LASSO Problem

Xiaochuan Gong

October 18, 2022

Contents

1 Continuation Strategy for LASSO Problem

Consider LASSO problem

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1, \tag{1.1}$$

where μ is the regularization parameter. If μ is relatively small, then the convergence rate of directly solving the LASSO problem will be very slow. The continuation strategy is that we gradually decrease from a larger regularization parameter μ_t to μ (i.e. $\mu_1 \geq \cdots \geq \mu_{t-1} \geq \mu_t \geq \cdots \geq \mu$), and solve the corresponding LASSO problem:

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|^2 + \mu_t\|x\|_1. \tag{1.2}$$

The advantage of continuation strategy is that when solving the optimization problem corresponding to μ_t, the solution of the optimization problem corresponding to μ_{t-1} (the μ_1 subproblem uses random initial point) can be used as a good approximation to solution corresponding to μ_t, and thus we can solve the μ_t subproblem in a short time. We summarize the continuation strategy as Algorithm 1 below.

Algorithm 1 Continuation Strategy for LASSO Problem

Choose initial iterate x_0, regularization parameter $\mu > 0$, factor $\gamma \in (0, 1)$, initialize parameter $\mu_0 > \mu > 0$, $k \leftarrow 0$.
while $\mu_k \geq \mu$ **do**
    Set x_k as initial iterate, solve the problem $x_{k+1} = \arg\min_x \left\{\frac{1}{2}\|Ax - b\|^2 + \mu_k\|x\|_1\right\}$.
    if $\mu_k = \mu$ **then**
        Stop iterating and output x_{k+1}.
    else
        Update regularization parameter $\mu_{k+1} \leftarrow \max\{\mu, \gamma\mu_k\}$.
        $k \leftarrow k + 1$.
    end if
end while

We konw that the larger the μ_t is, the easier the corresponding LASSO problem can be solved. Therefore, the continuation strategy is equivalent to speeding up solving the solution of the original problem (1.1) by quickly solving a series of simple problems (1.2) (complex problems become simpler with a good initial iterate).

2 Subgradient Method for LASSO Problem

Consider LASSO problem

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1, \tag{2.1}$$

it's easy to check that

$$g = A^\top(Ax - b) + \mu\text{sgn}(x)$$

is a subgradient of $f(x)$. Thus the sugradient method for LASSO problem can be written as

$$x_{k+1} = x_k - \alpha_k(A^\top(Ax_k - b) + \mu\text{sgn}(x_k)), \tag{2.2}$$

where α_k can be some fixed step-size or diminishing step-sizes.

3 Gradient Method for LASSO Problem

3.1 Barzilai-Borwein Method

The Barzilai-Borwein (BB) method is a gradient method with modified step sizes, which is motivated by Newton's method but not involves any Hessian. In 1988, Barzilai and Borwein [1] proposed the following two novel step-sizes that signifcantly improve the performance of the standard gradient descent method at nearly no extra cost:

$$\alpha_k^{BB1} = \frac{s_{k-1}^\top s_{k-1}}{s_{k-1}^\top y_{k-1}} \quad \text{and} \quad \alpha_k^{BB2} = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}, \tag{3.1}$$

where

$$s_{k-1} = x_k - x_{k-1} \quad \text{and} \quad y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}). \tag{3.2}$$

Note that since x_{k-1} and $\nabla f(x_{k-1})$ and thus s_{k-1} and y_{k-1} are unavailable at $k = 0$, we apply the standard gradient descent method at $k = 0$ and start BB method at $k = 1$. During the iteration process, we can use either α_k^{BB1} or α_k^{BB2} or alternate between them. For example, we can fix $\alpha_k = \alpha_k^{BB1}$ or $\alpha_k = \alpha_k^{BB2}$ for a few consecutive steps, or we can just use $\alpha_k = \alpha_k^{BB1}$ in odd steps and use $\alpha_k = \alpha_k^{BB2}$ in even steps.

Remark. *In this section, we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is a general differentiable objective function instead of the objective function for LASSO problem.*

BB method performs very well on minimizing quadratic and many other functions. It is proved that the BB method converges superlinearly for minimizing two-dimensional strongly convex quadratic functions [1]. Although the BB method does not decrease the objective function value monotonically, extensive numerical experiments show that it performs much better than some other optimization methods. For general problems, the step-sizes calculated by (3.1) may be too large or too small, so we also need to truncate the step-sizes by upper and lower bounds, that is, choose $0 < \alpha_k < \alpha_M$ such that for all $k \in \mathbb{N}$,

$$\alpha_m \le \alpha_k \le \alpha_M.$$

Recall that $f(x_k)$ and $\|\nabla f(x_k)\|$ are not monotonically decreasing, thus there is no convergence guarantee for general smooth convex problems. We often pair up BB method with non-monotone line search as a safeguard on these general problems. We give a general framework of BB method with non-monotone line search as shown in Algorithm 2. Common non-monotone line search methods include Grippo's non-monotone line search [3] and Zhang-Hager's nonmonotone line search [4]. We also give the frameworks of these two non-monotone line search methods as Algorithm 3 and 4, respectively.

For general problems, the convergence of the BB method needs further study. But even so, using BB step-sizes usually reduce the number of iterations of the algorithm. Therefore, choosing the BB step-size is one of the commonly used acceleration strategies in practice.

3.2 Huber Smoothing Gradient Method

In this section, we consider applying gradient method to solve LASSO problem (1.1),

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1.$$

The objective function f of the LASSO problem is not smooth, and the gradient cannot be obtained at some points, so the gradient method cannot be used directly to solve the original problem. Considering that the non-smooth term of the objective function is $\|x\|_1$, which is actually the sum of the absolute values of each component of x, i.e.,

$$\|x\|_1 = |x_1| + \cdots + |x_n|,$$

Algorithm 2 BB Method with Non-Monotone Line Search

Choose initial iterate x_0, initialize $\alpha_0 > 0$, upper and lower bound $\alpha_M > \alpha_m > 0$, $k \leftarrow 0$.
while not converged **do**
  while not satisfying certain conditions **do**
    Compute α_k by Grippo's or Zhang-Hager's non-monotone line search condition.
  end while
  Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$.
  Compute the BB step-size α_{k+1} as the initial step-size for the next line search iteration according to one of (3.1) and truncate it by lower bound α_m and upper bound α_M,

$$\alpha_{k+1} \leftarrow \min\left\{\max\left\{\alpha_m, \alpha_{k+1}^{BB}\right\}, \alpha_M\right\}.$$

  Update parameters of Grippo's or Zhang-Hager's non-monotone line search framework.
  Set $k \leftarrow k+1$.
end while

Algorithm 3 Grippo's Non-Monotone Line Search Framework

Choose initial iterate x_0, initialize $c_1, \gamma \in (0,1)$, $0 < \lambda_1 < \lambda_2$, $M > 0$, $m_0 \leftarrow 0$, $k \leftarrow 0$.
while not converged **do**
  Initialize $\bar{\alpha}_k \in (\lambda_1, \lambda_2)$ and set $\alpha_k \leftarrow \bar{\alpha}_k$.
  while $f(x_k + \alpha_k d_k) > \max_{0 \le j \le m_k} f(x_{k-j}) + c_1 \alpha_k \nabla f(x_k)^\top d_k$ **do**
    Set $\alpha_k \leftarrow \gamma \alpha_k$
  end while
  Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$.
  Choose m_{k+1} satisfying $0 \le m_k \le m_{k+1} \le \min\{m_k + 1, M\}$.
  Set $k \leftarrow k+1$.
end while

Algorithm 4 Zhang-Hager's Non-Monotone Line Search Framework

Choose initial iterate x_0, initialize $0 < c_1 < c_2 < 1$, $0 \le \eta_{min} \le \eta_{max} \le 1$, $C_0 \leftarrow f(x_0)$, $Q_0 \leftarrow 1$, $k \leftarrow 0$.
while not converged **do**
  Compute α_k satisfying the modified Wolfe conditions:

$$f(x_k + \alpha_k d_k) \le C_k + c_1 \alpha_k \nabla f(x_k)^\top d_k,$$

$$\nabla f(x_k + \alpha_k d_k)^\top d_k \le c_2 \nabla f(x_k)^\top d_k;$$

  or satisfying the modified Armijo condition by backtracking:

$$f(x_k + \alpha_k d_k) \le C_k + c_1 \alpha_k \nabla f(x_k)^\top d_k.$$

  Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$.
  Choose $\eta_k \in [\eta_{min}, \eta_{max}]$, and set

$$Q_{k+1} \leftarrow \eta_k Q_k + 1 \quad \text{and} \quad C_{k+1} \leftarrow \frac{1}{Q_{k+1}}\left(\eta_k Q_k C_k + f(x_{k+1})\right).$$

  Set $k \leftarrow k+1$.
end while

if we can find a smooth function to approximate the absolute value function, then the gradient method can be used to solve the LASSO problem. In practical applications, we can consider the following one-dimensional smooth function $l_\delta : \mathbb{R} \to \mathbb{R}$:

$$l_\delta(x) = \begin{cases} \frac{1}{2\delta}x^2, & |x| < \delta, \\ |x| - \frac{\delta}{2}, & |x| \geq \delta. \end{cases} \tag{3.3}$$

Definition (3.3) is actually a modification of Huber loss, when $\delta \to 0$, the smooth function $l_\delta(x)$ and the absolute value function $|x|$ will get closer.

Therefore, we construct the smoothed LASSO problem as

$$\min_x f_\delta(x) = \frac{1}{2}\|Ax - b\|^2 + \mu L_\delta(x), \tag{3.4}$$

where δ is the given smoothing parameter and $L_\delta : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$L_\delta(x) = \sum_{i=1}^n l_\delta(x_i).$$

It is easy to calculate the gradient of $f_\delta(x)$ as

$$\nabla f_\delta(x) = A^\top(Ax - b) + \mu \nabla L_\delta(x),$$

where $\nabla L_\delta : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$(\nabla L_\delta(x))_i = \begin{cases} \operatorname{sgn}(x_i), & |x_i| > \delta, \\ \frac{x_i}{\delta}, & |x_i| \leq \delta. \end{cases}$$

Thus the Huber smoothing gradient method for LASSO problem can be written as

$$x_{k+1} = x_k - \alpha_k \left(A^\top(Ax - b) + \mu \nabla L_\delta(x)\right). \tag{3.5}$$

Obviously, $\nabla f_\delta(x)$ is Lipschitz continuous since for any $x, x' \in \mathbb{R}^n$, we have

$$\begin{aligned} \|\nabla f_\delta(x) - \nabla f_\delta(x')\| &= \|A^\top A(x - x') + \mu(\nabla L_\delta(x) - \nabla L_\delta(x'))\| \\ &\leq \|A^\top A\|_2 \|x - x'\| + \frac{\mu}{\delta}\|x - x'\| \\ &= \left(\|A^\top A\|_2 + \frac{\mu}{\delta}\right)\|x - x'\|. \end{aligned}$$

Thus f_δ is L-smooth, where $L = \|A^\top A\|_2 + \frac{\mu}{\delta}$. By convergence property of the gradient method, the fixed step-size $\alpha_k = \alpha$ needs to be no more than $1/L$ to ensure the convergence of the algorithm. If δ is too small, then we need to choose a sufficiently small step size α_k to make the gradient method converge.

4 Proximal-Gradient Method for LASSO Problem

4.1 Proximal-Gradient Method

In this section, we consider applying proximal-gradient method to solve LASSO problem (1.1),

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1.$$

Let

$$f(x) = \frac{1}{2}\|Ax - b\|^2 \quad \text{and} \quad h(x) = \mu\|x\|_1,$$

then we have

$$\begin{aligned}\nabla f(x) &= A^\top(Ax - b),\\ \text{prox}_{\alpha_k h} &= \text{sgn}(x)\max\{|x| - \alpha_k\mu, 0\}.\end{aligned}$$

Therefor, the proximal-gradient method for solving the LASSO problem can be written as

$$\begin{aligned}y_k &= x_k - \alpha_k A^\top(Ax_k - b),\\ x_{k+1} &= \text{sgn}(y_k)\max\{|y_k| - \alpha_k\mu, 0\}.\end{aligned} \tag{4.1}$$

That is, we first do gradient descent step, then we do soft thresholding step. In particular, the soft thresholding operator in the second step guarantees the sparse structure of the solution of LASSO problem. This also explains why the proximal-gradient method works well.

4.2 FISTA

In this section, we consider applying a fast iterative shrinkage-thresholding algorithm (FISTA) [2] to solve LASSO problem (1.1),

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1.$$

As we all know, the convergence rate for the proximal-gradient method (4.1) for solving LASSO problem (1.1) is $O(1/k)$. Naturally, we want to accelerate the convergence speed for proximal-gradient method, which is FISTA introduced in this section. In fact, FISTA can achieve a $O(1/k^2)$ convergence rate [2].

FISTA for solving the LASSO problem can be written as

$$\begin{aligned}y_{k+1} &= x_k + \frac{\theta_k - 1}{\theta_{k+1}}(x_k - x_{k-1}),\\ x_{k+1} &= \text{prox}_{\alpha_k h}(y_{k+1} - \alpha_k \nabla f(y_{k+1})),\end{aligned} \tag{4.2}$$

where

$$\theta_0 = 1 \quad \text{and} \quad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}.$$

4.3 BB Step-Size with Line Search

Denote $F(x) \coloneqq f(x) + h(x)$, where f is differentiable and h is convex. In the previous settings, we choose $f(x) = \frac{1}{2}\|Ax - b\|^2$ and $h(x) = \mu\|x\|_1$. Obviously, f is L-smooth, where $L = \lambda_{max}(A^\top A)$. By convergence property of the proximal-gradient method, we know that if f is convex or strongly convex and we choose a fixed step-size $\alpha_k = \alpha$ with $0 < \alpha \le 1/L$, then F can converge to its minimum. However, sometimes it's hard for us to know the exact value of L, thus we consider using the BB step-size with line search in some of the settings. Since F is not necessarily differentiable, so when we use (3.1) and (3.2) to compute the BB step-size, we should take y_{k-1} as $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$ (here ∇f represents the gradient of the smooth part of the objective function F). Also, according to the convergence proof of the proximal-gradient method, we need to choose α_k satisfying the following inequality in the line search process (Zhang-Hager's framework),

$$F(x_{k+1}) \le C_k - \frac{c_1}{2\alpha_k}\|x_{k+1} - x_k\|^2,$$

where $c_1 \in (0, 1)$, $C_0 = F(x_0)$ and $C_{k+1} = \frac{1}{Q_{k+1}}\left(\eta_k Q_k C_k + F(x_{k+1})\right)$. We summarize the BB method and Zhang-Hager's line search framework for proximal-gradient method as Algorithm 5.

Algorithm 5 BB Step-Size with Zhang-Hager's Line Search for Proximal-Gradient Method

Choose initial iterate x_0, initialize $\alpha_0 > 0$, $\rho, \eta, c_1 \in (0,1)$, upper and lower bound $\alpha_M > \alpha_m > 0$, positive integer $nls.max > 0$, $C_0 \leftarrow F(x_0)$, $Q_0 \leftarrow 1$, $k \leftarrow 0$.
while not converged **do**
  Set $x_{k+1} \leftarrow \mathrm{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k))$.
  Set $nls \leftarrow 0$.
  while $F(x_{k+1}) > C_k - \frac{c_1}{2\alpha_k}\|x_{k+1} - x_k\|^2$ and $nls < nls.max$ **do**
    Set $\alpha_k \leftarrow \rho\alpha_k$, $nls \leftarrow nls + 1$.
    Set $x_{k+1} \leftarrow \mathrm{prox}_{\alpha_k h}(x_k - \alpha_k \nabla f(x_k))$.
  end while
  Compute the BB step-size α_{k+1} as the initial step-size for the next line search iteration according to one of (3.1) and truncate it by lower bound α_m and upper bound α_M,

$$\alpha_{k+1} \leftarrow \min\left\{\max\left\{\alpha_m, \alpha_{k+1}^{BB}\right\}, \alpha_M\right\}.$$

  Set $Q_{k+1} \leftarrow \eta Q_k + 1$ and $C_{k+1} \leftarrow \frac{1}{Q_{k+1}}(\eta Q_k C_k + F(x_{k+1}))$.
  Set $k \leftarrow k + 1$.
end while

Similarly, according to the convergence proof of FISTA, we need to choose α_k satisfying the following inequality in the line search process (Zhang-Hager's framework),

$$F(x_{k+1}) \le C_k - \frac{c_1}{2\alpha_k}\|x_{k+1} - y_{k+1}\|^2,$$

where $c_1 \in (0,1)$, $C_0 = F(x_0)$ and $C_{k+1} = \frac{1}{Q_{k+1}}(\eta_k Q_k C_k + F(x_{k+1}))$. We summarize the BB method and Zhang-Hager's line search framework for FISTA as Algorithm 6.

5 Experiments

In the MATLAB environment, we generate $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$ as follows:

```
m = 512; n = 1024;
A = randn(m, n);
u = sprandn(n, 1, r);
b = A * u;
```

Here r is used to control the sparsity of the true solution u (the ratio of the number of non-zero elements to the total number of elements in u is r). We take the sparsity $r = 0.1$ in the following discussions and μ is varied in different methods.

5.1 Subgradient Method

We take the regularization parameter $\mu = 1$ and now we use the subgraient method to solve the LASSO problem. We choose the fixed step-sizes $\alpha_k = 0.0005, 0.0002, 0.0001$ and the diminishing step-size $\alpha_k = 0.002/\sqrt{k}$ respectively to test the performance of the subgradient method. As can be seen from Figure 1, within 3000 iterations, if we use different fixed step-sizes, the subgradient method will eventually reach a sub-optimal solution, and the function value will stabilize around a certain value when it decreases to a certain extent; while if we use diminishing step-sizes, the subgradient method will eventually converge, though the convergence rate is slower. It can also be seen from the Figure 1

Algorithm 6 BB Step-Size with Zhang-Hager's Line Search for FISTA

Choose initial iterate x_{-1}, x_0, y_0, initialize $\alpha_0 > 0$, $\rho, \eta, c_1 \in (0,1)$, upper and lower bound $\alpha_M > \alpha_m > 0$, positive integer $nls.max > 0$, $C_0 \leftarrow F(x_0)$, $Q_0 \leftarrow 1$, $\theta_0 \leftarrow 1$, $k \leftarrow 0$.
while not converged **do**
  Set $\theta_{k+1} = \dfrac{1+\sqrt{1+4\theta_k^2}}{2}$
  Set $y_{k+1} \leftarrow x_k + \dfrac{\theta_k - 1}{\theta_{k+1}}(x_k - x_{k-1})$.
  Set $x_{k+1} \leftarrow \mathrm{prox}_{\alpha_k h}(y_{k+1} - \alpha_k \nabla f(y_{k+1}))$.
  Set $nls \leftarrow 0$.
  while $F(x_{k+1}) > C_k - \dfrac{c_1}{2\alpha_k}\|x_{k+1} - y_{k+1}\|^2$ and $nls < nls.max$ **do**
    Set $\alpha_k \leftarrow \rho\alpha_k$, $nls \leftarrow nls + 1$.
    Set $x_{k+1} \leftarrow \mathrm{prox}_{\alpha_k h}(y_{k+1} - \alpha_k \nabla f(y_{k+1}))$.
  end while
  Compute the BB step-size α_{k+1} as the initial step-size for the next line search iteration according to one of (3.1) and truncate it by lower bound α_m and upper bound α_M,

$$\alpha_{k+1} \leftarrow \min\left\{\max\left\{\alpha_m, \alpha_{k+1}^{BB}\right\}, \alpha_M\right\}.$$

  Set $Q_{k+1} \leftarrow \eta Q_k + 1$ and $C_{k+1} \leftarrow \dfrac{1}{Q_{k+1}}\left(\eta Q_k C_k + F(x_{k+1})\right)$.
  Set $k \leftarrow k + 1$.
end while

that the subgradient method itself is a non-monotonic method, so we need to record the optimal value $\hat{f}(x_k)$ of the objective function in the history as the final output of the algorithm.

5.2 Subgradient Method with Continuation Strategy

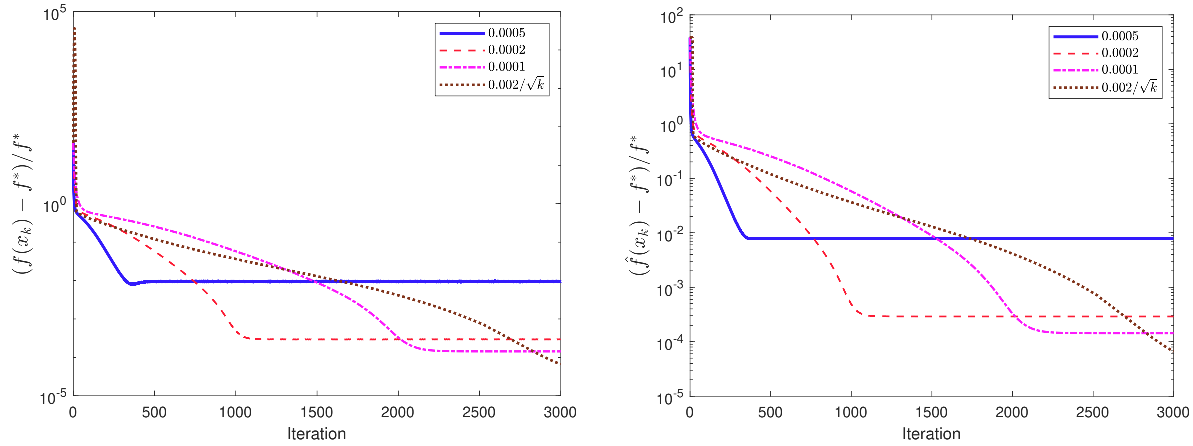
We take the regularization parameter $\mu = 10^{-2}, 10^{-3}$ respectively. As we know, when μ is relatively small, the convergence rate of the LASSO problem will be very slow if we solve it directly. Therefore, we use the subgradient method with continuation strategy to solve the LASSO problem. The iterative process is shown in Figure 2. It can be seen that the subgradient method with continuation strategy converges around 1200 steps.

5.3 Huber Smoothing Gradient Method with Continuation Strategy

We take the regularization parameter $\mu = 10^{-3}$. In order to speed up the convergence of the algorithm, the continuation strategy can be used along with the Huber smoothing gradient method. Specifically, for each μ_t, we use the Huber smoothing gradient method with BB step-size (here the smoothing parameter $\delta_t = 10^{-3}\mu_t$) to solve the corresponding subproblem. When the subproblem of μ_t is solved, we set $\mu_{t+1} = \max\{\eta\mu_t, \mu\}$ to solve the next subproblem, where $\eta \in (0,1)$ is the shrinking factor. As can be seen from Figure 3, within about 400 iterations, the Huber smoothing gradient method with continuation strategy converges to the solution of the original LASSO problem. Also, the solution solved by gradient method is sparse as we expected.

5.4 Proximal-Gradient Method with Continuation Strategy

We take the regularization parameter $\mu = 10^{-3}$ and now we use the proximal-gradient method and FISTA along with continuation strategy to solve the LASSO problem. We choose the fixed step size $\alpha_k = 1/L$, where $L = \lambda_{max}(A^\top A)$, and the BB step-size combined with line search, respectively. It



(a) (b)

Figure 1: Subgradient Method

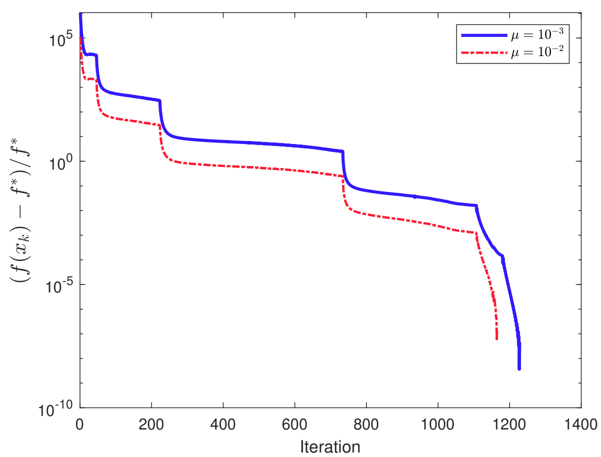
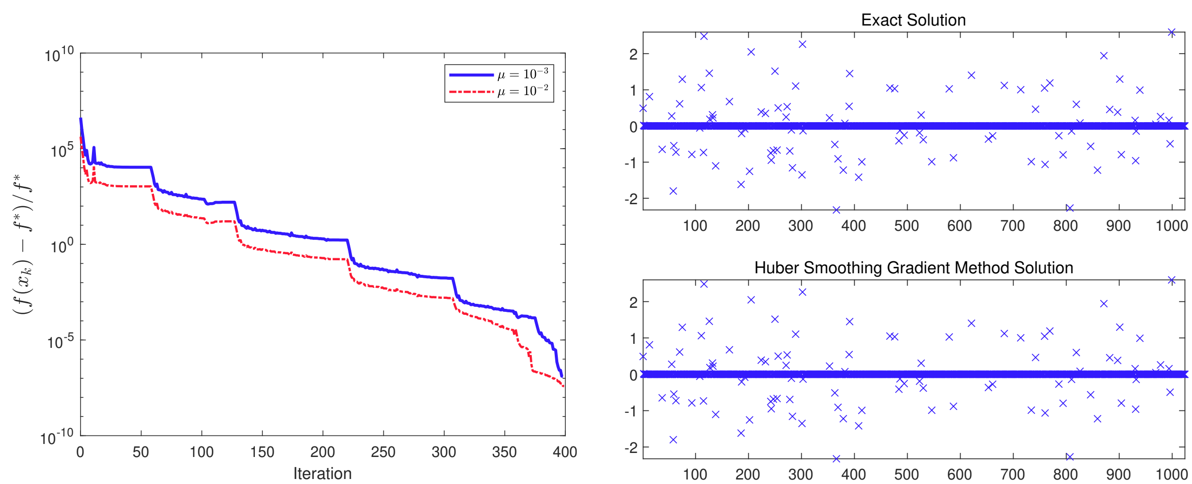


Figure 2: Subgradient Method with Continuation Strategy



(a) (b)

Figure 3: Huber Smoothing Gradient Method with Continuation Strategy

can be seen from Figure 4 that the BB step-size combined with line search can significantly improve the convergence speed of the algorithm, and it converges faster than the Huber smoothing gradient method.

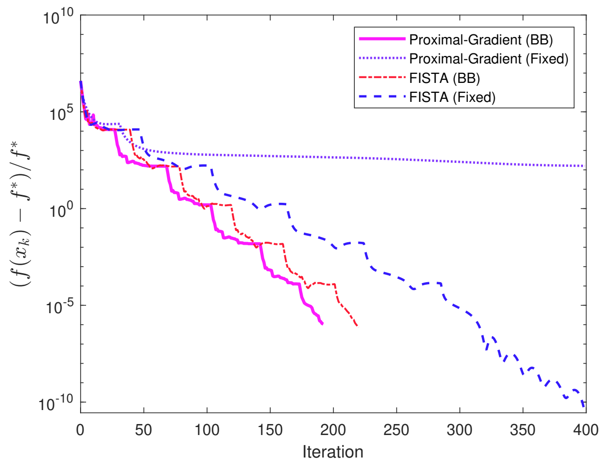


Figure 4: Proximal-Gradient Method with Continuation Strategy

References

[1] Jonathan Barzilai and Jonathan M. Borwein. "Two-Point Step Size Gradient Methods". In: *IMA Journal of Numerical Analysis* 8.1 (1988), pp. 141–148.

[2] Amir Beck and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[3] L. Grippo, F. Lampariello, and S. Lucidi. "A Nonmonotone Line Search Technique for Newton's Method". In: *SIAM Journal on Numerical Analysis* 23.4 (1986), pp. 707–716.

[4] Hongchao Zhang and William W. Hager. "A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization". In: *SIAM Journal on Optimization* 14.4 (2004), pp. 1043–1056.