\ 0, & \text{else} \end{cases} \tag{17}$$

If we rewrite the form into soft-threshold operator, we obtain that

$$y^{(k+1)} = y^* = S_{\frac{\lambda_1\tau}{2}}(x^{(k+1)}+\mu^{(k)})$$

(3) The original iteration is in the form that

$$\mu^{(k+1)} = \mu^{(k)}+\frac{1}{\tau}(x^{(k+1)}-y^{(k+1)})$$

However, when we implement the iterations in the program, we notice that if the coefficient $\frac{1}{\tau}$ is reserved by $0<\tau<1$, the solution will oscilate stably near the optimal solution, where it is hard to exit the loop. To simplify the model and obtain a more reliable convergence, we have tried to remove the coefficient $\frac{1}{\tau}$ and figure out a better convergence property. Therefore, we revise the iteration into

$$\mu^{(k+1)} = \mu^{(k)}+(x^{(k+1)}-y^{(k+1)}) \tag{18}$$

□

Algorithm 2 ADMM Algorithm

```
Predetermine τ
Predetermine N as threshold for maximum iterations
k ← 0
TOL ← 10^-6
Initialize β_a0, y_0, μ_0
while k < N do
    β_a ← A_1x * (A_2x + (y_0 − μ_0)/τ)
    y ← soft(β_a + μ_0, λ_1 * τ/2)
    if ‖β_a − β_a0‖_2 < TOL then
        break
    end if
    μ ← μ_0 = (β_a − y)
    β_a0 ← β_a
    y_0 ← y
    μ_0 ← μ
    k ← k + 1
end while
return β_a                ▷ The coefficient of burst flow
```

3.2 Proximal Gradient Method

In order to use soft-threshold operator to handle the optimal solution to the existance of 1-norm and 2-norm, we need the lemma for the next proposition. The idea of the lemma below refer to the paper by Yan, Paynabar and Shi[3].

Lemma 3.2. *The problem* $\beta_a^{(k)} = \arg\min_{\beta_a}\|z-\beta_a\|_2^2+\lambda_1\|\beta_a\|_1$ *has a closed-form solution in the form of* $\beta_a^{(k)} = S_{\frac{\lambda_1}{2}}(z)$ *where S is the soft-threshold operator.*

Proof. The first-order condition of the optimization problem can be expressed as $\nabla\|z-\beta_a\|_2^2+\lambda_1 g$ where

$$g=[g_i]=\begin{cases}\text{sgn}(\beta_{ai}) & \beta_{ai}\neq 0\\ [-1,1] & \beta_{ai}=0\end{cases} \tag{19}$$

The closed interval means that the operator can be selected as an arbitrary value in the inteval. The loss function can be simplified to $\|z\|_2^2-2z^T\beta_a+\|\beta_a\|_2^2$. Then, the first-order condition gives $\beta_a=z-\frac{1}{2}g$. Similar with the discussion in **Proposition 3.1**, the solution is equivalent to the form in soft-threshold operator$\beta_a^{(k)}=S_{\frac{\lambda_1}{2}}(z)$. □

It is practical to utilize linear approximation to represent the smooth part of the original function. Moreover, it provides higher speed for proximal gradient method.[4]

Proposition 3.3. *Notice that*

$$G(\beta_a)\approx G(\beta_a^{(k)})+\nabla G(\beta_a^{(k)})^T(\beta_a-\beta_a^{(k)}) \tag{20}$$

The proximal gradient method for the problem in **Proposition 2.1** *is equivalent to*

$$\begin{aligned}\beta_a^{(k+1)}=\arg\min_{\beta_a}\{&G(\beta_a^{(k)})+\nabla G(\beta_a^{(k)})^T(\beta_a-\beta_a^{(k)})\\&+\frac{L}{2}\|\beta_a-\beta_a^{(k)}\|_2^2+\lambda_1\|\beta_a\|_1\}\end{aligned} \tag{21}$$

where $\beta_a^{(k+1)}$ *has a closed form*

$$\beta_a^{(k+1)}=S_{\frac{\lambda}{L}}(\beta_a^{(k)}+\frac{2}{L}\Phi_a^T H(z-\Phi_a\beta_a^{(k)})) \tag{22}$$

Proof. After omitting constant items, the problem in Eq.21 can be translated into

$$\begin{aligned}&\arg\min_{\beta_a}\{\nabla G(\beta_a^{(k)})^T(\beta_a-\beta_a^{(k)})+\frac{L}{2}\|\beta_a-\beta_a^{(k)}\|_2^2\\&\quad+\lambda_1\|\beta_a\|_1\}\\&=\arg\min_{\beta_a}\{\frac{2}{L}\nabla G(\beta_a^k)^T(\beta_a-\beta_a^k)+\|\beta_a-\beta_a^{(k)}\|_2^2\\&\quad+\frac{1}{L^2}\|\nabla G(\beta_a^{(k)})\|_2^2+\frac{2\lambda_1}{L}\|\beta_a\|_1\}\\&=\arg\min_{\beta_a}\{\|\beta_a-\beta_a^{(k)}-\frac{1}{L}\nabla G(\beta_a^{(k)})\|_2^2+\frac{2\lambda_1}{L}\|\beta_a\|_1\}\end{aligned}$$

Notice that the function is similar with the form in **Lemma 3.2** where $z=\beta_a^{(k)}+\frac{1}{L}\nabla G(\beta_a)$, then we have

$$\beta_a^{(k+1)}=S_{\frac{\lambda_1}{L}}(\beta_a^{(k)}+\frac{1}{L}\nabla G(\beta_a^{(k)})) \tag{23}$$

If we compute the gradient of G, we obtain that

$$\nabla G(\beta_a^{(k)})=2\Phi_a^{-1}H(\Phi_a\beta_a^{(k)}-z) \tag{24}$$

Now we have the ultimate form of the solution. □

Algorithm 3 Proximal Gradient Algorithm

```
Predetermine τ
Predetermine N as threshold for maximum iterations
k ← 0
TOL ← 10^{-6}
L = 2 * ||A_2||_2
Initialize β_{a0}
while k < N do
    r ← Φ'_a * H * (z − Φ_a * β_{a0})
    β_a ← soft(β_a + 2/L * r, λ_1/L)
    if ||β_a − β_{a0}||_2 < TOL then
        break
    end if
    β_{a0} ← β_a
    k ← k + 1
end while
return β_a                ▷ The coefficient of burst flow
```

3.3 Accelerated Proximal Gradient Method

From **Lemma 2.5**, the convergence speed of proximal gradient method can reach an order of $O(1/k^2)$ if accelerated methods are adopted in each iteration. We introduced one of the accelerated methods, which is also known as FISTA, referring to the works by Beck and Teboulle[5].

Algorithm 4 Accelerated Proximal Gradient Algorithm

```
Predetermine τ
Predetermine N as threshold for maximum iterations
k ← 0
TOL ← 10^{-6}
L = 2 * ||A_2||_2
Initialize β_{a0}
γ ← β_{a0}
t_0 ← 0
t_1 ← 1
while k < N do
    r ← Φ'_a * H * (z − Φ_a * β_{a0})
    β_a ← soft(β_a + 2/L * r, λ_1/L)
    t_2 = 1 + sqrt(1 + 4 * t_1^2)/2
    if ||β_a − β_{a0}||_2 < TOL then
        break
    end if
    γ ← β_a + t_0/t_2 * (β_a − β_{a0})
    β_{a0} ← β_a
    t_0 ← t_1
    t_1 ← t_2
    k ← k + 1
end while
return β_a                ▷ The coefficient of burst flow
```

4 Re-analysis—Dual Problem

Proposition 4.1. *The optimal solution of the primal weighted LASSO problem in* **Proposition 2.1** *is equal to the optimal solution of its dual problem*

$$\begin{aligned}&\textit{minimize}\quad &&\frac{1}{4}\mu^T H^{-1}\mu-\mu^T z\\&\textit{subject to}\quad &&\|\Phi_a^T\mu\|_\infty\leq\lambda_1\end{aligned} \tag{25}$$

Proof. Reform the problem

$$\begin{aligned}&\text{minimize}\quad &&(z-y)^T H(z-y)+\lambda_1\|x\|_1\\&\text{subject to}\quad &&y=\Phi_a x\end{aligned} \tag{26}$$

We write down the Lagrangian Function

$$L(x,y,\mu) = (z-y)^T H(z-y) + \lambda_1 \|x\|_1 + \mu^T(y - \Phi_a x) \quad (27)$$

To compute the partial derivative,

$$\frac{\partial L}{\partial y} = -2Hz + \mu + 2Hy = 0 \quad \Rightarrow y = z - \frac{1}{2}H^{-1}\mu \quad (28)$$

due to the fact that H is full rank.

If $\inf_{x,y} L(x,y,\mu)$ is not $-\infty$, we have

$$\|\Phi_a^T \mu\|_\infty \leq \lambda_1 \quad (29)$$

This is the constraint for the dual problem. After simplification, we have the dual weighted LASSO problem in Eq.25.

Now we prove that there is no duality gap between Primal weighted LASSO and its dual problem. First, we analyze KKT conditions for the dual problem. To reduce inequality constraints into equality constraints, we introduce slack variables to reform the problem in Eq.25 and Lagrangian function.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{4}\mu^T H^{-1}\mu - \mu^T z \\ \text{subject to} \quad & \begin{pmatrix} \Phi_a^T \mu \\ -\Phi_a^T \mu \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 \mathbf{e} \\ \lambda_1 \mathbf{e} \end{pmatrix} \quad (30) \\ & \delta_1 \succeq 0, \quad \delta_2 \succeq 0 \end{aligned}$$

where $\mathbf{e}$ is vector of all 1's.

Now the Lagrangian function is

$$\begin{aligned} L(\mu, \delta_1, \delta_2, y_1, y_2, \eta_1, \eta_2) = \\ \frac{1}{4}\mu^T H^{-1}\mu - \mu^T z - \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}^T \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} \\ + \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}^T \left[\begin{pmatrix} \Phi_a^T \mu \\ -\Phi_a^T \mu \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} - \begin{pmatrix} \lambda_1 \mathbf{e} \\ \lambda_1 \mathbf{e} \end{pmatrix} \right] \end{aligned} \quad (31)$$

Then from KKT conditions,

$$\begin{aligned} & y_1 = \eta_1 \quad y_2 = \eta_2 \\ & \tfrac{1}{2}H^{-1}\mu + \Phi_a(y_1 - y_2) = z \\ & \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \odot \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \mathbf{0} \quad (32) \\ & \begin{pmatrix} \Phi_a^T \mu \\ -\Phi_a^T \mu \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 \mathbf{e} \\ \lambda_1 \mathbf{e} \end{pmatrix} \end{aligned}$$

where $\odot$ represents dot multiple and $\mathbf{0}$ means zero vector.

Notice that H is positive semidefinite and symmetric, we have eigenvalues decomposition. There exists orthogonal matrix P and nonnegative diagonal matrix D such that

$$H = P^T D P \quad (33)$$

Define that

$$H^{\frac{1}{2}} = P^T D^{\frac{1}{2}} P^T \quad (34)$$

If we denote that $u_1 = \max(0, x)$, $u_2 = \max(0, -x)$, then $x = u_1 - u_2$. By optimality conditions on the Lagrangian of the primal problem in Eq.27, condition $\lambda_1 \|x\|_1 - (\Phi_a^T \mu)^T x$ is equivalent to

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \odot \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \mathbf{0} \quad (35)$$

Therefore, u_1, u_2 are equal to y_1, y_2 and η_1, η_2. Now we write down the duality gap

$$G = (z - \Phi_a x)^T H (z - \Phi_a x) + \frac{1}{4}\mu^T H^{-1}\mu - \mu^T z \quad (36)$$

After simplification, Eq.36 is equal to

$$G = \|H^{\frac{1}{2}}(z - \Phi_a x - \frac{1}{2}H^{-1}\mu)\|_2^2 + \delta_1^T u_1 + \delta_2^T u_2 \quad (37)$$

From the second and third equations in Eq.32, if p^* denotes the optimal solution of the primal problem, q^* denotes the optimal solution of the dual problem,

$$G^* = p^* - d^* = 0 \quad (38)$$

□

From the conclusion of **Proposition 4.1**, it is equivalent to figure out the optimal solution of the problem in Eq.30 instead of handling the original non-smooth problem:

5 Experiment

In this section, the performance of ADMM algorithm, PG method and APG method are evaluated through simulations under actual water flow records. The dataset consists of four water flow records from one pipe, which are one-dimension. The simulated bursted flow records are available to test the reliability of optimal solutions, which derive from implements of the three algorithms. Ultimately, we will compare the outcomes from dual problem and make comparison.

5.1 Convergence Speed

If we memory the value of the cost function for each iteration, we can simply plot a figure to compare the convergence speed between the three algorithms. Due to the fact that the solution variate little after 30 iterations, we set the maximum of iterations as 30. To be more specific, we set $\lambda_1 = 4000$, $\lambda_2 = 20$ and $\tau = \frac{1}{2}$.

In Figure 1, PG algorithm and APG algorithm converge significantly faster than ADMM algorithm, but the difference between non-accelerated and accelerated PG algorithm is not explicitly illustrated. We delete the plot for ADMM algorithm to display the actual difference of the other two algorithms.

Now we can find out that APG Algorithm converges to the actual optimal solution faster than PG algorithm in the first 10 iterations. If the dimension of the original problem expands(**e.g.** change the 576 time points into 10000 time points), the acceleration will be more explicit.

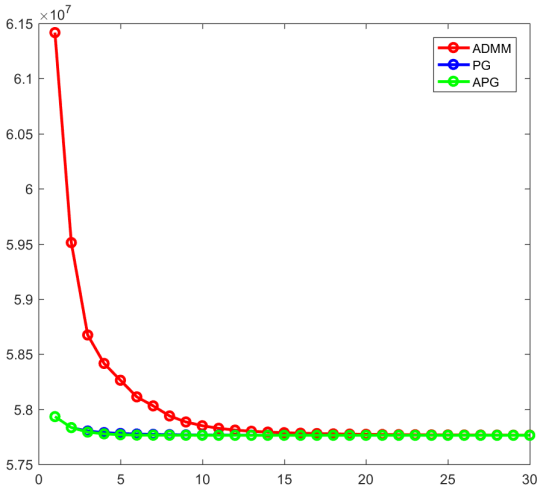

Figure 1: Comparison of Convergence between Three Algorithms

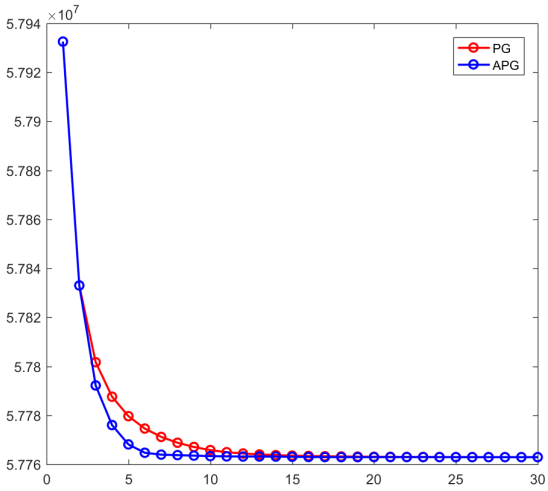

Figure 2: Comparison of Convergence between PG and APG Algorithms

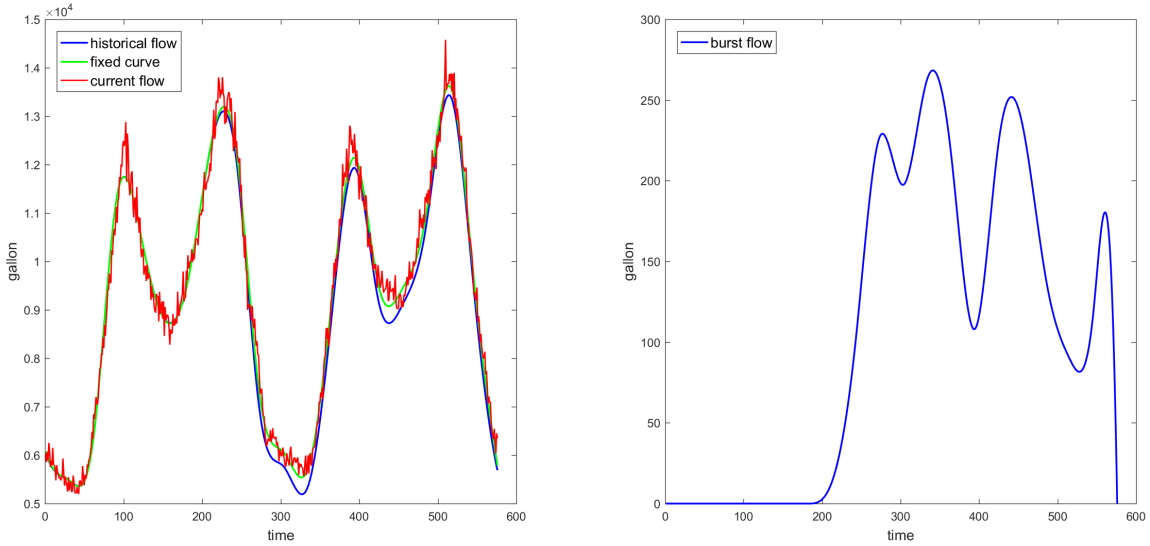

Figure 3: Comparison of Actual Flow and Burst Flow

5.2 Fitting Agreement

Sparse solution means that the solution is composed of many zeros, in which case the fitted burst flow agrees with the actual burst flow to the greatest extent. The most efficient way to test the reliablity of our algorithms is to compare the obtained burst flow values and actual burst flow records.

On the left hand side of Figure 3, the fitted curve generally agrees with the actual water flow. However, after time 300, additional flow is added on the general flow and shows deviation. The arise of deviation means that there exists a burst flow with significant water quantity records.

5.3 Anormal Detection Effect

Now we have four water flow records and four simulated burst flow records. The comparison is illustrated in Figure 4. The general trend of theoretical burst flow is consistent with the actual burst flow, but the range does not fit so well. We notice that in soft-threshold operator, the basic flow coefficients is substracted by $\frac{\lambda}{L}$. If we add one step between 9 and 10 in **Algorithm 3**,

$$\beta_{a0} \leftarrow \beta_{a0} + \mathbf{1}_{\{|\beta_a 0| \leq TOL\}}(\beta_{a0}) * \lambda_1 / L$$

revised burst flow is illustrated in Figure 5. It shows that our model and algorithm provides relatively precise detection.

5.4 Efficiency Comparison

In Table 1, we list the iteration times that every algorithm requires under the tolerance 10^{-6}. We conclude that the efficiency advantage between ADMM algorithm and PG algorithm depends on the property of actual data flow.

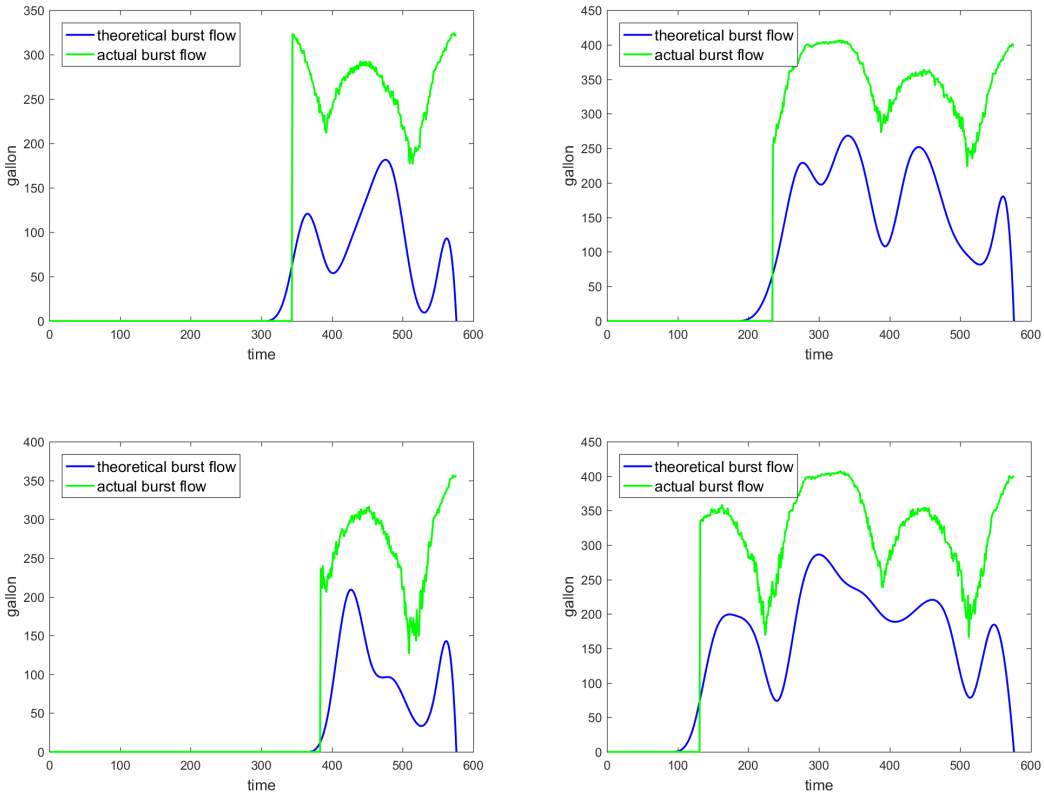

Figure 4: Comparison of Actual Burst Flow and Burst Flow

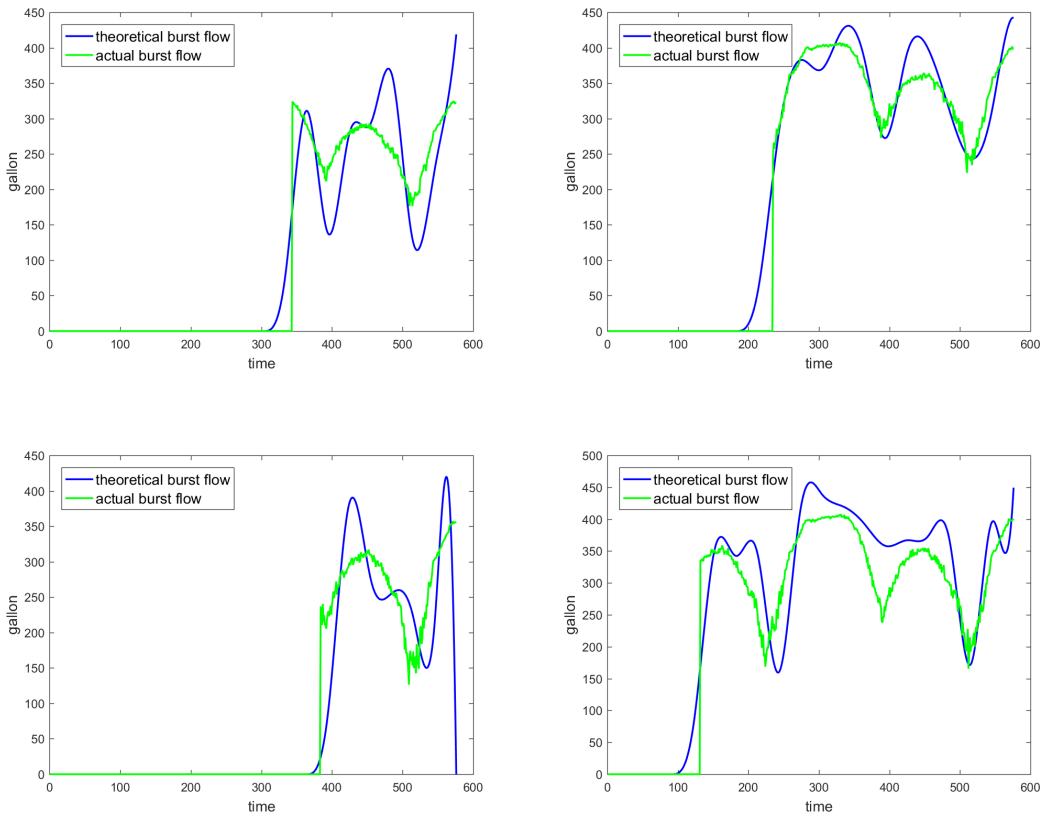

Figure 5: Comparison of Actual Burst Flow and Burst Flow(After Revision)

However, APG algorithm is significantly better than PG algorithm. Due to the fact that dual problem is fundamentally quadratic convex problem, utilizing the dual problem to solve the original problem possesses the highest efficiency.

DataFlow	ADMM	PG	APG	DUAL
1	97	74	42	7
2	95	138	86	7
3	101	124	65	7
4	82	86	55	6

Table 1: Efficiency Comparison

Moreover, if the dimension expands, the advantage relationship between these algorithm are still uncertain, which require further research. For instance, if the research object tranfer from one pipe into a number of pipes — pipe network, exponential times of computation is required and thus we need to avoid solving the inverse of a huge matrix. Even if in dual problem, we still need to figure out an optimal solution of a subproblem in a great dimension.

6 Conclusion

In this project, we have simulated basic theory in smooth-sparse decomposition applied in image segment research to realize a model in pipe burst detection problem. We have translated the original model into a weighted LASSO problem, proved several propositions as a basis of iteration algorithms. In genaral, ADMM method, which is universally utilized in a tremendous number of optimization problems, can reach the optimal solution at a relatively slow speed. While, proximal gradient method and its accelerated version, which to some extend borrow the idea of projection gradient method, can reach the same optimal solutions as well. To be more specific, accelerated proximal gradient method(APG) performs significantly better than ADMM method. However, we have calculated that the weighted LASSO problem has no duality gap between its primal problem and dual problem. If we adopt the dual problem to obtain the optimal solution, significantly less iterations are required, which is more efficient in this circumstance. What's more, further research is urgently needed to handle pipe network burst problems from another perspective.

References

[1] Ito N, Takeda A, Toh K C. A unified formulation and fast accelerated proximal gradient method for classification[J]. The Journal of Machine Learning Research, 2017, 18(1): 510-558.

[2] Elgabli A, Elghariani A, Al-Abbasi A O, et al. Two-stage LASSO admm signal detection algorithm for large scale mimo[C]//2017 51st Asilomar Conference on Signals, Systems, and Computers. IEEE, 2017: 1660-1664.

[3] Yan H, Paynabar K, Shi J. Anomaly detection in images with smooth background via smooth-sparse decomposition[J]. Technometrics, 2017, 59(1): 102-114.

[4] Parikh N, Boyd S. Proximal algorithms[J]. Foundations and Trends® in Optimization, 2014, 1(3): 127-239.

[5] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM journal on imaging sciences, 2009, 2(1): 183-202.