# Line search methods

Given a starting point $\boldsymbol{x}_k$ and a direction $\boldsymbol{d}_k$, we still need to decide on $\alpha_k$, i.e., how far to move. With $\boldsymbol{x}_k$ and $\boldsymbol{d}_k$ fixed, we can think of the remaining problem as a one-dimensional optimization problem where we would like to choose $\alpha$ to minimize (or at least reduce)

$$\phi(\alpha) = f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right).$$

Note that we don't necessarily need to find the true minimum – we aren't even sure that we are moving in the right direction at this point – but we would generally still like to make as much progress as possible before calculating a new direction $\boldsymbol{d}_{k+1}$. There are many methods for doing this, here are three:

## Fixed step size

We can just use a constant step size $\alpha_k = \alpha$. This will work if the step size is small enough, but usually this results in using more iterations than necessary. This is actually a very commonly used approach since if your problem is small enough, this may not matter.

## Exact line search

Another approach is to solve the one-dimensional optimization program

$$\underset{\alpha \geq 0}{\text{minimize}} \ \phi(\alpha).$$

There are a variety of strategies you could take here (e.g., applying a bisection search or some similar one-dimensional optimization strategy) to try to solve this problem. This is typically not worth

the trouble. However, there are certain instances (e.g., least squares and other unconstrained convex quadratic programs) when it can be solved analytically, in which case it is generally a good idea.

**Example: Minimizing a quadratic function** Suppose we wish to solve the optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \ \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b}.$$

For example, this optimization problem arises in the context of solving least squares problems. Suppose that we have selected a step direction $\boldsymbol{d}_k$. In this case

$$\phi(\alpha) = \frac{1}{2}(\boldsymbol{x}_k + \alpha\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{Q}(\boldsymbol{x}_k + \alpha\boldsymbol{d}_k) - (\boldsymbol{x}_k + \alpha\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{b}.$$

This is a quadratic function of $\alpha$, and thus we can compute the optimal step size by finding the $\alpha$ such that $\phi'(\alpha) = 0$. By expanding out the quadratic term, it is easy to show that

$$\phi'(\alpha) = \alpha\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{d}_k + \boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{b}.$$

Setting this equal to zero and solving for $\alpha$ yields the step size

$$\alpha_k = \frac{\boldsymbol{d}_k^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{Q}\boldsymbol{x}_k)}{\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{d}_k}.$$

## Backtracking

Exact line search is generally not worth the trouble, but the problem with a fixed step size is that we cannot guarantee convergence of $\alpha$ is too large, but when $\alpha$ is too small we may not make much progress on

each iteration. A popular strategy is to do a rudimentary search for an $\alpha$ that gives us sufficient progress as measured by the inequality

$$f\left(\boldsymbol{x}_k\right) - f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right) \geq -c_1 \alpha \left\langle \boldsymbol{d}_k, \nabla f\left(\boldsymbol{x}_k\right) \right\rangle,$$

where $c_1 \in (0,1)$. This is known as the **Armijo condition**. For $\alpha$ satisfying the inequality we have that the reduction in $f$ is proportional to both the step length $\alpha$ and the directional derivative in the direction $\boldsymbol{d}_k$.

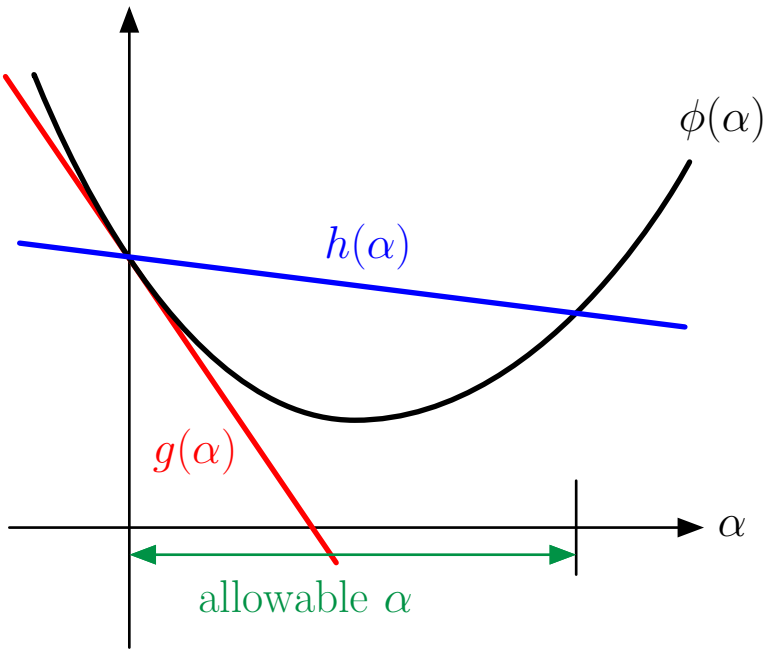
Note that we can equivalently write this condition as

$$\phi(\alpha) \leq h(\alpha) := \phi(0) + c_1 \alpha \phi'(0).$$

Recall that from convexity, we also have that

$$\phi(\alpha) \geq g(\alpha) := \phi(0) + \alpha \phi'(0).$$

Since $c_1 < 1$, we always have $\phi(\alpha) \leq h(\alpha)$ for sufficiently small $\alpha$. An example is illustrated below:



We still haven't said anything about how to actually use the Armijo condition to pick $\alpha$. Within the set of allowable $\alpha$ satisfying the condition, the (guaranteed) reduction in $f$ is proportional to $\alpha$, so we would generally like to select $\alpha$ to be large. Note that the Armijo condition protects us against setting $\alpha$ to be *too* large, however, it actually does not rule out setting $\alpha$ to be extremely small, which can be just as much of a problem in practice.

This inspires the following very simple **backtracking** algorithm: start with a large step size of $\alpha = \bar{\alpha}$, and then decrease by a factor of $\rho$ until the Armijo condition is satisfied.

**Backtracking line search**

Input: $\boldsymbol{x}_k$, $\boldsymbol{d}_k$, $\nabla f(\boldsymbol{x}_k)$, $\bar{\alpha} > 0$, $c_1 \in (0,1)$, and $\rho \in (0,1)$.

Initialize: $\alpha = \bar{\alpha}$

**while** Armijo condition not satisfied **do**

$\alpha = \rho\alpha$

**end while**

The backtracking line search tends to be cheap, and works very well in practice. A common choice for $\bar{\alpha}$ is $\bar{\alpha} = 1$, but this can vary somewhat depending on the algorithm. The choice of $c_1$ can range from extremely small ($10^{-4}$, encouraging larger steps) to relatively large (0.3, encouraging smaller steps), and typical values of $\rho$ range from 0.1, (corresponding to a relatively coarse search) to 0.8 (corresponding to a finer search).

# Convergence of gradient descent

Here we will discuss convergence guarantees for **gradient descent**, i.e., the version of our iterative algorithm where we set

$$\boldsymbol{d}_k = -\nabla f\left(\boldsymbol{x}_k\right),$$

resulting in the update rule

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f\left(\boldsymbol{x}_k\right).$$

It is hard to say much about the convergence properties of this approach for *arbitrary* convex functions. However, if $f$ satisfies certain "regularity conditions", then we can get very nice guarantees, even for a fixed step size. Here we will look at two different kinds of regularity assumptions on $f$, and translate them into convergence rates. Throughout, we will assume that $f$ is differentiable everywhere.[1]

## Smoothness and strong convexity

We will consider two related kinds of assumptions on $f$. One is that $f$ is **smooth** in a certain sense. Qualitatively, we would just like to assume that the gradient changes in a controlled manner as we move from point to point. Quantitatively, we will assume that $f$ has a **Lipschitz gradient**. This means that there exists an $M > 0$ such that

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq M\|\boldsymbol{x} - \boldsymbol{y}\|_2, \tag{1}$$

[1] Methods for nondifferentiable $f(\boldsymbol{x})$ are also of great interest, and will be covered later in the course. These methods are not much more involved algorithmically (although, you obviously will have to replace the gradient with something else), but they are slightly harder to analyze.

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. We will say that such a function is $M$-**smooth** or **strongly smooth**.

One can show that $f$ obeying (1) is actually equivalent to saying that

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

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$.

This provides some intuition for what kind of structure the Lipschitz gradient condition imposes on $f$. Recall that for any convex function, we have that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle,$$

so if $f$ is convex, then at any point $\boldsymbol{x}$ we can bound $f$ from *below* by a linear approximation. If $f$ has a Lipschitz gradient, (2) but we can also bound it from *above* using a quadratic approximation.

In the case that $f$ is twice differentiable, one can also show that (2) is equivalent to

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

i.e., that the largest eigenvalue of the Hessian is bounded by $M$ for all $\boldsymbol{x}$. Note, however, that the Lipschitz gradient condition and the analysis below does not require $f$ to be twice differentiable.

A closely related assumption that we can make is to assume that $f$ is **strongly convex** (with strong convexity parameter $m > 0$), meaning that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{m}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2. \tag{3}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. This tells us that not only is $f$ bounded below by a *linear* approximation (since it is convex), but also by a (nontrivial) convex *quadratic* approximation. Note also that strong convexity implies strict convexity, but strict convexity does not necessarily imply strong convexity.

In the case that $f$ is twice differentiable, strong convexity is equivalent to

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

That is, the eigenvalues of the Hessian are bounded below by $m > 0$ for all $\boldsymbol{x}$. When combined with the assumption of $M$-smoothness, this bounds the conditioning of the Hessian matrix so that its eigenvalues are bounded between $m > 0$ and $M < \infty$. However, again note that strong convexity does not require $f$ to be twice differentiable.

## Convergence of gradient descent for smooth and strongly convex $f$

If you look back to when we analyzed the convergence of gradient descent for the case where $f$ was a quadratic function (as in least squares), you may notice that our analysis centered entirely around the identity

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

where in this case $\boldsymbol{H} = \nabla^2 f(\boldsymbol{x})$. In fact, the main convergence result follows from a pair of inequalities that follow directly from this identity. Specifically, that

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

and

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle + \frac{\lambda_{\min}(\boldsymbol{H})}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2.$$

By assuming that $f$ is $M$-smooth and strongly convex, we are essentially making precisely the assumptions that we need to reproduce this analysis. Another perspective is that our assumptions imply that $f$ is not too different from a quadratic function, and so we have similar convergence guarantees. Specifically, using the same basic approach as before one can show (try this at home!) that for $\alpha_k = 1/M$ we have

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

This is an example of *linear convergence*, and using a simple argument from the homework this implies that

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

as long as

$$k \geq \frac{\log((f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star))/\epsilon)}{\log(1/(1 - m/M))}.$$

## Convergence of gradient descent for smooth $f$

It is also possible to get weaker convergence guarantees without making the assumption that $f$ is strongly convex. In the technical addendum at the end of these notes, we show that by combining the assumption of $M$-smoothness with the definition of convexity and doing some clever manipulations, we can get a guarantee of the form

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star) \leq \frac{M}{2k} \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2.$$

Thus, for $M$-smooth functions, we can guarantee that the error is $O(1/k)$ after $k$ iterations. Another way to put this is to say that we can guarantee accuracy

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star) \leq \epsilon$$

as long as

$$k \geq \frac{M}{2\epsilon}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2.$$

Note that if $\epsilon$ is very small, this says we can expect to need a very large number of iterations. This is **much** slower convergence than what we obtained by assuming strong convexity – it is $O(\epsilon^{-1})$ versus $O(\log \epsilon^{-1})$. As an example, if we wanted to set $\epsilon = 10^{-6}$, $\epsilon^{-1} = 10^6$ (versus $\log \epsilon^{-1} \approx 14$). Of course, to get the stronger guarantee we had to make a much stronger assumption (strong convexity), which may not always be applicable depending on the objective function you are optimizing.

## Technical Details: Convergence analysis for $M$-smooth functions

Here we provide the convergence analysis for gradient descent on $M$-smooth functions that are not necessarily strongly convex. From our assumption that $f$ is $M$-smooth, we know that $f$ satisfies (2), and thus plugging in $\boldsymbol{y} = \boldsymbol{x}_{k+1}$, we obtain

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &\leq f(\boldsymbol{x}_k) + \left\langle -\frac{1}{M}\nabla f(\boldsymbol{x}_k), \nabla f(\boldsymbol{x}_k) \right\rangle + \frac{M}{2}\left\| \frac{1}{M}\nabla f(\boldsymbol{x}_k) \right\|_2^2 \\ &= f(\boldsymbol{x}_k) - \frac{1}{M}\|\nabla f(\boldsymbol{x}_k)\|_2^2 + \frac{1}{2M}\|\nabla f(\boldsymbol{x}_k)\|_2^2 \\ &= f(\boldsymbol{x}_k) - \frac{1}{2M}\|\nabla f(\boldsymbol{x}_k)\|_2^2. \end{aligned} \tag{4}$$

Moreover, by the convexity of $f$,

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

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

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}^\star) + \langle \boldsymbol{x}_k - \boldsymbol{x}^\star, \nabla f(\boldsymbol{x}) \rangle - \frac{1}{2M} \|\nabla f(\boldsymbol{x}_k)\|_2^2.$$

Substituting $\nabla f(\boldsymbol{x}_k) = M(\boldsymbol{x}_k - \boldsymbol{x}_{k+1})$ then yields

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^\star) \leq M \langle \boldsymbol{x}_k - \boldsymbol{x}^\star, \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle - \frac{M}{2} \|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|_2^2. \quad (5)$$

We can re-write this in a slightly more convenient way using the fact that

$$\|\boldsymbol{a} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{a}\|_2^2 - 2\langle \boldsymbol{a}, \boldsymbol{b} \rangle + \|\boldsymbol{b}\|_2^2$$

and thus

$$2\langle \boldsymbol{a}, \boldsymbol{b} \rangle - \|\boldsymbol{b}\|_2^2 = \|\boldsymbol{a}\|_2^2 - \|\boldsymbol{a} - \boldsymbol{b}\|_2^2.$$

Setting $\boldsymbol{a} = \boldsymbol{x}_k - \boldsymbol{x}^\star$ and $\boldsymbol{b} = \boldsymbol{x}_k - \boldsymbol{x}_{k+1}$ and applying this to (5), we obtain the bound

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^\star) \leq \frac{M}{2} \left( \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\|_2^2 \right).$$

This result bounds how far away $f(\boldsymbol{x}_{k+1})$ is from the optimal $f(\boldsymbol{x}^\star)$ in terms (primarily) of the error in the previous iteration: $\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2$. We can use this result to bound $f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^\star)$ in terms of the initial error $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2$ by a clever argument.

Specifically, this bound holds not only for iteration $k$, but for all iterations $i = 1, \ldots, k$, so we can write down $k$ inequalities and then sum them up to obtain

$$\sum_{i=1}^{k} f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star) \leq \frac{M}{2} \left( \sum_{i=1}^{k} \|\boldsymbol{x}_{i-1} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 \right).$$

The right-hand side of this inequality is what is called a *telescopic sum*: each successive term in the sum cancels out part of the previous term. Once you write this out, all the terms cancel except for two (one component from the $i = 1$ term and one from the $i = k$ term) giving us:

$$\begin{aligned}\sum_{i=1}^{k} f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star) &\leq \frac{M}{2}\left(\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2\right) \\ &\leq \frac{M}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2.\end{aligned}$$

Since, as noted above, $f(\boldsymbol{x}_i)$ is monotonically decreasing in $i$, we also have that

$$k\left(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star)\right) \leq \sum_{i=1}^{k} f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star),$$

and thus

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star) \leq \frac{M}{2k}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2,$$

which is exactly what we wanted to show.