Subgradient Method

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

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

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

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

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

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 $x^{(k)}_{\text{best}}$ 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
- Stochastic subgradient method

Step size choices

- Fixed step sizes: $t_k = t$ all $k = 1, 2, 3, \ldots$
- 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: step sizes are typically pre-specified, not adaptively computed

Convergence analysis

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

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

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

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

Theorem: For diminishing step sizes, subgradient method satisfies

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

Basic inequality

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

- Using definition of subgradient,

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

- Iterating last inequality,

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

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

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

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

Convergence rate

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

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

With fixed step size t, this gives

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

For this to be $\le \epsilon$, let's make each term $\le \epsilon/2$. 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 $i = 1, \ldots n$, consider the logistic regression loss:

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

This is a smooth and convex, with

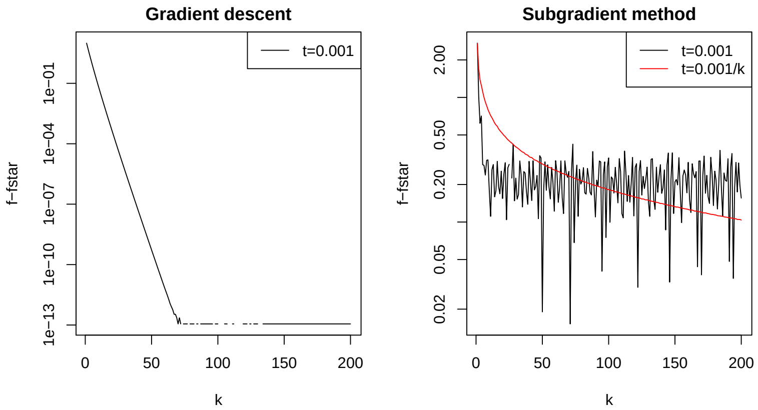
$$\nabla f(\beta) = \sum_{i=1}^n \big(y_i - p_i(\beta) \big) x_i$$

where $p_i(\beta) = \exp(x_i^T \beta)/(1 + \exp(x_i^T \beta))$, $i = 1, \ldots n$. We will 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 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 $f^\star$ 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:

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

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 $x^\star \in C_1 \cap \ldots \cap C_m$, 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

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

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

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

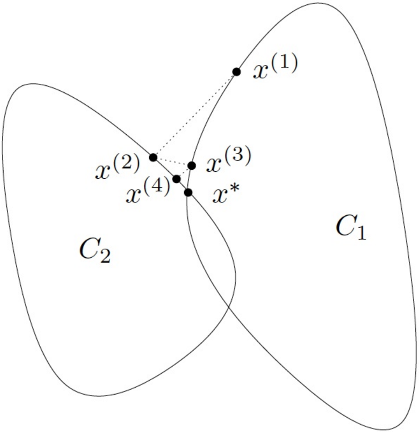
$$g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2}$$

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



(From Boyd's lecture notes)

Projected subgradient method

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

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

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

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

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 ...

Stochastic subgradient method

Similar to our setup for stochastic gradient descent. Consider sum of convex functions

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

Stochastic subgradient method repeats:

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

where $i_k \in \{1, \ldots m\}$ is some chosen index at iteration k, chosen by either by the random or cyclic rule, and $g_i^{(k-1)} \in \partial f_i(x^{(k-1)})$ (this update direction is used in place of the usual $\sum_{i=1}^m g_i^{(k-1)}$)

Note that when each f_i, $i = 1, \ldots, m$ is differentiable, this reduces to stochastic gradient descent (SGD)

Convergence of stochastic methods

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

For fixed step sizes $t_k = t$, $k = 1, 2, 3, \ldots$, cyclic and randomized[1] stochastic subgradient methods both satisfy

$$\lim_{k \to \infty} f(x_{\mathsf{best}}^{(k)}) \le f^\star + 5m^2G^2t/2$$

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

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

[1] For randomized rule, results hold with probability 1

How about convergence rates? This is where things get interesting

Looking back carefully, 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 rule[2]: 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

[2]For 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 (now we're talking SGD):

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

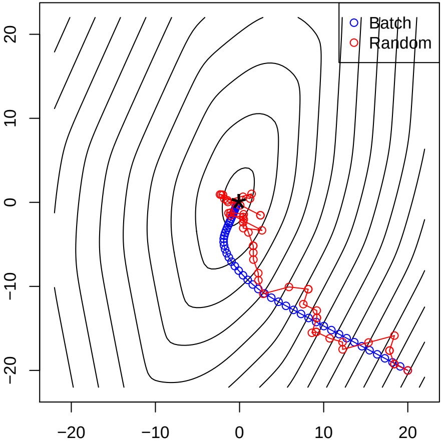
The gradient computation $\nabla f(\beta) = \sum_{i=1}^n \big(y_i - p_i(\beta)\big) x_i$ is doable when n is moderate, but not when $n \approx 500$ million. Recall:

- 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

(Even 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 updating $x^{(k)}$ in

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

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

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

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