First-order methods in optimization - Evaluation

Olivier Leblanc, Guillaume Thiran.

1 Part 1 - Slide 45

1.1 6. (with code)

$$f(\boldsymbol{x}) = 2x_{[1]} + x_{[2]} = \max_{\boldsymbol{y}} \left\{ \sum_i y_i x_i; \sum_i y_i = 3, 0 \le y_i \le 2 \right\} = \sigma_{\{\boldsymbol{y}|\mathbf{1}^\top \boldsymbol{y}=3, \mathbf{0}\le \boldsymbol{y} \le 2\mathbf{1}\}}.$$

Hence,

$$\text{prox}_f(\boldsymbol{x}) = \boldsymbol{x} - \mathcal{P}_{\{\boldsymbol{y}|\mathbf{1}^\top \boldsymbol{y}=3, \mathbf{0}\le \boldsymbol{y}\le 2\mathbf{1}\}}(\boldsymbol{x}).$$

Writing $C = \{\boldsymbol{y}|\mathbf{1}^\top \boldsymbol{y} = 3, \mathbf{0} \le \boldsymbol{y} \le 2\mathbf{1}\}$, it can be compared with $H_{\boldsymbol{a},b} \bigcap \text{Box}[\boldsymbol{l}, \boldsymbol{u}]$, with $\boldsymbol{a} = \mathbf{1}, b = 3, \boldsymbol{l} = \mathbf{0}, \boldsymbol{u} = \mathbf{2}$.

```
import numpy as np
def error_fct(a,b,l,u,x,mu):
    y = projbox(x-mu*a,l,u)
    return error = a@y-b

def projbox(x, l, u):
    return np.minimum(np.maximum(x,l), u)

def proj_H_inter_box(a,b,l,u,x):
    mu_low = -1 #start with guesses for mu-levels
    mu_high = 1
    #check that the levels give respectively negative and positive values
    #for the function error = a@y-1 with y = proj_box(l,u,x-mu*a)
    #positive for lower bound
    j=0
    j_max =100
    error_l = -1
    while (error_l<0) & (j<j_max) :
        mu_low = mu_low*2 #more negative
        error_l=error_fct(a,b,l,u,x,mu_low)
        j=j+1
    #negative for upper bound
    k=0
    k_max =10
    error_h = 1
    while (error_h>0) & (k<k_max) :
        mu_high = mu_high*2 #more positive
        error_h=error_fct(a,b,l,u,x,mu_high)
        k=k+1

    i = 0
    i_max = 100
    tol = 1e-8
    error = 2*tol
    while (np.abs(error)>tol) & (i<i_max) :
        mu_mid = (mu_low+mu_high)/2
        error = error_fct(a,b,l,u,x,mu_mid)
        if error>0:
            mu_low = mu_mid
        else:
            mu_high = mu_mid
        i=i+1

    #Compute the solution with the good level
    return projbox(x-mu_mid*a,l,u)

def proxf(x):
    return x - proj_H_inter_box(np.ones(len(x)),3,np.zeros(len(x)),2*np.ones(len(x)),x)

x = np.array([2,1,4,1,2,1])
```

```
print(proxf(x))
```

The output is : $[1.5, 1., 2., 1., 1.5, 1.]$.

1.2 8.

$$f(t) = \begin{cases} 1/t, & t > 0, \\ \infty, & \text{else.} \end{cases}$$

$$\operatorname{prox}_{\lambda f}(t) = \arg\min_u \begin{cases} \lambda/u, & u > 0, \\ \infty, & \text{else.} \end{cases} + \tfrac{1}{2} \|u - t\|_2^2$$

Clearly, the minimum occurs when $u > 0$, i.e. on the differentiable part. Hence,

$$\frac{-\lambda}{u^2} + u - t = 0 \Leftrightarrow u^3 - tu^2 - \lambda = 0,$$

and it can be checked the second derivative is always positive on $u > 0$, implying it exists a unique solution of the above and it corresponds to a minimum. Finally,

$$\operatorname{prox}_{\lambda f}(t) = \{u > 0 | u^3 - tu^2 - \lambda = 0\}. \tag{1}$$

1.3 9.

$$f(\boldsymbol{X}) = \begin{cases} \operatorname{tr} \boldsymbol{X}^{-1}, & \boldsymbol{X} \succ 0, \\ \infty, & \text{else.} \end{cases} = \begin{cases} \sum_{i=1}^n \frac{1}{\lambda_i}, & \boldsymbol{X} \succ 0, \\ \infty, & \text{else.} \end{cases}$$

As one can write $f(\boldsymbol{X}) = g(\lambda_1(\boldsymbol{X}), \cdots, \lambda_n(\boldsymbol{X})) = \sum_{i=1}^n h(\lambda_i)$, f is a *symmetric spectral function*. With the EigenValue Decomposition (EVD) of $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U} \operatorname{diag} \lambda(\boldsymbol{X}) \boldsymbol{U}^T$, this gives

$$\begin{aligned} \operatorname{prox}_{\lambda f}(\boldsymbol{X}) &= \boldsymbol{U} \operatorname{diag}(\operatorname{prox}_{\lambda g}[\lambda_1, \cdots, \lambda_n]) \boldsymbol{U}^\top \\ &= \boldsymbol{U} \operatorname{diag}(\operatorname{prox}_{\lambda h}(\lambda_1), \cdots, \operatorname{prox}_{\lambda h}(\lambda_n)) \boldsymbol{U}^\top \\ &= \boldsymbol{U} \operatorname{diag}(\{\{u > 0 | u^3 - \lambda_i u^2 - \lambda = 0\}\}_{i=1}^n) \boldsymbol{U}^\top, \end{aligned}$$

with $h(t) = \frac{1}{t}$ for $t > 0$.

1.4 10.

$$\lambda f(\boldsymbol{x}) = \lambda(\|\boldsymbol{x}\|_2 - 1)^2 = \lambda \|\boldsymbol{x}\|_2^2 - 2\lambda \|\boldsymbol{x}\|_2 + \lambda.$$

Using the provided tables, one identifies it with $g(\boldsymbol{x}) + \frac{c}{2} \|\boldsymbol{x}\|_2^2 + \langle \boldsymbol{a}, \boldsymbol{x} \rangle + \gamma$, with $g(\boldsymbol{x}) = -2\lambda \|\boldsymbol{x}\|_2, c = 2\lambda, \boldsymbol{a} = \boldsymbol{0}, \gamma = 0$. Hence,

$$\operatorname{prox}_{\lambda f}(\boldsymbol{x}) = \operatorname{prox}_{\frac{-2\lambda\|\cdot\|_2}{1+2\lambda}} \left(\frac{\boldsymbol{x}}{1 + 2\lambda} \right) = \begin{cases} \left(1 + \frac{2\lambda}{\|\boldsymbol{x}\|_2}\right) \frac{\boldsymbol{x}}{1+2\lambda}, & \boldsymbol{x} \neq \boldsymbol{0}, \\ \{\boldsymbol{u} : \|\boldsymbol{u}\| = \frac{2\lambda}{1+2\lambda}\}, & \boldsymbol{x} = \boldsymbol{0}. \end{cases}$$

2 Part 2 - Exercise 0 - slide 40

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \tfrac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \underbrace{\frac{\lambda_1}{2} \|\boldsymbol{x}\|_2^2 + \lambda_2 \|\boldsymbol{x}\|_1}_{\text{elastic net}},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\lambda_1, \lambda_2 > 0$. Choosing $f(\boldsymbol{x}) = \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\lambda_1}{2} \|\boldsymbol{x}\|_2^2$ as the σ-strongly convex and differentiable part and $g(\boldsymbol{x}) = \lambda_2 \|\boldsymbol{x}\|_1$ as the closed convex but non differentiable part, one has $\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \lambda_1 \boldsymbol{x}$ and $\operatorname{prox}_{\alpha g}(\boldsymbol{x}) = \mathcal{T}_{\alpha\lambda_2}(\boldsymbol{x})$.

- (*Proximal Gradient*)

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \operatorname{prox}_{\frac{1}{L} g}(\boldsymbol{x}^k - \tfrac{1}{L} \nabla f(\boldsymbol{x}^k)) \\ &= \mathcal{T}_{\frac{\lambda_2}{L}} \left(\boldsymbol{x}^k - \tfrac{1}{L}(\boldsymbol{A}^\top \boldsymbol{A} + \lambda_1 \boldsymbol{I}) \boldsymbol{x}^k + \tfrac{1}{L} \boldsymbol{A}^\top \boldsymbol{b} \right), \end{aligned}$$

  with $L = \left\| \boldsymbol{A}^\top \boldsymbol{A} + \lambda_1 \boldsymbol{I} \right\| = \lambda_{\max}(\boldsymbol{A}^\top \boldsymbol{A}) + \lambda_1 = \|\boldsymbol{A}\|_2^2 + \lambda_1$.

- (*FISTA*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2}{L}}\left(\boldsymbol{y}^k - \frac{1}{L}(\boldsymbol{A}^\top\boldsymbol{A} + \lambda_1\boldsymbol{I})\boldsymbol{y}^k + \frac{1}{L}\boldsymbol{A}^\top\boldsymbol{b}\right) \\ t_{k+1} & = \frac{1+\sqrt{1+4t_k^2}}{2} \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{t_k-1}{t_{k+1}}\right)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \end{cases}$$

- (*V-FISTA*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2}{L}}\left(\boldsymbol{y}^k - \frac{1}{L}(\boldsymbol{A}^\top\boldsymbol{A} + \lambda_1\boldsymbol{I})\boldsymbol{y}^k + \frac{1}{L}\boldsymbol{A}^\top\boldsymbol{b}\right) \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases}$$

  with $\kappa = L/\sigma$, and $\sigma = \lambda_{\min}(\boldsymbol{A}^\top\boldsymbol{A}) + \lambda_1 = \lambda_1$ if $\boldsymbol{A}$ is not full rank.

Now, if one chooses $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ as the differentiable (but not strongly convex) part and $\alpha g(\boldsymbol{x}) = \frac{\alpha\lambda_1}{2}\|\boldsymbol{x}\|_2^2 + \alpha\lambda_2\|\boldsymbol{x}\|_1$ as the closed convex but non differentiable part, one has $\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$ and, by identification with $g'(\boldsymbol{x}) + \frac{c}{2}\|\boldsymbol{x}\|_2^2 + \langle \boldsymbol{a}, \boldsymbol{x}\rangle + \gamma$ with $g'(\boldsymbol{x}) = \alpha\lambda_2\|\boldsymbol{x}\|_1$, $c = \alpha\lambda_1, \boldsymbol{a} = \boldsymbol{0}, \gamma = 0$, $\mathrm{prox}_{\alpha g}(\boldsymbol{x}) = \mathrm{prox}_{\frac{\alpha}{\alpha\lambda_1+1}\lambda_2\|\cdot\|_1}(\frac{\boldsymbol{x}}{\alpha\lambda_1+1}) = \mathcal{T}_{\frac{\alpha\lambda_2}{\alpha\lambda_1+1}}(\frac{\boldsymbol{x}}{\alpha\lambda_1+1})$.

- (*V-FISTA2*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2/L_2}{\lambda_1/L_2+1}}\left(\frac{\boldsymbol{y}^k - \frac{1}{L_2}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b})}{\frac{\lambda_1}{L_2}+1}\right) = \mathcal{T}_{\frac{\lambda_2}{\lambda_1+L_2}}\left(\frac{L_2\boldsymbol{y}^k - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b})}{\lambda_1+L_2}\right) \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases}$$

  with $L_2 = \|\boldsymbol{A}\|_2^2$, $\kappa_2 = L_2/\sigma$, and $\sigma = \lambda_1$ (given by the statement).

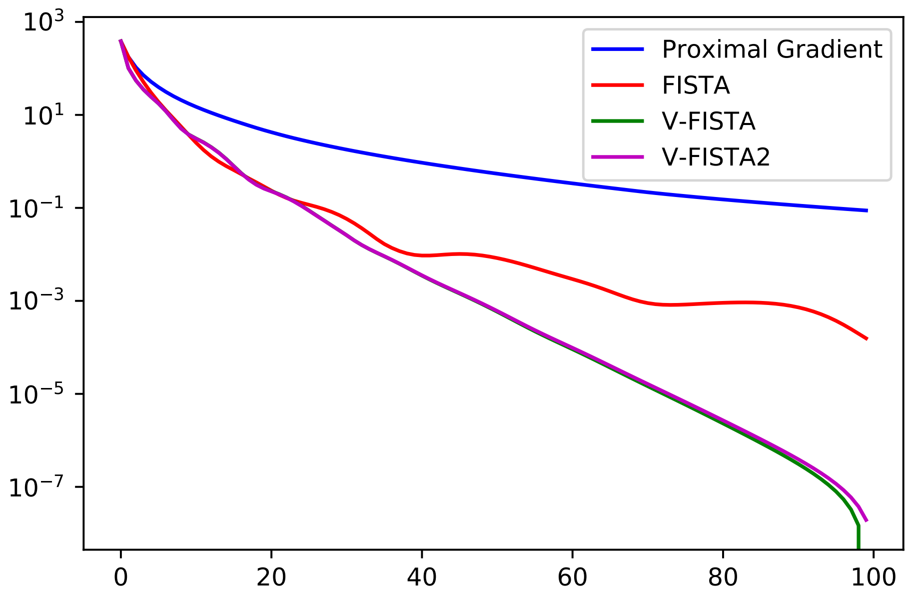

Figure 1: $F(\boldsymbol{x}^k) - F_{\text{opt}}$ in log-scale along the y-axis for the first 100 iterations of each of the methods with all-zeros vectors and a constant stepsize.

As can be observed in Fig. 2, the V-FISTA methods worked the best, with a slight improvement in V-FISTA which does not follow the theory. The four first elements are:

- (-0.40403765 0.18475212 0.97264407 -0.99645397])
- (-0.43969331 0.01974521 1.42280231 -0.87819581)

- (-0.4319773 0.02881602 1.43373682 -0.9066518)
- (-0.43210834 0.0295975 1.43437285 -0.90583213)
- (-0.43210892 0.02959727 1.43437048 -0.9058291)

for the Ground truth, and the four methods, PG, FISTA, V-FISTA, V-FISTA2, respectively.

3 Part 2 - Exercise 1 - slide 41

$$\min_{\boldsymbol{x}\in\mathbb{R}^{30}} \sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c} + 0.2 \left\|\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}\right\|_1,$$

where $\boldsymbol{Q} \in \mathbb{R}^{30\times 30}$, $\boldsymbol{b} \in \mathbb{R}^{30}$, $c \in \mathbb{R}$, $\boldsymbol{D} \in \mathbb{R}^{30\times 30}$. The matrix $\boldsymbol{Q}$ is positive definite.

(a) The first step of the problem is to show it is well-defined (i.e. $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c \geq 0$ if $c > \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}$.). To that aim, letting the Cholesky factorisation of $\boldsymbol{Q}$ being denoted as $\boldsymbol{Q} = \boldsymbol{L}^\top \boldsymbol{L}$,

$$\begin{aligned}
\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c &= \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{L}^\top \boldsymbol{L}^{-\top}\boldsymbol{b} + \boldsymbol{b}^\top \boldsymbol{L}^{-1}\boldsymbol{L}^{-\top}\boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{L}^\top \boldsymbol{L}^{-\top}\boldsymbol{b} + c,\\
&= \left\|\boldsymbol{L}\boldsymbol{x} + \boldsymbol{L}^{-\top}\boldsymbol{b}\right\|_2^2 + c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}.
\end{aligned}$$

From the above, as a norm is always nonnegative, one can conclude the problem is well-defined if $c > \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}$.

(b) Starting from the norm expression, one can obtain

$$\sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c} + 0.2 \left\|\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}\right\|_1 = \left\| \begin{matrix} \boldsymbol{L}\boldsymbol{x} + \boldsymbol{L}^{-\top}\boldsymbol{b} \\ \sqrt{c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}} \end{matrix} \right\|_2 + 0.2 \left\|\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}\right\|_1.$$

As a norm is convex, as a composition with a linear mapping preserves convexity and since a sum of convex functions is convex, the problem is convex.

(c) To fit the framework of FISTA, we denote by $f(\boldsymbol{x}) = \sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c}$ and $g(\boldsymbol{x}) = 0.2 \left\|\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}\right\|_1$ where g is proper, closed and convex, and f is L_f-smooth and convex. More precisely, the Lipschitz constant of f can be identify by computing its Hessian:

$$\begin{aligned}
\boldsymbol{\nabla} f(\boldsymbol{x}) &= \frac{\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}}{\sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c}},\\
\boldsymbol{\nabla}^2 f(\boldsymbol{x}) &= \frac{\boldsymbol{Q}}{\sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c}} - \frac{(\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b})(\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b})^\top}{\left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c\right)^{\frac{3}{2}}}.
\end{aligned}$$

The latter can be bounded as

$$\boldsymbol{\nabla}^2 f(\boldsymbol{x}) \preceq \frac{\boldsymbol{Q}}{\sqrt{\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c}} = \frac{\boldsymbol{Q}}{\sqrt{c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}}\sqrt{1 + \frac{\left\|\boldsymbol{L}\boldsymbol{x} + \boldsymbol{L}^{-\top}\boldsymbol{b}\right\|_2^2}{c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}}}} \preceq \frac{\boldsymbol{Q}}{\sqrt{c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}}},$$

leading to $L_f = \frac{\lambda_{\max}(\boldsymbol{Q})}{\sqrt{c - \boldsymbol{b}^\top \boldsymbol{Q}^{-1}\boldsymbol{b}}}$. In the case of the exercise, we obtain $L_f = 53.54$ and thus a step size of 0.019.

As $\boldsymbol{D}\boldsymbol{D}^\top = \boldsymbol{I}$, one can compute the proximal operator of αg as

$$\begin{aligned}
\operatorname{prox}_{\alpha g}(\boldsymbol{x}) &= \boldsymbol{x} + \boldsymbol{D}^\top \left(\mathcal{T}_{0.2\alpha}(\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}) - \boldsymbol{D}\boldsymbol{x} - \boldsymbol{1}\right),\\
&= \boldsymbol{D}^\top \mathcal{T}_{0.2\alpha}(\boldsymbol{D}\boldsymbol{x} + \boldsymbol{1}) - \boldsymbol{D}^\top \boldsymbol{1}.
\end{aligned}$$

This leads to

- (*Proximal gradient*):

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \operatorname{prox}_{\frac{1}{L_f}g}(\boldsymbol{x}^k - \tfrac{1}{L_f}\nabla f(\boldsymbol{x}^k))\\
&= \boldsymbol{D}^\top \mathcal{T}_{\frac{0.2}{L_f}}\left[\boldsymbol{D}(\boldsymbol{x}^k - \tfrac{1}{L_f}\nabla f(\boldsymbol{x}^k)) + \boldsymbol{1}\right] - \boldsymbol{D}^\top \boldsymbol{1}\\
&= \boldsymbol{D}^\top \mathcal{T}_{\frac{0.2}{L_f}}\left[\boldsymbol{D}(\boldsymbol{x}^k - \tfrac{1}{L_f}\frac{\boldsymbol{Q}\boldsymbol{x}^k + \boldsymbol{b}}{\sqrt{(\boldsymbol{x}^k)^\top \boldsymbol{Q}\boldsymbol{x}^k + 2\boldsymbol{b}^\top \boldsymbol{x}^k + c}}) + \boldsymbol{1}\right] - \boldsymbol{D}^\top \boldsymbol{1}.
\end{aligned}$$

- (*FISTA*):

$$\begin{cases} \boldsymbol{x}^{k+1} &= \boldsymbol{D}^\top \mathcal{T}_{\frac{0.2}{L_f}} \left[\boldsymbol{D}(\boldsymbol{y}^k - \frac{1}{L_f} \frac{\boldsymbol{Q}\boldsymbol{y}^k + \boldsymbol{b}}{\sqrt{(\boldsymbol{y}^k)^\top \boldsymbol{Q}\boldsymbol{y}^k + 2\boldsymbol{b}^\top \boldsymbol{y}^k + c}}) + \boldsymbol{1} \right] - \boldsymbol{D}^\top \boldsymbol{1} \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \boldsymbol{y}^{k+1} &= \boldsymbol{x}^{k+1} + \left(\frac{t_k - 1}{t_{k+1}} \right) (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \end{cases}$$

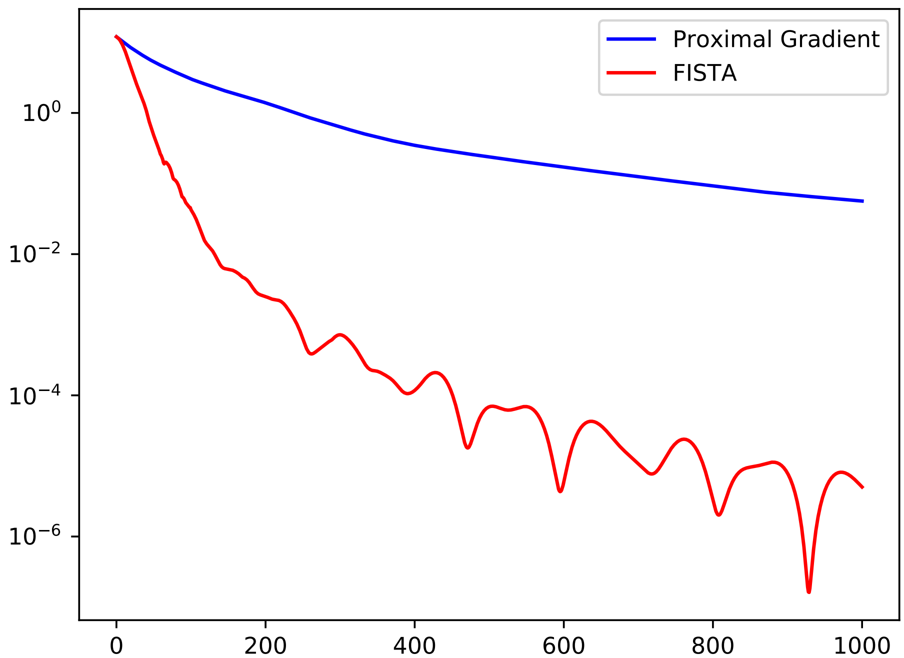

Figure 2: $F(\boldsymbol{x}^k) - F_{\text{opt}}$ in log-scale along the y-axis for the first 1001 iterations of each of the methods with all-zeros vectors and a constant stepsize. The optimal solution has been computed as the best one across 10000 iterations.

Vector found by PG:

$$\boldsymbol{x}^{*,\text{PG}} = [0.406, -0.093, -0.093, -0.874, -1.025, -0.044, 0.521, -1.16, 0.877, 0.129, -0.242, 1.664, 1.32, -0.561, -0.079, -1.764, \\ -1.351, -0.387, -1.158, 0.844, 0.43, -0.715, -0.349, -0.037, 1.408, -0.971, 1.206, 0.795, 0.568, 1.284].$$

Vector found by FISTA:

$$\boldsymbol{x}^{*,\text{FISTA}} = [-0.455, -0.389, 0.291, -1.149, -1.309, -0.43, 0.667, -1.434, 1.082, -0.019, -0.416, 1.732, 1.293, -0.541, -0.143, \\ -1.704, -1.396, -0.742, -1.539, 0.335, -0.218, -0.662, -0.151, -0.085, 1.801, -0.566, 1.321, 0.849, 0.701, 1.091].$$

4 Part 2 - Exercise 3 - slides 71-72

Given a set of data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ and corresponding labels $y_1, y_2, \cdots, y_n$. The soft margin SVM problem is given by

$$\min \left\{ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{n} \max \left\{ 0, 1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i \right\} \right\}$$

(a)

(b)

5 Part 3 - Exercise 2 - slides 35-36