

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, $\text{dom}(f) = \mathbb{R}^n$. **Gradient descent:**
choose initial $x^{(0)} \in \mathbb{R}^n$, repeat:

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

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 $\text{dom}(f) = \mathbb{R}^n$, but not necessarily differentiable

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

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

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

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

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

Outline

Today:

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

Step size choices

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

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

i.e., square summable but not summable. Important here that 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

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

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

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

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

Theorem: For diminishing step sizes, subgradient method satisfies

$$\lim_{k \rightarrow \infty} f(x_{\text{best}}^{(k)}) = f^*$$

Basic inequality

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

- Using definition of subgradient,

$$\begin{aligned} \|x^{(k)} - x^*\|_2^2 &\leq \\ &\|x^{(k-1)} - x^*\|_2^2 - 2t_k(f(x^{(k-1)}) - f(x^*)) + t_k^2\|g^{(k-1)}\|_2^2 \end{aligned}$$

- Iterating last inequality,

$$\begin{aligned} \|x^{(k)} - x^*\|_2^2 &\leq \\ \|x^{(0)} - x^*\|_2^2 - 2 \sum_{i=1}^k t_i (f(x^{(i-1)}) - f(x^*)) &+ \sum_{i=1}^k t_i^2 \|g^{(i-1)}\|_2^2 \end{aligned}$$

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

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

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

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

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

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

With fixed step size t , this gives

$$f(x_{\text{best}}^{(k)}) - f^* \leq \frac{R^2}{2kt} + \frac{G^2 t}{2}$$

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

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

Example: regularized logistic regression

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

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

This is a smooth and convex function with

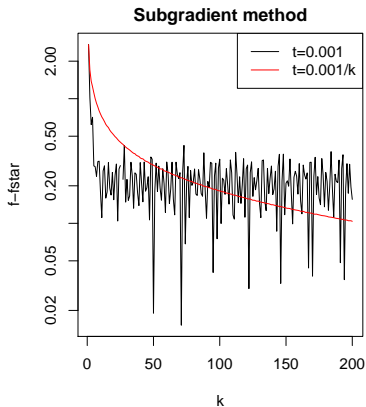
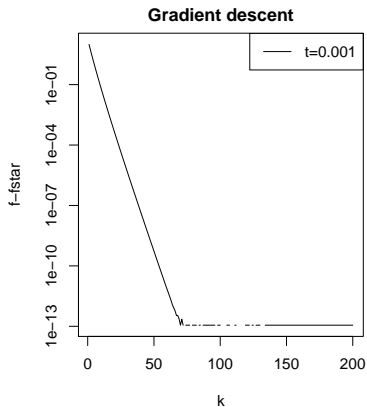
$$\nabla f(\beta) = \sum_{i=1}^n (y_i - p_i(\beta)) x_i$$

where $p_i(\beta) = \exp(x_i^T \beta) / (1 + \exp(x_i^T \beta))$, $i = 1, \dots, n$. 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^* is known, take

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

Can be motivated from first step in subgradient proof:

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

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 $x^* \in C_1 \cap \cdots \cap C_m$, i.e., find a point in intersection of closed, convex sets C_1, \dots, C_m

First define

$$f_i(x) = \text{dist}(x, C_i), \quad i = 1, \dots, m$$
$$f(x) = \max_{i=1, \dots, m} f_i(x)$$

and now solve

$$\min_x f(x)$$

Check: is this convex?

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

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

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

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

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

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

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 $f_i(x) = \text{dist}(x, C_i)$: if C_i is farthest set from x (so $f_i(x) = f(x)$), and

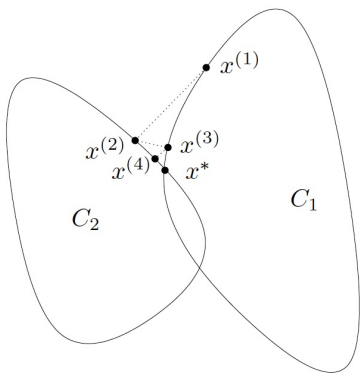
$$g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2}$$

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** algorithm¹, i.e., just keep projecting back and forth



(From Boyd's lecture notes)

¹von 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 ,

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

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

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

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

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

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

where subgradients $g^{(0)}, g^{(1)}, \dots, g^{(k-1)}$ 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

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

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

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

where g is convex and differentiable, h is convex and nonsmooth but “simple”

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

References and further reading

- S. Boyd, Lecture notes for EE 264B, Stanford University, Spring 2010-2011
- Y. Nesterov (1998), “Introductory lectures on convex optimization: a basic course”, Chapter 3
- B. Polyak (1987), “Introduction to optimization”, Chapter 5
- L. Vandenberghe, Lecture notes for EE 236C, UCLA, Spring 2011-2012