1 Outline

In this lecture, we study

- Nesterov's acceleration and FISTA.

2 LASSO: Least Absolute Shrinkage and Selection Operator

Recall the formulation of LASSO, given by

$$\min_{\theta} \quad \frac{1}{n}\|y - X\theta\|_2^2 + \lambda\|\theta\|_1.$$

Here, the objective function is non-differentiable because of the ℓ_1-regularization term $\lambda\|\theta\|_1$, and therefore, it is non-smooth. On the other hand, the objective is convex, and we have a characterization of the subdifferential of $\|\theta\|_1$, so we can simply apply the subgradient method. To bound the additive error by ϵ, the subgradient method requires $O(1/\epsilon^2)$ iterations.

As discussed in the last lecture, we know that the first part is smooth, and the other part is something whose subdifferential is well understood. Hence, we may apply proximal gradient descent with $h(\theta) = \lambda\|\theta\|_1$ whose associated prox operator is given by

$$\text{prox}_{\eta\lambda\|\cdot\|_1}(\theta) = \left(\underbrace{\max\{0, |\theta_i| - \eta\lambda\}}_{\text{shirinkage operator}} \cdot \text{sign}(\theta_i) \right)_{i\in[d]}$$

The proximal gradient algorithm applies to this composite problem proceeds with the following update rule.

$$\theta_{t+1} = \text{prox}_{\eta h}(\theta_t - \eta\nabla g(\theta_t)).$$

Proximal Gradient Descent applied to LASSO is referred to as Iterative Shrinkage-Thresholding Algorithm (ISTA).

3 Nesterov's Acceleration and FISTA

We observed that proximal gradient descent achieves a convergence rate of $O(1/T)$, and therefore, ISTA solves LASSO with a convergence rate of $O(1/T)$. In fact, we may deduce a faster convergence rate based on Nesterov's acceleration. We mentioned that Nesterov's accelerated gradient descent guarantees a convergence rate of $O(1/T^2)$ for smooth convex minimization. We will show that an accelerated version of proximal gradient descent achives a rate of $O(1/T^2)$ for the composite convex minimization where g is smooth and convex.

Recall that proximal gradient descent for minimizing $g + h$ where g is β-smooth and convex and h is convex follows the update rule of

$$x_{t+1} = \text{prox}_{h/\beta}\left(x_t - \frac{1}{\beta}\nabla g(x_t)\right)$$

from a given point x_t. Instead of applying the gradient descent update to x_t, we move a bit further from x_t along the momentum direction that we took from x_{t-1} to x_t. Let $\gamma_t > 0$ be a weight, and

$$y_t = x_t + \gamma_t(x_t - x_{t-1}).$$

Then we apply the primal gradient descent update on y_t to obtain the next point x_{t+1}, as follows.

$$x_{t+1} = \text{prox}_{h/\beta}\left(y_t - \frac{1}{\beta}\nabla g(y_t)\right).$$

Algorithm 1 summarizes the accelerated version of proximal gradient descent that we just explained.

Algorithm 1 Accelerated Proximal Gradient Descent

Initialize $x_1 \in \mathbb{R}^d$.
Set $x_0 = x_1$.
for $t = 1, \ldots, T$ **do**
  $y_t = x_t + \gamma_t(x_t - x_{t-1})$ for some $\gamma_t > 0$.
  $x_{t+1} = \text{prox}_{h/\beta}\left(y_t - \frac{1}{\beta}\nabla g(y_t)\right)$.
end for
Return x_{T+1}.

To provide a convergence result of the accelerated proximal gradient descent method, we need the following lemma.

Lemma 13.1. *Let $u, v \in \mathbb{R}^d$. Then for all $z \in \mathbb{R}^d$,*

$$\frac{1}{\eta}(\text{prox}_{\eta h}(x) - x)^\top (z - \text{prox}_{\eta h}(x)) + h(z) \geq h(\text{prox}_{\eta h}(x)).$$

Proof. Note that

$$\text{prox}_{\eta h}(x) = \underset{z \in \mathbb{R}^d}{\text{argmin}} \left\{ h(z) + \frac{1}{2\eta}\|x - z\|_2^2 \right\}.$$

By the optimality condition, it follows that for any $z \in \mathbb{R}^d$ and $g \in \partial h(\text{prox}_{\eta h}(x))$,

$$\left(g + \frac{1}{\eta}\left(\text{prox}_{\eta h}(x) - x\right)\right)^\top (z - \text{prox}_{\eta h}(x)) \geq 0.$$

This implies that

$$\frac{1}{\eta}(\text{prox}_{\eta h}(x) - x)^\top (z - \text{prox}_{\eta h}(x)) + g^\top (z - \text{prox}_{\eta h}(x)) \geq 0.$$

Here, since h is convex, we have

$$h(z) \geq h(\text{prox}_{\eta h}(x)) + g^\top (z - \text{prox}_{\eta h}(x)).$$

Adding the two inequalities, we prove the desired bound of this lemma. □

Theorem 13.2. *Let $f = g + h$ where g is a β-smooth convex function in the ℓ_2 norm and h is convex. We set η and γ_t as*

$$\eta = \frac{1}{\beta}, \quad \gamma_t = \frac{t-2}{t+1}.$$

Then x_{T+1} returned by Accelerated Proximal Gradient Descent (Algorithm 1) satisfies

$$f(x_{T+1}) - f(x^*) \leq \frac{2\beta}{(T+1)^2}\|x_1 - x^*\|_2^2$$

where x^* is an optimal solution to $\min_{x\in\mathbb{R}^d} f(x)$.

Proof. Note that Algorithm 1 is equivalent to

$$\begin{aligned} y_t &= (1-\lambda_t)x_t + \lambda_t v_t \\ x_{t+1} &= \mathrm{prox}_{h/\beta}\left(y_t - \frac{1}{\beta}\nabla g(y_t)\right) \\ v_{t+1} &= x_t + \frac{1}{\lambda_t}(x_{t+1} - x_t) \end{aligned}$$

where

$$\lambda_t = \frac{2}{t+1}.$$

This is because $y_t = x_t + \lambda_t(v_t - x_t)$ and

$$\lambda_t(v_t - x_t) = \lambda_t\left(\left(\frac{1}{\lambda_{t-1}} - 1\right)x_t + \left(1 - \frac{1}{\lambda_{t-1}}\right)x_{t-1}\right) = \frac{\lambda_t(1-\lambda_{t-1})}{\lambda_{t-1}}(x_t - x_{t-1}) = \gamma_t(x_t - x_{t-1}).$$

Moreover, we have $\lambda_1 = 1$, and for $t \geq 2$,

$$\frac{1-\lambda_t}{\lambda_t^2} \leq \frac{1}{\lambda_{t-1}^2}.$$

First, as g is β-smooth,

$$g(x_{t+1}) \leq g(y_t) + \nabla g(y_t)^\top(x_{t+1} - y_t) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2.$$

Next, Lemma 13.1 implies that for any $z \in \mathbb{R}^d$,

$$\begin{aligned} h(x_{t+1}) &\leq h(z) + \beta\left(x_{t+1} - y_t + \frac{1}{\beta}\nabla g(y_t)\right)^\top (z - x_{t+1}) \\ &= h(z) + \nabla g(y_t)^\top(z - x_{t+1}) + \beta\,(x_{t+1} - y_t)^\top (z - x_{t+1}). \end{aligned}$$

Adding these two inequalities, we deduce that

$$\begin{aligned} f(x_{t+1}) &\leq h(z) + g(y_t) + \nabla g(y_t)^\top(z - y_t) + \beta\,(x_{t+1} - y_t)^\top (z - x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2 \\ &\leq f(z) + \beta\,(x_{t+1} - y_t)^\top (z - x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2 \end{aligned}$$

where the second inequality follows from convexity of g. By setting $z = x^*$ and $z = x_t$, we have

$$\begin{aligned} f(x_{t+1}) - f(x^*) &\leq \beta\,(x_{t+1} - y_t)^\top (x^* - x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2 \\ f(x_{t+1}) - f(x_t) &\leq \beta\,(x_{t+1} - y_t)^\top (x_t - x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2. \end{aligned}$$

Summing up the first inequality multiplied by λ_t and the second one multiplied by $(1-\lambda_t)$, we get

$$\begin{aligned}
&f(x_{t+1}) - f(x^*) - (1-\lambda_t)(f(x_t) - f(x^*)) \\
&\leq \beta\,(x_{t+1} - y_t)^\top\,(\lambda_t x^* + (1-\lambda_t)x_t - x_{t+1}) + \frac{\beta}{2}\|x_{t+1} - y_t\|_2^2 \\
&= \frac{\beta}{2}(x_{t+1} - y_t)^\top(2\lambda_t x^* + 2(1-\lambda_t)x_t - x_{t+1} - y_t) \\
&= \frac{\beta}{2}\|y_t - (1-\lambda_t)x_t - \lambda_t x^*\|_2^2 - \frac{\beta}{2}\|x_{t+1} - (1-\lambda_t)x_t - \lambda_t x^*\|_2^2 \\
&= \frac{\beta\lambda_t^2}{2}\|v_t - x^*\|_2^2 - \frac{\beta\lambda_t^2}{2}\|v_{t+1} - x^*\|_2^2.
\end{aligned}$$

This implies that

$$\begin{aligned}
\frac{1}{\lambda_t^2}(f(x_{t+1}) - f(x^*)) + \frac{\beta}{2}\|v_{t+1} - x^*\|_2^2 &\leq \frac{1-\lambda_t}{\lambda_t^2}(f(x_t) - f(x^*)) + \frac{\beta}{2}\|v_t - x^*\|_2^2 \\
&\leq \frac{1}{\lambda_{t-1}^2}(f(x_t) - f(x^*)) + \frac{\beta}{2}\|v_t - x^*\|_2^2 \\
&\vdots \\
&\leq \frac{1}{\lambda_1^2}(f(x_2) - f(x^*)) + \frac{\beta}{2}\|v_2 - x^*\|_2^2 \\
&\leq \frac{1-\lambda_1}{\lambda_1^2}(f(x_1) - f(x^*)) + \frac{\beta}{2}\|v_1 - x^*\|_2^2 \\
&= \frac{\beta}{2}\|v_1 - x^*\|_2^2 \\
&= \frac{\beta}{2}\|x_1 - x^*\|_2^2.
\end{aligned}$$

Therefore, it follows that

$$f(x_{T+1}) - f(x^*) \leq \frac{\beta\lambda_T^2}{2}\|x_1 - x^*\|_2^2 = \frac{2\beta}{(T+1)^2}\|x_1 - x^*\|_2^2,$$

as required. □

Hence, the convergence rate is $O(1/T^2)$, which matches the oracle lower bound. The number of required iterations to bound the error by ϵ is $O(1/\sqrt{\epsilon})$.

FISTA stands for Fast ISTA, that is an accelerated version of ISTA. Basically, FISTA is the accelerated proximal gradient descent method applied to LASSO. ISTA requires $O(1/\epsilon)$ iterations, while FISTA needs $O(1/\sqrt{\epsilon})$ iterations to converge to an ϵ-approximate solution.

We generated a random instance with 300 feature variables and 100 data samples. The figure compares the subgradient method, ISTA, and FISTA for the random LASSO instance. We can see that FISTA has the fastest rate of convergence while ISTA is also faster than the subgradient method.

4 Proximal point algorithm

Remember that the proximal gradient method works for the following composite minimization problem.

$$\text{minimize} \quad f(x) = g(x) + h(x).$$

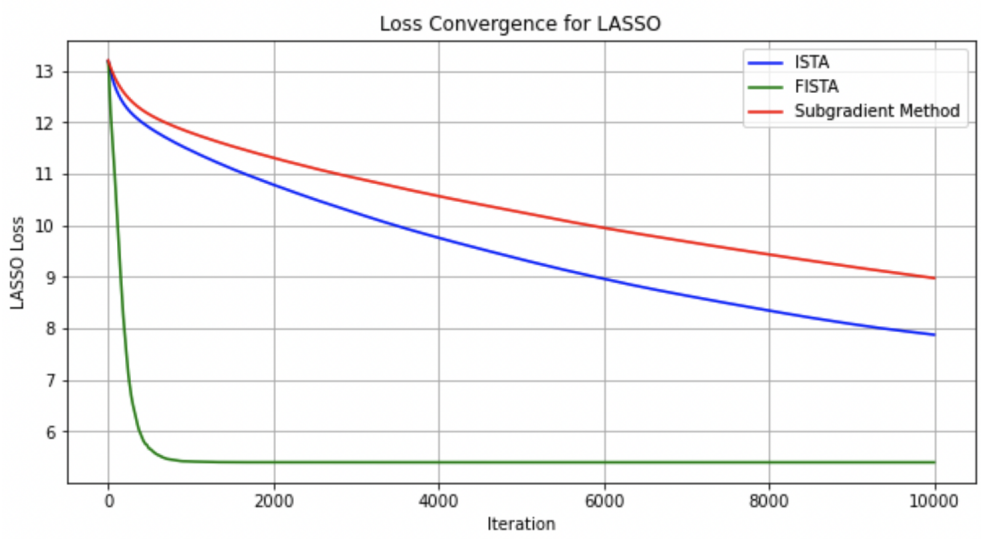

Figure 13.1: Comparing the subgradient method, ISTA, and FISTA for LASSO

The proximal gradient method proceeds with the update rule

$$x_{t+1} = \text{prox}_{\eta h}(x_t - \eta \nabla g(x)).$$

In this section, we discuss the proximal point method, which is a special case of proximal gradient, and its application to the dual problem. Note that minimizing a closed convex function f can be written as a (trivial) composite minimization as follows.

$$\text{minimize} \quad f(x) = 0 + f(x).$$

Here, the first part is $g = 0$, which is trivially smooth, and the second part is $h = f$. Then the corresponding proximal gradient update is given by

$$x_{t+1} = \text{prox}_{\eta f}(x_t).$$

The algorithm with this update rule is referred to as the proximal point method. As $g = 0$ is smooth, the proximal point algorithm converges with a rate of $O(1/T)$.

Algorithm 2 Proximal point algorithm

Initialize x_1.
for $t = 1, \ldots, T$ **do**
  Update $x_{t+1} = \text{prox}_{\eta f}(x_t)$.
end for
Return x_{T+1}.

Theoretically, we can use any function h_t to run the proximal point algorithm, even if the objective is not h_t, in which case, the update rule corresponds to

$$x_{t+1} = \text{prox}_{\eta h_t}(x_t).$$

Hence, at each time step t, we may use a different function h_t hypothetically. Let us consider the first-order approximation of the objective function f at $x = x_t$.

$$h_t(x) = f(x_t) + \nabla f(x_t)^\top (x - x_t).$$

We know that $f(x) \geq h_t(x)$ for all x by convexity. Then what is the proximal point update with h_t? Note that

$$\begin{aligned}\text{prox}_{\eta h_t}(x_t) &= \underset{u}{\text{argmin}} \left\{ f(x_t) + \nabla f(x_t)^\top (u - x_t) + \frac{1}{2\eta} \|u - x_t\|_2^2 \right\} \\ &= x_t - \eta \nabla f(x_t).\end{aligned}$$

Therefore, the proximal point algorithm with the first-order approximation of f is precisely gradient descent. Hence, one can interpret gradient descent as an instance of the proximal point algorithm.

Let us now compare the proximal point algorithm with the objective f and gradient descent.

Lemma 13.3. $\text{prox}_{\eta f}(x) = (I + \eta \partial f)^{-1}(x)$.

Proof. Let $u = \text{prox}_{\eta f}(x)$. Remember that $u = \text{prox}_{\eta f}(x)$ if and only if $x - u \in \eta \partial f(u)$. Note that $x - u \in \eta \partial f(u)$ is equvialent to $x \in (I + \eta \partial f)(u)$, which is equivalent to $u \in (I + \eta \partial f)^{-1}(x)$. In summary,

$$u = \text{prox}_{\eta f}(x) \quad \leftrightarrow \quad u \in (I + \eta \partial f)^{-1}(x).$$

Since u is unique, it follows that $u = (I + \eta \partial f)^{-1}(x)$. □

By this lemma, the proximal point update rule can be written as

$$x_{t+1} = \text{prox}_{\eta f}(x_t) = (I + \eta \partial f)^{-1}(x_t).$$

This is equivalent to $x_t = (I + \eta \partial f)(x_{t+1}) = x_{t+1} + \eta \nabla f(x_{t+1})$, which is

$$x_{t+1} = x_t - \eta \nabla f(x_{t+1}).$$

In contrast to gradient descent that proceeds with $x_{t+1} = x_t - \eta \nabla f(x_t)$, we use the gradient at x_{t+1}.