IE 598: Big Data Optimization **Fall 2016**

Lecture 19: Proximal Gradient Method and Its Acceleration

Lecturer: Niao He *Scriber: Lucas Buccafusca*

Overview: In this lecture, we introduce and analyze the Proximal Gradient method as a way of solving non-smooth convex problems with specific composite structure. We analyze the convergence rate and discuss the corresponding accelerated methods. Lastly, some simulations are done to demonstrate these algorithms for solving LASSO problem and compare algorithms we learned for nonsmooth optimization.

19.1 Motivations for Proximal Gradient Method

Our goal is to continue to address $\min_{x\in\mathbb{R}^n} f(x)$. Previously, we have discussed severak methods to solve it, based on the properties of f, assuming f is convex. For instance,

- f is smooth: Gradient Descent Method.

$$x_{t+1} = x_t - \gamma_t \nabla f(x_t) = \operatorname*{argmin}_x \big\{ \frac{1}{2\gamma_t} \|x - [x_t - \gamma_t \nabla f(x_t)]\|_2^2 \big\}$$

- f is nonsmooth: Proximal Point Algorithm.

$$x_{t+1} = \text{prox}_{\gamma_t f}(x_t) = \operatorname*{argmin}_x \big\{ \frac{1}{2\gamma_t} ||x - x_t||_2^2 + f(x) \big\}$$

In this lecture, we will consider the special type of nonsmooth problems when the objective is the sum of a smooth and a nonsmooth function:

$$\min_x F(x) = \min_x \big\{ f(x) + g(x) \big\}$$

where $f(x)$ is a smooth and convex function and $g(x)$ is a non-smooth and convex function.

Note that neither gradient descent nor proximal point algorithm works well in this situation. We will introduce a new algorithm, **proximal gradient method**, which essentially combines gradient descent and proximal point algorithm and works as follows

$$x_{t+1} = \text{prox}_{\gamma_t g}(x_t - \gamma_t \nabla f(x_t)) = \operatorname*{argmin}_x \big\{ \frac{1}{2\gamma_t} ||x - [x_t - \gamma_t \nabla f(x_t)]||_2^2 + g(x) \big\}$$

We will discuss the details below.

19.2 Convex Composite Minimization

Problem setting. We consider the following convex minimization problem

$$\min_{x\in\mathbb{R}^n} F(x) := f(x) + g(x)$$

where:

- $f(x)$ is L-smooth (i.e. gradient is Lipschitz continuous) and convex
- $g(x)$ is convex, 'simple' and non-smooth (By 'simple', we mean that the proximal operation of g is easy to calculate)

Applications. This type of optimization problem are found everywhere:

- Machine Learning: many supervised learning problems can be cast into optimization problems in the form
$$\min_{h\in\mathbb{H}} \sum_{i=1}^{n} l(h(x_i), y_i) + p(h)$$
where h is the prediction function to be learned, the loss function $\ell(\cdot,\cdot)$ is many cases are smooth functions, and the $p(h)$ is known as a penalty or regularization, which is often nonsmooth, e.g. L_1, L_2 or a combination of $L_1\&L_2$ norms. Specific examples of supervised learning include: ridge regression, LASSO, logisitic regression, etc.
- Signal and Image Processing: many signal/image recover problems can be formulated as the general form
$$\min_x ||Ax - b||_2^2 + r(x)$$
where the first term is the data fidelity term and the second term is some regularization. Such examples include compressive sensing, image deblurring, image denoising, etc.

Special cases: Note that the above problem covers many problems we have discussed as special cases

- nonsmooth convex minimization: when $f(x) = 0$
- smooth convex minimization : when $g(x) = 0$
- constrained convex minimizaton : when $g(x) = \delta_X(x)$ is an indicator function of a closed convex set X

19.3 Proximal Gradient Method

The proximal gradient (PG) method (dates back to [B75]) is fairly simple: at each iteration $t = 0, 1, 2, \ldots,$

$$x_{t+1} = \text{prox}_{\gamma_t g}(x_t - \gamma_t \nabla f(x_t)) = \underset{x}{\text{argmin}} \left\{ \frac{1}{2\gamma_t} ||x - [x_t - \gamma_t \nabla f(x_t)]||_2^2 + g(x) \right\} \tag{19.1}$$

Note that

- $f(x) = 0$: PG $\Rightarrow$ proximal point algorithm
- $g(x) = 0$: PG $\Rightarrow$ gradient descent
- $g(x) = \delta_X(x)$: PG $\Rightarrow$ projected gradient descent

LASSO Example of Proximal Gradient Method We provide a special example to help demonstrate how this algorithm works on the LASSO problem:

$$\min_x \underbrace{\frac{1}{2}||Ax-b||_2^2}_{f(x)} + \underbrace{\lambda||x||_1}_{g(x)}$$

First, we wish to calculate $\nabla f(x)$ and $\text{prox}_{\mu||\cdot||_1}(x)$. By inspection we have:

$$\nabla f(x) = A^T(Ax-b)$$
$$\text{prox}_{\mu||\cdot||_1}(x) = \underset{y}{\text{argmin}} \left\{ \frac{1}{2\mu}\sum_i (x_i - y_i)^2 + \sum_i |y_i| \right\}$$

We can separate this into each of the ith coordinates:

$$[\text{prox}_{\mu||\cdot||_1}(x)]_i = \underset{y_i}{\text{argmin}} \left\{ \tfrac{1}{2\mu}(x_i - y_i)^2 + |y_i| \right\}$$

The solution to this type of problem is well-known as the *soft thresholding operator*.

$$S_\mu(x_i) = \begin{cases} x_i - \mu & \text{if } x_i > \mu \\ 0 & \text{if } |x_i| \le \mu \\ x_i + \mu & \text{if } x_i < -\mu \end{cases}$$

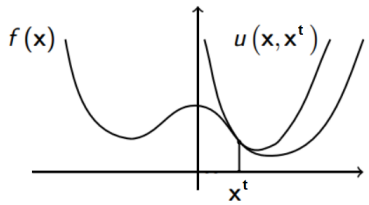
For each coordinates, we have the solution in the form of this soft thresholding operator, so we can rewrite our LASSO problem in Proximal Gradient form as

$$x_{t+1} = S_{\lambda\gamma_t}(x_t - A^T(Ax_t - b))$$

This is known as the Iterative Soft Thresholding Algorithm (ISTA).

Interpretation of Proximal Gradient Method. There are several ways in which we can interpret the Proximal Gradient Method:

1. *Majorization and Minimization:* A more general framework to solving minimization problems is to find an upper bound to your function and then try to minimize that bound. This framework is called majorization and minimization. This is done even in non-convex instances. Pictorally, we represent the Majorization and Minimization (otherwise known as MM) method below, where u(x) is the upper bound to our function [P16].



   Note that the updates (19.1) of proximal gradient method at each iteration can be rewritten as

$$x_{t+1} = \underset{x}{\text{argmin}} \left\{ f(x_t) + \nabla f(x_t)^T(x - x_t) + \frac{1}{2\gamma_t}||x - x_t||_2^2 + g(x) \right\}$$

If $\gamma_t \leq \frac{1}{L}$, then due to smoothness we have:

$$f(x_t) + \nabla f(x_t)^T(x - x_t) + \frac{1}{2\gamma_t}||x - x_t||_2^2 \geq f(x)$$

so we have an upper bound function that we minimize over, with equivalence at $x = x_t$.

2. *Fixed Point Iteration:* The proximal gradient method can also be treated as an iterative algorithm for a fixed point problem.

   Lemma 19.1 *x^* is optimal if and only if $\forall \gamma > 0$: $x^* = \text{prox}_{\gamma g}(x^* - \gamma \nabla f(x^*))$.*

   Proof: On the one hand, we have

   $$x^* \text{ is optimal } \Leftrightarrow 0 \in \nabla f(x^*) + \partial g(x^*).$$

   On the other hand, we have

   $$x^* = \text{prox}_{\gamma g}(x^* - \gamma \nabla f(x^*)) \Leftrightarrow 0 \in \frac{1}{\gamma}(x^* - (x^* - \gamma \nabla f(x^*))) + \partial g(x^*) \Leftrightarrow 0 \in \nabla f(x^*) + \partial g(x^*).$$

   ■

3. *Forward-Backward Operator:* We can equivalently write the proximal gradient operation as:

   $$x_{t+1} = (I + \gamma_t \partial g)^{-1}(I - \gamma_t \nabla f)(x_t)$$

   The $(I - \gamma_t \nabla f)$ is the 'forward' portion of the algorithm, e.g. a gradient step, and $(I + \gamma_t \partial g)^{-1}$ is the 'backward' portion of the algorithm, i.e. the proximal operator. This concept is typically used to solve general problems given by the sum of two maximal monotone operators.

19.4 Convergence Rate

Now we analyze the convergence of this algorithm. We show that just like gradient descent or proximal point algorithm, the proximal gradient method attains the $O(1/t)$ convergence rate.

Theorem 19.2 *Proximal gradient method with fixed step size $\gamma_t = \frac{1}{L}$ satisfies the following convergence rate:*

$$F(x_t) - F(x^*) \leq \frac{L||x_0 - x^*||_2^2}{2t}$$

Proof: For notation purposes, we can write: $x_{t+1} = x_t - \gamma_t G_{\gamma_t}(x_t)$ where $G_\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_{\gamma g}(x - \gamma \nabla f(x)))$ From Lipschitz smoothness of $f(x)$, we know that the quadratic upper bound of any function is:

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \tfrac{L}{2}||y - x||_2^2$$

Using the above property for $y = x - \gamma_t G_\gamma(x)$

$$f(x - \gamma_t G_\gamma(x)) \leq f(x) + \nabla f(x)^T((x - \gamma_t G_\gamma(x)) - x) + \tfrac{L}{2}||x - \gamma_t G_\gamma(x) - x||_2^2$$

$$\leq f(x) - \gamma_t \nabla f(x)^T G_\gamma(x) + \tfrac{L\gamma_t^2}{2}||G_\gamma(x)||_2^2$$

$$\leq f(x) - \gamma_t \nabla f(x)^T G_\gamma(x) + \tfrac{\gamma_t}{2}||G_\gamma(x)||_2^2 \text{ for } \gamma_t \leq \tfrac{1}{L}$$

Claim: For any y, we have

$$F(x - \gamma_t G_\gamma(x)) \leq F(y) + \nabla G_\gamma(x)^T (x - y) - \tfrac{\gamma_t}{2}||G_\gamma(x)||_2^2$$

Proof of Claim: From above, when applied to our function F:
$$F(x - \gamma_t G_\gamma(x)) \leq f(x) - \gamma_t \nabla f(x)^T G_\gamma(x) + \tfrac{\gamma_t}{2}||G_\gamma(x)||_2^2 + g(x - \gamma_t G_\gamma(x))$$

From the convexity of f and g and the fact that $G_\gamma(x) - \nabla f(x) \in \partial g(x - \gamma_t G_\gamma(x))$,

$$\begin{aligned} F(x - \gamma_t G_\gamma(x)) \leq \quad & f(y) + \nabla f(x)^T (x - y) - \gamma_t \nabla f(x)^T G_\gamma(x) + \tfrac{\gamma_t}{2}||G_\gamma(x)||_2^2 \\ & + g(y) + (G_\gamma(x) - \nabla f(x))^T (x - y - \gamma_t G_\gamma(x)) \end{aligned}$$

Simplifying and canceling terms nets:
$$F(x - \gamma_t G_\gamma(x)) \leq f(y) + g(y) + G_\gamma(x)^T (x - y) - \tfrac{\gamma_t}{2}||G_\gamma(x)||_2^2$$

Thus we have proved the Claim.

So now, applying the inequality f at $x = x_t$ an $y = x^*$, we have:

$$\begin{aligned} F(x_{t+1}) - F(x^*) &\leq G_{\gamma_t}(x_t)^T (x_t - x^*) - \frac{\gamma_t}{2}||G_{\gamma_t}(x_t)||_2^2 && (19.2) \\ &= \frac{1}{2\gamma_t}\left[||x_t - x^*||_2^2 - ||x_t - x^* - \gamma_t G_{\gamma_t}(x_t)||_2^2\right] && (19.3) \\ &= \frac{1}{2\gamma_t}\left[||x_t - x^*||_2^2 - ||x_{t+1} - x^*||_2^2\right] && (19.4) \end{aligned}$$

Take sums of both sides over all t and consider $\gamma_t = \frac{1}{L}$
$$\textstyle\sum_{i=1}^t (F(x_i) - F(x^*)) \leq \frac{L}{2} \sum_{i=1}^t \left[||x_i - x^*||_2^2 - ||x_{i+1} - x^*||_2^2\right] \leq \frac{L}{2}||x_0 - x^*||_2^2$$

Now we get:

$$F(x_t) - F(x^*) \leq \frac{1}{t}\sum_{i=1}^t (F(x_i) - F^*) \leq \frac{L}{2t}||x_0 - x^*||_2^2$$

■

Note that this convergence rate is the same as gradient descent and proximal point algorithm. Similarly, this is not the best rate achievable. We can also have accelerated proximal gradient descent methods.

19.5 Acceleration

19.5.1 Proximal Gradient with Backtracking Line-Search

Often times during our analysis, we set $\gamma_t = \frac{1}{L}$, but in practice, we do not know L a priori, or it is difficult to solve for. For example, for the LASSO problem, $L = \lambda_{max}(A^T A)$, but A may be large and difficult to work with. In practice we use backtracking line-search to find the local Lipschitz constant.

Backtracking line-search for Lipschitz constant Here is how the line-search works

- we initialize $L_0 = 1$ and some $\alpha > 1$.
- At each iteration t, we find the smallest integer i such that $L = \alpha^i L_{t-1}$ satisfies the Lipschitz condition, specifically:
$$F(x^+) \leq F(x_t) + \nabla f(x_t)(x^+ - x_t) + \tfrac{L}{2}||x^+ - x_t||_2^2$$
where $x^+ = \text{prox}_{\frac{g}{L}}(x_t - \frac{1}{L}\nabla f(x_t))$.
Then update $L_t = L$ and $x_{t+1} = x^+$

19.5.2 Accelerated Proximal Gradient Method

Originally developed by [Nesterov, 2007] and [Beck and Teboulle, 2009] , we can accelerate the proximal gradient method simply as follows

$$x_{t+1} = \text{prox}_{\gamma_t g}(y_t - \gamma_t \nabla f(y_t))$$

$$y_{t+1} = x_{t+1} + \beta_t(x_{t+1} - x_t)$$

Some simple choices for β: $\beta_t = \frac{t}{t+3}$ or $\beta_t = \frac{\lambda_t - 1}{\lambda_{t+1}}$ where $\lambda_0 = 0, \lambda_{t+1} = \frac{1+\sqrt{1+4\lambda_t^2}}{2}$.

The latter choice of acceleration is =called the Fast Iterative Soft Thresholding Algorithm (FISTA) derivation [BT09]. It was shown in [BT09] that

Theorem 19.3 *The sequences $x_t, F(x_t)$ generated via FISTA with either a constant or backtracking (with ratio $\alpha \geq 1$) stepsize rule satisfy*
$$F(x_t) - F(x^*) \leq \frac{2\alpha L}{t^2}||x_0 - x^*||_2^2$$

Remarks: By far, we have shown that when solving the composite convex minimization problem $\min_x \ f(x)+ g(x)$, we can the accelerated proximal algorithm attains the optimal $O(1/t^2)$ convergence rate.

19.6 Simulation

Some simulations of the LASSO problem were demonstrated in class [6].

$$\min_x \ \frac{1}{2}||Ax - b||_2^2 + \lambda||x||_1$$

We implement and compare all the algorithms we learned so far for solving nonsmooth problems, including

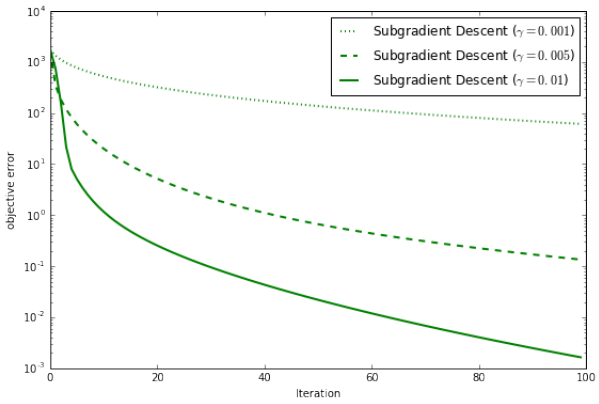
- Subgradient descent
- Nesterov's smoothing
- Proximal gradient (w/o backtracking)
- Accelerated proximal gradient (w/o backtracking)

We summarize the convergence rates of each of the five methods in the table below:

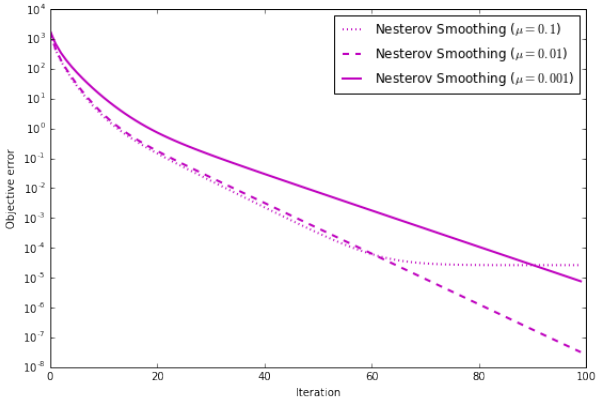
Method	Convergence	Parameter
Subgradient Descent	$O(\frac{1}{\sqrt{t}})$	stepsize $\gamma_t = \frac{\gamma}{\sqrt{t}}$
Nesterov Smoothing + GD	$O(\frac{1}{\mu t})$	smoothness $\mu > 0$
Nesterov Smoothing + AGD	$O(\frac{1}{\mu t^2})$	smoothness $\mu > 0$
Proximal Gradient Descent	$O(\frac{1}{t})$	stepsize $\gamma_t = \frac{1}{L}$ or line-search
Accelerated Proximal Gradient	$O(\frac{1}{t^2})$	stepsize $\gamma_t = \frac{1}{L}$ or line-search

We provide the results below:

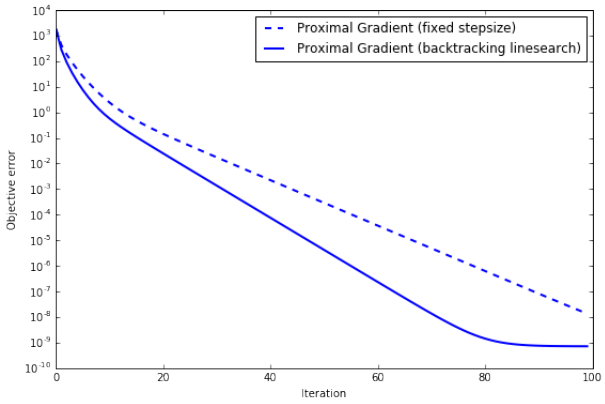
- Subgradient Descent - since the optimal stepsize is not computable, we set $\gamma_t = \frac{\gamma}{\sqrt{t}}$ and adjust γ.



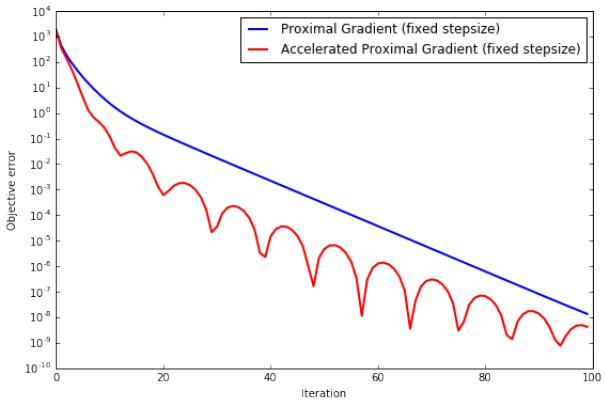
- Nesterov Smoothing - we approximate the nonsmooth L_1 norm by the Huber function and solve the resulting smoothed problem with Gradient Descent and adjust the smoothness parameter μ.



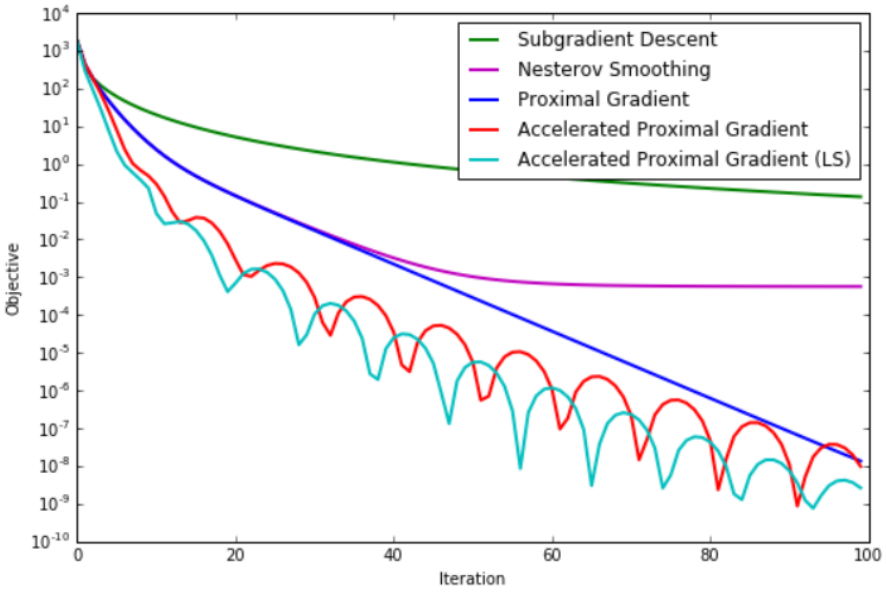
- Proximal Gradient - the only parameter here is the stepsize. We apply both fixed stepsize and backtracking line search. We see that backtracking helps.



- Accelerated Proximal Gradient - We can improve on Proximal Gradient Descent by applying the Accelerated method. With the same parameters of a fixed stepsize, we can immediately see the improvement.



Below is the overall comparison of the all five methods applied on the LASSO problem:



References

[B75] Bruck Jr, Ronald E., "An iterative solution of a variational inequality for certain monotone operators in Hilbert space." Bulletin of the American Mathematical Society 81, no. 5 (1975): 890-892.

[P16] Palomar, Daniel , "ELEC 5470-Convex Optimization Course Notes", 2016.

[CP09] Combettes, Patrick L.; Pesquet, Jean-Christophe, "Proximal Splitting Methods in Signal Processing." (2009) arXiv:0912.3522

[N07] Nesterov, Yu , "Gradient methods for minimizing composite objective function" No. CORE Discussion Papers (2007/76). UCL, 2007.

[BT09] Beck, A., and Teboulle, M. , "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." SIAM journal on imaging sciences, 2(1), 183-202, 2009.

[H16] He, Niao , "Python demo for LASSO " `http://niaohe.ise.illinois.edu/IE598/lasso_demo/index.html`, 2016.