10-725: Optimization **Fall 2013**

Lecture 7: September 17

Lecturer: Ryan Tibshirani *Scribes: Serim Park, Yiming Gu*

7.1 Recap.

The drawbacks of Gradient Methods are: (1) requires f is differentiable; (2) relatively slow convergence. Subgradient methods have better property in (1) and sometimes better in (2).

$$x^{(k)} = x^{(k-1)} - t_k g^{(k-1)}, k = 1, 2, 3, ..., \tag{7.1}$$

Subgradient of convex function f at x is any g s.t.

$$f(y) \leq f(x) + g^T(y - x), \forall\, y \tag{7.2}$$

7.1.1 Subgradient for Indicator Function

Indicator function is given as,

$$I_C(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{if } x \notin C. \end{cases}$$

Subgradient g for indication function should satisfy the following condition,

$$I_C(y) \geq I_C(x) + g^T(y - x)$$

The case for $y \notin C$ is trivial because then $I_C(y) = \infty$ and holds for $\forall g$. In case of $y \in C$, $I_C(y) = 0$ and then all possible g that satisfies $g^T x \geq g^T y$ for $\forall y$ forms a subgradient set for point x; $\partial I_C(x) = \{g : g^T(x - y) \geq 0\} = N_C(x)$, which is called a normal cone at point x (if $g^T(x - y) \geq 0$ is true, then this holds for any αg for $\alpha > 0$ too, and thus g forms a cone).

7.1.2 Subgradient Examples

Example 1. Subgradient for $f(x) = max(f_1(x), f_2(x))$ (where both $f_1(x)$ and $f_2(x)$ are differentiable) is,

$$\partial f(x) = \begin{cases} \{\nabla f_1(x)\} & \text{if } f_1(x) > f_2(x) \\ \{\nabla f_2(x)\} & \text{if } f_1(x) > f_2(x) \\ conv\{\nabla f_1(x), \nabla f_2(x)\} & \text{if } f_1(x) = f_2(x). \end{cases}$$

This immediately follows from the subgradient property $\partial f(x) = conv(\bigcup_{f_i:\text{active}} \partial f_i(x))$.

Example 2. Subgradient for $\|x\|_p = max_{y\ s.t. \|y\|_q \leq 1} y^T x$ for dual norm $\|x\|_p$ and $\|x\|_q$ is,

$$\partial \|x\|_p = \arg \max_{\|y\|_q \leq 1} y^T x = \frac{x}{\|x\|_q}.$$

Since $\partial f(x) = cl(conv(\bigcup_{f_s:\text{active}} \partial f_s(x)))$ where $f_s(x) = y^T x$ and $\partial f_s(x) = y$.

7.1.3 Extension to Constrained Optimization

$$\min_{x \in C} f(x), \qquad C = \{x : g_i(x) \leq 0, h_i(x) = 0\}$$
$$= \min_{x \in R^n} f(x) + Z_C(x)$$

x^* is a minimizer
$\Leftrightarrow 0 \in \partial f(x^*) + \partial I_C(x^*)$
$\Leftrightarrow -v \in N_C(x^*)$ for some $v \in \partial f(x^*)$ for differentiable f
$\Leftrightarrow -\nabla f(x^*) \in N_C(x^*)$
$\Leftrightarrow \nabla f(x^*)^T(y - x) \geq 0, \qquad \forall y \in C.$

Example 1.

$$\min_{x \in C} \|y - x\|^2 = \min_{x \in R^n} \frac{1}{2}\|y - x\|^2 + I_C(x).$$

$\nabla f(x) = x - y$.x is optimal if: $y - x \in N_C(x)$
$\Leftrightarrow (y - x^*)^T x* \geq (y - x*)^T u, \qquad \forall u \in C$
$\Leftrightarrow (y - x^*)^T(x^* - u) \geq 0, \qquad \forall y \in C.$

Example 2.

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2}\|y - \beta\|_2^2 + \lambda\|\beta\|_1$$

There is a unique minimizer since quadratic part is strictly convex. Let $f(\beta) = \frac{1}{2}\|y - \beta\|_2^2 + \lambda\|\beta\|_1$, then unique minimizer β^* should satisfy,

$$\partial f(\beta^*) \ni \beta^* - y + \lambda\partial\|\beta^*\|_1 = 0.$$

Soft thresholding function $\beta = S_\lambda(y)$ is a unique minimizer by checking the subgradient.

$$S_\lambda(y) = \begin{cases} y_i - \lambda & \text{if } y_i > \lambda \\ 0 & \text{if } y_i \in [-\lambda, \lambda] \\ y_i + \lambda & \text{y}_i < -\lambda. \end{cases}$$

$$\partial f(\beta) = \begin{cases} y_i - \lambda - y_i + \lambda\partial\|\beta\|_1 = 0 & \text{if } y_i > \lambda \Rightarrow \\ y_i + \lambda - y_i + \lambda\partial\|\beta\|_1 = 0 & \text{if } y_i < -\lambda \\ 0 - y_i + \lambda\partial\|\beta\|_1 = 0 & \text{if } y_i \in [-\lambda, \lambda]. \end{cases}$$

7.2 Subgradient Method

7.2.1 Subgradient method

For convex f, not necessarily differentiable, subgradient method finds the lowest value of the criterion by:

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

where $g^{(k-1)}$ is any subgradient of f at $x^{(k-1)}$. Note that it is not a decent method, that the next iterative doesn't always find the lower criterion. So we need to keep the best lowest criterion value at every iteration, i.e., $f(x^{(k)}_{\text{best}}) = \min_i f(x^{(i)})$.

7.2.2 Choosing the step size

i) Fixed step size: $t_k = t \ \ \forall k$.
However, for subgradient method, we do not typically chose fixed step size.

ii) Diminishing step size (Standard): choose t_k that is square summable but not summable.

$$\sum_{k=1}^{\infty} t_k^2 < \infty, \ \ \sum_{k=1}^{\infty} = \infty.$$

Note that step sizes are all pre-defined, not adaptively computed during the optimization iteration.

7.2.3 Convergence analysis

i) Fixed step size: Suboptimal Convergence.
For convex, not differentiable function f, if the function itself is Lipschitz with constant G such as,

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

subgradient method using fixed step size t would give a point that is suboptimal such as,

$$\lim_{k \to \infty} f(x^{(k)}_{\text{best}}) \leq f(x^*) + G^2 \frac{t}{2}.$$

In other words, the smaller the step size, the smaller the difference would be between the optimal and sub-optimal convergence.

ii) Diminishing step size that is square summable: Optimal Convergence.

$$\lim_{k \to \infty} f(x^{(k)}_{\text{best}}) = f(x^*).$$

Note that subgradient method is applicable to functions that may not look like Lipschitz, since the over the bounded set the function can be Lipschitz.

7.2.4 Polyak step size

When the optimal value $f(x^*)$ is known:

$$t_k = \frac{f(x^{(k-1)}) - f(x^*)}{\|g^{(k-1)}\|_2^2}, \ \ k = 1, 2, 3, \cdots$$

It is kind of impractical using the optimal value in the step size but it is important in that it gives the convergence rate for the subgradient method. With this choose of the step size, if we want to compute an estimate of the optimum value within the epsilon such as,

$$f(x_{\text{best}}^{(k)}) - f(x^*) \leq \epsilon$$

then we need $O(\frac{1}{\epsilon^2})$ iterations (substantially smaller than $O(\frac{1}{\epsilon})$ iterations in gradient descend method).

7.2.5 Finding intersection of convex sets

We want to find a point $x^* \in C_1 \cap \cdots \cap C_m$, where C_i is a closed convex set, using subgradient method. First define a convex optimization problem:

$$\min_{x \in C} f(x) = \min_{x \in C} \max_{i=1,\cdots,m} f_i(x)$$

Where $f_i(x)$ is the minimum distance between the point x and the set C_i, i.e. $f_i(x) = \min_{y \in C_i} \|y - x\|_2$. Note that $f_i(x)$ is convex function, and thus $f(x)$ is also a convex function.

Subgradient $g \in \partial f(x)$ would be the subgradient of $f_i(x)$, where $f(x) = f_i(x)$. Let $P_{C_i}(x)$ is the point that minimizes the distance between x and $y \in C_i$, i.e. $P_{C_i}(x) = \arg\min_{y \in C_i} \|x - y\|_2$. Then the gradient g would be:

$$g = \bigtriangledown f_i(x) = \frac{(x - P_{C_i}(x))}{\|x - P_{C_i}(x)\|_2},$$

Applying subgradient method with Polyak step size,

$$\begin{aligned} x^{(k)} &= x^{(k-1)} - f(x^{(k-1)}) \frac{(x^{(k-1)} - P_{C_i}(x^{(k-1)}))}{\|x^{(k-1)} - P_{C_i}(x^{(k-1)})\|_2} \\ &= P_{C_i}(x^{(k-1)}). \end{aligned}$$

Thus, the next update in the iteration would be finding the point that gives the minimum distance among all sets. In case of just two sets, this is equivalent to alternating projection algorithm.

7.2.6 Projected subgradient method

Projected subgradient method can be used to minimize a convex function over a convex set C:

$$\min_{x \in C} f(x)$$

It is same as usual subgradient update except we project the solution back on to C every time so that at every iteration we move in the direction of the subgradient but still lies in the set C.

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

Alternative method:

$$\min_{x \in C} f(x) = \min_{x \in \mathbb{R}^n} f(x) + I_C(x)$$

Examples for projection onto solution set C:
i) $C = \{y : \ y_i \geq \ \forall i\} \Rightarrow [P_C(x)]_i = \max\{x_i, 0\}$.

ii) $C = \{x : Ax = b\} = x_0 + \text{null}(A)$.

$$\begin{aligned} P_C(x) &= \arg\min_v \|x_0 + v - x\|_2^2 \quad \text{s.t.} \quad v \in \text{null}(A) \\ &= B(B^TB)^{-1}B^T(x - x_0) + x_0 \\ &= P_{\text{null}}(A)(x - x_0) + x_0 (P_{\text{null}}(A) = B(B^TB)^{-1}B^T) \\ &= (I - P_{\text{row}}(A))(x - x_0) + x_0 \\ &= (I - A^T(AA^T)^{-1}A)(x - x_0) \\ &= (I - A^T(AA^T)^{-1}A)(x - A^T(AA^T)^{-1}b) + A^T(AA^T)^{-1}b \\ &= x + A^T(AA^T)^{-1}(b - Ax) \quad \text{(A has full row rank)} \end{aligned}$$

Therefore, $Ax = b, x_0 = A^T(AA^T)^{-1}b$.

7.2.7 Basic Pursuit Problem

We can use projected subgradient method to solve the basic pursuit problem:

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_1 \text{ s.t. } X\beta = y.$$

In this case, the solution set is $C = \{\beta : X\beta = y\}$.
The projection on to solution set C is $P_C(\beta) = \beta + X^T(XX^T)^{-1}(y - X\beta)$ as shown in example 2 above.
Projected subgradient method performs step

$$\begin{aligned} \beta^{(k)} &= P_C(\beta^{(k-1)} - t_k g^{(k-1)}) \\ &= \beta^{(k-1)} - t_k g^{(k-1)} + X^(XX^T)^{-1}(y - X\beta^{(k-1)} + Xt_k g^{(k-1)}) \\ &= \beta^{(k-1)} - (I - X^T(XX^T)^{-1}X)t_k g^{(k-1)} \end{aligned}$$

Where, $g^{(k-1)} \in \partial\|\beta^{(k-1)}\|_1$.