

Subgradient Method

Ryan Tibshirani

Convex Optimization 10-725/36-725

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 Lipschitz, gradient descent has convergence rate $O(1/\epsilon)$

Downsides:

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

Subgradient method

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

Subgradient method: like gradient descent, but replacing gradients with subgradients. I.e., 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, so 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
- Stochastic 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 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 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 bound, and convergence rate

Letting $R = \|x^{(0)} - x^*\|_2$, after k steps, we have the basic bound

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

Previous theorems follow from this. With fixed step size t , we have

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

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

I.e., subgradient method has convergence rate $O(1/\epsilon^2)$... compare this to $O(1/\epsilon)$ rate of gradient descent

Example: regularized logistic regression

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

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

This is a smooth and convex, with

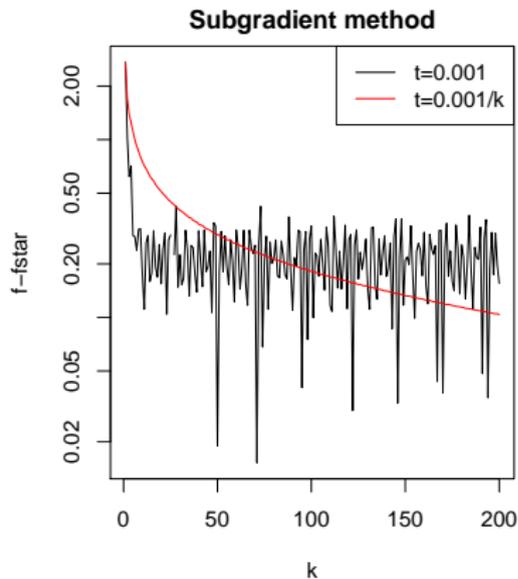
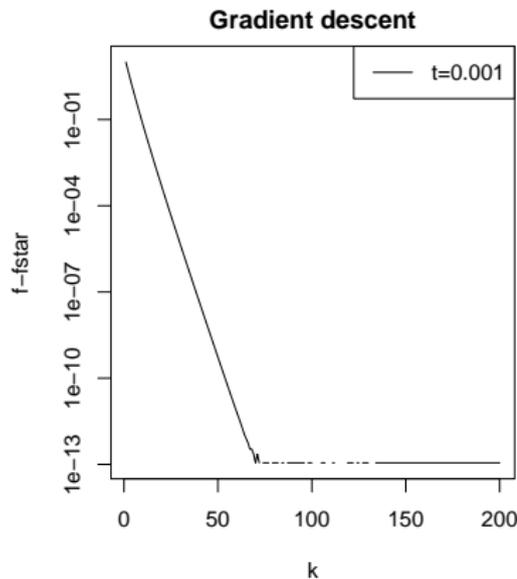
$$\nabla f(\beta) = \sum_{i=1}^n (y_i - p_i(\beta)) x_i$$

where $p_i(\beta) = \exp(x_i^T \beta) / (1 + \exp(x_i^T \beta))$, $i = 1, \dots, n$. We will consider the regularized problem:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \cdot P(\beta)$$

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

Ridge problem: use gradients; lasso problem: use subgradients.
Data example with $n = 1000$, $p = 20$:



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 $x^* \in C_1 \cap \dots \cap C_m$, 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

$$\min_x f(x)$$

Note that $f^* = 0 \Rightarrow x^* \in C_1 \cap \dots \cap C_m$. Check: is this problem convex?

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

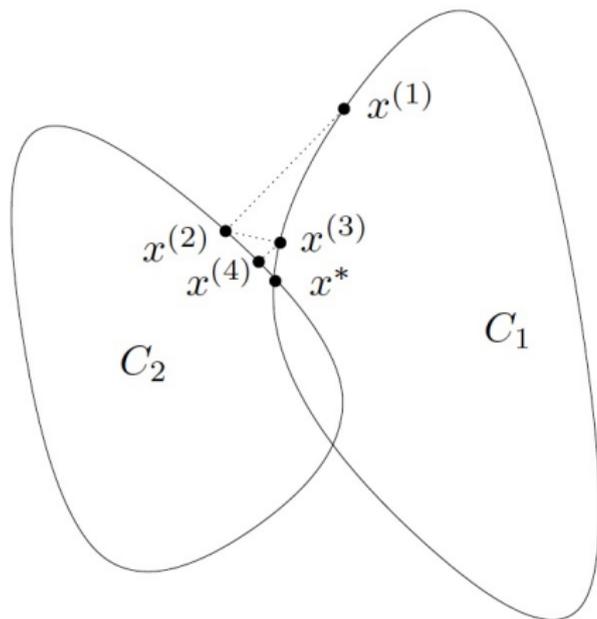
$$g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2}$$

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



(From Boyd's lecture notes)

Stochastic subgradient method

Consider sum of functions

$$\min_x \sum_{i=1}^m f_i(x)$$

Recall that $\partial \sum_{i=1}^m f_i(x) = \sum_{i=1}^m \partial f_i(x)$, and subgradient method would repeat

$$x^{(k)} = x^{(k-1)} - t_k \cdot \sum_{i=1}^m g_i^{(k-1)}, \quad k = 1, 2, 3, \dots$$

where $g_i^{(k-1)} \in \partial f_i(x^{(k-1)})$. In comparison, **stochastic subgradient method** (or incremental subgradient) repeats

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

where $i_k \in \{1, \dots, m\}$ is some chosen index at iteration k

Stochastic gradient descent: special case when f_i , $i = 1, \dots, m$ are differentiable, so $g_i^{(k-1)} = \nabla f_i(x^{(k-1)})$

Two rules for choosing index i_k at iteration k :

- **Cyclic rule:** choose $i_k = 1, 2, \dots, m, 1, 2, \dots, m, \dots$
- **Randomized rule:** choose $i_k \in \{1, \dots, m\}$ uniformly at random

Randomized rule is more common in practice

What's the difference between stochastic and usual (called batch) methods? Computationally, m stochastic steps \approx one batch step. But what about progress?

Consider smooth cyclic rule, for simplicity: here we cycle through a descent step on each f_i individually. After m steps (i.e., one cycle), assuming constant step sizes $\alpha_{k+1} = \dots \alpha_{k+m}$, the update is

$$x^{(k+m)} = x^{(k)} - t \sum_{i=1}^m \nabla f_i(x^{(k+i-1)})$$

One batch step is instead

$$x^{(k+1)} = x^{(k)} - t \sum_{i=1}^m \nabla f_i(x^{(k)})$$

so difference in direction is $\sum_{i=1}^m [\nabla f_i(x^{(k+i-1)}) - \nabla f_i(x^{(k)})]$

We can believe that the stochastic method still converges if $\nabla f(x)$ doesn't vary wildly with x

Convergence of stochastic methods

Assume each f_i , $i = 1, \dots, m$ is convex and Lipschitz with constant $G > 0$

For fixed step sizes $t_k = t$, $k = 1, 2, 3, \dots$, cyclic and randomized¹ stochastic subgradient methods both satisfy

$$\lim_{k \rightarrow \infty} f(x_{\text{best}}^{(k)}) \leq f^* + 5m^2G^2t/2$$

Note: mG can be viewed as Lipschitz constant for whole function $\sum_{i=1}^m f_i$, so this is comparable to batch bound

For diminishing step sizes, cyclic and randomized methods satisfy

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

¹For randomized rule, results hold with probability 1

How about convergence rates? This is where things get interesting

Recall that the batch subgradient method rate was $O(G_{\text{batch}}^2/\epsilon^2)$, where Lipschitz constant G_{batch} is for whole function

- Cyclic rule: iteration complexity is $O(m^3G^2/\epsilon^2)$. Therefore number of cycles needed is $O(m^2G^2/\epsilon^2)$, comparable to batch
- Randomized rule²: iteration complexity is $O(m^2G^2/\epsilon^2)$. Thus number of random cycles needed is $O(mG^2/\epsilon^2)$, **reduced by a factor of m !**

This is a convincing reason to use randomized stochastic methods, for problems where m is big

²For randomized rule, result holds in expectation, i.e., bound is on expected number of iterations

Example: stochastic logistic regression

Back to the logistic regression problem:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) = \sum_{i=1}^n \underbrace{\left(-y_i x_i^T \beta + \log(1 + \exp(x_i^T \beta)) \right)}_{f_i(\beta)}$$

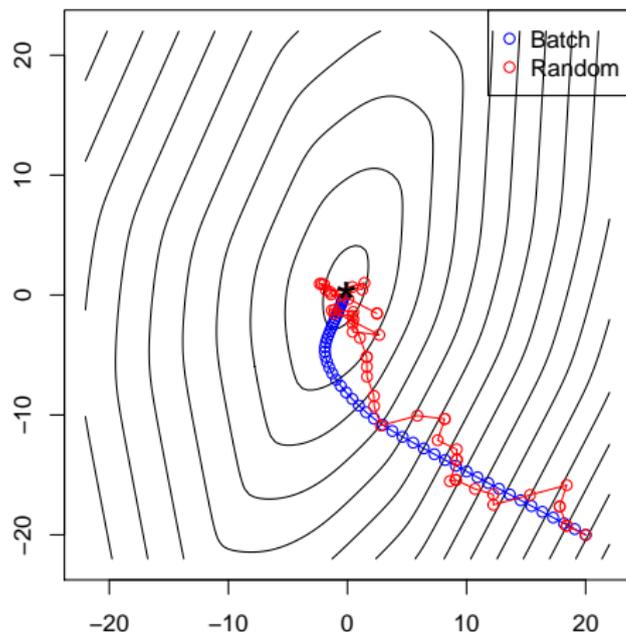
The gradient computation $\nabla f(\beta) = \sum_{i=1}^n (y_i - p_i(\beta)) x_i$ is doable when n is moderate, but **not when $n \approx 500$ million**. Recall:

- One batch update costs $O(np)$
- One stochastic update costs $O(p)$

So clearly, e.g., 10K stochastic steps are much more affordable

Also, we often take fixed step size for stochastic updates to be $\approx n$ what we use for batch updates. (Why?)

The “classic picture”:



Blue: batch steps, $O(np)$

Red: stochastic steps, $O(p)$

Rule of thumb for stochastic methods:

- generally thrive far from optimum
- generally struggle close to optimum

(More on stochastic methods later in the course ...)

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 that start with $x^{(0)}$ and update $x^{(k)}$ 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

- D. Bertsekas (2010), “Incremental gradient, subgradient, and proximal methods for convex optimization: a survey”
- 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