5.1 Last time: Subgradient

Subgradients are alternatives to gradients when the function $f$ is non-smooth or non-differentiable. For convex and differentiable $f$:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \forall x, y$$

A subgradient of a convex function $f$ at $x$ is any $g \in \mathbb{R}^n$ such that:

$$f(y) \geq f(x) + g^T (y - x), \forall x, y$$

5.2 Subgradient Method

Now consider $f$ convex, having $\text{dom}(f) = \mathbb{R}^n$, but not necessarily differentiable. Our objective is to minimize $f$. Subgradient method is like gradient descent, but we replace gradients with subgradients, i.e. initialize $x^{(0)}$, repeat:

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

where $g^{(k-1)} \in \partial f(x^{(k-1)})$ is any subgradient of $f$ at $x^{(k-1)}$, and $\partial f$ represents the subdifferential of $f$.

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

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

5.2.1 Step size choices

1. Fixed step sizes: $t_k = t$, for all $k = 1, 2, 3, ...$

2. Diminishing step sizes: choose to meet conditions

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

These two inequalities, square summable but not summable, are important here to ensure that step sizes diminish to zero, but not too fast.
3. Polyak step sizes: when the optimal value $f^*$ is known, take
\[ t_k = \frac{f(x(k-1)) - f^*}{\|g(k-1)\|^2}, \quad k = 1, 2, 3, ... \]
Polyak step size minimizes the right-hand side of
\[ \|x^{(k)} - x^*\|_2^2 \leq \|x^{(k-1)} - x^*\|_2^2 - 2t_k(f(x^{(k-1)}) - f(x^*)) + t_k^2\|g^{(k-1)}\|^2 \]

5.2.2 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.,
\[ |f(x) - f(y)| \leq G|x - y|_2, \quad \forall x, y \]

**Theorem 5.1 Convergence for fixed step size:** For a fixed step size $t$, subgradient method satisfies
\[ \lim_{k \to \infty} f(x^{(k)}_{\text{best}}) \leq f^* + \frac{G^2 t}{2} \]

**Theorem 5.2 Convergence for diminishing step size:** For diminishing step sizes that satisfy the conditions from Section 5.2.1, subgradient method satisfies
\[ \lim_{k \to \infty} f(x^{(k)}_{\text{best}}) = f^* \]

**Proof:** Can prove both the theorems from a basic inequality.

For a convex, $G$-Lipschitz function $f$, a subgradient has bounded norm. That is,
\[ g \in \partial f(x) \Rightarrow \|g\|_2 \leq G \]

From the definition of a subgradient,
\[ \|x^{(k)} - x^*\|_2^2 = \|x^{(k-1)} - t_kg^{(k-1)} - x^*\|_2^2 \]
\[ = \|x^{(k-1)} - x^*\|_2^2 + t_k^2\|g^{(k-1)}\|^2 - 2t_k(g^{(k-1)})^T(x^{(k-1)} - x^*) \]
\[ \leq \|x^{(k-1)} - x^*\|_2^2 + t_k^2G^2 - 2t_k(f(x^{(k-1)}) - f(x^*)) \]

Where we use the definition of a subgradient in the last term on the right hand side.

\[ f(x^*) \geq f(x^{(k-1)}) + (g^{(k-1)})^T(x^{(k-1)} - x^*) \]
\[ \Rightarrow (g^{(k-1)})^T(x^{(k-1)} - x^*) \leq f(x^*) - f(x^{(k-1)}) \]

Iterating last inequality, we can get
\[ \|x^{(k)} - x^*\|_2^2 \leq \|x^{(0)} - x^*\|_2^2 + \sum_{i=1}^{k}t_i^2G^2 - 2\sum_{i=1}^{k}t_i(f(x^{(i-1)}) - f(x^*)) \]
\[ \Rightarrow 2\sum_{i=1}^{k}t_i(f(x^{(i-1)}) - f(x^*)) \leq R^2 + \sum_{i=1}^{k}t_i^2G^2 \]

Each term in the summation on the left hand side
\[ t_i(f(x^{(i-1)}) - f(x^*)) \geq t_i(f(x^{(\text{best})}) - f(x^*)) \]
\[ \Rightarrow 2\sum_{i=1}^{k}t_i(f(x^{(\text{best})}) - f(x^*)) \leq R^2 + \sum_{i=1}^{k}t_i^2G^2 \]
\[ \Rightarrow f(x^{(\text{best})}) - f(x^*) \leq \frac{R^2 + \sum_{i=1}^{k}t_i^2G^2}{2\sum_{i=1}^{k}t_i} \]
where \( f(x^{(k)}_{\text{best}}) = \min_{i=0,...,k} f(x^{(i)}) \) is the objective value at the best iterate \( x^{(k)}_{\text{best}} \).

This equation is the basic inequality we can use to derive convergence results for different step sizes.

1. For \( t_i = t, \forall i \)

\[
f(x^{(k)}_{\text{best}}) - f(x^*) \leq \frac{R^2 + t^2 k G^2}{2tk} \xrightarrow{k \to \infty} \frac{R^2}{2tk} + \frac{G^2 t}{2}
\]

2. For diminishing \( t_i \)

\[
f(x^{(k)}_{\text{best}}) - f(x^*) \leq \frac{R^2 + \sum_{i=1}^k t_i^2 G^2}{2 \sum_{i=1}^k t_i} \xrightarrow{k \to \infty} \frac{R^2 + \sum_{i=1}^k t_i^2}{2 \sum_{i=1}^k t_i} \to \infty
\]

This concludes the proof.

**Convergence rate** The basic inequality tells us that after \( k \) steps, we have

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

With fixed step size \( t \), this gives

\[
f(x^{(k)}_{\text{best}}) - f(x^*) \leq \frac{R^2}{2tk} + \frac{G^2 t}{2}
\]

For this to be \( \leq \epsilon \), let's make each term \( \leq \epsilon/2 \). So we can choose \( t = \epsilon/G^2 \), and \( k = R^2/t \cdot 1/\epsilon = R^2 G^2/\epsilon^2 \).

This shows that subgradient method has convergence rate \( O(1/\epsilon^2) \) (compare this to convergence rate of \( O(1/\epsilon) \) for gradient descent).

### 5.2.3 Projected subgradient method

To optimize a convex function \( f \) over a convex set \( C \),

\[
\min f(x) \text{ subject to } x \in C
\]

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

\[
x^{(k)} = P_C(x^{(k-1)} - t_k \cdot g^{(k-1)}), \ k = 1, 2, 3, ...
\]

Assuming we can do this projection, we get the same convergence guarantees as the usual subgradient method, with the same step size choices.

There are many types of sets \( C \) that are easy to project onto, 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 \( C = \{ x : Ax \leq b \} \).

### 5.2.4 Improving on the subgradient method

The upside of the subgradient method is that it has broad applicability. The downside is that the convergence rate \( O(1/\epsilon^2) \) is slow over the problem class of convex, Lipschitz functions. We will see if we can improve the convergence rate.

Nonsmooth first-order methods are the iterative methods that update \( x^{(k)} \) in the following way:

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

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

**Theorem 5.3 (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

\[
f(x^{(k)}) - f^* \geq \frac{RG}{2(1 + \sqrt{k + 1})}
\]

From Nesterovs theorem we can find that \( f(x^{(k)}) - f^* \) has a lower bound, which gives the convergence rate \( O(1/\epsilon^2) \). In summary, we cannot do better than the \( O(1/\epsilon^2) \) convergence rate for the 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 of simple form.

For a lot of problems (i.e., functions \( h \)), we can recover the \( O(1/\epsilon) \) rate of gradient descent with a simple algorithm, which has important practical consequences.

### 5.3 Proximal Gradient Descent

Suppose \( f(x) \) is decomposable:

\[
f(x) = g(x) + h(x)
\]

Where \( g \) is convex, differentiable, \( \text{dom}(g) = \mathbb{R}^n \); \( h \) is convex, but not necessary differentiable.

If \( f \) were differentiable, then gradient descent update would be:

\[
x^+ = x - t \cdot \nabla f(x)
\]
We can do quadratic approximation to get:

\[ x^+ = \arg \min_z f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|^2 \]

If we apply this quadratic approximation to \( g \) and keep \( h \) the same, we get:

\[ x^+ = \arg \min_z \frac{1}{2t} \|z - (x - t \nabla g(x))\|^2 + h(z) \]

The idea is to stay close to gradient update for \( g \) and also make \( h \) small. This function is defined as proximal mapping. Rewrite as follows:

\[ \text{prox}_t(x) = \arg \min_z \frac{1}{2t} \|x - z\|^2 + h(z) \]

This function has unique solution because the square term is strictly convex and \( h(x) \) is convex. So proximal gradient descent is just repeat following steps:

\[ x^{(k)} = \text{prox}_{t_k}(x(k - 1) - t_k \nabla g(x^{(k - 1)})), \quad k = 1, 2, 3, ... \]

To make this update step look familiar, can rewrite it as

\[ x^{(k)} = x^{(k - 1)} - t_k \cdot G_t(x^{(k - 1)}) \]

where \( G_t \) is the generalized gradient of \( f \), (Nesterovs Gradient Mapping)

\[ G_t(x) = \frac{x - \text{prox}_t(x - t \nabla g(x))}{t} \]

Key point is that \( \text{prox}_t(\cdot) \) is can be computed analytically for a lot of important functions \( h \). Note that:

- Mapping \( \text{prox}_t(\cdot) \) does not depend on \( g \) at all, only on \( h \).
- Smooth part \( g \) can be complicated, we only need to compute its gradients.

### 5.3.1 Backtracking line search

Backtracking for prox gradient descent works similar as before (in gradient descent), but operates on \( g \) and not \( f \). Choose parameter \( 0 < \beta < 1 \). At each iteration, start at \( t = t_{\text{init}} \), and while

\[ g(x - t G_t(x)) > g(x) - t \nabla g(x)^T G_t(x) + \frac{t}{2} \|G_t(x)\|^2 \]

shrink \( t = \beta t \), for some \( 0 < \beta < 1 \). Otherwise perform proximal gradient update.

### 5.3.2 Convergence analysis

**Theorem 5.4** Proximal gradient descent with fixed step size \( t \leq 1/L \) satisfies

\[ f(x^{(k)}) - f^* \leq \frac{\|x^{(0)} - x^*\|^2}{2tk} \]

and same result holds for backtracking, with \( t \) replaced by \( \beta/L \).

So proximal gradient descent has convergence rate \( O(1/k) \) or \( O(1/\epsilon) \), which is the same as gradient descent. But we need to consider prox cost too.
5.3.3 Special cases

Proximal gradient descent also called composite gradient descent, or generalized gradient descent. It is called generalized because of several special cases:

- $h = 0$: gradient descent
- $h = I_C$: projected gradient descent
- $g = 0$: proximal point algorithm

5.3.3.1 Projected gradient descent

Given closed, convex set $C \in \mathbb{R}^n$,

$$\min_{x \in C} g(x) \iff \min_{x} g(x) + I_C(x)$$

where $I_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$ is the indicator function of $C$. Hence,

$$\text{prox}_t(x) = \arg \min_z \frac{1}{2t}||x - z||^2_2 + I_C(z)$$

$$= \arg \min_z \in C ||x - z||^2_2$$

I.e., $\text{prox}_t(x) = P_C(x)$, projection operator onto $C$. Therefore proximal gradient update step is:

$$x^+ = P_C(x - t\nabla g(x))$$

5.3.3.2 Proximal point algorithm

When $g = 0$, gradient of $g$ is also zero, so the update is just

$$x^+ = \arg \min_z \frac{1}{2t}||x - z||^2_2 + h(z)$$

Called proximal minimization algorithm. Faster than subgradient method, but not implementable unless we know $\text{prox}$ in closed form.

In practice, if we cannot evaluate $\text{prox}_t$, we can consider to approximate it if we know how to control the error.

5.3.4 Acceleration

As before, consider:

$$\min_x g(x) + h(x)$$

where $g$ convex, differentiable, and $h$ convex. Accelerated proximal gradient method: choose initial point $x^{(0)} = x^{(1)} \in \mathbb{R}^n$, repeat:

$$v = x^{(k-1)} + \frac{k - 2}{k + 1}(x^{(k-1)} - x^{(k-2)})$$

$$x^{(k)} = \text{prox}_{t_k}(v - t_k \nabla g(v))$$

for $k = 1, 2, 3, ...$
• First step $k = 1$ is just usual proximal gradient update
• After that, $v = x^{(k-1)} + \frac{k-2}{k+1} (x^{(k-1)} - x^{(k-2)})$ carries some momentum from previous iterations
• $h = 0$ gives accelerated gradient method

5.3.4.1 Backtracking line search

Simple approach: fix $\beta < 1, t_0 = 1$. At iteration $k$, start with $t = t_{k-1}$, and while

$$g(x^+) > g(v) + \nabla g(v)^T (x^+ - v) + \frac{1}{2t} ||x^+ - v||_2^2$$

shrink $t = \beta t$, and let $x^+ = \text{prox}_t(v - t\nabla g(v))$. Otherwise keep $x^+$.

5.3.4.2 Convergence analysis

**Theorem 5.5** Accelerated proximal gradient method with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f^* \leq \frac{2||x^{(0)} - x^*||^2_2}{2t(k+1)^2}$$

and same result holds for backtracking, with $t$ replaced by $\beta/L$.

Achieves optimal rate $O(1/k^2)$ or $O(1/\sqrt{\epsilon})$ for first-order methods.

References
