1) Please use gradient descent method, FISTA and Newton's method to solve problem (2) separately. Write down their updating scheme and plot figures of function value at each iteration $f(x^k) - f^*$ to compere their performance. (Hint: $P(x)$ is L-smooth in this case, so you can use a constant step size $\frac{1}{L}$ for these algorithms);

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-b_i \cdot a_i^T x)\right) + \frac{\lambda}{2}\|x\|_2^2 \tag{2}$$

$$\nabla P(x) = \frac{1}{n}\sum_{i=1}^{n}\left(-b_i a_i + \frac{b_i a_i}{1 + e^{-b_i a_i^T x}}\right) + \lambda x$$

$$\nabla^2 P(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{b_i^2 a_i^T a_i}{2(1 + e^{-b_i a_i^T x})} - \frac{b_i^2 a_i^T a_i}{2(1 + e^{-b_i a_i^T x})^2}\right) + \lambda I$$

$$1 + e^{-b_i a_i^T x} \in (1, +\infty)$$

$$b_i^2 = 1$$

$$\nabla^2 P(x) = \lambda I + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{b_i^2 a_i^T a_i}{2(1 + e^{-b_i a_i^T x})} - \frac{b_i^2 a_i^T a_i}{2(1 + e^{-b_i a_i^T x})^2}\right) \leq \lambda I + \frac{1}{8n}\sum_{i=1}^{n} a_i^T a_i$$

$$L = \lambda + \frac{1}{8n}\sum_{i=1}^{n} eigvals(a_i^T a_i).max$$

$$\lambda = 0.1$$

Algorithm 1 Gradient descent method

initial point x_0
$L = \lambda + \frac{1}{8n}\sum_{i=1}^{n} maxeig(a_i^T a_i)$
for $k = 0, 1, 2, \ldots,$ **do**
    $\nabla P(x_k) = \frac{1}{n}\sum_{i=1}^{n}\left(-b_i a_i + \frac{b_i a_i}{1 + e^{-b_i a_i^T x_k}}\right) + \lambda x_k$
    update $x_{k+1} = x_k - \nabla P(x_k)/L$
    if $\|x^{k+1} - x^k\|_2 \leq eps$ **or** $k \geq K$ **then**
        stop
    endif
endfor

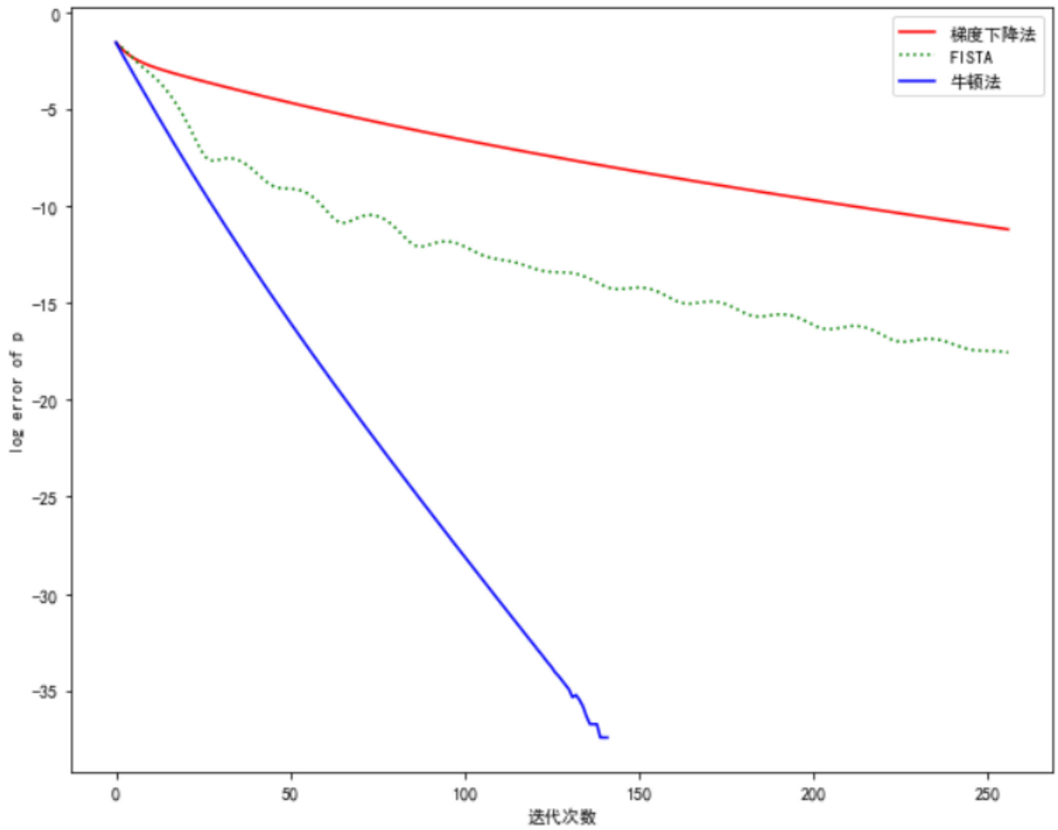
Algorithm 2 Fista

1. initial point x_0, x_{-1}
2. $L = \lambda + \frac{1}{8n} \sum_{i=1}^{n} maxeig(a_i^T a_i)$
3. **for** $k = 0, 1, 2, \ldots,$ **do**
4. $\nabla P(x_k) = \frac{1}{n} \sum_{i=1}^{n} \left(-b_i a_i + \frac{b_i a_i}{1+e^{-b_i a_i^T x_k}} \right) + \lambda x_k$
5. $y = x_k + \frac{k-2}{k+1}(x_k - x_{k-1})$
6. update $x_{k+1} = y - \nabla P(x_k)/L$
7. **if** $\left\| x^{k+1} - x^k \right\|_2 \leq eps$ **or** $k \geq K$ **then**
8. stop
9. **endif**
10. **endfor**

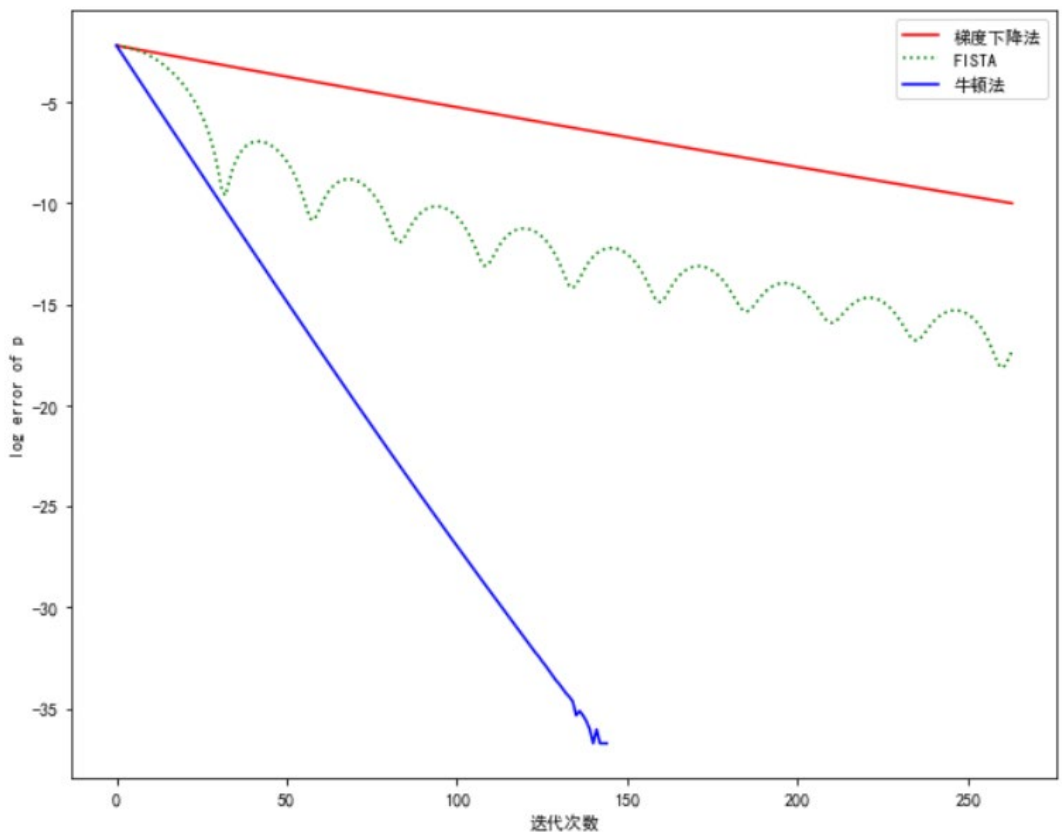
Algorithm 3 Newton's method

1. initial point x_0, x_{-1}
2. $L = \lambda + \frac{1}{8n} \sum_{i=1}^{n} maxeig(a_i^T a_i)$
3. **for** $k = 0, 1, 2, \ldots,$ **do**
4. $\nabla P(x_k) = \frac{1}{n} \sum_{i=1}^{n} \left(-b_i a_i + \frac{b_i a_i}{1+e^{-b_i a_i^T x_k}} \right) + \lambda x_k$
4. $\nabla^2 P(x) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{b_i^2 a_i^T a_i}{2\left(1+e^{-b_i a_i^T x}\right)} - \frac{b_i^2 a_i^T a_i}{2\left(1+e^{-b_i a_i^T x}\right)^2} \right) + \lambda I$
6. $d = -\nabla^2 P(x)^{-1} \nabla P(x_k)$
7. update $x_{k+1} = x_k + d/L$
8. **if** $\left\| x^{k+1} - x^k \right\|_2 \leq eps$ **or** $k \geq K$ **then**
9. stop
10. **endif**
11. **endfor**

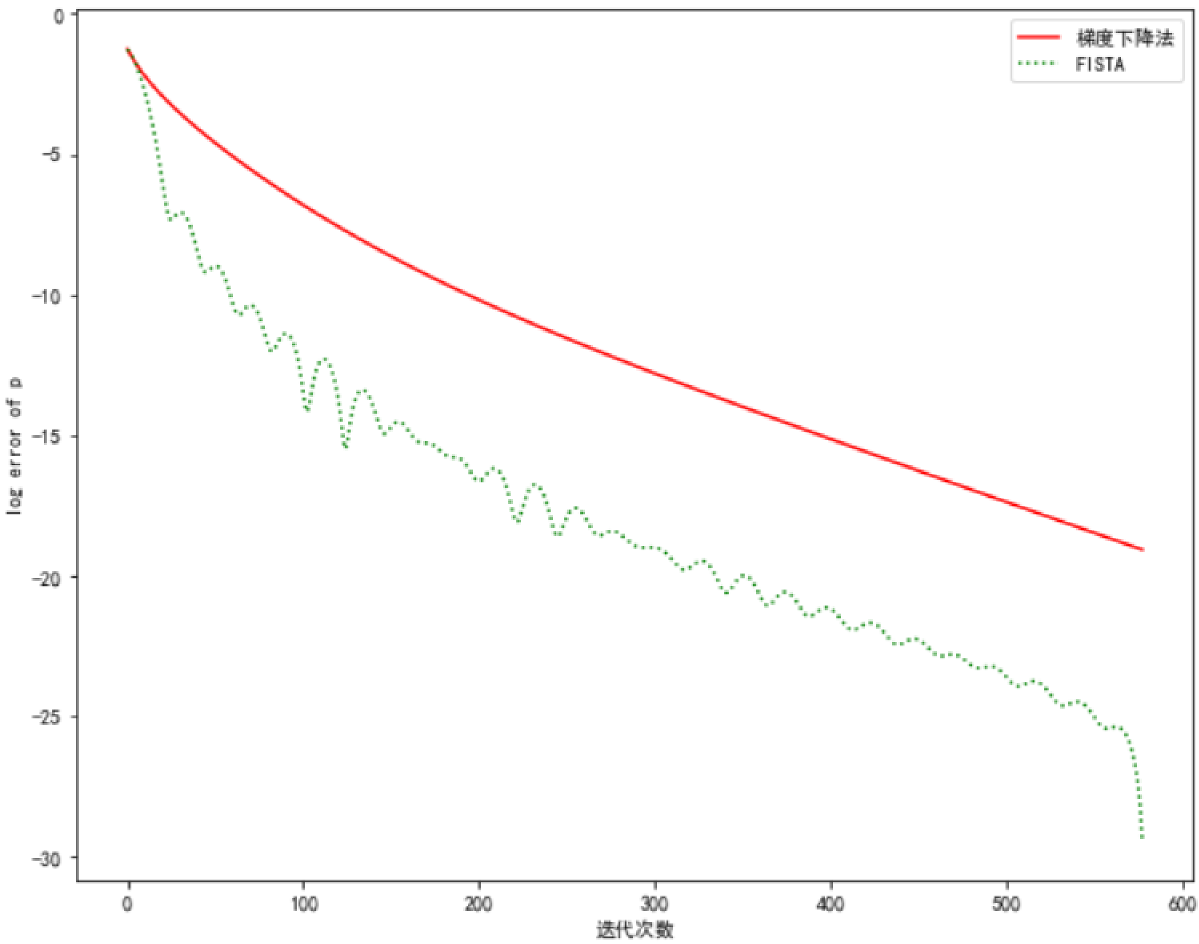
Dataset a2a:



Dataset ijcnn1:



Dataset w8a(Newton method:TLE):



2) Please use subgradient method (with dinimishing step size) and ADMM to solve problem (3). Write down their updating scheme and plot figures of the error of sequence $\|x^k - x^*\|_2$ to compere their performance; (Hint: A possible reformulation of problem (3) is

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{\lambda}{2}\|x\|_2^2 \\ \text{s.t.} \quad & y_i \geq 1 - b_i \cdot a_i^T x,\ y_i \geq 0,\ \forall i = 1, \dots, n. \end{aligned}$$

One can further introduce $s_i \geq 0$ to change the inequality $y_i \geq 1 - b_i \cdot a_i^T x$ to be an equality $y_i = 1 - b_i \cdot a_i^T x + s_i$.)

- ℓ_2-regularized support vector machine (SVM) with hinge loss:

$$P(x) = \frac{1}{n}\sum_{i=1}^{n}[1 - b_i \cdot a_i^T x]_+ + \frac{\lambda}{2}\|x\|_2^2$$

where the operator $[y]_+ = \max\{0, y\}$;

$$\partial P(x) = \begin{cases} \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(-b_i a_i) + \lambda x & if \;\; b_i a_i^T x < 1 \\ \lambda x & if \;\; b_i a_i^T x \geq 1 \end{cases}$$

Algorithm 1 Subgradient method

1. initial point x_0
2. **for** $k = 0, 1, 2, \dots,$ **do**
3. $\quad \partial P(x_k) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(-b_i a_i) + \lambda x_k & \text{if } b_i a_i^T x_k < 1 \\ \lambda x_k & \text{if } b_i a_i^T x_k \geq 1 \end{cases}$
4. $\quad$ update $x_{k+1} = x_k - \partial P(x_k)/(k+1)$
5. $\quad$ **if** $\|x^{k+1} - x^k\|_2 \leq eps$ **or** $k \geq K$ **then**
6. $\quad\quad$ stop
7. $\quad$ **endif**
8. **endfor**

ADMM

原问题

$$\min \qquad P(x) = \frac{1}{n}\sum_{i=1}^{n} [1 - b_i \cdot a_i^T x]_+ + \frac{\lambda}{2} \| x \|_2^2$$

Reformulation：

$$min \quad \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{\lambda}{2} \| x \|_2^2$$

$$\text{s.t.} \quad y_i = 1 - b_i \cdot a_i^T x + s_i, s_i \geq 0, \forall i = 1, \dots, n.$$

Lagrangian function:

$$\mathcal{L}_t(x, Y, S, \Lambda) = \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{\lambda}{2} \| x \|_2^2 + I_{s_i \geq 0}(S) - < \Lambda, Y - \mathbf{1} + bAx - S > + \frac{t}{2}$$

$$\| Y - \mathbf{1} + bAx - S \|_2^2$$

Updating scheme:

$$x^{k+1} = \mathrm{argmin}_x \frac{\lambda}{2} \| x \|_2^2 + \frac{t}{2} \| Y^k - \mathbf{1} + bAx - S^k - \frac{\Lambda^k}{t} \|_2^2$$

$$Y^{k+1} = \mathrm{argmin}_Y \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{t}{2} \| Y - \mathbf{1} + bAx^{k+1} - S^k - \frac{\Lambda^k}{t} \|_2^2$$

$$S^{k+1} = \mathrm{argmin}_S I_{s_i \geq 0}(S) + \frac{t}{2} \| Y^{k+1} - \mathbf{1} + bAx^{k+1} - S - \frac{\Lambda^k}{t} \|_2^2$$

$$\Lambda^{k+1} = \Lambda^k - t(Y^{k+1} - \mathbf{1} + bAx^{k+1} - S)$$

计算得到

$$x^{k+1} = t(\lambda I + t(bA)^T(bA))^{-1}(bA)^T\left(\mathbf{1} + S^k + \frac{\Lambda^k}{t} - Y^k\right)$$

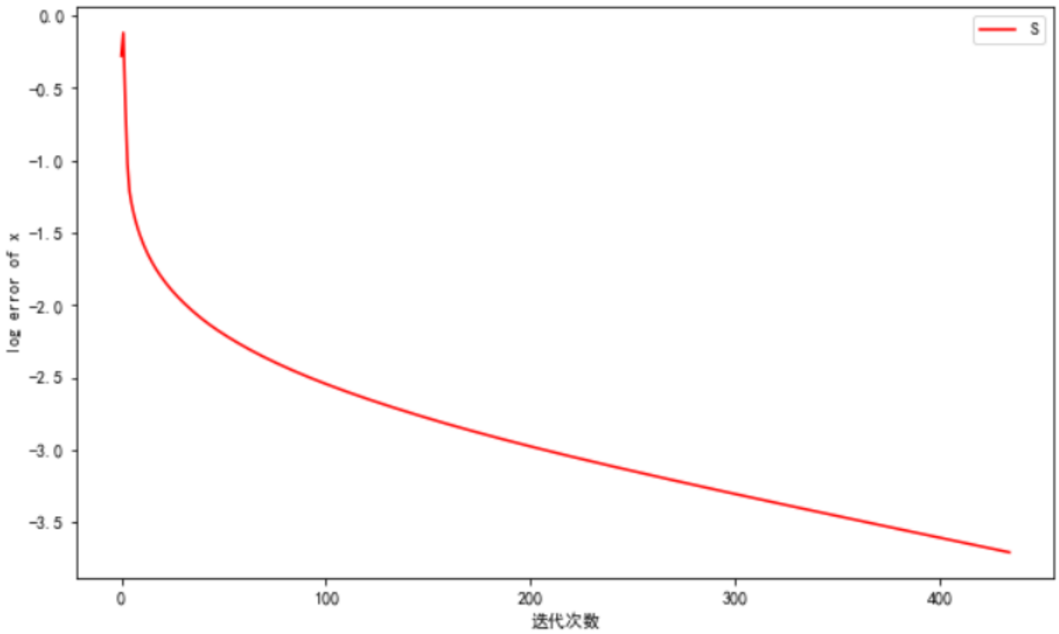
$$Y^{k+1} = \mathbf{1} - bAx^{k+1} + S^k + \frac{\Lambda^k}{t} - \frac{t}{n} * \mathbf{1}$$

$$s_i^{k+1} = \begin{cases} y_i^{k+1} - 1 + bAx_i^{k+1} - \frac{\lambda_i^k}{t} & if \geq 0 \\ 0 \qquad if < 0 \end{cases}$$

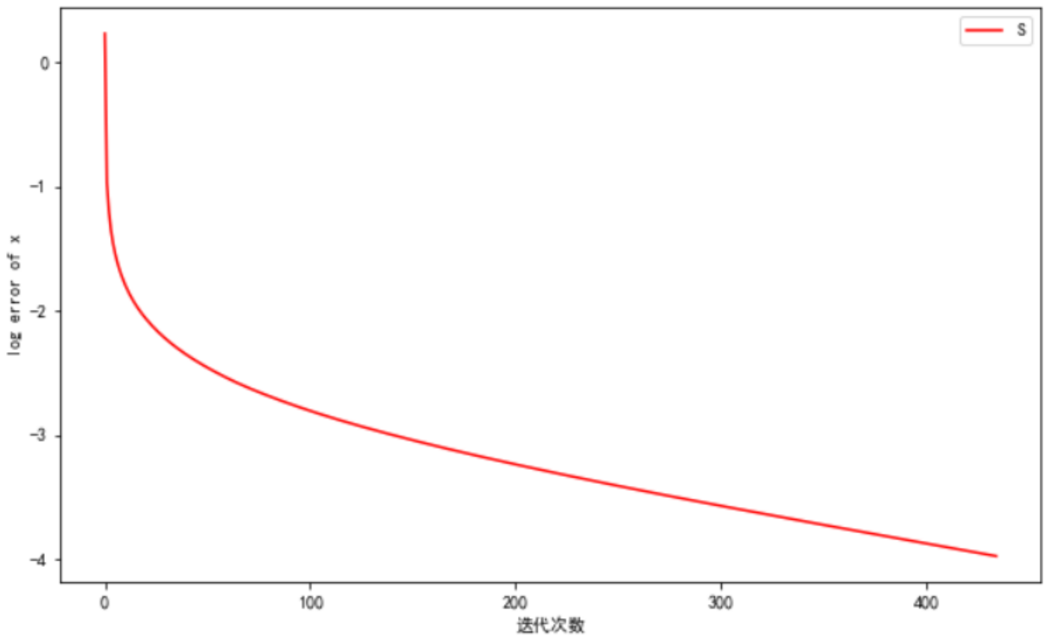
$$\Lambda^{k+1} = \Lambda^k - t(Y^{k+1} - \mathbf{1} + bAx^{k+1} - S^{k+1})$$

由于神秘原因，ADMM 无法收敛，故无数据。

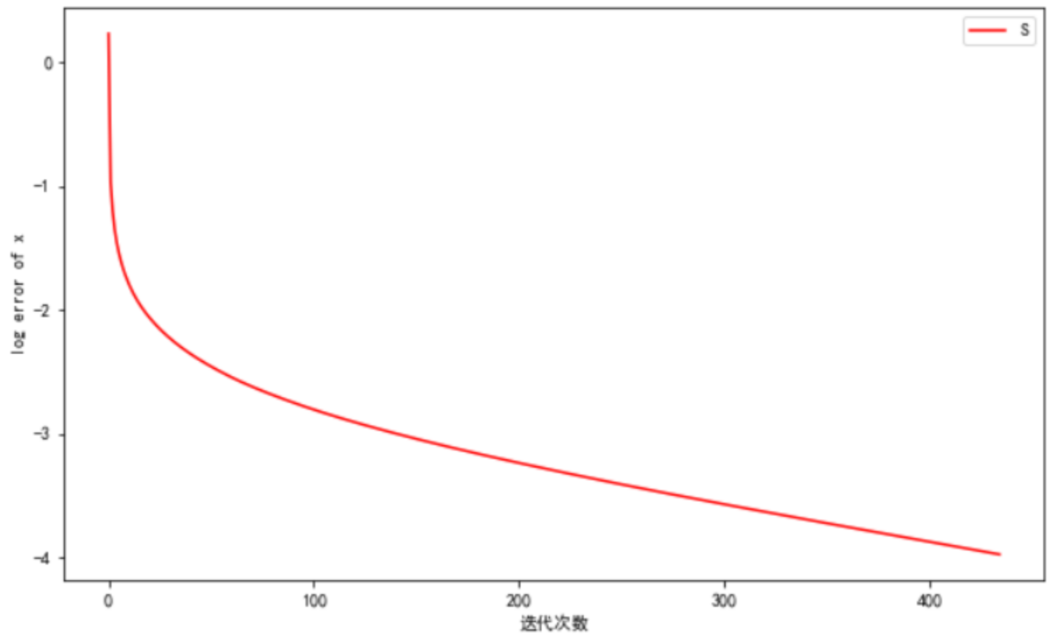
Dataset a2a:



Dataset w8a:



Dataset ijcnn1:



3) Please use proximal gradient method, FISTA and FISTA with restarting strategy to solve problem (4). Write down their updating scheme and plot figures of function value at each iteration $f(x^k) - f^*$ to compere their performance. (Hint: A backtracking line-search process may be needed to determine the step size).

- ℓ_1-regularized support vector machine (SVM) with squared hinge loss:

$$P(x) = \frac{1}{n}\sum_{i=1}^{n}\left([1 - b_i \cdot a_i^T x]_+\right)^2 + \frac{\lambda}{2}\|x\|_1. \tag{4}$$

$$g(x) = \frac{1}{n}\sum_{i=1}^{n}\left([1 - b_i \cdot a_i^T x]_+\right)^2$$

$$h(x) = \frac{\lambda}{2}\parallel x \parallel_1$$

$$\nabla g(x) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(-2b_i a_i(1 - b_i \cdot a_i^T x)) & if\ \ b_i a_i^T x < 1 \\ 0 & if\ \ b_i a_i^T x \geq 1 \end{cases}$$

$$prox_{\frac{\lambda}{2}\|x\|_1}(u)_i = \begin{cases} u_i - \frac{\lambda}{2} & u_i > \frac{\lambda}{2} \\ 0 & -\frac{\lambda}{2} \leq u_i \leq \frac{\lambda}{2} \\ u_i + \frac{\lambda}{2} & u_i < -\frac{\lambda}{2} \end{cases}$$

Algorithm 1 proximal gradient method

initial point x_0

for $k = 0, 1, 2, \ldots,$ **do**

$t = 1$

$$\partial g(x) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\left(-2b_i a_i\left(1 - b_i \cdot a_i^T x\right)\right) & \text{if } b_i a_i^T x < 1 \\ 0 & \text{if } b_i a_i^T x \geq 1 \end{cases}$$

update $x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(x_k))$

while $g(x_{k+1}) > g(x_k) + \nabla g(x)^T(x_{k+1} - x_k) + \frac{1}{2t}\|x_{k+1} - x_k\|_2^2$

$t = \beta t$

$x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(x_k))$

if $\left\|x^{k+1} - x^k\right\|_2 \leq eps$ **or** $k \geq K$ **then**

stop

endif

endfor

Algorithm 2 FISTA

initial point x_0

for $k = 0, 1, 2, \ldots,$ **do**

$t = 1$

$$\partial g(x) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\left(-2b_i a_i\left(1 - b_i \cdot a_i^T x\right)\right) & \text{if } b_i a_i^T x < 1 \\ 0 & \text{if } b_i a_i^T x \geq 1 \end{cases}$$

$y = x_k + \frac{k-2}{k+1}(x_k - x_{k-1})$

update $x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(y))$

while $g(x_{k+1}) > g(x_k) + \nabla g(x)^T(x_{k+1} - x_k) + \frac{1}{2t}\|x\|_2^2$

$t = \beta t$

$x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(y))$

if $\left\|x^{k+1} - x^k\right\|_2 \leq eps$ **or** $k \geq K$ **then**

stop

endif

endfor

Algorithm 3 FISTA with restarting strategy

initial point x_0

for $k = 0, 1, 2, \ldots,$ **do**

$\quad t = 1$

$\quad \partial g(x) = \begin{cases} \frac{1}{n} \sum_{i=1}^{n} \left(-2b_i a_i \left(1 - b_i \cdot a_i^T x\right)\right) & \text{if } b_i a_i^T x < 1 \\ 0 & \text{if } b_i a_i^T x \geq 1 \end{cases}$

$\quad y = x_k + \frac{kt-2}{kt+1}(x_k - x_{k-1})$

$\quad$ update $x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(y))$

$\quad$ **if** $kt > 50$

$\quad\quad kt = 0$

$\quad$ **else** $k = kt + 1$

$\quad$ **while** $g(x_{k+1}) > g(x_k) + \nabla g(x)^T(x_{k+1} - x_k) + \frac{1}{2t}\|x\|_2^2$

$\quad\quad t = \beta t$

$\quad\quad x_{k+1} = \text{prox}_{\frac{\lambda}{2}\|x\|_1}(x_k - t\partial P(y))$

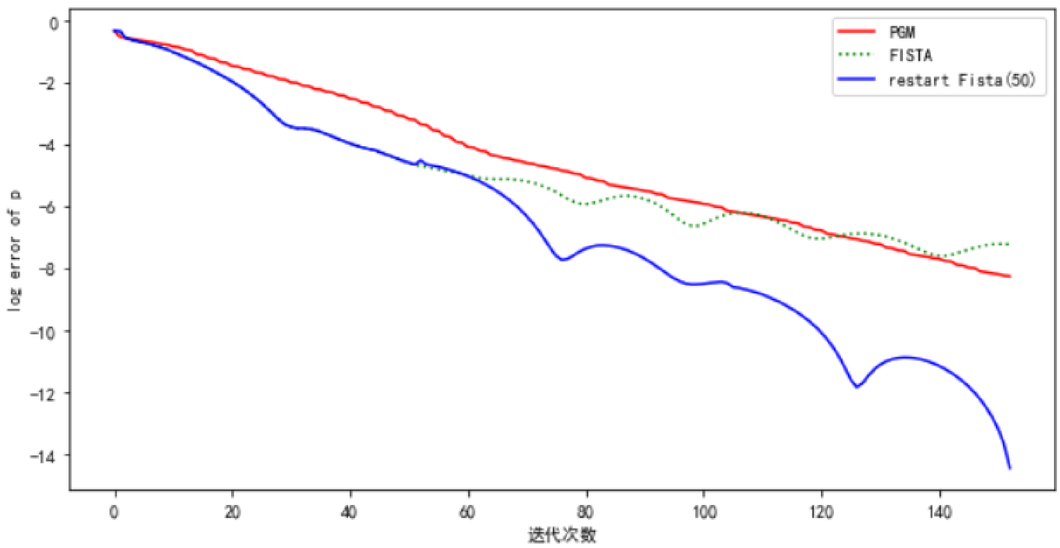
$\quad$ **if** $\left\|x^{k+1} - x^k\right\|_2 \leq eps$ **or** $k \geq K$ **then**

$\quad\quad$ stop

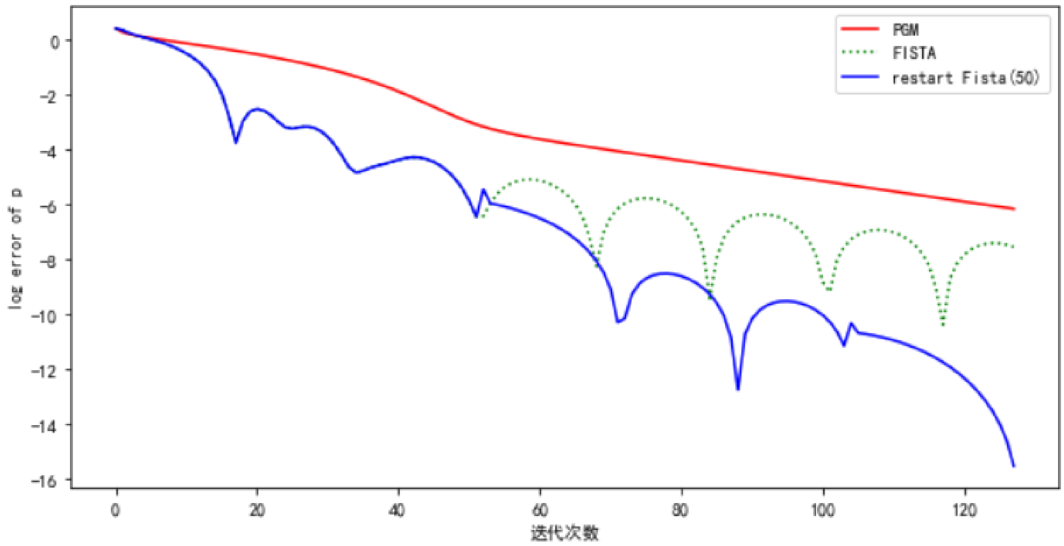
$\quad$ **endif**

endfor

Dataset a2a:



Dataset ijcnn1:



Dataset w8a：