9

Subgradient methods

In some cases the objective function of our optimization problem is indeed convex, but not smooth. In those cases, we cannot use the gradient methods, nor any of its variants, because the function is not always differentiable, and thus the gradient might not exist. This prompts the development of new concepts and methods, overviewed in this chapter.

9.1 Examples

We start by providing two examples of convex non-smooth optimization problems that arise frequently in applications. The first occurs when we want to minimize a $\|\cdot\|_1$ or $\|\cdot\|_\infty$ norm. A very common case goes under the name of *L1 regularization*, and consists in optimizing the function $f(x)+\lambda\|x\|_1$, instead of the original objective $f(x)$: the penalty term $\lambda\|x\|_1$ enforces sparsity, a very desirable property in many domains.

Another application is the Lagrangian relaxation of combinatorial optimization problems. Consider for example a pure binary linear problem:

$$\begin{aligned} \min\ & c^\top x \\ & Ax \geq b \\ & x \in \{0,1\}^n \end{aligned}$$

We can introduce Lagrangian multipliers $\lambda \geq 0$ for all linear constraints and construct the Lagrangian problem:

$$\begin{aligned} \min\ & c^\top x + \lambda^\top (b - Ax) \\ & x \in \{0,1\}^n \end{aligned}$$

As usual, we denote with $l(\lambda)$ the optimal objective of the Lagrangian subproblem, as a function of λ, and are interested in solving the Lagrangian dual:

$$\begin{aligned} \max\ & l(\lambda) \\ & \lambda \geq 0 \end{aligned}$$

As well known, the function $l(\lambda)$ is always concave, but the discrete nature of the Lagrangian subproblem makes it non-smooth: for each $x \in \{0,1\}^n$, the objective expression is a linear function of λ, and we are taking the point-wise minimum over finitely many terms, thus $l(\lambda)$ is piecewise-linear.

9.2 Subgradients

Even though the function is non-differentiable, the fact that it is convex allows us to introduce the following generalization:

Definition 9.1. *A vector d is a* subgradient *of a convex function $f(x)$ at a point x if:*

$$f(y) \geq f(x) + d^\top (y - x) \quad \forall y$$

In other words, d defines an globally valid under-approximation of our function. In general, at any given point x, we might have multiple subgradients: we call the corresponding set *subdifferential*, and we denote it with $\partial f(x)$. For convex functions, the subdifferential $\partial f(x)$ has the following properties:

- it is always non-empty (because of convexity);
- if $f(x)$ is differentiable at x, we have $\partial f(x) = \{\nabla f(x)\}$;
- $\partial f(x)$ is always a convex set.

In addition, we have the following theorem:

Theorem 9.1. *A point x^* is a global minimizer of the convex function $f(x)$ if and only if $0 \in \partial f(x^*)$.*

In other words, the subdifferential acts as a generalization of the gradient for convex functions.

9.3 Iterative methods

An obvious approach to extend the gradient methods that we have studied in the smooth case is to use a(ny) subgradient in the place of the gradient in an iterative scheme:

$$x^{k+1} = x^k - t_k d_k \quad d_k \in \partial f(x^k)$$

While very natural, there are at least 3 issues with this approach:

We cannot use automatic differentiation, for example.

1. Computing a subgradient is not necessarily straightforward. While there is a calculus of subdifferentials, that allows us to compute the subdifferential of a composite function from the subdifferentials of simpler ones, this is in general far from trivial. On the other hand, in many cases we can resort to application-specific methods to compute a subgradient at a given point. The most famous example is, again, the Lagrangian dual of an integer linear program. For a fixed λ, let x^* be any minimizer of $L(x, \lambda)$. Then we have that $s = b - Ax^*$ is a subgradient of $l(\lambda)$.

Note that we need to compute x^* in order to evaluate $l(\lambda)$.

Proof. We need to show that $l(v) \leq l(u) + s^\top (v - u)$, for any $v \geq 0$.

$$\begin{aligned} l(v) &= \min_{x \in \{0,1\}^n} \{c^\top x + v^\top (b - Ax)\} \\ &\leq c^\top x^* + v^\top (b - Ax^*) \\ &= c^\top x^* + u^\top (b - Ax^*) + (v - u)^\top (b - Ax^*) \\ &= l(u) + s^\top (v - u) \end{aligned}$$

□

Incidentally, this also shows that when the optimum is unique for $L(x, \lambda)$ the function $l(\lambda)$ is differentiable.

2. $d \in \partial f(x^k)$ is *not* necessarily a descent direction for f at x^k. However, we can show that the *distance* to the optimal solution always decreases for a small enough step size t_k:

$$\|x^{k+1} - x^*\| < \|x^k - x^*\|$$

3. The subgradient does *not* converge to zero as we approach a minimizer x^*. This means that we need diminishing step sizes to guarantee convergence, and this destroys the performance of the method, as we will see.

9.4 Convergence

Let us analyze a basic subgradient method. As usual, we assume that the function f is Lipschitz continuous with coefficient M. This implies a bound on the subgradient norm:

$$\|d_k\| \leq M \quad \forall d_k \in \partial f(x^k)$$

The starting point is expressing the new distance from the optimal solution in terms of the old distance:

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &= \|(x^k - t_k d_k) - x^*\|^2 \\ &= \|(x^k - x^*) - t_k d_k\|^2 \\ &= \|x^k - x^*\|^2 - 2t_k d_k^\top (x^k - x^*) + t_k^2 \|d_k\|^2\end{aligned}$$

Since d_k is a subgradient, we have that $f(x^*) \geq f(x^k) + d_k^\top (x^* - x^k)$, and thus $-d_k^\top (x^k - x^*) \leq -(f(x^k) - f(x^*))$. Plugged into the above we obtain:

$$\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - 2t_k d_k^\top (f(x^k) - f(x^*)) + t_k^2 M^2$$

We can now rearrange and sum up the first s terms (from 0 to $s-1$), exploit the usual telescoping sum, and get:

$$2 \sum_{k=0}^{s-1} \underbrace{(f(x^k) - f(x^*))}_{\geq f(x^{best}) - f(x^*)} \leq \underbrace{\|x^0 - x^*\|^2}_{=D} - \underbrace{\|x^s - x^*\|^2}_{\geq 0} + M^2 \sum_{k=0}^{s-1} t_k^2$$

After some final rearrangement we obtain:

$$f(x^{best}) - f(x^*) \leq \frac{D^2 + M^2 \sum_{k=0}^{s-1} t_k^2}{2 \sum_{k=0}^{s-1} t_k}$$

This is very similar to the analysis of stochastic gradient descent: in particular, we get convergence if:

$$\sum t_k \to +\infty \quad \text{and} \quad \sum t_k^2 \to O(1)$$

Note, it is not $\sqrt{k}$.

What is the speed of convergence? The optimal choice is given by:

$$t_k = \frac{D}{M}\frac{1}{\sqrt{s}}$$

After plugging it in we obtain:

$$\frac{D^2 + M^2 s \frac{D^2}{M^2}\frac{1}{s}}{2s\frac{D}{M}\frac{1}{\sqrt{s}}} = \frac{DM}{\sqrt{s}} \leq \varepsilon$$

which implies $s \geq \frac{DM}{\varepsilon^2} = O(1/\varepsilon^2)$. Note that this is significantly worse than in the smooth case, where we obtained an iteration complexity of $O(1/\varepsilon)$.

This does not change we worst case complexity, though.

It turns out that this step size is quite slow not only in theory but also in practice. In implementations, a different strategy is usually exploited, that goes under the name of *Polyak step rule*. Assuming we know the optimal value p^* (in practice, we use some estimate), the rule prescribes the step size:

$$t_k = \frac{f(x^k) - p^*}{\|d_k\|^2}$$

Let us see the convergence proof under this rule. We start again from:

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &\leq \|x^k - x^*\|^2 - 2t_k d_k^\top (f(x^k) - p^*) + t_k^2 \|d_k\|^2 \\ &= \|x^k - x^*\|^2 - \frac{(f(x^k) - p^*)^2}{\|d_k\|^2}\end{aligned}$$

Notice that this makes it explicit that the distance to the optimal solution decreases at each step. Rearranging, summing up the first s terms, and exploiting the familiar telescoping sum we obtain:

$$\sum_{k=0}^{s-1} \frac{(f(x^k) - p^*)^2}{\|d_k\|^2} \leq \|x^0 - x^*\|^2 - \|x^s - x^*\|^2 \leq D^2$$

Each term in the summation on the left can be lower bounded by the constant term

$$\frac{(f(x^{best}) - p^*)^2}{M^2}$$

so in the end we get:

$$f(x^{best}) - p^* \leq \frac{DM}{\sqrt{s}}$$

and the same convergence speed of $O(1/\varepsilon^2)$.

Note that we can, of course, consider the same variants as we did in the smooth case. In particular:

- we can do *stochastic subgradient descent*: in this case, there is nothing to lose in iteration complexity;
- we can do *projected subgradient* for simple sets, e.g., $\lambda \geq 0$ in Lagrangian duals.

9.5 Zigzagging

The practical convergence of the method is usually slowed down by *zigzagging*. In particular, we distinguish zigzagging of *type I*, occurring when consecutive subgradients form a negative angles with each other, i.e., $d_k d_{k-1} < 0$, and zigzagging of *type II*, which is induced by the projection operation.

We will not investigate type II, but we provide some details for zigzagging of type I. In this case, the idea is to deflect the subgradient, using a linear combination of the current subgradient s_k and the previous direction d_{k-1} to compute the current direction d_k. In details, instead of choosing $d_k = s_k$, for some $s_k \in \partial f(x^k)$, we use:

$$d_k = s_k + \delta_k d_{k-1}$$

for some scalar $\delta_k \geq 0$. The same convergence results of the standard subgradient method can be obtained with a proper choice of the parameter δ_k, for example using:

$$\delta_k = \begin{cases} -\tau_k \frac{s_k d_{k-1}}{\|d_{k-1}\|^2} & \text{if} s_k d_{k-1} < 0 \\ 0 & \text{otherwise} \end{cases}$$

for some $0 \geq \tau_k \geq 2$.