Unconstrained minimization of smooth functions

We want to solve

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \; f(\boldsymbol{x}),$$

where f is convex. In this section, we will assume that f is differentiable (so its gradient exists at every point), and is **smooth** (we will consider a few different definitions of "smooth" — qualitatively, this just means that the gradient changes in a controlled manner as we move from point to point).

While many problems are smooth, methods for nonsmooth $f(\boldsymbol{x})$ are also of great interest, and will (hopefully) be covered later in the course. Nonsmooth methods are not much more involved algorithmically, but they are slightly harder to analyze.

Since f is convex, a necessary and sufficient condition for $\boldsymbol{x}^\star$ to be a minimizer is that the gradient vanishes:

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}.$$

It is not a given that such a $\boldsymbol{x}^\star$ exists — it is possible that $f(\boldsymbol{x})$ is unbounded below. In this section, we will assume that f does have (at least one) minimizer, and we will denote the optimal value as $p^\star = f(\boldsymbol{x}^\star)$.

Every general-purpose optimization algorithm we will look at in this course is **iterative** — they will all have the basic form:

```
$\boldsymbol{x}^{(0)}$ = initial guess
$k = 0$
do
    $k = k + 1$
    calculate a direction to move $\boldsymbol{d}^{(k)}$
    calculate a step size $t_k \geq 0$
    $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + t_k\,\boldsymbol{d}^{(k)}$
    check convergence criteria
until converged
```

In the coming lectures, we will focus primarily on two methods for computing the direction $\boldsymbol{d}^{(k)}$.

1. **Gradient descent**: we take

$$\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k-1)}).$$

   This is the direction of "steepest descent" (where "steepest" is defined relative to the Euclidean norm). Gradient descent iterations are **cheap**, but typically many iterations are required for convergence.

2. **Newton's method**: we build a quadratic model around $\boldsymbol{x}^{(k)}$ then compute the exact minimizer of this quadratic by solving a system of equations. This corresponds to taking

$$\boldsymbol{d}^{(k)} = -(\nabla^2 f(\boldsymbol{x}^{(k-1)}))^{-1}\nabla f(\boldsymbol{x}^{k-1}),$$

   that is, the inverse of the Hessian evaluated at $\boldsymbol{x}^{(k-1)}$ applied to the gradient evaluated at the same point. Newton iterations tend to be **expensive** (as they require a system solve), but they typically converge in far fewer iterations than gradient descent.

Whichever direction we choose, it should be a **descent direction**; $\boldsymbol{d}^{(k)}$ should satisfy

$$\langle \boldsymbol{d}^{(k)}, \nabla f(\boldsymbol{x}^{(k-1)})\rangle \leq 0.$$

Since f is convex, it is always true that

$$f(\boldsymbol{x} + t\boldsymbol{d}) \geq f(\boldsymbol{x}) + t\langle \boldsymbol{d}, \nabla f(\boldsymbol{x})\rangle,$$

and so to decrease the value of the functional while moving in direction $\boldsymbol{d}$, it is necessary that the inner product above be negative.

Line search

After the step direction is chosen, we need to compute how far to move. There are many methods for doing this, here are three:

Exact: Solve the 1D optimization program

$$\min_{s\geq 0} f(\boldsymbol{x}^{(k-1)} + s\boldsymbol{d}^{(k)}).$$

This is typically not worth the trouble, but there are instances (i.e. unconstrained convex quadratic programs) when it can be solved analytically.

Backtracking: Start with a step size of $t = 1$, then decrease by a factor of β until we are below a certain line.

```
Fix $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$.
Given a starting point $\boldsymbol{x}$ and direction $\boldsymbol{d}$,
$t = 1$
repeat
    if $f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}) + \alpha\, t\langle \boldsymbol{d}, \nabla f(\boldsymbol{x})\rangle$
        converged
    else
        $t = \beta t$
    endif
until converged
```

Picture:

The backtracking line search tends to be cheap, and works very well in practice. Typically, we take α small (to encourage a large step) and $\beta \in [0.3, 0.8]$.

Fixed: Finally, we can just use a constant step size $t_k = t$. This will actually work if the step size is small enough, but usually this results in way too many iterations.

Convergence of gradient descent

How quickly does gradient descent converge?

I'm glad you asked!

We will look at two different smoothness assumptions on f, and translate them into convergence rates. In the first case, we assume that f is twice differentiable (so that its Hessian exists everywhere), and that

$$m\mathbf{I} \ \preceq \ \nabla^2 f(\boldsymbol{x}) \ \preceq \ M\mathbf{I}.$$

That is, the eigenvalues of the Hessian (which is always in S^N for a convex function) are bounded between $m > 0$ and $M < \infty$. We call this assumption **strong convexity** (note that the lower bound by itself means that f is strictly convex).

Recall that the main consequence of convexity is that we have a way to compute a linear global lower bound at every point:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle, \quad \text{for all } \boldsymbol{x}, \boldsymbol{y}.$$

The main consequence of strong convexity is that we have in addition a quadratic lower and upper bound. By the Taylor theorem, we know that for any $\boldsymbol{x}, \boldsymbol{y}$,

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{x})^{\mathrm{T}} \nabla^2 f(\boldsymbol{z})(\boldsymbol{y} - \boldsymbol{x})$$

for some point $\boldsymbol{z}$ on the line segment between $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus we have the bounds

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{m}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2, \tag{1}$$

and

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{M}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2. \tag{2}$$

An immediate consequence of the stronger lower bound in (1) is that we can tell at any point $\boldsymbol{x}$ how close to optimal we are. The smallest value the right hand side can take in that inequality is when $\tilde{\boldsymbol{y}} = \boldsymbol{x} - m^{-1}\nabla f(\boldsymbol{x})$; plugging that value into the right hand side yields

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) - \frac{1}{2m}\|\nabla f(\boldsymbol{x})\|_2^2,$$

a bound which holds for every $f(\boldsymbol{y})$. In particular, it holds for the optimal value $p^\star$, and so

$$p^\star \geq f(\boldsymbol{x}) - \frac{1}{2m}\|\nabla f(\boldsymbol{x})\|_2^2. \tag{3}$$

Thus we get a bound on $f(\boldsymbol{x}) - p^\star$ just by calculating the norm of the gradient. It also re-iterates that if $\|\nabla f(\boldsymbol{x})\|_2$ is small, we are close to the solution.

We can also bound how close $\boldsymbol{x}$ is to the minimizer $\boldsymbol{x}^\star$. Using (1) with $\boldsymbol{y} = \boldsymbol{x}^\star$,

$$\begin{aligned} p^\star = f(\boldsymbol{x}^\star) &\geq f(\boldsymbol{x}) + \langle \boldsymbol{x}^\star - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{m}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2^2 \\ &\geq f(\boldsymbol{x}) - \|\boldsymbol{x}^\star - \boldsymbol{x}\|_2 \, \|\nabla f(\boldsymbol{x})\|_2 + \frac{m}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2^2. \end{aligned}$$

Then since $p^\star \leq f(\boldsymbol{x})$,

$$-\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2 \, \|\nabla f(\boldsymbol{x})\|_2 + \frac{m}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2^2 \leq 0,$$

and so

$$\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2 \leq \frac{2}{m}\|\nabla f(\boldsymbol{x})\|_2.$$

These facts are very useful for defining stopping criteria.

Suppose we have a strongly convex function that we minimize using gradient descent with an **exact line search**. We can show that at each iteration, the gap $f(\boldsymbol{x}^{(k)}) - p^\star$ gets cut down by a fixed factor. We consider a single iteration — we will use $\boldsymbol{x}$ to denote the current point, and $\boldsymbol{x}^+ = \boldsymbol{x} - t_{\text{exact}}\nabla f(\boldsymbol{x})$ to denote the result of the gradient step. We choose t_{exact} by minimizing the following function:

$$\tilde{f}(t) = f(\boldsymbol{x} - t\nabla f(\boldsymbol{x})).$$

By strong convexity, we know that

$$\tilde{f}(t) \leq f(\boldsymbol{x}) - t\|\nabla f(\boldsymbol{x})\|_2^2 + \frac{Mt^2}{2}\|\nabla f(\boldsymbol{x})\|_2^2.$$

By definition of t_{exact}, we know

$$f(\boldsymbol{x}^+) = \tilde{f}(t_{\text{exact}}) \leq \tilde{f}(1/M) \leq f(\boldsymbol{x}) - \frac{1}{2M}\|\nabla f(\boldsymbol{x})\|_2^2.$$

From (3), we also know that

$$\|\nabla f(\boldsymbol{x})\|_2^2 \geq 2m(f(\boldsymbol{x}) - p^\star),$$

and so

$$f(\boldsymbol{x}^+) - p^\star \ \leq \ f(\boldsymbol{x}) - p^\star - \frac{m}{M}(f(\boldsymbol{x}) - p^\star),$$

which means

$$\frac{f(\boldsymbol{x}^+) - p^\star}{f(\boldsymbol{x}) - p^\star} \leq \left(1 - \frac{m}{M}\right).$$

That is, the gap between the current functional evaluation and the optimal value has been cut down by a factor of $1 - m/M < 1$.

Applying this relationship recursively, we see that after k iterations of gradient descent, we have

$$\frac{f(\boldsymbol{x}^{(k)}) - p^\star}{f(\boldsymbol{x}^{(0)}) - p^\star} \leq \left(1 - \frac{m}{M}\right)^k.$$

Another way to say this is that we can achieve accuracy

$$f(\boldsymbol{x}^{(k)}) - p^\star \leq \epsilon,$$

by taking

$$k \geq \frac{\log(E_0/\epsilon)}{\log(1 - m/M)}, \quad E_0 = f(\boldsymbol{x}^{(0)}) - p^\star,$$

steps.

There are similar results for gradient descent on strongly convex functions using backtracking. They are similar in nature; they have the same linear convergence but with constants that depend on α and β along with m and M. See [**?**, p. 468].

Lipschitz gradient condition

We can also get (much weaker) convergence results when f is not strongly convex (or even necessarily twice differentiable), but rather has a **Lipschitz gradient**. This means that there exists an $L > 0$ such that

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2. \tag{4}$$

The upshot here is that we still have a quadratic upper bound for f at every point, but not the lower bound.

To be precise, if f obeys (4), then

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2.$$

This follows immediately from the fundamental theorem of calculus[1]:

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \int_0^1 \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f\left((1-t)\boldsymbol{x} + t\boldsymbol{y}\right) \rangle \ \mathrm{d}t,$$

and so

$$\begin{aligned} f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle &= \int_0^1 \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f((1-t)\boldsymbol{x} + t\boldsymbol{y}) - \nabla f(\boldsymbol{x}) \rangle \ \mathrm{d}t \\ &\leq \|\boldsymbol{y} - \boldsymbol{x}\|_2 \int_0^1 \|\nabla f((1-t)\boldsymbol{x} + t\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|_2 \ \mathrm{d}t \\ &\leq L\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \int_0^1 t \ \mathrm{d}t \\ &= \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2. \end{aligned}$$

Now, let's consider running gradient descent on such a function with a **fixed step size** $t \leq 1/L$. As before, we denote the current iterate as $\boldsymbol{x}$ and the next iterate at $\boldsymbol{x}^+ = \boldsymbol{x} - t\nabla f(\boldsymbol{x})$. We have

$$\begin{aligned} f(\boldsymbol{x}^+) &\leq f(\boldsymbol{x}) - \left(1 - \frac{Lt}{2}\right) t\|\nabla f(\boldsymbol{x})\|_2^2 \\ &\leq f(\boldsymbol{x}) - \frac{t}{2}\|\nabla f(\boldsymbol{x})\|_2^2, \end{aligned}$$

[1] The 1D function $\tilde{f}(t) = f((1-t)\boldsymbol{x} + t\boldsymbol{y})$ has derivative $\tilde{f}'(t) = \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f\left((1-t)\boldsymbol{x} + t\boldsymbol{y}\right)\rangle$. This is just a simple application of the chain rule.

for the range of t we are considering. By convexity of f,

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x}^\star) + \langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla f(\boldsymbol{x})\rangle,$$

where $\boldsymbol{x}^\star$ is a minimizer of f, and so

$$f(\boldsymbol{x}^+) \leq f(\boldsymbol{x}^\star) + \langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla f(\boldsymbol{x})\rangle - \frac{t}{2}\|\nabla f(\boldsymbol{x})\|_2^2,$$

and then substituting $\nabla f(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^+)/t$ yields

$$\begin{aligned} f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star) &\leq \frac{1}{t}\langle \boldsymbol{x} - \boldsymbol{x}^\star, \boldsymbol{x} - \boldsymbol{x}^+\rangle - \frac{1}{2t}\|\boldsymbol{x} - \boldsymbol{x}^+\|_2^2 \\ &= \frac{1}{2t}\left(\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^+ - \boldsymbol{x}^\star\|_2^2\right). \end{aligned}$$

Summing this difference over k iterations yields:

$$\begin{aligned} \sum_{i=1}^k f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star) &\leq \frac{1}{2t}\left(\sum_{i=1}^k \|\boldsymbol{x}^{(i-1)} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^\star\|_2^2\right) \\ &= \frac{1}{2t}\left(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2^2\right) \\ &\leq \frac{1}{2t}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2. \end{aligned}$$

Since $f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star)$ is monotonically decreasing in i, the kth term will be smaller than the average:

$$\begin{aligned} f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) &\leq \frac{1}{k}\sum_{i=1}^k f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star) \\ &\leq \frac{1}{2tk}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2. \end{aligned}$$

Note that this convergence guarantee is much slower — it is $O(1/k)$ in place of $O(c^k)$ for some $c < 1$. This is the price we pay for allowing f to not be as smooth.