Subgradient method

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

Recall gradient descent

We want to solve

$$\min_{x \in \mathbb{R}^n} f(x),$$

for f convex and differentiable

Gradient descent: choose initial $x^{(0)} \in \mathbb{R}^n$, repeat

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

If ∇f Lipschitz, gradient descent has convergence rate $O(1/k)$

Downsides:

- Requires f differentiable $\leftarrow$ this lecture
- Can be slow to converge $\leftarrow$ next lecture

Subgradient method

Given convex $f : \mathbb{R}^n \to \mathbb{R}$, not necessarily differentiable

Subgradient method: just like gradient descent, but replacing gradients with subgradients. I.e., initialize $x^{(0)}$, then repeat

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

where $g^{(k-1)}$ is any subgradient of f at $x^{(k-1)}$

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

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

Step size choices

- Fixed step size: $t_k = t$ all $k = 1, 2, 3, \ldots$
- Diminishing step size: choose t_k to satisfy

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

i.e., square summable but not summable

Important that step sizes go to zero, but not too fast

Other options too, but important difference to gradient descent: all step sizes options are pre-specified, not adaptively computed

Convergence analysis

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex, 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

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

Theorem: For diminishing step sizes, subgradient method satisfies

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

Basic inequality

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

- Using definition of subgradient,

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

- Iterating last inequality,

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

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

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

- Introducing $f(x^{(k)}_{\mathrm{best}}) = \min_{i=0,\ldots k} f(x^{(i)})$, and rearranging,

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

  We call this our basic inequality

For different step sizes choices, convergence results can be directly obtained from this basic inequality. E.g., theorems for fixed and diminishing step sizes follow

Polyak step sizes

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

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

Can be motivated from first step in subgradient proof:

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

Polyak step size minimizes the right-hand side

With this choice of step size, error complexity after k iterations is

$$f(x^{(k)}_{\text{best}}) - f(x^\star) = O(1/\sqrt{k})$$

I.e., to get $f(x^{(k)}_{\text{best}}) - f(x^\star) \leq \epsilon$, need $O(1/\epsilon^2)$ iterations

Intersection of sets

Example (from Boyd's lecture notes): suppose we want to find $x^\star \in C_1 \cap \ldots \cap C_m$, i.e., find point in intersection of closed, convex sets $C_1, \ldots C_m$

First define

$$f(x) = \max_{i=1,\ldots m} \operatorname{dist}(x, C_i),$$

and now solve

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

Note that $f(x^\star) = 0 \;\Rightarrow\; x^\star \in C_1 \cap \ldots \cap C_m$

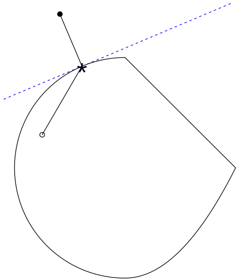
Recall distance to set C,

$$\operatorname{dist}(x, C) = \min\{\|x - u\|_2 : u \in C\}$$

For closed, convex C, there is a unique point minimizing $\|x - u\|_2$ over $u \in C$. Denoted $u^\star = P_C(x)$, so $\operatorname{dist}(x, C) = \|x - P_C(x)\|_2$

Recall optimality characterization:
$u^\star = P_C(x)$ if and only if

$\langle x - u^\star, u^\star - u \rangle \geq 0$ for all $u \in C$



Consider $h(x) = \operatorname{dist}(x, C)$. For $x \notin C$,

$$\nabla h(x) = \frac{x - P_C(x)}{\|x - P_C(x)\|_2}$$

Follows from definition of subgradients, and above characterization

Now write $f_i(x) = \mathrm{dist}(x, C_i)$ for $i = 1, \ldots m$, and

$$f(x) = \max_{i=1,\ldots m} f_i(x)$$

We know how to compute subgradient $g \in \partial f(x)$: first find set C_i with $f_i(x) = f(x)$, then let

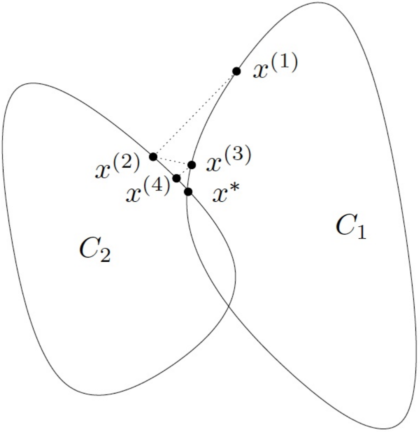
$$g = \nabla f_i(x) = (x - P_{C_i}(x))/\|x - P_{C_i}(x)\|_2$$

Can apply subgradient method, with Polyak step $t_k = f(x^{(k-1)})$

At iteration k, we find C_i so that $x^{(k-1)}$ is farthest from C_i. Then 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 exactly the famous alternating projections algorithm, i.e., just keep projecting back and forth



(From Boyd's notes)

Projected subgradient method

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

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

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

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

Assuming we can do this projection, 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 $C = \{Ax + b : x \in \mathbb{R}^n\}$
- Solution set of linear system $C = \{x \in \mathbb{R}^n : Ax = b\}$
- Nonnegative orthant $C = \{x \in \mathbb{R}^n : x \geq 0\} = \mathbb{R}^n_+$
- Norm balls $C = \{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$, 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., it is generally hard to project onto solution set of arbitrary linear inequalities, i.e, arbitrary polyhedron $C = \{x \in \mathbb{R}^n : Ax \leq b\}$

Basis pursuit

Recall the basis pursuit problem

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_1 \quad \text{subject to} \quad X\beta = y$$

Here $C = \{\beta : X\beta = y\}$ and $P_C(\beta) = \beta + X^T(XX^T)^{-1}(y - X\beta)$ (assuming that $\mathrm{rank}(X) = n$)

Hence, projected subgradient method repeats

$$\begin{aligned} \beta^{(k)} &= P_C\big(\beta^{(k-1)} - t_k s^{(k-1)}\big) \\ &= \beta^{(k-1)} - t_k\big(I - X^T(XX^T)^{-1}X\big)s^{(k-1)} \end{aligned}$$

where $s^{(k-1)} \in \partial\|\beta^{(k-1)}\|_1$, i.e.,

$$s_i^{(k-1)} \in \begin{cases} \{\mathrm{sign}(\beta^{(k-1)})\} & \beta_i^{(k-1)} \neq 0 \\ [-1, 1] & \text{otherwise} \end{cases}$$

Can we do better?

Strength of subgradient method: broad applicability. Downside: $O(1/\sqrt{k})$ convergence rate over problem class of convex, Lipschitz functions is really slow

Nonsmooth first-order methods: iterative methods that start with $x^{(0)}$ and update $x^{(k)}$ in

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

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

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

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

Improving on the subgradient method

So we cannot generically do better than the subgradient method, unless we go beyond nonsmooth first-order methods

Instead of trying to better 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 $O(1/k)$ rate of gradient descent with a natural algorithm, having big practical consequences

References

- S. Boyd, Lecture Notes for EE 264B, Stanford University, Spring 2010-2011
- Y. Nesterov (2004), "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