[M2, MVA]
Convex Optimization, Algorithms and Applications

Maha ELBAYAD
maha.elbayad@student.ecp.fr

Homework 3

December 2, 2015

1 Second order methods for dual problem

1. The dual of Lasso

Let us consider the LASSO problem

$$\underset{w}{\text{minimize}} \frac{1}{2}\|Xw - y\|_2^2 + \lambda\|w\|_1 \tag{LASSO}$$

where $w \in \mathbb{R}^d, X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^n$ abd $\lambda > 0$ the regularization parameter.
We rewrite the problem as:

$$\underset{w,v}{\text{minimize}} \frac{1}{2}\|Xw - y\|_2^2 + \lambda\|v\|_1, \text{ subject to } w = v.$$

For which Slater's condition is satisfied, thus the strong duality holds.
In fact if we consider $u = [w, v]^T$, $\bar{X} = [X, \mathbf{0}_{n,d}]$, $A = [\mathbf{0}_{d,d}, I_d]$ and $E = [I_d, -I_d]$ the problem can be transformed into the convex problem with affine equality contraint:

$$\underset{u}{\text{minimize}} \frac{1}{2}\|\bar{X}u - y\|_2^2 + \lambda\|Au\|_1, \text{ subject to } Eu = \mathbf{0}.$$

We consider the Lagrange multiplier μ, the lagrangian is:

$$\begin{aligned} L(w, v, \mu) &= \frac{1}{2}\|Xw - y\|_2^2 + \lambda\|v\|_1 + \mu^T(w - v) \\ &= (\frac{1}{2}\|Xw - y\| + \mu^T w) + (\lambda\|v\|_1 - \mu^T v) \end{aligned}$$

We minimize L with respect to the two variables w, v:

$$\nabla_w(\frac{1}{2}\|Xw - y\| + \mu^T w) = \mu + X^T Xw - X^T y = \mathbf{0}$$

Thus,

$$\min_w(\frac{1}{2}\|Xw - y\| + \mu^T w) = -\frac{1}{2}y^T XH(X^T y - \mu) - \frac{1}{2}\mu^T H(X^T y - \mu) + \frac{1}{2}y^T y$$

where $H = (X^T X)^{-1} \in \mathbb{R}^d$, and:

$$\min_v(\lambda\|v\|_1 - \mu^T v) = \begin{cases} 0\, if\ \|\mu\|_\infty \leq \lambda \\ -\infty\, otherwise \end{cases}$$

The dual problem would be:

$$\begin{aligned} \underset{\mu}{\text{minimize}} \quad & \frac{1}{2}\mu^T H\mu - (HX^T y)^T \mu \\ \text{subject to} \quad & \lambda \leq \mu_i \leq \lambda,\ i = 1...m \end{aligned} \tag{1}$$

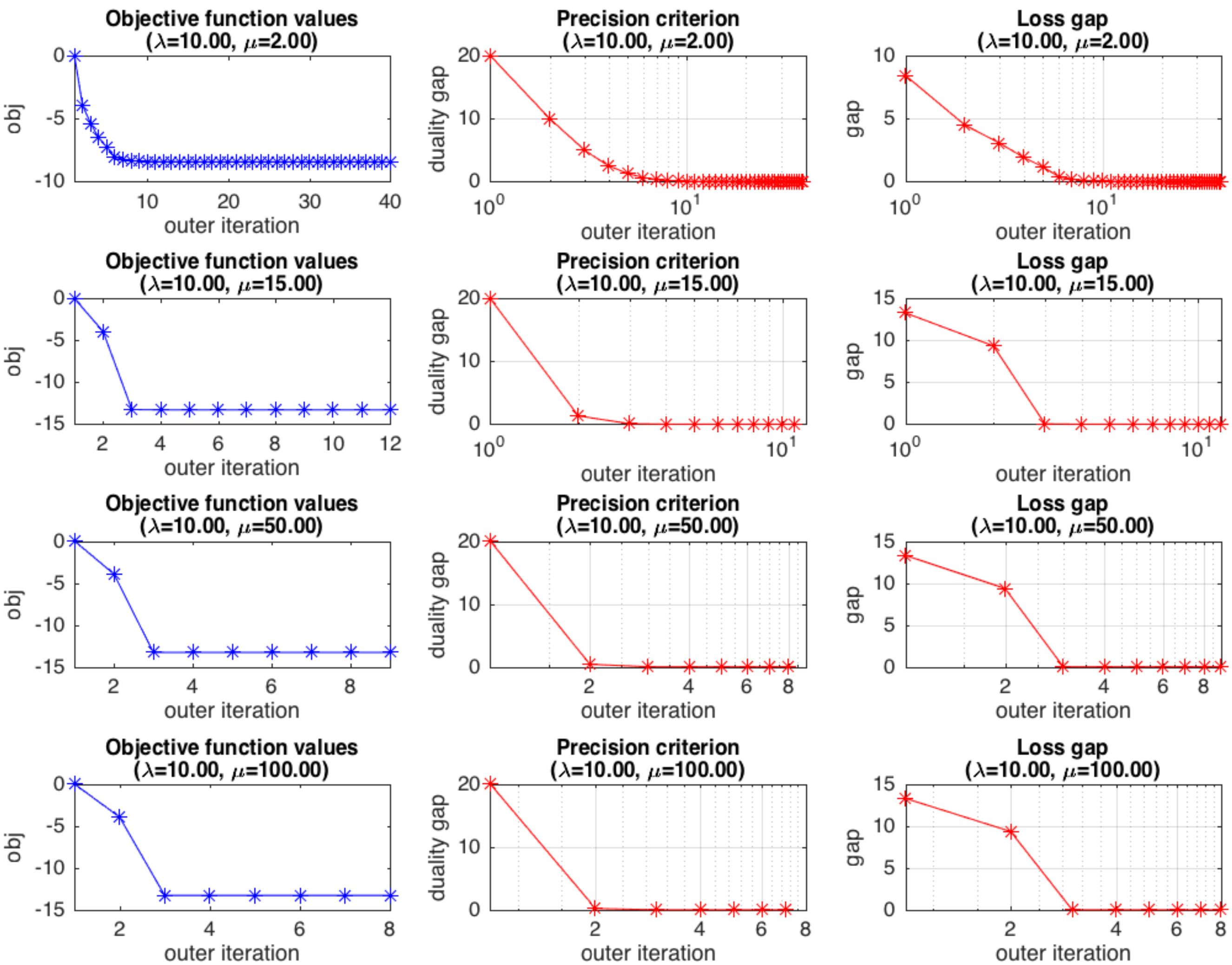
Which is a quadratic problem in the form:

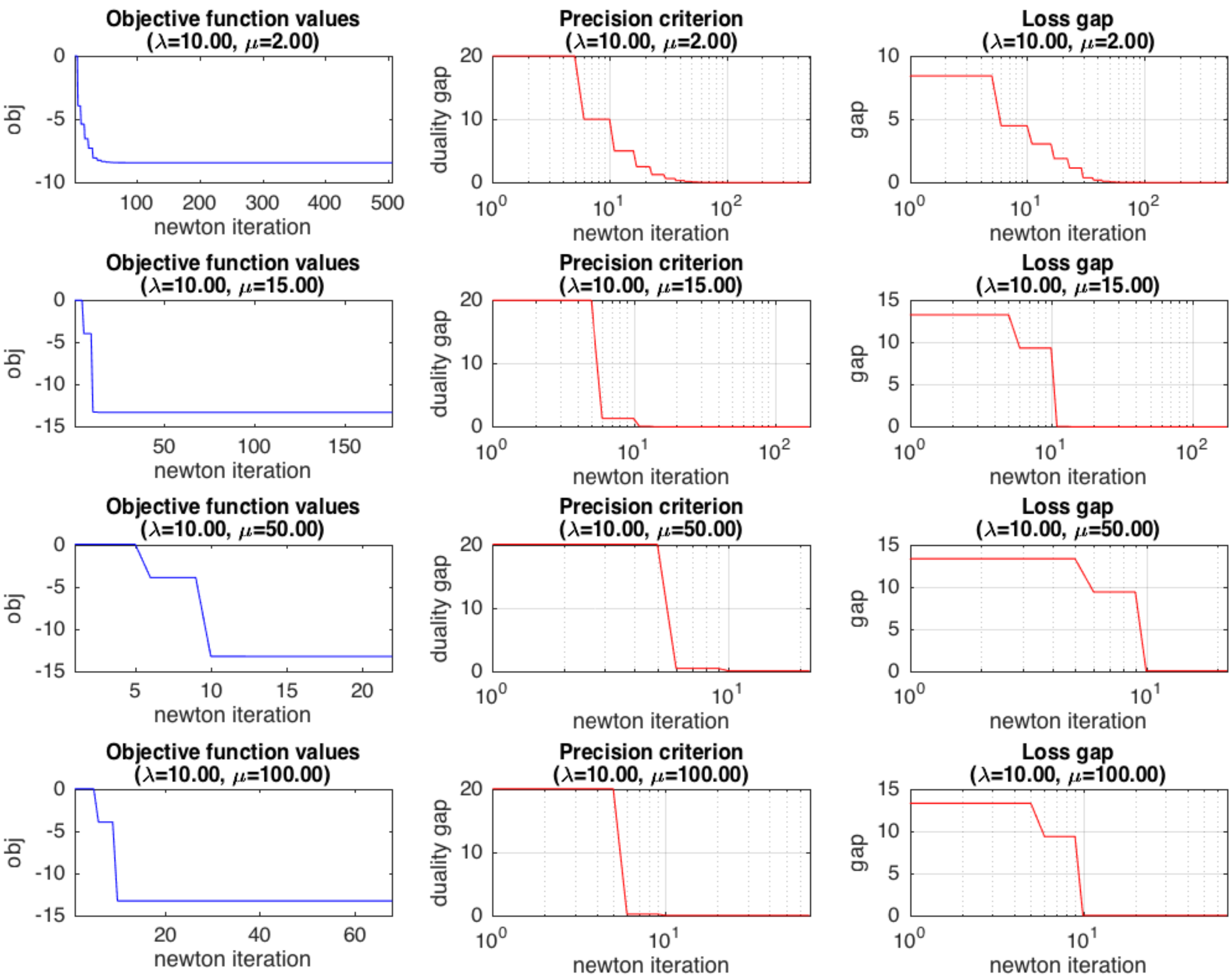
$$\begin{aligned} \text{minimize} \quad & v^T Qv + p^T v \\ \text{subject to} \quad & Av \preceq b \end{aligned} \tag{QP}$$

with $Q = H/2 \in \mathbf{S}^d_{++}$, $p = -HX^T y$, $b = \lambda\mathbf{1}_{2d}$ and $A = [I_d, -I_d]^T \in \mathbb{R}^{2d\times d}$

2. Test

We test the impelmented log-barrier method on randomly generated samples, the results are shown in the figures below. In the current situation, the most appropritate choice of μ is 50 which, among the tested values $\{2, 15, 50, 100\}$, requires the fewest iterations.


Objective function values
(λ=10.00, μ=2.00)
obj
outer iteration
Precision criterion
(λ=10.00, μ=2.00)
duality gap
outer iteration
Loss gap
(λ=10.00, μ=2.00)
gap
outer iteration
Objective function values
(λ=10.00, μ=15.00)
obj
outer iteration
Precision criterion
(λ=10.00, μ=15.00)
duality gap
outer iteration
Loss gap
(λ=10.00, μ=15.00)
gap
outer iteration
Objective function values
(λ=10.00, μ=50.00)
obj
outer iteration
Precision criterion
(λ=10.00, μ=50.00)
duality gap
outer iteration
Loss gap
(λ=10.00, μ=50.00)
gap
outer iteration
Objective function values
(λ=10.00, μ=100.00)
obj
outer iteration
Precision criterion
(λ=10.00, μ=100.00)
duality gap
outer iteration
Loss gap
(λ=10.00, μ=100.00)
gap
outer iteration



2 First order methods for primal problem

1. The sub-gradient descent algorithm for LASSO

To implement the function `subgrad` we use the gradient of the least-squares $\|Xw - y\|_2^2$ with a subgradient of the $l1$-norm:

$$\partial l1(w) = \{g |\ \|g\|_\infty \leq 1,\ g^T w = \|w\|_1\}$$

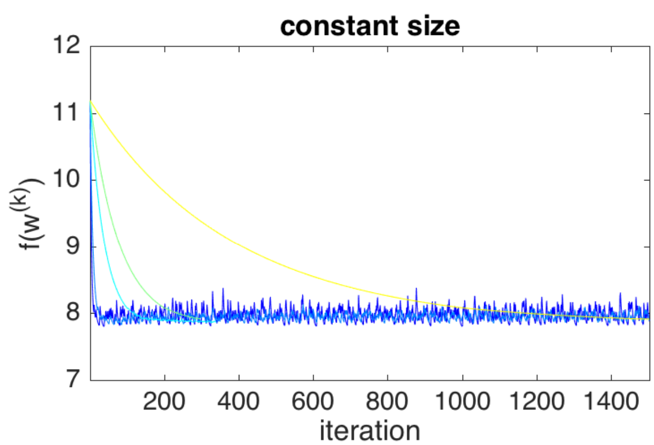
We can simply take $g = sign(w) = \begin{cases} +1, \; w > 0 \\ 0, \; w = 0 \\ -1, \; w < 0 \end{cases}$

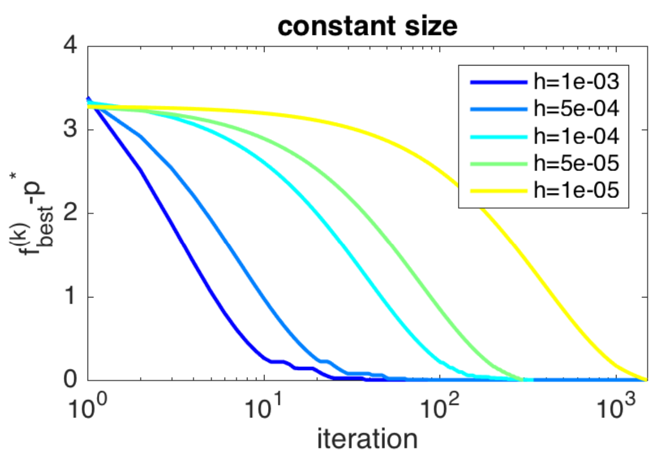
For a randomly genetated sample (`n=100, d=10;` λ`=10`) we plot the loss function values at each itearion and $f_{best}^{(k)} - p^*$ the best value found yet at iteration k compared to the final best value.

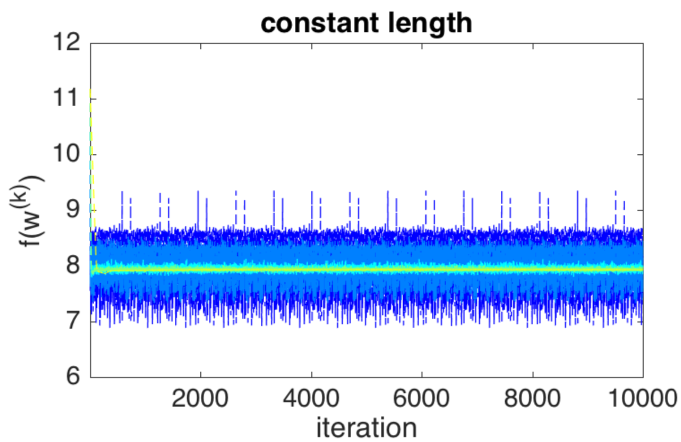
At each iteration we update:

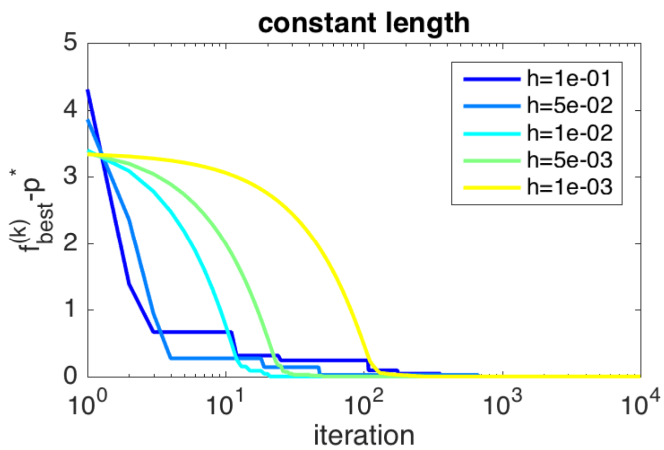
$$w^{(k+1)} = w^{(k)} - \alpha_k . g^{(k)}$$

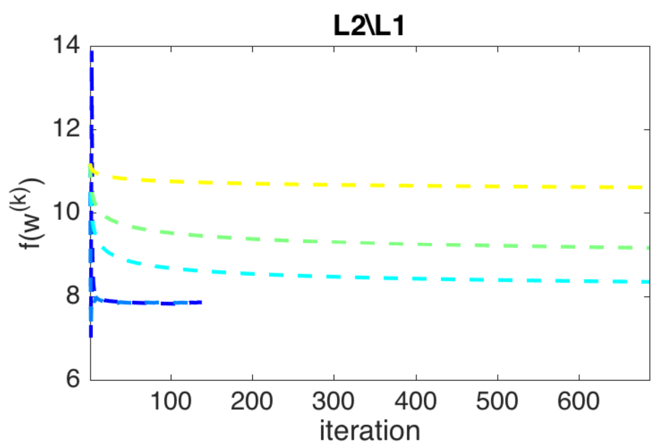
Constant step size: $\alpha_k = h$
Constant step length: $\alpha^{(k)} = h\|g^{(k)}\|_2$
Square summable but not summable $\alpha^{(k)} = \dfrac{h}{k}$
Nonsummable diminishing $\alpha^{(k)} = \dfrac{h}{\sqrt{k}}$

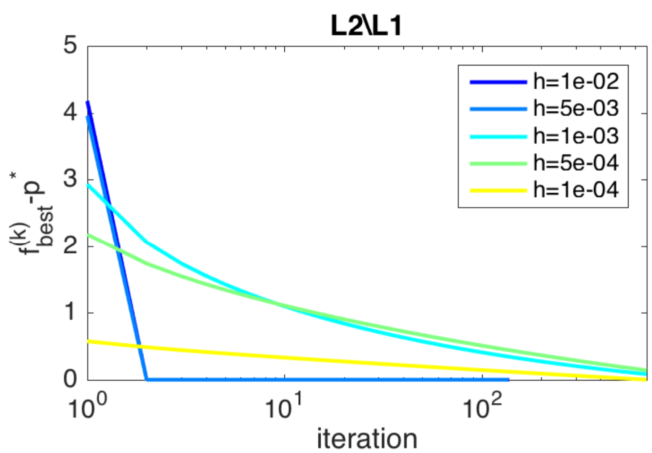


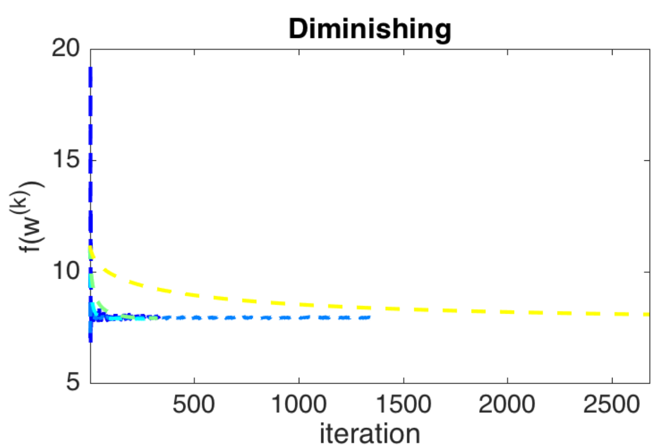


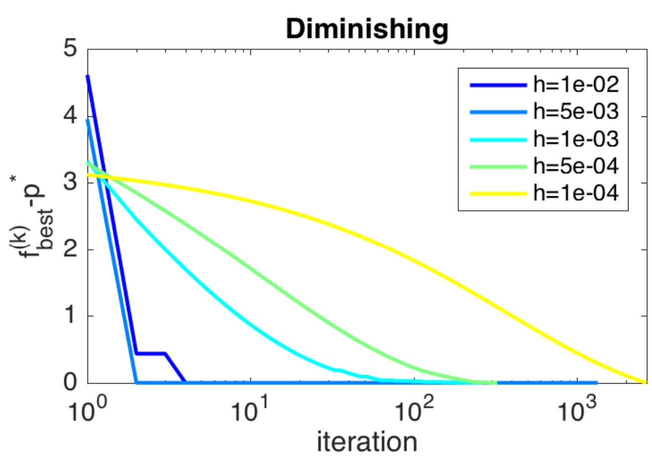

constant length
f(w^(k))
iteration
12
11
10
9
8
7
6
2000
4000
6000
8000
10000


constant length
f_best^(k) - p*
iteration
h=1e-01
h=5e-02
h=1e-02
h=5e-03
h=1e-03


L2\L1
f(w^(k))
iteration
14
12
10
8
6
100
200
300
400
500
600


L2\L1
f_best^(k) - p*
iteration
h=1e-02
h=5e-03
h=1e-03
h=5e-04
h=1e-04


Diminishing
f(w^(k))
iteration
20
15
10
5
500
1000
1500
2000
2500


Diminishing
f_best^(k) - p*
iteration
h=1e-02
h=5e-03
h=1e-03
h=5e-04
h=1e-04

2. The coordinate descent algorithm for the LASSO dual

We iterates over the coordinates (i) and update:

$$\mu_i^{(k+1)} = \arg\min_{-\lambda \leq \mu_i \leq \lambda} \left[\frac{1}{2}\mu^T H \mu - (HX^T y)^T \mu\right]$$

$$\mu_j^{(k+1)} = \mu_j^{(k)},\ j \neq i$$

We have

$$(\nabla_\mu f)_i = H_i^T \mu - (HX^T y)_i$$

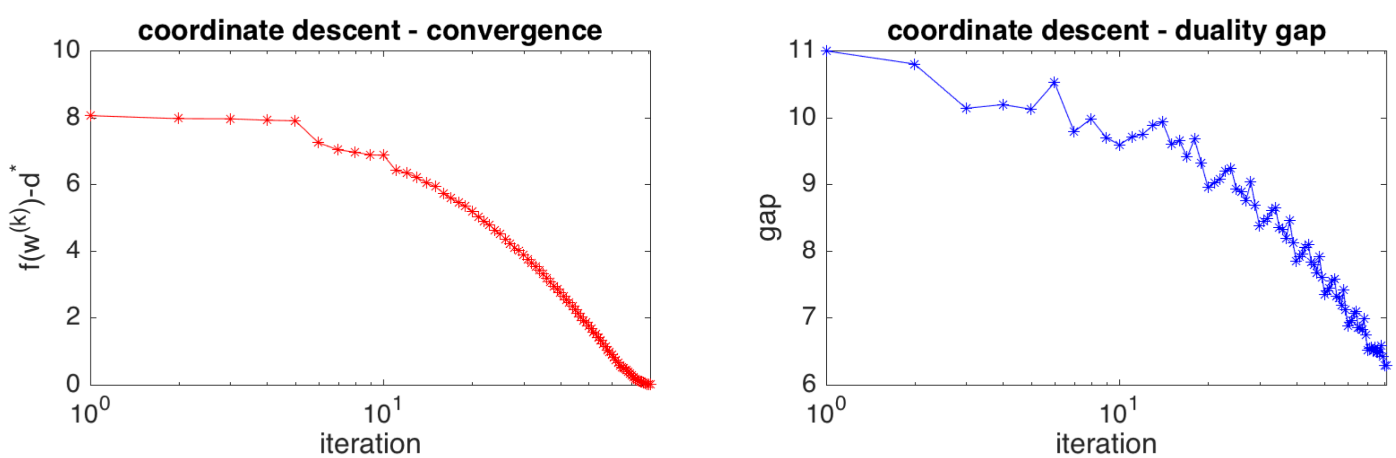
Thus:

$$\mu_i = \frac{(HX^T y)_i - H_{-i}^T \mu_{-i}}{H_{ii}}$$

To satisfy the box constraint we truncate the computed μ_i

$$\mu_i = T_\lambda\left(\frac{(HX^T y)_i - H_{-i}^T \mu_{-i}}{H_{ii}}\right),\ T_\lambda(x) = \begin{cases} \lambda,\ if\ x > \lambda \\ -\lambda,\ if\ x < -\lambda \\ x,\ otherwise \end{cases}$$

The convergence of the method is shown in the figure below.



We compare the CPU time/ Number of required iterations of the subgradient method and the coordinate descent at different precision levels for a sample $(n = 100, d = 10)$.

precision ϵ	1×10^{-3}	1×10^{-6}	1×10^{-10}
Subgradient method($L2\backslash L1$)	0.0009/13	0.0085/313	didn't converge
Coordinate descent	0.0121/8	0.0051/8	0.0038/8

The subgradient method seems more effective for low precision level whilst the coordinate method is more robust with very high precision.

3 Proximal methods for primal problem

1. For the LASSO problem with $A \in \mathbb{R}^{n\times d}$: $f(x) = \frac{1}{2}\|Ax - y\|_2^2$ of hessian $\nabla^2 f(x) = A^T A$. f is strongly convex if there exists $m > 0$ such that $X^T X \succeq mI$ i.e $A^T A$ is positive-definite.
This means $\forall w \in \mathbb{R}^d$, $x^T A^T A x = \mathbf{0} \iff x = \mathbf{0}$ and since $x^T A^T A x = \mathbf{0}$ implies $Ax = \mathbf{0}$ we must have $rank(A) = d$ and consequently $d \leq n$.
If f is strongly convex then the maximum eigenvalue of $\nabla f^2(x)$ is a continuous bounded function of x which means

$$\exists M > 0,\ \forall x \in \mathbb{R}^d,\ \nabla^2 f(x) \preceq MI$$

The tightest choice of m and M would be

$$\begin{cases} m = \lambda_{min}(A^T A) \\ M = \max \lambda_{max}(A^T A) \end{cases}$$

If $n \ll d$ then the hessian is sigular and f is not strongly convex.
2. For the indicator of a convex set I_C

$$\text{prox}_{I_C,P}(x) = \arg\min_z \frac{P}{2}\|z - x\|_2^2 + I_C(z)$$

$\min_z \frac{P}{2}\|z - x\|_2^2 + I_C(z) = \frac{P}{2}\min_z \|z - x\|_2^2$
Thus $\text{prox}_{I_C,P}(x) = \frac{P}{2}.p_C(x) \propto$ the projection of x on the convex set C.
For $h(x) = \|x\|_1$

$$\text{prox}_{h,P}(x) = \arg\min_z \frac{P}{2}\|z - x\|_2^2 + \|z\|_1$$

The optimality condition is:

$$\mathbf{0} \in P(z - x) + \partial(\|z\|_1)$$

h is separable so we can consider each element apart. for i, if $z_i \neq 0$ $\partial(|z_i|) = sign(z_i)$ therefore $z_i := x_i - \frac{1}{P}.sign(z_i)$
if $z_i < 0$ then $x_i < -\frac{1}{P} < 0$ and if $z_i > 0$ then $x_i > \frac{1}{P} > 0$ which means $sign(z_i) = sign(x_i)$ therefore $z_i = x_i - \frac{1}{P}sign(x_i)$ with $|x_i| > \frac{1}{P}$

if $z_i = 0$ the optimality condition becomes $\mathbf{0} \in -Px_i + [-1,1]$ i.e $|x_i| < \frac{1}{P}$.

$$\text{prox}_{h,P}(x)_i = \begin{cases} x_i - \frac{1}{P},\ if\ x_i > \frac{1}{P} \\ 0,\ if\ |x_i| < \frac{1}{P} \\ x_i + \frac{1}{P},\ if\ x_i < -\frac{1}{P} \end{cases}$$

3. For $z, x \in \mathbb{R}^d$

$$f(z) = f(x) + \nabla f(x)^T(z-x) + \frac{1}{2}(z-x)^T\nabla^2 f(y)(z-x)$$

for some $y = tx + (1-t)z,\ t \in [0,1]$.
Assuming the smoothness of f ($\exists M > 0\ \nabla^2 f(x) \preceq MI$) we would have:

$$f(z) \leq f(x) + \nabla f(x)^T(z-x) + \frac{m}{2}\|z-x\|_2^2$$

Therefore:

$$\phi(z) \leq g_{x,M}(z)$$

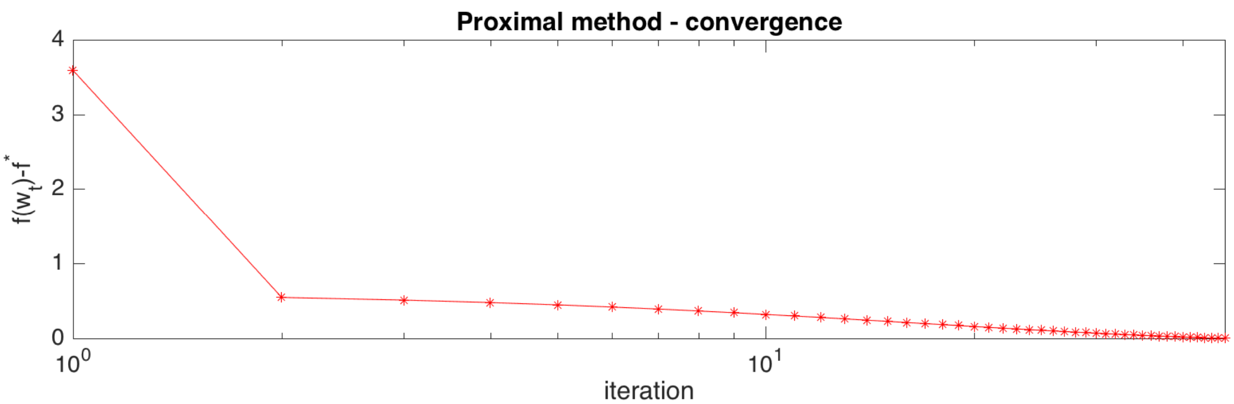
holds for any $M > \lambda_{max}(\nabla^2 f(x))$.
The iteration scheme is:

$$\begin{aligned} x_{t+1} &= \arg\min_z g_{x_t,M}(z) \\ &= \arg\min_z \nabla f(x)^T(z-x_t) + \frac{M}{2}\|z-x_t\|_2^2 + h(z) \\ &= \arg\min_z \frac{M}{2}\|z - x_t + \frac{1}{M}\nabla f(x_t)\|_2^2 + h(z) \\ &= \text{prox}_{h,M}(x_t - \frac{1}{M}\nabla f(x_t)) \end{aligned}$$

For $h = 0$ $\text{prox}_{0,M} = Id$ thus $x_{t+1} = x_t - \frac{1}{M}\nabla f(x_t)$ which is the gradient descent update.
For $h = I_C$ $x_{t+1} = p_C(x_t - \frac{1}{M}\nabla f(x_t))$ the gradient projection update.
4. We implement the proximal method for the LASSO problem and track the CPU time needed to converge as well as the number of iterations.

precision ϵ	1×10^{-3}	1×10^{-6}	1×10^{-10}
Subgradient method($L2\backslash L1$)	0.0009/13	0.0083/313	didn't converge
Coordinate descent	0.0121/80	0.0051/80	0.0038/80
Proximal	0.0158/114	0.0654/435	0.1311/869

The performance of the proximal method is illustrated on a random sample ($n = 100, d = 10, eps = 1e-5$)



5. We rewrite the update as:

$$x_{t+1} = x_t - \frac{1}{M}F(x_t)$$

with

$$F(x_t) = M(x_t - prox_{h,M}(x_t - \frac{1}{M}\nabla f(x_t))$$

From the definition of the proximal operator:

$$u = prox_{h,M}(x) \iff M(x-u) \in \partial h(u)$$

Therefore,

$$M(x_t - \frac{1}{M}\nabla f(x_t) - prox(x_t - \frac{1}{M}\nabla f(x_t))) \in \partial h(prox(x_t - \frac{1}{M}\nabla f(x_t)))$$

and

$$\begin{aligned} F(x_t) = M(x_t - prox(x_t - \frac{1}{M}\nabla f(x_t))) &\in \nabla f(x_t) + \partial h(prox(x_t - \frac{1}{M}\nabla f(x_t))) \\ &\in \nabla f(x_t) + \partial h(x_t - \frac{1}{M}F(x_t)) \end{aligned}$$

For this descent to be a point-fix algorithm we need to prove:

$$F(x^*) = 0 \iff x^* \text{ minimizes } \phi(x) = f(x) + h(x)$$

From the smoothness/strong convexity we get:

$$f(x_t - \frac{1}{M}F(x_t)) \leq f(x_t) - \frac{1}{M}\nabla f(x)^T F(x_t) + \frac{1}{2M}\|F(x_t)\|_2^2$$

Thus from $F(x_t) - \nabla f(x_t) \in \partial h(x_t - \frac{1}{M}F(x_t))$, forall z:

$$\begin{aligned}\phi(x_t - \frac{1}{M}F(x_t)) &\leq f(z) + \nabla f(x)^T(x-z) - \frac{1}{M}\nabla f(x_t)^T F(x_t) + \frac{1}{2M}\|F(x_t)\|_2^2 \\ &+ h(z) + (F(x_t) - \nabla f(x_t))^T(x - z - \frac{1}{M}F(x_t)) \\ &= \phi(z) + F(x_t)^T(x_t - z) - \frac{1}{2M}\|F(x_t)\|_2^2\end{aligned}$$

In particular for $z = x_t$ and $z = x^* = \arg\min \phi(x)$

$$\phi(x_{t+1}) \leq \phi(x_t) - \frac{1}{2M}\|F(x_t)\|_2^2$$

$$\begin{aligned}\phi(x_{t+1}) &\leq \phi(x^*) + F(x_t)^T(x_t - x^*) - \frac{1}{2M}\|F(x_t)\|_2^2 \\ &\leq \phi(x^*) + \frac{M}{2}\left(\|x_t - x^*\|_2^2 - \|x_t - x^* - \frac{1}{M}F(x_t)\|_2^2\right) \\ \phi(x_{t+1}) - \phi(x^*) &\leq \frac{M}{2}(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2)\end{aligned}$$

Which means the sequence $(\phi(x_t))_t$ is non-increasing and we're getting closer to x^*. Then we sum over the n past iterations:

$$\begin{aligned}\sum_{t=1}^{n}(\phi(x_t) - \phi(x^*)) &\leq \sum_{t=1}^{n}\frac{M}{2}(\|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2) \\ &\leq \frac{M}{2}(\|x_0 - x^*\|_2^2 - \|x_n - x^*\|_2^2) \leq \frac{M}{2}\|x_0 - x^*\|_2^2\end{aligned}$$

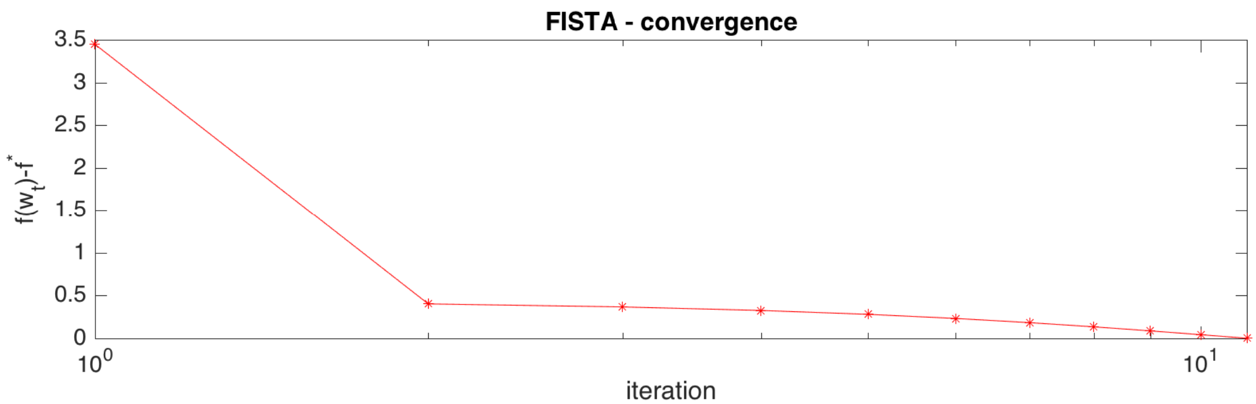
And

$$\phi(x_n) - \phi(x^*) \leq \frac{1}{n}\sum_{t=1}^{n}(\phi(x_t) - \phi(x^*)) \leq \frac{M}{2n}\|x_0 - x^*\|_2^2$$

Therefore the proximal gradient method is a point-fix algorithm that convergs in $\mathcal{O}(1/\epsilon)$.
6. We implement the accelrated proximal method for the LASSO problem and track the CPU time needed to converge as well as the number of iterations.

precision ϵ	1×10^{-3}	1×10^{-6}	1×10^{-10}
Subgradient method($L2\backslash L1$)	0.0009/13	0.0083/313	didn't converge
Coordinate descent	0.0121/80	0.0051/80	0.0038/80
Proximal	0.0158/114	0.0654/435	0.1311/869
Proximal acc	0.0083/10	0.0061/10	0.0052/10

The fast convergence of the accelrated proximal method is shown in the figure below.


FISTA - convergence
f(w_t)-f*
3.5
3
2.5
2
1.5
1
0.5
0
10^0
10^1
iteration