Report of Software Implementation Project for "Convex Optimization"

Hanju Wu
Sun Yat-Sen University
wuhj29@mail2.sysu.edu.cn

2022 年 9 月 19 日

目录

1 Background

Consider the composite optimization problem

$$\min_x f(x) + h(x)$$

where $f(x)$ is differentiable and $h(x)$ is a function whose proximal operator is easily available.

Both $f(x)$ and $h(x)$ may be nonconvex.

Let $h(x)$ be a proper and close function, and $\inf_{x \in \mathrm{dom}\, h} h(x) > -\infty$.

The proximal operator of $h(x)$ is defined as

$$\mathrm{prox}_h(x) = \arg\min_x h(u) + \frac{1}{2}\|u - x\|_2^2$$

2 Conjugate Function

Definition 2.1. *conjugate function of a proper function*

$$f^*(y) = \sup_{x \in \operatorname{dom} f} \left\{ y^{\mathrm{T}} x - f(x) \right\}$$

Example 2.2. *Indicative functions of convex sets*

$$I_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C \end{cases}$$

$$I_C^*(y) = \sup_x \left\{ y^{\mathrm{T}} x - I_C(x) \right\} = \sup_{x \in C} y^{\mathrm{T}} x.$$

where $I_C^*(y)$ call the support function of convex set C

Example 2.3. *Norm*

$$f(x) = \|x\|, f^*(y) = \begin{cases} 0, & \|y\|_* \leqslant 1 \\ +\infty, & \|y\|_* > 1 \end{cases}$$

where $\|y\|_* = \sup_{\|x\| \leqslant 1} x^{\mathrm{T}} y$

Example 2.4. *operation rules of conjugate function*

(1)

$$f\left(x_1, x_2\right) = g\left(x_1\right) + h\left(x_2\right), \quad f^*\left(y_1, y_2\right) = g^*\left(y_1\right) + h^*\left(y_2\right)$$

(2)

$$f(x) = \alpha g(x), \quad f^*(y) = \alpha g^*(y / \alpha)$$

3 Proximal Operator

3.1 Some method to calculate Proximal Operator

Theorem 3.1. *h is a proper and close convex function defined on $\mathbb{R}^n$, then we have:*

$$u = \operatorname{prox}_h(x) \Longleftrightarrow x - u \in \partial h(u).$$

Lemma 3.2. *(Moreau) f is a proper and close convex function defined on $\mathbb{R}^n$, $x \in \mathbb{R}^n$,*

$$x = \operatorname{prox}_f(x) + \operatorname{prox}_{f^*}(x);$$

$$x = \operatorname{prox}_{\lambda f}(x) + \lambda \operatorname{prox}_{\lambda^{-1} f^*}\left(\frac{x}{\lambda}\right) = \operatorname{prox}_{\lambda f}(x) + \operatorname{prox}_{\lambda f^*\left(\frac{x}{\lambda}\right)}(x)\ (\lambda > 0)$$

if the proximal operator of conjugate function is easy to calculate,there we have

$$\operatorname{prox}_f(x) = x - \operatorname{prox}_{f^*}(x);$$

Example 3.3. *(operation rules of proximal operator)*

(1)

$$h(x) = g(\lambda x + a), \quad \operatorname{prox}_h(x) = \frac{1}{\lambda}\left(\operatorname{prox}_{\lambda^2 g}(\lambda x + a) - a\right); (\lambda \neq 0)$$

(2)

$$h(x) = \lambda g\left(\frac{x}{\lambda}\right), \quad \operatorname{prox}_h(x) = \lambda \operatorname{prox}_{\lambda^{-1} g}\left(\frac{x}{\lambda}\right); (\lambda > 0)$$

(3)

$$h(x) = g(x) + a^{\mathrm{T}} x, \quad \operatorname{prox}_h(x) = \operatorname{prox}_g(x - a).$$

(4)

$$h(x) = g(x) + \frac{u}{2}\|x - a\|_2^2, \operatorname{prox}_h(x) = \operatorname{prox}_{\theta g}(\theta x + (1-\theta)a), \theta = \frac{1}{1+u}, (u > 0)$$

(5)

$$h\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \varphi_1(x) + \varphi_2(y), \quad \operatorname{prox}_h\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} \operatorname{prox}_{\varphi_1}(x) \\ \operatorname{prox}_{\varphi_2}(y) \end{bmatrix}.$$

(6)

$$h(x) = ag(x) + b, a > 0, \operatorname{prox}_h(x) = \operatorname{prox}_{ag}(x)$$

(7)

$$h(x) = g(Ax+b)$$

if

$$AA^{\mathrm{T}} = \frac{1}{\alpha} I$$

we have

$$\operatorname{prox}_h(x) = \left(I - \alpha A^{\mathrm{T}} A\right) x + \alpha A^{\mathrm{T}} \left(\operatorname{prox}_{\alpha^{-1} g}(Ax+b) - b\right)$$

Example 3.4. *(Projection on a closed convex set)*

$$I_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & \textit{else} \end{cases}$$

$$\begin{aligned} \operatorname{prox}_{I_C}(x) &= \arg\min_u \left\{ I_C(u) + \frac{1}{2}\|u-x\|^2 \right\} \\ &= \arg\min_{u \in C} \|u-x\|^2 = \mathcal{P}_C(x) \end{aligned}$$

Lemma 3.5. *(KKT Condition)*

$$\min_{x \in \mathcal{X}} f(x)$$

with

$$\mathcal{X} = \left\{ x \in \mathbb{R}^n \mid c_i(x) \leqslant 0, i \in \mathcal{I} \text{ 且 } c_i(x) = 0, i \in \mathcal{E} \right\}.$$

if x^* is a local optimal solution, and $T_{\mathcal{X}}(x^*) = \mathcal{F}(x^*)$,

then there exist λ_i^* satisfy the following condition:

$$\begin{aligned} &\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + \textstyle\sum_{i \in \mathcal{I} \cup \mathcal{E}} \lambda_i^* \nabla c_i(x^*) = 0 \\ &c_i(x^*) = 0, \forall i \in \mathcal{E} \\ &c_i(x^*) \leqslant 0, \forall i \in \mathcal{I} \\ &\lambda_i^* \geqslant 0, \forall i \in \mathcal{I} \\ &\lambda_i^* c_i(x^*) = 0, \forall i \in \mathcal{I} \end{aligned}$$

3.2 Proximal Operator of $h(x)$

3.2.1 General functions

Example 3.6. *(General functions)*

(1) $\ell_0 - norm$:

$$h(x) = \|x\|_0, \quad \operatorname{prox}_{th}(x) = \begin{cases} 0 & \text{if } x^2 < 2t \\ \{x, 0\} & \text{if } x^2 = 2t \\ x & \text{otherwise} \end{cases}$$

(2) sum-of-norms (for group lasso):

$$h(x) = \sum_{g \in \mathcal{G}} \|x_g\|_2 \,, \mathcal{G} \text{ is a division of } [n]$$

$$(\operatorname{prox}_{th}(x))_g = \left(1 - \frac{t}{\|x_g\|_2}\right)_+ x_g$$

$$x_+ = \max\{x, 0\}$$

(3) elastic-net:

$$h(x) = \|x\|_1 + (\lambda/2)\|x\|_2^2, \quad \operatorname{prox}_{th}(x) = \frac{1}{1 + \lambda t} \operatorname{prox}_{t\|x\|_1}(x)$$

(4) log-barrier and its variant:

$$h(x) = -\sum_{i=1}^{n} \ln x_i, \quad (\operatorname{prox}_{th}(x))_i = \frac{x_i + \sqrt{x_i^2 + 4t}}{2}, \quad i = 1, 2, \cdots, n.$$

$$h(x) = -\sum_{i=1}^{n} \ln x_i - (\lambda/2)\|x\|_2^2, (\operatorname{prox}_{th}(x))_i = \frac{x_i + \sqrt{x_i^2 + 4t(1 - \lambda t)}}{2(1 - \lambda t)}, \quad i = 1, 2, \cdots, n.$$

(5) ReLU function:

$$h(x) = \sum_{i=1}^{n} \max(0, x_i), (\operatorname{prox}_{th}(x))_i = \begin{cases} x_i & \text{if } x_i < 0 \\ 0 & \text{if } 0 \le x \le t \\ x_i - t & \text{otherwise} \end{cases}$$

(6) quadratic function:

$$h(x)=\frac{1}{2}x^{\mathrm{T}}Ax+b^{\mathrm{T}}x+c,\quad \operatorname{prox}_{th}(x)=(I+tA)^{-1}(x-tb).$$

where A is positive semidefinite

(7) maximal function:

$$h(x)=\max_i x_i,\quad (\operatorname{prox}_{th}(x))_i=\min\{x_i,\lambda\}$$

where $\lambda\in\mathbb{R}$ is such that $\sum_{i=1}^n \max\{0,x_i-\lambda\}=t$

Algorithm 1: Proximal Operator of maximal function

Input: $\mathbf{x}\in R^n$, t

1 Descending sort of $\mathbf{x}\longrightarrow\boldsymbol{\mu}:\mu_1\geq\mu_2\geq\ldots\geq\mu_n$

2 Find $\rho(\boldsymbol{\mu})=\max\left\{j\in[n]:\mu_j-\frac{1}{j}\left(\sum_{r=1}^j\mu_r-t\right)>0\right\}$

3 Defined $\theta=\frac{1}{\rho}\left(\sum_{i=1}^\rho\mu_i-t\right)$

Output: $(\operatorname{prox}_{th}(x))_i=\min\{x_i,\theta\}$

(8) conjugate of a proper closed function:

$$h(x)=f^*,\operatorname{prox}_{th}(x)=x-t\operatorname{prox}_{t^{-1}f}\left(\frac{x}{t}\right)=x-\operatorname{prox}_{tf\left(\frac{x}{t}\right)}(x)$$

(9) $\ell_1-norm:$

$$h(x)=\|x\|_1,\quad \operatorname{prox}_{th}(x)=\operatorname{sign}(x)\max\{|x|-t,0\}.$$

(10) $\ell_2-norm:$

$$h(x)=\|x\|_2,\quad \operatorname{prox}_{th}(x)=\begin{cases}\left(1-\frac{t}{\|x\|_2}\right)x, & \|x\|_2\geqslant t,\\ 0, & \text{其他}.\end{cases}$$

(11) $\ell_\infty-norm:$

$$h(x)=\|x\|_\infty,(\operatorname{prox}_{th}(x))_i=(\operatorname{sign}(x_i)\min\{|x_i|,\lambda\})_{1\leq i\leq n}$$

where $\lambda\in\mathbb{R}$ is such that $\sum_{i=1}^n\max\{0,|x_i|-\lambda\}=t$

Algorithm 2: Proximal Operator of $\ell_\infty - norm$

Input: $\mathbf{x} \in R^n$, t

1 $u_i = |x_i|$,Descending sort of $\mathbf{u} \longrightarrow \boldsymbol{\mu} : \mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$

2 Find $\rho(t, \boldsymbol{\mu}) = \max \left\{ j \in [n] : \mu_j - \frac{1}{j} \left(\sum_{r=1}^{j} \mu_r - t \right) > 0 \right\}$

3 Defined $\theta = \frac{1}{\rho} \left(\sum_{i=1}^{\rho} \mu_i - t \right)$

Output: $(\text{prox}_{th}(x))_i = (\text{sign}\,(x_i) \min \{|x_i|, \theta\})_{1 \leq i \leq n}$

or

$$\text{prox}_{th}(x) = x - \text{prox}_{I_C}(x)$$

where

$$I_C(x) = \begin{cases} 0, & \|x\|_1 \leqslant t \\ +\infty, & \|x\|_1 > t \end{cases}$$

3.2.2 Indicator functions

Example 3.7. *(Indicator functions)*

(1) $\ell_\infty - ball$:

$$I_C(x) = \begin{cases} 0, & \|x\|_\infty \leqslant t \\ +\infty, & \|x\|_\infty > t \end{cases}$$

$$\text{prox}_{I_C}(x) = x - \text{prox}_{t\|x\|_1}(x) = \text{sign}\,(x) \min \{|x|, t\}$$

(2) $\ell_2 - ball$:

$$I_C(x) = \begin{cases} 0, & \|x\|_2 \leqslant t \\ +\infty, & \|x\|_2 > t \end{cases}$$

$$\text{prox}_{I_C}(x) = x - \text{prox}_{t\|x\|_2}(x) = \frac{t}{\max \{t, \|x\|_2\}} x = \begin{cases} \frac{t}{\|x\|_2} x, & \|x\|_2 \geqslant t, \\ x, & \text{其他}. \end{cases}$$

(3) simple box:

$$I_C(x) = \begin{cases} 0, & l \leq x_i \leq u, 1 \leq i \leq n \\ +\infty, & else \end{cases}$$

$$(\text{prox}_{I_C}(x))_i = [\max \{l, \min \{x_i, u\}\}]_{1 \leq i \leq n}$$

(4) hyperplane:

$$I_C(x) = \begin{cases} 0, & a^T x = b \\ +\infty, & else \end{cases}$$

$$\operatorname{prox}_{I_C}(x) = x + \frac{b - a^T x}{\|a\|_2^2} a$$

(5) half space:

$$I_C(x) = \begin{cases} 0, & a^T x \leq b \\ +\infty, & else \end{cases}$$

$$\operatorname{prox}_{I_C}(x) = \begin{cases} x + \frac{b - a^T x}{\|a\|_2^2} a & \text{if } a^T x > b \\ x & \text{if } a^T x \leq b \end{cases}$$

(6) affine set:

$$I_C(x) = \begin{cases} 0, & Ax = b \\ +\infty, & else \end{cases}$$

$$\operatorname{prox}_{I_C}(x) = x + A^T \left(AA^T\right)^{-1} (b - Ax)$$

(7) probability simplex:

$$I_C(x) = \begin{cases} 0, & \sum_{i=1}^n x_i = 1, x_i \geq 0 \\ +\infty, & else \end{cases}$$

$$(\operatorname{prox}_{I_C}(x))_i = \max\{0, x_i - \lambda\}_{1 \leq i \leq n} \text{ with } \lambda \in \mathbb{R} \text{ such that } \sum_{i=1}^n \max\{0, x_i - \lambda\} = 1$$

Algorithm 3: Projection to simplex

Input: $\mathbf{x} \in R^n$

1 Descending sort of $\mathbf{x} \longrightarrow \boldsymbol{\mu} : \mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$

2 Find $\rho(\boldsymbol{\mu}) = \max\left\{j \in [n] : \mu_j - \frac{1}{j}\left(\sum_{r=1}^j \mu_r - 1\right) > 0\right\}$

3 Defined $\theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho} \mu_i - 1\right)$

Output: $(\operatorname{prox}_{I_C}(x))_i = \max\{x_i - \theta, 0\}$

---

(8) $\ell_1 -$ ball :

$$I_C(x) = \begin{cases} 0, & \|x\|_1 \leqslant t \\ +\infty, & \|x\|_1 > t \end{cases}$$

$$\operatorname{prox}_{I_C}(x) = x - \operatorname{prox}_{t\|x\|_\infty}(x)$$

Algorithm 4: Projection to $\ell_1 - ball$

Input: $\mathbf{x} \in R^n$ and t

if $\|x\|_1 \leqslant t$ **then**

  $\operatorname{prox}_{I_C}(x) = x$

else

  $u_i = |x_i|$,Descending sort of $\mathbf{u} \longrightarrow \boldsymbol{\mu} : \mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$

  Find $\rho(t, \boldsymbol{\mu}) = \max\left\{j \in [n] : \mu_j - \frac{1}{j}\left(\sum_{r=1}^{j} \mu_r - t\right) > 0\right\}$

  Defined $\theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho} \mu_i - t\right)$

  $(\operatorname{prox}_{I_C}(x))_i = \operatorname{sign}(x_i) \max\{|x_i| - \theta, 0\}$

end

(9) $\ell_0 - ball$:

$$I_C(x) = \begin{cases} 0, & \|x\|_0 \leqslant t \\ +\infty, & \|x\|_0 > t \end{cases}$$

Algorithm 5: Projection to $\ell_0 - ball$

Input: $\mathbf{x} \in R^n$ and t

Descending sort of $\mathbf{x} \longrightarrow \boldsymbol{\mu} : \mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$

Set $\mathbf{y} = \mathbf{x}$

Set $y_k = 0$, where x_k is the one to one correspondence of μ_i $\lfloor t \rfloor < i \leq n$

Output: $\operatorname{prox}_{I_C}(x) = \mathbf{y}$

4 Gradient of $f(x)$

4.1 Least squares

$$f(x) = \frac{1}{2}\|Ax-b\|_2^2 \text{ or } f(x) = \frac{1}{2}\|Ax-b\|_F^2$$
$$\nabla f(x) = A^{\mathrm{T}}(Ax-b)$$

4.2 Variant of Huber loss

$$(f(x))_i = \begin{cases} \frac{1}{2\delta}x_i^2, & |x_i| < \delta \\ |x_i| - \frac{\delta}{2}, & \text{otherwise} \end{cases}$$
$$(\nabla f(x))_i = \begin{cases} \frac{1}{\delta}x_i, & |x_i| < \delta \\ -1, & x_i \leq -\delta \\ 1, & x_i \geq \delta \end{cases}$$

4.3 Logistic regression

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \ln\left(1 + \exp\left(-b_i a_i^T x\right)\right)$$
$$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{m} \frac{\exp\left(-b_i a_i^T x\right)}{1 + \exp\left(-b_i a_i^T x\right)} - b_i a_i$$

Due to computer byte limit, we have to calculate:

$$\ln\left(1 + \exp\left(x\right)\right) = \begin{cases} x + \ln\left(1 + \exp\left(-x\right)\right), & x \geq 0 \\ \ln\left(1 + \exp\left(x\right)\right), & x < 0 \end{cases}$$

$$\frac{\exp\left(x\right)}{1 + \exp\left(x\right)} = \begin{cases} \frac{1}{1+\exp(-x)}, & x \geq 0 \\ \frac{\exp(x)}{1+\exp(x)}, & x < 0 \end{cases}$$

5 Function

5.1 $f(x)$

(1) Least squares

head file in *include/OptSuite/Base/func/axmb_norm_sqr.h*

sourse codes in *src/Base/func/axmb_norm_sqr.cpp*

(2) Logistic regression

head file in *include/OptSuite/Base/func/LogisticRegression.h*

sourse codes in *src/Base/func/LogisticRegression.cpp*

5.2 $h(x)$

5.2.1 general function

(1) ℓ_0 and ℓ_∞ norm

head file in *include/OptSuite/Base/func/shrinkage_lp.h*

sourse codes in *src/Base/func/shrinkage_lp.cpp*

(2) maximal function

head file in *include/OptSuite/Base/func/maximal.h*

sourse codes in *src/Base/func/maximal.cpp*

5.2.2 indicator function

(1) ℓ_0,ℓ_1,ℓ_2,ℓ_∞ ball

head file in *include/OptSuite/Base/func/ball_lp.h*

sourse codes in *src/Base/func/ball_lp.cpp*

(2) probability simplex

head file in *include/OptSuite/Base/func/probability_simplex.h*

sourse codes in *src/Base/func/probability_simplex.cpp*

6 Model

6.1 logistic_regression_l1 and logistic_regression_l2

6.1.1 code

Composite/Model/logistic_regression_l1.cpp
Composite/Model/logistic_regression_l2.cpp

6.1.2 test

example/LogisticRegression_L1.cpp
example/LogisticRegression_L2.cpp

```
    int m = 256;
    int n = 512;
    int l = 1;

    // set rng seed
    rng(233);

    Mat A = randn(n, m);
    Mat b = randn(m, l);
    for (Index i = 0; i < m; i++){
        b.coeffRef(i,0) = (rand() % 2 == 0 ? 1 : -1);
    }
    Scalar mu = 0.001;
    Mat x0 = randn(n, l);

    Composite::Model::LogisticRegression_L1 model(A, b, mu);
    //Composite::Model::LogisticRegression_L2 model(A, b, mu)
        ;
    Composite::Solver::ProxGBB<> alg;
```

7 Solver

7.1 FISTA

7.1.1 Code

Composite/Solver/proxgbb_FISTA

7.1.2 Calculate y^k

$$y^k = x^{k-1} + \frac{k-2}{k+1}\left(x^{k-1} - x^{k-2}\right)$$

```
template<typename xtype = mat_wrapper_t>
mat_t getmat(var_t_ptr x)  {
    xtype* sol = dynamic_cast<xtype*>(x.get());
    OPTSUITE_ASSERT(sol);
    return (*sol).mat();
}
```

```
template<typename dtype>
void ProxGBBFISTA<dtype>::set_y(var_t_ptr x, var_t_ptr xp,
  var_t_ptr y, Scalar theta){
   mat_t A = getmat(x) + theta * (getmat(x) - getmat(xp));
   const var_t& a=mat_wrapper_t(A);
   y->assign(a);
}
```

7.1.3 Compute BB step size

calculate:

$$\begin{aligned} dyg &= \langle y - yp, gy - gyp\rangle \\ &= y\_gy + yp\_gyp - y\_gyp - yp\_gy \end{aligned}$$

$$\mathrm{d}y^k = y^{k+1} - y^k,\ \mathrm{d}g^k = g^{k+1} - g^k$$

if (bb_variant == 0 and iter % 2 == 0) or bb_variant == 1

$$\tau=\frac{\left(\mathrm{d}y^k\right)^{\top}\mathrm{d}y^k}{\left(\mathrm{d}y^k\right)^{\top}\mathrm{d}g^k}$$

if (bb_variant == 0 and iter % 1 == 0) or bb_variant == 2

$$\tau=\frac{\left(\mathrm{d}y^k\right)^{\top}\mathrm{d}g^k}{\left(\mathrm{d}g^k\right)^{\top}\mathrm{d}g^k}$$

7.1.4 Line search

(1) "Zhang, Hager" Rule:

$$\psi\left(x^k\right)\leq C_{k-1}-\frac{1}{2}t_k\rho\left\|x^k-y^k\right\|_2^2$$

with:

$$C^0=f\left(x^0\right),C^k=\frac{1}{Q^k}\left(\eta Q^{k-1}C^{k-1}+\psi\left(x^k\right)\right)$$

where:

$$\left\{Q^k\right\}\text{ satisfy }Q^0=1,Q^{k+1}=\eta Q^k+1,\eta\in(0,1)$$

(2) "Standard" Rule:

$$f\left(x^k\right)\leqslant f\left(y^k\right)+\left\langle\nabla f\left(y^k\right),x^k-y^k\right\rangle+\frac{1}{2t_k}\left\|x^k-y^k\right\|_2^2$$

7.2 Nesterov's 2nd method

7.2.1 Code

Composite/Solver/Nesterov_2nd

7.2.2 iteration

define $\gamma_k=\frac{2}{k+1}$, renew $\{x^k\}\{y^k\}\{z^k\}$

$$\begin{aligned}
z^k&=(1-\gamma_k)x^{k-1}+\gamma_k y^{k-1},\\
y^k&=\mathrm{prox}_{(t_k/\gamma_k)h}(y^{k-1}-\tfrac{t_k}{\gamma_k}\nabla f(z^k)),\\
x^k&=(1-\gamma_k)x^{k-1}+\gamma_k y^k.
\end{aligned}$$

7.3 Numerical test

7.3.1 Code

test/test.cpp

```
int m = 256;
int n = 512;
int l = 1;

// set rng seed
rng(233);

Mat A = randn(m, n);
SpMat u = sprandn_c(n, l, 0.1);
Mat b = A * u;
Scalar mu = 0.001;
Mat x0 = randn(n, l);

Composite::Model::LASSO model(A, b, mu, Model::LASSOType
    ::Standard);
//Composite::Solver::ProxGBB<> alg;
//Composite::Solver::FISTA<> alg;
//Composite::Solver::Nesterov_2nd<> alg;
```

Power Method in Evaluating All the Maximum Modulus Eigenvalues of Square Matrix is include in "OptSuite/LinAlg/power_method.h"

If you use Nesterov_2nd, you need to calculate const step tau = 1/L(if possible)

7.3.2 Result

method	iter	message return	elapsed time	relative error
ProxGBB(BB step with line search)	382	optimal objdiff/Xdiff	0.015747 sec	5.74019e-06
FISTA(BB step with line search)	456	optimal objdiff/Xdiff	0.039219 sec	5.73793e-06
Nesterov's 2nd(const step)	813	optimal objdiff/Xdiff	0.0493599 sec	5.75413e-06