10-425/625: Introduction to Convex Optimization (Fall 2023)

Lecture 10: Subgradients / The Subgradient Method

Instructor:[1] *Matt Gormley* *September 25, 2023*

10.1 Subgradients

10.1.1 Some basic subgradient calculus

In many ways subgradients behave just like gradients provided you interpret "set valued" operations correctly. Here are a couple of facts (you will explore a couple more on your HW):

1. **Scaling:** $\partial(af) = a \times \partial f(x)$, when $a > 0$.
2. **Addition:** $\partial(f_1 + \ldots + f_m) = \partial f_1 + \ldots + \partial f_m$.
3. **Affine composition:** if $g(x) = f(Ax + b)$, then
$$\partial g(x) = A^T \partial f(Ax + b)$$
4. **Finite pointwise maximum:** if $f(x) = \max_{i=1,\ldots,m} f_i(x)$, then
$$\partial f(x) = \text{conv}\left(\bigcup_{i: f_i(x) = f(x)} \partial f_i(x) \right)$$
convex hull of union of subdifferentials of active functions at x

10.1.2 Soft Thresholding

A closely related optimization problem to the LASSO, but one where we can in fact find the optimal solution in closed form using the optimality conditions is the following program:

$$x^* = \arg\min_x \frac{1}{2}\|y - x\|_2^2 + \lambda\|x\|_1.$$

[1]These notes were originally written by Siva Balakrishnan for 10-725 Spring 2023 (original version: here) and were edited and adapted for 10-425/625.

Now, the optimality conditions tell us that x^* is optimal iff

$$0 \in -(y - x^*) + \lambda \text{sign}(x^*),$$

or equivalently,

$$(y - x^*) \in \lambda \text{sign}(x^*).$$

For just a single element x_j^*, we have that

$$0 \in -(y_j - x_j^*) + \lambda \text{sign}(x_j^*),$$

So we observe that,

$$x_j^* = \begin{cases} \text{if } x_j^* > 0, & x_j^* = y_j - \lambda \\ \text{if } x_j^* < 0, & x_j^* = y_j + \lambda \\ \text{if } x_j^* = 0, & x_j^* = 0 \end{cases}$$

Lets define the soft-thresholding operation for a scalar y_i, and $\lambda > 0$,

$$S_\lambda(y_i) = \begin{cases} y_i - \lambda & \text{if } y_i > \lambda \\ 0 & \text{if } -\lambda \leq y_i \leq \lambda \\ y_i + \lambda & \text{if } y_i < -\lambda. \end{cases}$$

Now, if you stared at the above optimality conditions you would see that,

$$x^* = S_\lambda(y),$$

where we apply soft-thresholding element-wise satisfies the optimality conditions (i.e. is an optimal solution). It's not difficult to convince yourself that since the objective is strictly convex it is also the unique optimal solution.

This idea that we can efficiently, in closed-form, solve this simplified optimization problem will be useful to us when we discuss proximal methods.

10.1.3 LASSO Optimality Conditions

Now, we'll briefly look at the optimality conditions for the LASSO program, and try to see another example where optimality conditions are a useful lens for understanding an optimization problem.

The LASSO program is simply least squares with an ℓ_1 penalty, i.e. we solve:

$$x^* = \arg\min_{x \in \mathbb{R}^d} \frac{1}{2}\|b - Ax\|_2^2 + \lambda\|x\|_1.$$

Now, by our optimality conditions we know that any solution must have zero subgradient, i.e. it must be the case that,

$$0 \in -A^T(b - Ax^*) + \lambda \text{sign}(x^*).$$

Coordinatewise this gives us the condition that,

$$A_j^T(b - Ax^*) \in \lambda \text{sign}(x_j^*).$$

One intuitive fact that one can glean from this is that at optimum if $x_j^* = 0$ then we know that, $|A_j^T(b - Ax^*)| \leq \lambda$, i.e. roughly the 0s of the LASSO solution will correspond to variables A_j which have a small correlation with the residuals $(b - Ax^*)$. You can with a bit more effort glean many other nice factoids about the LASSO solution from these conditions – but unfortunately (unlike ridge, or ordinary least squares) one cannot solve them in closed form to solve the LASSO.

10.1.4 Using Subgradients to Derive Constrained, Non-Smooth Optimality Conditions

We have already seen one way to derive the optimality conditions for non-smooth constrained optimization. Now we will see a different way, reducing it to an unconstrained problem where we know the optimality conditions.

Observe that,

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

is equivalent to the unconstrained problem:

$$\min_{x \in \mathbb{R}^d} f(x) + \mathbb{I}_C(x).$$

So x^* is optimal for this program iff $0 \in \partial(f(x^*) + \mathbb{I}_C(x^*))$, i.e. $0 \in \partial f(x^*) + \partial I_C(x^*) = \partial f(x^*) + N_C(x^*)$, which is exactly the condition we derived before.

10.2 Subgradient Method

Exactly like gradient descent, but we replace gradients by subgradients, i.e. we initialize at x^0, and iterate:

$$x^{t+1} = x^t - \eta_t g_{x^t},$$

where $g_{x^t} \in \partial f(x^t)$, is any subgradient of f at x^t.

Since it often will not be a descent method, we'll usually keep track of the best iterate found so far, and output:

$$x^{\text{best}} = \arg \min_{t \in \{0,\ldots,k\}} f(x^t).$$

It is common to use the term subgradient method (instead of subgradient descent) since often the method is not a descent method (i.e. in most cases where we apply the method, and for reasonable choices of the step-size, function values can go up between iterations).

It's a bit beyond the scope of this course so I'll just mention this in passing:

1. Subgradient directions need not be descent directions, i.e. they can be directions such that for any step-size $\eta > 0$, the function value increases. We already know the example of $f(x) = |x|$ where at the point 0, -1 (for instance) is a valid subgradient, but moving in that direction will only increase the objective function.

   A slightly more interesting example is to take the function $f(x) = |x_1| + 10|x_2|$, then at the point $(1, 0)$ (which is not the minimizer), the vector $(1, 10)$ is a valid subgradient, but for any $\eta > 0$ our next iterate would be at $(1 - \eta, -10\eta)$ which would be a strictly worse point (i.e. the negative subgradient direction is a direction of increase for the function).

2. It will turn out to be the case that if you were very clever in your choice of subgradient (and particularly, picked a direction in $\partial f(x^t)$ which has minimum ℓ_2 norm) then these special subgradient directions are descent directions, and one could hope to design a subgradient descent method (i.e. by carefully choosing the step-sizes).
3. Finding the full sub-differential and then finding the minimum norm element is hard in a lot of examples. So our goal will instead be to show that a naive algorithm that simply follows an arbitrary subgradient at each iteration will in fact converge to a globally optimal solution, under some assumptions, provided we are careful about step-size choice.

10.3 Step Sizes

One of the basic differences between GD and the subgradient method is that we typically use different methods for choosing the step-size.

Since, as we discussed above subgradient directions need not be descent directions, in practice people don't usually use adaptive step-sizes (i.e. backtracking line searches) since those are based on finding step-sizes which ensure "sufficient descent".

Instead people typically use:

1. **Fixed Step-Sizes:** If we're going to run k iterations of the subgradient method, one can use a fixed step-size η for all iterations. As will be more clear when we look at some convergence proofs a typical choice will be $\eta \sim 1/\sqrt{k}$, i.e. if we're going to do many steps we'll take a smaller step-size.
2. **Step-Size Schedules:** In practice, it is more common to use decaying step sizes, e.g. $\eta_t = 1/t$. For reasons that will be clear in a little while, we usually choose the step-sizes $\eta_t \to 0$ so that two conditions are satisfied:

$$\sum_{t=0}^{\infty} \eta_t = \infty \tag{10.1}$$

$$\sum_{t=0}^{\infty} \eta_t^2 < \infty. \tag{10.2}$$

3. Question: Why wouldn't we using line search for the subgradient method?

10.4 Lipschitz Functions

To analyze the subgradient method we'll assume that our objective function is G-Lipschitz, i.e. for any x, y

$$|f(x) - f(y)| \leq G\|x - y\|_2.$$

A more useful (for us) equivalent characterization for convex functions f will be that for any $g \in \partial f(x)$,

$$\|g\|_2 \leq G.$$

We are not assuming that the function is smooth (or even differentiable), rather we are assuming that the subgradients are bounded. Notice that all the non-smooth functions we've encountered so far (things like the absolute value function) satisfy this assumption.

We can check one direction of this equivalence: by convexity, we know that for any y,

$$g_x^T(x - y) \leq f(x) - f(y),$$

so choosing $y = x - g_x$, we see that,

$$\|g_x\|_2^2 \leq f(x) - f(x - g_x) \leq |f(x) - f(x - g_x)| \leq G\|g_x\|_2.$$

10.5 Convergence of Subgradient Method

Our goal will now to be to prove the following theorem:

Theorem 10.1. *Suppose f is convex, and G-Lipschitz. Denote by x^{best} the best iterate of the subgradient method amongst $x^0, \ldots, x^k$. Then:*

1. *If the step-size is chosen to be a constant $\eta = R/(G\sqrt{k})$*

$$f(x^{best}) - f(x^*) \leq \frac{GR}{\sqrt{k}}.$$

2. *For any sequence of step-sizes, which satisfy the conditions in* (10.1), (10.2) *we have that,*

$$f(x^{best}) - f(x^*) \to 0, \quad as\ k \to \infty.$$

Proof: The proof is similar to our earlier proofs so hopefully you're getting the hang of the manipulations.

$$\begin{aligned}\|x^{t+1} - x^*\|_2^2 &= \|x^t - \eta_t g_{x^t} - x^*\|_2^2 \\ &= \|x^t - x^*\|_2^2 + \eta_t^2 \|g_{x^t}\|_2^2 - 2\eta g_{x^t}^T (x^t - x^*) \\ &\leq \|x^t - x^*\|_2^2 + \eta_t^2 G^2 - 2\eta_t g_{x^t}^T (x^t - x^*).\end{aligned}$$

By convexity we know that,

$$\begin{aligned}f(x^*) &\leq f(x^t) + g_{x^t}^T (x^* - x^t), \\ \Rightarrow -2\eta_t g_{x^t}^T (x^t - x^*) &\leq 2\eta_t (f(x^*) - f(x^t)).\end{aligned}$$

and as a consequence we obtain,

$$\|x^{t+1} - x^*\|_2^2 \leq \|x^t - x^*\|_2^2 + \eta_t^2 G^2 + 2\eta_t (f(x^*) - f(x^t)).$$

Summing from $t = \{0, \ldots, k-1\}$ and dropping a negative term we obtain that,

$$2\sum_{t=0}^{k-1} \eta_t (f(x^t) - f(x^*)) \leq \|x^0 - x^*\|_2^2 + G^2 \sum_{t=0}^{k-1} \eta_t^2.$$

Recalling our definition of the best iterate seen we see that,

$$f(x^{\text{best}}) - f(x^*) \leq \frac{\|x^0 - x^*\|_2^2 + G^2 \sum_{t=0}^{k-1} \eta_t^2}{2\sum_{t=0}^{k-1} \eta_t}.$$

From this expression we can directly see all of our conclusions. Suppose that $\|x^0 - x^*\|_2 \leq R$.

1. If we choose a constant step-size, $\eta_t = R/(G\sqrt{k})$ then we see that,

$$f(x^{\text{best}}) - f(x^*) \leq \frac{GR}{\sqrt{k}}.$$

2. To get an ϵ error, we need $O(1/\epsilon^2)$ iterations.
3. More generally, for any sequence of step-sizes which satisfy (10.1), (10.2) we have that $f(x^{\text{best}}) - f(x^*) \to 0$ (which is some type of "convergence" of the method).
4. Finally, if we don't know the number of steps in advance (and we don't know the parameters R and G) we might choose, $\eta_t = 1/\sqrt{t}$, to obtain a similar guarantee,

$$f(x^{\text{best}}) - f(x^*) \lesssim \frac{G^2 \log(k) + R^2}{\sqrt{k}}.$$

This is worse than the fixed choice (by a log factor) but allows one to iterate beyond an a-priori fixed number of iterations.

■