EE 381V: Large Scale Optimization **Fall 2012**

Lecture 22 — November 15

Lecturer: Caramanis & Sanghavi *Scribe: Namyoon Lee and Tianyang Bai*

22.1 Recap

We briefly recap what was covered in the previous lecture.

22.1.1 Convergence Rate of Gradient Descent Method

Recall that to solve a convex optimization problem,

$$\min f(x),$$

we found that gradient descent (and subgradient descent for non-smooth problems) achieved certain performance guarantees, depending on the certain properties of the function f. Specifically, we saw the following:

1. If $f(x)$ is only known to be convex, using the subgradient descent method, the convergence rate is $O(1/\sqrt{k})$, i.e., to achieve ϵ accuracy, we need $O(1/\epsilon^2)$ iterations.
2. If $f(x)$ is known to be convex and smooth (assuming L-Lipschitz gradient in the proof), using the gradient descent method, the convergence rate is $O(1/k)$, i.e., to achieve ϵ accuracy, we need $O(1/\epsilon)$ iterations.
3. If $f(x)$ is strongly convex and smooth, using the gradient descent method, we have the following results:

$$||x_k - x^*||_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k ||x_0 - x^*||,$$

where $\kappa = \frac{M}{m}$ is the condition number of function f.

22.1.2 Lower Bound on the Iteration Times

Given a problem class and an algorithm, we have been able to provide a convergence analysis. These are effectively upper bounds on the running times, or iterations required to achieve some given sub optimality. It is also possible to discuss lower bounds. In order to do this, we (briefly) describe the Information Complexity *Gradient Oracle Model*, first pioneered (to the best of our knowledge) by Nemirovski and Yudin (see also the 2004 textbook by Nesterov). This allows us to give algorithm-free lower bounds on the number of calls needed to an oracle that provides function evaluation and a gradient or sub gradient. We note, however, that these lower bounds are worst-case, over the corresponding class of functions.

In summary: the lower bounds are given in terms of the number of function and subgradient evaluations, required to obtain an ϵ-optimal solution to any optimization problem within the class.

An oracle is a function $\phi : \mathcal{S} \to \mathcal{I}$ that answers any query $x \in \mathcal{S}$ by returning an element $\phi(x)$ in an information set $\mathcal{I}$. In our case, $\phi(x) = \{f(x), \partial f(x)\}$, the function and subgradient of f at x.

Specifically, at any given iteration t, the optimization method $\mathcal{M}$ queries at $x_t \in \mathcal{S}$, and the oracle returns the information $\phi(x_t, f) = \{f(x), \partial f(x)\}$. The method $\mathcal{M}$ then uses the information $\{\phi(x_1, f), \phi(x_2, f), ..., \phi(x_t, f)\}$ to determine x_{t+1}, the point at which the next query will be made.

To achieve ϵ-optimal solution, we have the following results regarding the iteration (query) times:

1. if $f(x)$ is convex, the lower bound iteration times is $O(1/\epsilon^2)$, i.e., $O(1/\sqrt{k})$ in terms of convergence rate;
2. if $f(x)$ is convex and smooth, the lower bound iteration times is $O(1/\sqrt{\epsilon})$, i.e., $O(1/k^2)$ in terms of convergence rate;
3. if $f(x)$ is strongly convex and smooth with condition number $\kappa = \frac{M}{m}$, then
$$||x_k - x^*||_2 \le \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k ||x_0 - x^*||.$$

Note that in case 2 and case 3, a gradient descent method does not achieve the lower bound convergence rate. Hence this motivates us to find "smarter" methods with an improved convergence rate.

22.1.3 Methods with Momentum term

It turns out that the lower bounds given above are indeed achievable. The algorithms that achieve them are essentially no more computationally demanding than the gradient and sub gradient algorithms we have already seen. The key difference is the addition of what has been come to be called a *momentum term*, whereby the next iterate x_{k+1} depends not only on the gradient and previous point x_k, but also on the point previous to that, x_{k-1}. This dependence comes through a momentum term $\beta_k(x_k - x_{k-1})$ at iteration $k+1$. Thus, generically, the update looks like:

$$x_{k+1} = x_k - t_k \nabla f + \beta_k(x_k - x_{k-1}).$$

There are three popular methods of the form above, one of which we have already seen in detail:

1. FISTA: we will explore its convergence rate for f convex and smooth.
2. Heavy Ball: we will explore its convergence rate when f is strongly convex and smooth.
3. Conjugate gradient: recall that we had already considered this earlier in the class, although at that point we did not explore its momentum interpretation, or acceleration properties.

Exercise 1 Show that there is a momentum term in the non-linear conjugate gradient method.

Proof: Recall that in the conjugate gradient method:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k p_k, && (22.1)\\ p_k &= -\nabla f_k + \beta_k p_{k-1}. && (22.2) \end{aligned}$$

By (22.1), we have

$$p_{k-1} = \frac{x_k - x_{k-1}}{\alpha_{k-1}}. \qquad (22.3)$$

Substituting (22.2) and (22.3) for (22.1), we have that

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k \nabla f_k + \alpha_k \beta_k p_{k-1}\\ &= x_k - \alpha_k \nabla f_k + \frac{\alpha_k \beta_k}{\alpha_{k-1}} \left(x_k - x_{k-1}\right). \end{aligned}$$

□

22.1.4 Proximal Gradient Descent

Recall from last time the proximal operator of a closed convex function h is defined as

$$\text{Prox}_h(\mathbf{x}) = \arg\min_u \{h(\mathbf{u}) + ||\mathbf{u} - \mathbf{x}||^2\}.$$

In the proximal gradient method, we assume

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}),$$

where $g(\mathbf{x})$ and $h(\mathbf{x})$ are convex functions and $h(\mathbf{x})$ possibly non-smooth function. Then, the update is

$$\begin{aligned} \mathbf{x}_+ &= \text{Prox}_{th}(\mathbf{x} - t\nabla g(\mathbf{x}))\\ &= \arg\min_{\mathbf{u}} \left\{ h(\mathbf{u}) + g(\mathbf{x}) + \nabla g(\mathbf{x})^{\mathrm{T}}(\mathbf{u} - \mathbf{x}) + \frac{1}{2t}||\mathbf{u} - \mathbf{x}||_2^2 \right\}, \end{aligned}$$

where intuitively, $g(\mathbf{x}) + \nabla g(\mathbf{x})^{\mathrm{T}}(\mathbf{u} - \mathbf{x}) + \frac{1}{2t}||\mathbf{u} - \mathbf{x}||_2^2$ can be viewed as the quadratic approximation of $g(\mathbf{u})$ given $g(\mathbf{x})$.

22.2 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

In this lecture, we introduce a class of *non-descent* algorithm called FISTA for solving a non-smooth convex optimization problem.

22.2.1 Algorithm

Let us consider the following problem:

$$\min_{x} f(\mathbf{x}),$$

where $f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$. Here, $g(\mathbf{x})$ and $h(\mathbf{x})$ are convex functions and $h(\mathbf{x})$ possibly non-smooth function. To solve this problem, we use FISTA algorithm. Basically, the general step of FISTA is of the form

$$\mathbf{x}^{(k)} = \mathrm{Prox}_{t_k h}(\mathbf{y} - t_k \nabla g(\mathbf{y})), \tag{22.4}$$

where $\mathbf{y} = \mathbf{x}^{(k-1)} + \frac{k-2}{k+1}\left(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}\right)$, t_k is a step-size, and initial condition is $\mathbf{x}^{(0)} = \mathbf{x}^{(-1)}$. We can re-write algorithm in 22.4 into an equivalent form as

$$\mathbf{x}^{(k)} = \mathrm{Prox}_{t_k h}(\mathbf{y} - t_k \nabla g(\mathbf{y})),$$

where $\mathbf{y} = (1-\theta_k)\mathbf{x}^{(k-1)} + \theta_k \mathbf{v}^{(k-1)}$, $\theta_k = \frac{2}{k+1}\mathbf{v}^{(k)}$, $\mathbf{v}^{(k-1)} = \mathbf{x}^{(k-2)} + \frac{1}{\theta_{k-1}}(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)})$, and $\mathbf{v}^{(0)} = \mathbf{x}^{(0)}$.

22.2.2 Interpretation by Comparison with Proximal Gradient Method

From an algorithm perspective, the main difference between FISTA and the proximal gradient algorithm introduced in the previous lecture is that the $\mathrm{Prox}_{t_k h}(\cdot)$ operator is not employed on the previous point $\mathbf{x}^{(k-1)}$, but rather at the point $\mathbf{y}$ which exploits a specific linear combination of the previous two points $\left\{\mathbf{x}^{(k-1)}, \mathbf{x}^{(k-2)}\right\}$, i.e., we use two steps of memory at each iteration. Thus, each iteration of FISTA algorithm can be interpreted as a proximal mapping using extrapolated point $\mathbf{y}$ as shown in Fig.22.1. Here, one thing to notice is that $\mathbf{x}^{(k)}$ is feasible in **dom** h but $\mathbf{y}$ may be infeasible in **dom** h because of extrapolation process. From a computational complexity point of view, the additional required computation for FISTA is clearly marginal. This is because the main computational effort at each iteration in both proximal gradient method and FISTA remains the same: The $\mathrm{Prox}_{t_k h}(\cdot)$ operator requires most computation efforts at each step.

22.3 Convergence Analysis

In this lecture we will prove the convergence of FISTA algorithm. The following Theorem is the main result for the convergence.

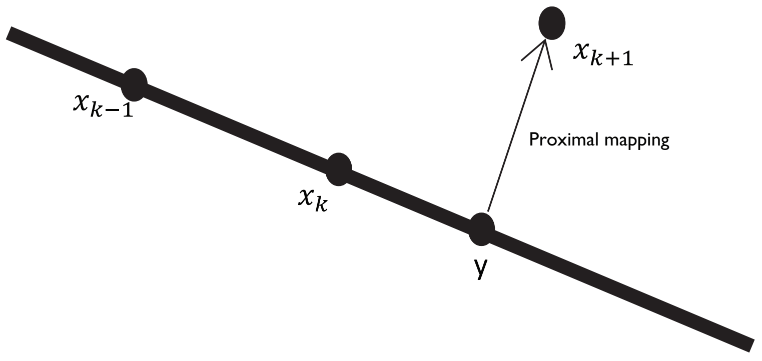

Figure 22.1. In FISTA $\mathbf{x}_{k+1}$ is the proximal mapping of $\mathbf{y}$, one linear combination of $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$; while in proximal gradient, $\mathbf{x}_{k+1}$ is just the proximal mapping of $\mathbf{x}_k$.

Theorem 22.1.

$$\frac{t_i}{\theta_i^2}\left(f(\mathbf{x}^{(i)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(i)} - \mathbf{x}^*\| \leq \frac{1-\theta_i}{\theta_i^2} t_i \left(f(\mathbf{x}^{(i-1)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(i-1)} - \mathbf{x}^*\| \tag{22.5}$$

Before providing proof for Theorem 22.1, let us first consider two key corollaries used for proof.

Corollary 22.2. *Let us consider a smooth and convex function $g(\mathbf{x})$ with L-Lipschitz condition, i.e., $\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$. If $t < \frac{1}{L}$, then*

$$g(\mathbf{x}^+) \leq g(\mathbf{y}) + \nabla g(\mathbf{y})^T(\mathbf{x}^+ - \mathbf{y}) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2.$$

Proof: This proof is done by the definition of convexity of $g(\mathbf{x})$. □

Corollary 22.3. $h(\mathbf{x}^+) \leq h(\mathbf{z}) + \nabla g(\mathbf{y})^T(\mathbf{z} - \mathbf{x}^+) + \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^+)$

Proof: If $\mathbf{u} = \text{Prox}_h(\mathbf{x})$,

$$\begin{aligned}
&\iff \quad \mathbf{x} - \mathbf{u} \in \partial h(\mathbf{x}) \\
&\iff \quad h(\mathbf{z}) \geq h(\mathbf{u}) + (\mathbf{x} - \mathbf{u})^T(\mathbf{z} - \mathbf{u}) \quad \forall \mathbf{z}. \\
&\iff \quad h(\mathbf{u}) \leq h(\mathbf{z}) + \frac{1}{t}(\mathbf{w} - \mathbf{u})^T(\mathbf{u} - \mathbf{w}) \quad \forall \mathbf{w} \quad \text{and} \quad \forall \mathbf{z}.
\end{aligned} \tag{22.6}$$

Recall that $\mathbf{x}^+ = \text{Prox}_{th}(\mathbf{y} - t\nabla g(\mathbf{y}))$. Therefore, from 22.6, we have

$$\begin{aligned}
h(\mathbf{x}^+) &\leq h(\mathbf{z}) + \frac{1}{t}(\mathbf{y} - t\nabla g(\mathbf{y}) - \mathbf{x}^+)^T(\mathbf{x}^+ - \mathbf{z}) \\
&= h(\mathbf{z}) + \nabla g(\mathbf{y})^T(\mathbf{z} - \mathbf{x}^+) + \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^+).
\end{aligned}$$

This completes the proof. □

Now, we are ready to prove Theorem 22.1 by using Corollaries 22.2 and 22.3.

Proof: Since $f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$, the upper bound of $f(\mathbf{x}^+)$ is given by

$$\begin{aligned} f(\mathbf{x}^+) &\leq h(\mathbf{z}) + g(\mathbf{y}) + \nabla g(\mathbf{y})^T(\mathbf{z}-\mathbf{y}) + \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\mathbf{z}-\mathbf{x}^+) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2, \\ &\leq h(\mathbf{z}) + g(\mathbf{z}) + \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\mathbf{z}-\mathbf{x}^+) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2, \quad \forall \mathbf{z} \\ &= f(\mathbf{z}) + \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\mathbf{z}-\mathbf{x}^+) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2. \quad \forall \mathbf{z} \end{aligned} \tag{22.7}$$

Since the inequality in 22.7 holds for all $\mathbf{z}$, we can compute convex combination of upper bounds for 1) $\mathbf{z} = \mathbf{x}$ and 2) $\mathbf{z} = \mathbf{x}^*$, which is

$$\begin{aligned} & f(\mathbf{x}^+) - f^* - (1-\theta)(f(\mathbf{x}) - f^*) \\ = \;& f(\mathbf{x}^+) - \theta f^* - (1-\theta) f(\mathbf{x}) \\ \leq \;& \frac{1}{t}(\mathbf{x}^+ - \mathbf{y})^T(\theta\mathbf{x}^* + (1-\theta\mathbf{x}^+)) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2. \end{aligned}$$

By plugging the definition of $\mathbf{y} = (1-\theta)\mathbf{x} + \theta\mathbf{v}$ and $\mathbf{v}^+ = \mathbf{x} + \frac{1}{\theta}(\mathbf{x}^+ - \mathbf{x})$, we have

$$\begin{aligned} & f(\mathbf{x}^+) - f^* - (1-\theta)(f(\mathbf{x}) - f^*) \\ \leq \;& \frac{1}{2t}\left[\|\mathbf{y} - (1-\theta)\mathbf{x} - \theta\mathbf{x}^*\|_2^2 - \|\mathbf{x}^+ - (1-\theta)\mathbf{x} - \theta\mathbf{x}^*\|_2^2\right] \\ = \;& \frac{\theta^2}{2t}\left[\|\mathbf{v} - \mathbf{x}^*\|_2^2 - \|\mathbf{v}^+ - \mathbf{x}^*\|_2^2\right]. \end{aligned}$$

If the inequality (22.2) holds at iteration i, then we have

$$\frac{t_i}{\theta_i^2}\left(f(\mathbf{x}^{(i)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(i)} - \mathbf{x}^*\|_2^2 \leq \frac{1-\theta_i}{\theta_i^2} t_i \left(f(\mathbf{x}^{(i-1)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(i-1)} - \mathbf{x}^*\|_2^2,$$

which completes the proof. □

22.3.1 Analysis for Fixed Step Size

For a fixed step size $t_i = t = \frac{1}{L}$, substituting $(1-\theta_i)/\theta_i^2 \leq 1/\theta_{i-1}^2$ for Theorem 22.1 , we have that

$$\begin{aligned} \frac{t}{\theta_k^2}\left(f(\mathbf{x}^{(k)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(k)} - \mathbf{x}^*\|_2^2 &\leq \frac{1-\theta_1}{\theta_1^2} t\left(f(\mathbf{x}^{(0)}) - f^*\right) + \frac{1}{2}\|\mathbf{v}^{(0)} - \mathbf{x}^*\|_2^2 \\ &= \frac{1}{2}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2. \end{aligned}$$

Therefore,

$$f(\mathbf{x}^{(k)}) - f^* \leq \frac{\theta_k^2}{2t}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 = \frac{2L}{(k+1)^2}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2. \tag{22.8}$$

From the inequality (22.8), we conclude that FISTA algorithm reaches $f(\mathbf{x}^{(k)}) - f^* \leq \epsilon$ after $O(1/\sqrt{\epsilon})$ iterations with the fixed step size.

22.3.2 Analysis for Backtracking Line Search

In general cases, L is difficult to obtain. Therefore, we have to consider alternative ways to choose step size t_i. Line search method is a good candidate for selecting a reasonable step size when we do not know L. The exit condition plays an important role in characterizing line search method. Here, we use the exit condition, which according to Corollary 22.2 is

$$g(\mathbf{x}^+) \leq g(\mathbf{y}) + \nabla g(\mathbf{y})^T(\mathbf{x}^+ - \mathbf{y}) + \frac{1}{2t}\|\mathbf{x}^+ - \mathbf{y}\|_2^2).$$

We select step size $t^{(k)}$ by keep reducing as $t^{(k)} = \beta t^{(k-1)}$ where $\beta < 1$ until the exit condition is satisfied. Hence, in the backtracking line search case, we have

$$t_k \geq \min\{1, \beta/L\},\ \forall k \in \mathbb{N}_+$$

Therefore,

$$f(\mathbf{x}^{(k)}) - f^* \leq \frac{\theta_k^2}{2\min\{1, \beta/L\}}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 \tag{22.9}$$

we conclude that with backtracking line search FISTA algorithm still reaches $f(\mathbf{x}^{(k)}) - f^* \leq \epsilon$ after $O(1/\sqrt{\epsilon})$ iterations.

22.3.3 From a Lyapunov Function Point of View

We interpret the convergence proof for Theorem 22.1 from a Lyapunov function perspective. Recall that FISTA algorithm belongs to a class of non-descent algorithm. Therefore, we cannot show that $\|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)\|$ is monotonically decreasing as in the gradient descent algorithms. In order to prove the convergence of FISTIA algorithm, we considered a type of Lyapunov functions that includes two terms: 1) $\left|f(\mathbf{x}^{(k)}) - f^*\right|$ and 2) $\|\mathbf{v}^{(k)} - \mathbf{x}^*\|$. Instead looking at $\|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)\|$, we examined and showed that one specific linear sum of the two terms (we can view such linear combination as the "Lyapunov function" of FISTA method.) is monotonically decreasing. We will see this type of convergence proof in the next lecture with more details.