Subgradient Method

Ryan Tibshirani
Convex Optimization 10-725

Last last time: gradient descent

Consider the problem

$$\min_x \; f(x)$$

for f convex and differentiable, $\mathrm{dom}(f) = \mathbb{R}^n$. Gradient descent: choose initial $x^{(0)} \in \mathbb{R}^n$, repeat:

$$\boxed{x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f(x^{(k-1)}),} \quad k = 1, 2, 3, \ldots$$

Step sizes t_k chosen to be fixed and small, or by backtracking line search

If ∇f is Lipschitz, gradient descent has convergence rate $O(1/\epsilon)$. Downsides:

- Requires f differentiable — addressed this lecture
- Can be slow to converge — addressed next lecture

Subgradient method

Now consider f convex, having $\mathrm{dom}(f) = \mathbb{R}^n$, but not necessarily differentiable

Subgradient method: like gradient descent, but replacing gradients with subgradients. Initialize $x^{(0)}$, repeat:

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

where $g^{(k-1)} \in \partial f(x^{(k-1)})$, any subgradient of f at $x^{(k-1)}$

항상 작아지지는 않는다.

Subgradient method is not necessarily a descent method, thus we keep track of best iterate $x^{(k)}_{\text{best}}$ among $x^{(0)}, \ldots, x^{(k)}$ so far, i.e.,

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

Outline

Today:

- How to choose step sizes
- Convergence analysis
- Intersection of sets
- Projected subgradient method

Step size choices

$x^{k+1} = x^k - t_k g^k$

- Fixed step sizes: $t_k = t$ all $k = 1, 2, 3, \ldots$
- Diminishing step sizes: choose to meet conditions

줄어들긴하는데
너무빠르게
줄어들지 않는다.

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

ex) $\frac{1}{k}$ $\quad \frac{1}{k^2}$ (X)

i.e., square summable but not summable. Important here that $\boxed{\text{step sizes go to zero, but not too fast}}$

There are several other options too, but key difference to gradient descent: step sizes are pre-specified, not adaptively computed

Convergence analysis

(G.D에선 ∇f가 Lipschitz)

Assume that f convex, $\mathrm{dom}(f) = \mathbb{R}^n$, and also that f is Lipschitz continuous with constant $G > 0$, i.e.,

$$|f(x) - f(y)| \leq G\|x - y\|_2 \quad \text{for all } x, y$$

$f(x) \geq f(y) + \partial f(y)^t (x-y) \Rightarrow$ $g^t(x-y) \leq |f(x) - f(y)| \leq G\|x-y\|_2$.

$\|g\|_2 \|x-y\|_2 \cos(\theta) \leq G\|x-y\|_2$

Theorem: For a fixed step size t, subgradient method satisfies

$\Rightarrow \|g\|_2 \cos(\theta) \leq G$.

$\|g\|_2^2 \cos^2(\theta) \leq G^2$ $\forall \theta$.

$\Rightarrow \|g\|_2^2 \leq G^2$

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

Theorem: For diminishing step sizes, subgradient method satisfies

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

Basic inequality

$$\left(\begin{array}{l} f(x) \ge f(x^{(k-1)}) + g^{(k-1)T}(x - x^{(k-1)}) \\ -(f(x^{(k-1)}) - f(x^*)) \ge -g^{(k-1)T}(x^{(k-1)} - x^*) \end{array} \right)$$

Can prove both results from same basic inequality. Key steps:

- Using definition of subgradient,

$$= \|x^{(k-1)} - t_k g^{(k-1)} - x^*\|_2^2 = \|x^{(k-1)} - x^*\|_2^2 - 2t_k g^{(k-1)T}(x^{(k-1)} - x^*) + t_k^2 \|g^{(k-1)}\|_2^2$$

$$\underbrace{\|x^{(k)} - x^\star\|_2^2}_{a_k} \le$$

$$\underbrace{\|x^{(k-1)} - x^\star\|_2^2}_{a_{k-1}} - 2t_k\big(f(x^{(k-1)}) - f(x^\star)\big) + t_k^2 \|g^{(k-1)}\|_2^2$$

$a_k \le a_{k-1} + b_k \Rightarrow a_k - a_{k-1} \le b_k$

- Iterating last inequality,

$$\left(\begin{array}{l} \sum (a_k - a_{k-1}) \le \sum b_k \\ a_k \le a_0 + \sum b_k \end{array} \right)$$

$$\|x^{(k)} - x^\star\|_2^2 \le$$

$$\|x^{(0)} - x^\star\|_2^2 - 2\sum_{i=1}^{k} t_i\big(f(x^{(i-1)}) - f(x^\star)\big) + \sum_{i=1}^{k} t_i^2 \|g^{(i-1)}\|_2^2$$

starting 고정값   모르지만 고정값

- Using $\|x^{(k)} - x^\star\|_2 \geq 0$, and letting $R = \|x^{(0)} - x^\star\|_2$,

$$0 \leq R^2 - 2\sum_{i=1}^{k} t_i\big(f(x^{(i-1)}) - f(x^\star)\big) + G^2 \sum_{i=1}^{k} t_i^2$$

$\sum_{i=1}^{k} t_i(f(x_{best}^{(k)}) - f(x^*)) \le \sum t_i(f(x^{(i-1)}) - f(x^*))$

- Introducing $f(x_{\text{best}}^{(k)}) = \min_{i=0,\ldots,k} f(x^{(i)})$, and rearranging, we have the basic inequality

$$f(x_{\text{best}}^{(k)}) - f(x^\star) \leq \frac{R^2 + G^2 \sum_{i=1}^{k} t_i^2}{2\sum_{i=1}^{k} t_i} \to 0$$

수렴

발산

For different step sizes choices, convergence results can be directly obtained from this bound, e.g., previous theorems follow

Convergence rate

The basic inequality tells us that after k steps, we have

$$f(x^{(k)}_{\text{best}}) - f(x^\star) \le \frac{R^2 + G^2 \sum_{i=1}^k t_i^2}{2 \sum_{i=1}^k t_i}$$

With fixed step size t, this gives

$\le \frac{\epsilon}{2}$ $\le \frac{\epsilon}{2}$

$$f(x^{(k)}_{\text{best}}) - f^\star \le \underline{\left(\frac{R^2}{2kt} + \frac{G^2 t}{2}\right)}$$ 상계 $= \epsilon$.

For this to be $\le \epsilon$, let's make each term $\le \epsilon/2$. So we can choose $t = \epsilon/G^2$, and $k = R^2/t \cdot 1/\epsilon = R^2G^2/\epsilon^2$

That is, subgradient method has convergence rate $O(1/\epsilon^2)$ ... note that this is <u>slower than $O(1/\epsilon)$ rate of gradient descent</u> 느리다..

Example: regularized logistic regression

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

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

This is a smooth and convex function with

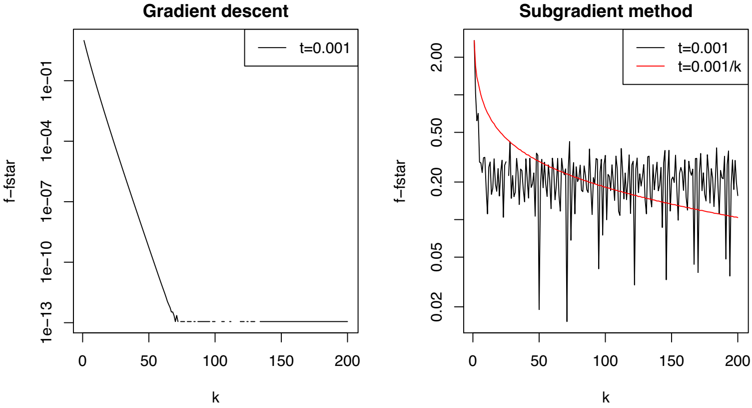
$$\nabla f(\beta) = \sum_{i=1}^{n} \big(y_i - p_i(\beta) \big) x_i$$

where $p_i(\beta) = \exp(x_i^T \beta)/(1 + \exp(x_i^T \beta))$, $i = 1, \ldots, n$. Consider the regularized problem:

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

where $P(\beta) = \|\beta\|_2^2$, ridge penalty; or $P(\beta) = \|\beta\|_1$, lasso penalty

Ridge: use gradients; lasso: use subgradients. Example here has $n = 1000$, $p = 20$:



Step sizes hand-tuned to be favorable for each method (of course comparison is imperfect, but it reveals the convergence behaviors)

Polyak step sizes

Polyak step sizes: when the optimal value $f^\star$ is known, take

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

Can be motivated from first step in subgradient proof:

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

→ goes zero

Polyak step size minimizes the right-hand side

With Polyak step sizes, can show subgradient method converges to optimal value. Convergence rate is still $O(1/\epsilon^2)$

Example: intersection of sets

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

First define

$$\begin{aligned} f_i(x) &= \mathrm{dist}(x, C_i), \quad i = 1, \ldots, m \\ f(x) &= \max_{i=1,\ldots,m} \; f_i(x) \end{aligned}$$

and now solve

$$\min_x \; f(x)$$

Check: is this convex?

Note that $f^\star = 0 \iff x^\star \in C_1 \cap \cdots \cap C_m$

Recall the distance function $\mathrm{dist}(x, C) = \min_{y \in C} \|y - x\|_2$. Last time we computed its gradient

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

where $P_C(x)$ is the projection of x onto C

Also recall subgradient rule: if $f(x) = \max_{i=1,\ldots,m} f_i(x)$, then

$$\partial f(x) = \mathrm{conv}\Bigg(\bigcup_{i : f_i(x) = f(x)} \partial f_i(x) \Bigg)$$

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

Put these two facts together for intersection of sets problem, with $f_i(x) = \mathrm{dist}(x, C_i)$: if C_i is farthest set from x (so $f_i(x) = f(x)$), and

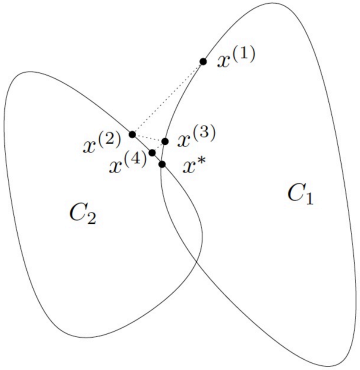
$$g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2}$$

then $g_i \in \partial f(x)$

Now apply subgradient method, with Polyak size $t_k = f(x^{(k-1)})$. At iteration k, with C_i farthest from $x^{(k-1)}$, we perform update

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

For two sets, this is the famous alternating projections algorithm[1], i.e., just keep projecting back and forth

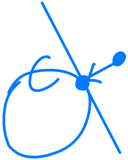


(From Boyd's lecture notes)

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

Projected subgradient method

To optimize a convex function f over a convex set C,



$$\min_x \; f(x) \quad \text{subject to} \quad x \in C$$

we can use the projected subgradient method. Just like the usual subgradient method, except we project onto C at each iteration:

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

만약 C에 없으면 가장 가까운 곳으로 projection

Assuming we can do this projection, we get the same convergence guarantees as the usual subgradient method, with the same step size choices

(5) Supporting Hyperplane theorem

$K = \alpha_1 b_1 + \alpha_2 b_2 + \sim \alpha_n b_n$

$= B\alpha$

$P_C(X) = P_K(X) = B(B^t B)^{-1} B^t X$

What sets C are easy to project onto? Lots, 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

Warning: it is easy to write down seemingly simple set C, and P_C can turn out to be very hard! E.g., generally hard to project onto arbitrary polyhedron $C = \{x : Ax \leq b\}$

Note: projected gradient descent works too, more next time ...

Can we do better?

Upside of the subgradient method: broad applicability. Downside: $O(1/\epsilon^2)$ convergence rate over problem class of convex, Lipschitz functions is really slow

Nonsmooth first-order methods: iterative methods updating $x^{(k)}$ in

$$x^{(0)} + \mathrm{span}\{g^{(0)}, g^{(1)}, \ldots, g^{(k-1)}\}$$

where subgradients $g^{(0)}, g^{(1)}, \ldots, g^{(k-1)}$ come from weak oracle

Theorem (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

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

Improving on the subgradient method

In words, we cannot do better than the $O(1/\epsilon^2)$ rate of subgradient method (unless we go beyond nonsmooth first-order methods)

So instead of trying to improve across the board, we will focus on minimizing composite functions of the form

미분되는건 GD로 빨리하고
미분안되는것만 sub GD로 최적화

$$f(x) = g(x) + h(x)$$

where g is convex and differentiable, h is convex and nonsmooth but "simple" ⇒ Proximal Gradient descent.

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