10-725/36-725: Convex Optimization **Fall 2018**

Lecture 7: September 19

Lecturer: Ryan Tibshirani *Scribes: Dhruv Mauria Saxena, Weiran Yao, Yang Yang*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

7.1 Subgradient Concepts Revisited

Subgradients are alternatives to gradients when the function f is non-smooth or non-differentiable. Recall that for convex and differentiable f:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \text{ for all } x, y$$

The linear approximation always underestimates f. A subgradient of a convex function f at x is any $g \in \mathbb{R}^n$ such that:

$$f(y) \geq f(x) + g^T (y - x) \text{ for all } x, y$$

7.2 Subgradient Method

Now consider f convex, having $\text{dom}(f) = \mathbb{R}^n$, but not necessarily differentiable. Our objective is to minimize f. Subgradient method is like gradient descent, but we replace gradients with subgradients, i.e. initialize $x^{(0)}$, repeat:

$$x^{(k)} = x^{(k-1)} - t_k g^{(k-1)}, \quad k = 1, 2, 3, \cdots$$

where $g^{(k-1)} \in \partial f(x^{(k-1)})$ is any subgradient of f at $x^{(k-1)}$, and ∂f represents the subdifferential of f.

7.2.1 Differences to gradient descent

Gradient descent method requires f to be convex and differentiable. Subgradient method only requires function f to be convex and the subgradient $g^{(k-1)}$ of f to exist at point $x^{(k-1)}$.

The step size used in gradient descent is chosen to be fixed and small, or by backtracking line search. For subgradient method, we generally choose fixed step size or diminishing step size, which will be discussed in the next section. Key differences are that step sizes are not adaptively computed for subgradient method.

Gradient descent, as its name suggests, is a descent method, which means the objective value is expected to decrease for every iteration with appropriate step size (such as $t \leq L$). However, subgradient method is not necessarily a descent method, so we keep track of best iterate $x^{(k)}_{\text{best}}$ among $x^{(0)}, x^{(1)}, ..., x^{(k)}$ so far, i.e.:

$$f(x^{(k)}_{\text{best}}) = \min_{i \in [0,k]} f(x^{(i)})$$

7.2.2 Step size choices

Three kinds of step size choices are covered in the lecture for subgradient methods: (1) fixed step sizes; (2) diminishing step sizes, and (3) Polyak step sizes.

Fixed step sizes $t_k = t$ for all $k = 1, 2, 3, ...$

Diminishing step sizes Choose t_k to meet conditions $\sum_{k=1}^{\infty} t_k^2 < \infty$ and $\sum_{k=1}^{\infty} t_k = \infty$. These two inequalities, square summable but not summable, ensure that step sizes diminish to zero, but not too fast.

Polyak step sizes when the optimal value $f^\star$ is known, take $t_k = \frac{f(x^{(k-1)}) - f^\star}{\|g^{(k-1)}\|_2^2}$. The step size can be derived by minimizing the RHS of subgradient proof:

$$\|x^{(k)} - x^\star\|_2^2 \leq \|x^{(k-1)} - x^\star\|_2^2 - 2t_k(f(x^{(k-1)}) - f(x^\star)) + t_k^2\|g^{(k-1)}\|_2^2$$

7.2.3 Convergence analysis

As in Section 7.2, we assume f to be convex, $\text{dom}(f) = \mathbb{R}^n$, but also additionally f is G-Lipschitz where $G > 0$ is some constant. This means,

$$|f(x) - f(y)| \leq G\|x - y\|_2$$

We can provide the following theorems concerning asymptotic convergence bounds:

Theorem 7.1 *(**Convergence for fixed step size**) For a fixed step size t, subgradient method satisfies*

$$\lim_{k\to\infty} f\left(x_{\text{best}}^{(k)}\right) \leq f^\star + \frac{G^2 t}{2}$$

Theorem 7.2 *(**Convergence for diminishing step size**) For diminishing step sizes that satisfy the conditions from Section 7.2.2, subgradient method satisfies*

$$\lim_{k\to\infty} f\left(x_{\text{best}}^{(k)}\right) = f^\star$$

Proof: The proof derives a 'basic inequality' that is used for both theorems above.

For a convex, G-Lipschitz function f, a subgradient has bounded norm. That is,

$$\begin{aligned} g \in \partial f(x) \implies \|g\|_2 &\leq G \\ \text{using} \quad f(y) &\geq f(x) + g^T(y - x) \\ \text{and} \quad |f(x) - f(y)| &\leq G\|x - y\|_2 \end{aligned} \tag{7.1}$$

From the definition of a subgradient,

$$\begin{aligned}\|x^{(k)} - x^\star\|_2^2 &= \|x^{(k-1)} - t_k g^{(k-1)} - x^\star\|_2^2 \\ &= \|x^{(k-1)} - x^\star\|_2^2 + \underbrace{t_k^2 \|g^{(k-1)}\|_2^2}_{\leq t_k^2 G^2, \text{ using Equation 7.1}} - 2t_k \left(g^{(k-1)}\right)^T \left(x^{(k-1)} - x^\star\right) \\ &\leq \|x^{(k-1)} - x^\star\|_2^2 + t_k^2 G^2 - 2t_k \left(f\left(x^{(k-1)}\right) - f\left(x^\star\right)\right) \end{aligned} \tag{7.2}$$

Where we use the definition of a subgradient in the last term on the right hand side.

$$\begin{aligned} f\left(x^\star\right) &\geq f\left(x^{(k-1)}\right) + \left(g^{(k-1)}\right)^T \left(x^{(k-1)} - x^\star\right) \\ \implies \left(g^{(k-1)}\right)^T \left(x^{(k-1)} - x^\star\right) &\leq -f\left(x^{(k-1)}\right) + f\left(x^\star\right) \end{aligned}$$

Iterate over Equation 7.2 for $i = 0, \cdots, k$ to get

$$\begin{aligned} \underbrace{0 \leq \|x^{(k)} - x^\star\|_2^2}_{\text{non-negative distance}} &\leq \underbrace{\|x^{(0)} - x^\star\|_2^2}_{\text{some constant } R^2} + \sum_{i=1}^k t_i^2 G^2 - 2\sum_{i=1}^k t_i \left(f\left(x^{(i-1)}\right) - f\left(x^\star\right)\right) \\ \implies 0 &\leq R^2 + \sum_{i=1}^k t_i^2 G^2 - 2\sum_{i=1}^k t_i \left(f\left(x^{(i-1)}\right) - f\left(x^\star\right)\right) \\ \implies 2\sum_{i=1}^k t_i \left(f\left(x^{(i-1)}\right) - f\left(x^\star\right)\right) &\leq R^2 + \sum_{i=1}^k t_i^2 G^2 \end{aligned}$$

But each term in the summation on the left hand side

$$\begin{aligned} t_i \left(f\left(x^{(i-1)}\right) - f\left(x^\star\right)\right) &\geq t_i \left(f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right)\right) \\ \implies 2\sum_{i=1}^k t_i \left(f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right)\right) &\leq R^2 + \sum_{i=1}^k t_i^2 G^2 \\ \implies f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right) &\leq \frac{R^2 + \sum_{i=1}^k t_i^2 G^2}{2\sum_{i=1}^k t_i} \end{aligned} \tag{7.3}$$

where $f\left(x_{\text{best}}^{(k)}\right) = \min_{i=0,\cdots,k} f\left(x^{(i)}\right)$ is the objective value at the best iterate $x_{\text{best}}^{(k)}$. Equation 7.3 is the basic inequality we can use to derive convergence results for different step sizes.

For $t_i = t, \forall i$,

$$f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right) \leq \frac{R^2 + t^2 k G^2}{2tk} \xrightarrow{\text{as } k \to \infty} \frac{G^2 t}{2} \tag{7.4}$$

For diminishing t_i, using the assumptions from Section 7.2.2,

$$f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right) \leq \frac{R^2 + \sum_{i=1}^k t_i^2 G^2}{2\sum_{i=1}^k t_i} \xrightarrow{\text{as } k \to \infty} \frac{R^2 + G^2 \underbrace{\sum_{i=1}^k t_i^2}_{<\infty}}{2\underbrace{\sum_{i=1}^k t_i}_{\to \infty}} \to 0$$

This concludes the proof. ■

Convergence Rate From Equation 7.4, say we want an ϵ-suboptimal solution, i.e.

$$f\left(x_{\text{best}}^{(k)}\right) - f\left(x^\star\right) \leq \frac{R^2}{2tk} + \frac{tG^2}{2} \leq \epsilon$$

We can make each of the terms in the middle equation to be $\leq \frac{\epsilon}{2}$, which would mean their sum is $\leq \epsilon$.

$$\frac{tG^2}{2} = \frac{\epsilon}{2} \implies t = \frac{\epsilon}{G^2}$$
$$\frac{R^2}{2tk} = \frac{\epsilon}{2} \implies k = \frac{R^2}{t\epsilon} = \frac{R^2G^2}{\epsilon^2}$$

This shows that subgradient method has convergence rate $O(1/\epsilon^2)$ (cf. convergence rate of $O(1/\epsilon)$ for gradient descent).

Remark: This derivation for the convergence rate assumes knowledge of $f\left(x^\star\right)$. However, the analysis is tight, i.e. we get the same rate for an optimal choice of step sizes t_i, $\forall\, i = 1, \cdots, k$ without knowledge of $f\left(x^\star\right)$.

7.2.3.1 Example: Regularized logistic regression

Given $(x_i, y_i) \in \mathbb{R}^p \times \{0, 1\}$ for $i = 1, \cdots, n$, the logistic regression loss is

$$f(\beta) = \sum_{i=1}^{n} \big(- y_i x_i^T \beta + \log(1 + \exp(x_i^T \beta))\big) \tag{7.5}$$

The loss function is smooth and convex. We have

$$\nabla f(\beta) = \sum_{i=1}^{n} (y_i - p_i(\beta)) x_i \tag{7.6}$$

where $p_i(\beta) = \exp(x_i^T \beta)/(1 + \exp(x_i^T \beta))$, $i = 1, \cdots, n$.

Consider the regularized problem:

$$\min_{\beta} f(\beta) + \lambda \cdot P(\beta) \tag{7.7}$$

where $P(\beta)$ is the penalty. If $P(\beta) = \|\beta\|_2^2$, it is ridge penalty. If $P(\beta) = \|\beta\|_1$, it is lasso penalty.

With ridge penalty, the objective function is smooth and convex. So we can use gradient descent method to solve the problem.

With lasso penalty, the objective function is nonsmooth, because $\|\beta\|_1$ is nonsmooth. Then we use the subgradient method to solve the problem.

The example of comparing the convergence rate of the gradient descent method and the subgradient method is shown in Figure 7.1. In the example, $n = 1000$, $p = 20$. Step sizes are hand-tuned to be favorable for each method. From the example we can observe that the subgradient method is much slower in convergence than the gradient descent method.

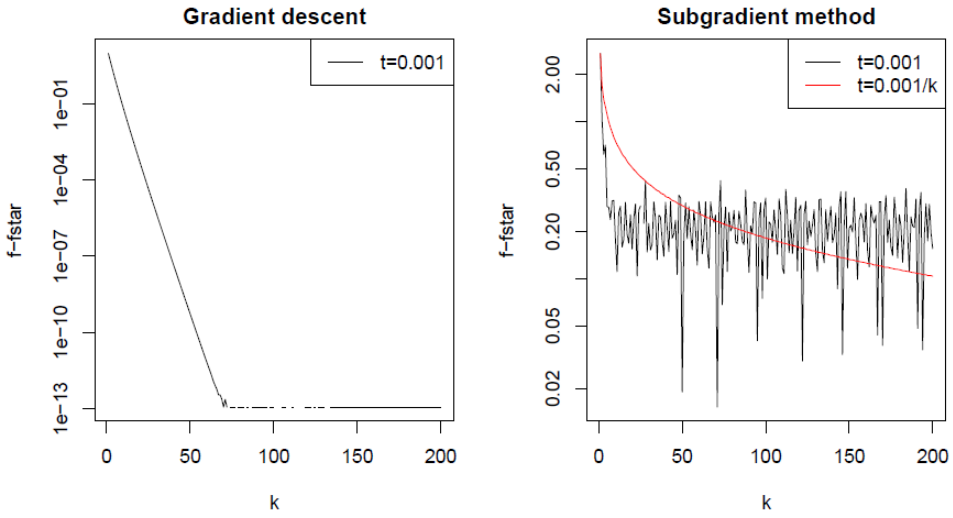

Figure 7.1: . Example of the regularized logistic regression. Left: Ridge penalty, gradient descend method used. Right: Lasso penalty, subgradient method used.

7.2.4 Example: intersection of sets

Suppose we want to find $x^* \in C_1 \cap \cdots \cap C_m$, i.e., a point in the intersection of closed, convex sets $C_1, \cdots, C_m$.

We define

$$f_i(x) = \text{dist}(x, C_i), \quad i = 1, \cdots, m \tag{7.8}$$

$$f(x) = \max_{i=1,\cdots,m} f_i(x) \tag{7.9}$$

where $\text{dist}(x, C_i) = \min_{y \in C_i} \|y - x\|_2$, and solve

$$\min_x f(x) \tag{7.10}$$

We have that $f^* = 0 \iff x^* \in C_1 \cap \cdots \cap C_m$. We can check that $f(x)$ is convex.

The gradient of the distance function $\text{dist}(x, C)$ is

$$\nabla \text{dist}(x, C) = \frac{x - P_C(x)}{\|x - P_C(x)\|_2} \tag{7.11}$$

where $P_C(x)$ is the projection of x onto C, i.e., $P_C(x) = \arg\min_{y \in C} \|y - x\|_2$.

Using the subgradient rule, if $f(x) = \max_{i=1,\cdots,m} f_i(x)$, then

$$\partial f(x) = \text{conv}\left(\bigcup_{i: f_i(x) = f(x)} \partial f_i(x) \right) \tag{7.12}$$

So if $f_i(x) = f(x)$ and $g_i \in \partial f_i(x)$, then $g_i \in \partial f(x)$.

Therefore, if C_i is the farthest set from x, i.e., $f_i(x) = f(x)$ and

$$g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2} \tag{7.13}$$

then we have $g_i \in \partial f(x)$.

We apply the subgradient method with Polyak size $t_k = \frac{f(x^{(k-1)}) - f^\star}{\|g^{(k-1)}\|_2^2} = f(x^{(k-1)})$. We have that the optimal value $f^\star = 0$ and $\|g^{(k-1)}\|_2^2 = 1$.

At iteration k, suppose C_i is farthest from $x^{(k-1)}$. Then we have

$$f(x^{(k-1)}) = f_i(x^{(k-1)}) = \|x^{(k-1)} - P_{C_i}(x^{(k-1)})\|_2$$

We perform the update

$$\begin{aligned} x^{(k)} &= x^{(k-1)} - t_k g^{(k-1)} \\ &= x^{(k-1)} - f(x^{(k-1)}) \frac{x^{(k-1)} - P_{C_i}(x^{(k-1)})}{\|x^{(k-1)} - P_{C_i}(x^{(k-1)})\|_2} \\ &= x^{(k-1)} - \left(x^{(k-1)} - P_{C_i}(x^{(k-1)})\right) \\ &= P_{C_i}(x^{(k-1)}) \end{aligned} \tag{7.14}$$

For finding the intersection of two sets, this is the famous alternating projections algorithm [1]. An example of the illustration of this algorithm is shown in Figure 7.2.

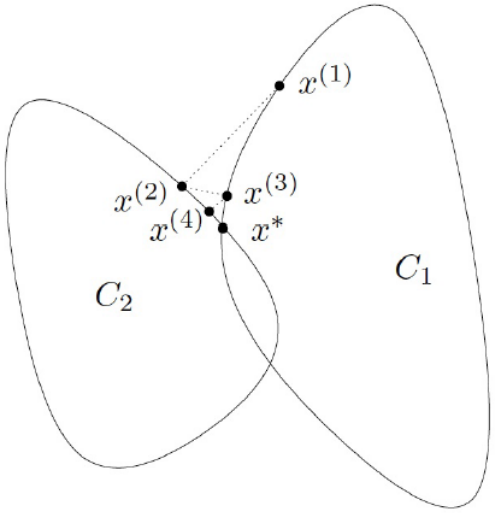

Figure 7.2: Illustration of the alternating projections algorithm (from Boyd's lecture notes).

7.2.5 Projected subgradient method

We can use the projected subgradient method to optimize a convex function f over a convex set C,

$$\min_x \ f(x) \text{ subject to } x \in C \tag{7.15}$$

The projected subgradient method is similar to the usual subgradient method, except that we project onto C at each iteration:

$$x^{(k)} = P_C(x^{(k-1)} - t_k \cdot g^{(k-1)}),\ k = 1, 2, 3, \cdots \tag{7.16}$$

Assuming we can perform this projection, we have the same convergence guarantees as the usual subgradient method, with the same step size choices.

There are many types of sets C that are easy to project onto, e.g.,

- Affine images: $\{Ax + b : x \in \mathbb{R}^n\}$
- Solution set of linear system: $\{x : Ax = b\}$
- Nonnegative orthant: $\mathbb{R}^n_+ = \{x : x \geq 0\}$
- Some norm balls: $\{x : \|x\|_p \leq 1\}$ for $p = 1, 2, \infty$
- Some simple polyhedra and simple cones

Although it is easy to write down seemingly simple set C, finding projection onto C can be very hard. For example, it is generally hard to project onto arbitrary polyhedron $C = \{x : Ax \leq b\}$.

Similarly, projected gradient descent method also works, which will be introduced in the next lecture.

7.2.6 Improving on the subgradient method

The upside of the subgradient method is that it has broad applicability. The downside is that the convergence rate $O(1/\epsilon^2)$ is slow over the problem class of convex, Lipschitz functions. We will see if we can improve the convergence rate.

Nonsmooth first-order methods are the iterative methods that update $x^{(k)}$ in the following way:

$$x^{(0)} + \text{span}\{g^{(0)}, g^{(1)}, \cdots, g^{(k-1)}\} \tag{7.17}$$

where subgradients $g^{(0)}, g^{(1)}, \cdots, g^{(k-1)}$ come from weak oracle. Weak oracle means that we are given the subgradient and do not have choices of subgradients. In practice it may be difficult to compute all the subgradients.

Theorem 7.3 ***(Nesterov)*** *For any $k \leq n - 1$ and starting point $x^{(0)}$, there is a function in the problem class such that any nonsmooth first-order method satisfies*

$$f(x^{(k)}) - f^* \geq \frac{RG}{2(1 + \sqrt{k+1})} \tag{7.18}$$

where R, G are defined in section 7.2.3. From Nesterov's theorem we can find that $f(x^{(k)}) - f^*$ has a lower bound, which gives the convergence rate $O(1/\epsilon^2)$. In summary, we cannot do better than the $O(1/\epsilon^2)$ convergence rate for the subgradient method unless we go beyond nonsmooth first-order methods.

Therefore, instead of trying to improve across the broad class of subgradient method, we will focus on minimizing composite functions of the form

$$f(x) = g(x) + h(x) \tag{7.19}$$

where g is convex and differentiable, h is convex and nonsmooth but of simple form. For example, the objective function with the lasso penalty in the regularized logistic regression problem in section 7.2.3.1 is a composite function of this form.

For a lot of problems (i.e., functions h), we can recover the $O(1/\epsilon)$ convergence rate of the gradient descent method with a simple algorithm, which has important practical consequences.

References

[1] von Neumann, "Functional operators, volume II: The geometry of orthogonal spaces", 1950.