Lecture 3

Most modern optimization methods are iterative: they generate a sequence of points $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$ in $\mathbb{R}^n$ in the hope that this sequences will converge to a local or global minimizer $\boldsymbol{x}^*$ of a function $f(\boldsymbol{x})$. A typical rule for generating such a sequence would be to start with a vector $\boldsymbol{x}_0$, chosen by an educated guess, and then for $k \geq 0$, move from step k to $k+1$ by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k,$$

in a way that ensures that $f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k)$. The parameter α_k is called the **step length**, while $\boldsymbol{p}_k$ is the **search direction**. In this lecture we discuss one such method, the method of gradient descent, or steepest descent, and discuss how to select the right step length.

3.1 Gradient descent

In the method of gradient descent, the search direction is chosen as

$$\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k). \tag{3.1}$$

To see why this makes sense, let $\boldsymbol{p}$ be a direction and consider the Taylor expansion

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}) = f(\boldsymbol{x}_k) + \alpha \langle \boldsymbol{p}, \nabla f(\boldsymbol{x}_k) \rangle + O(\alpha^2).$$

Considering this as a function of α, the rate of change in direction $\boldsymbol{p}$ at $\boldsymbol{x}_k$ is the derivative of this function at $\alpha = 0$,

$$\frac{\mathrm{d} f(\boldsymbol{x}_k + \alpha \boldsymbol{p})}{\mathrm{d}\alpha}|_{\alpha=0} = \langle \boldsymbol{p}, \nabla f(\boldsymbol{x}_k) \rangle,$$

also known as the **directional derivative** of f at $\boldsymbol{x}_k$ in the direction $\boldsymbol{p}$. This formula indicates that the rate of change is *negative*, and we have a **descent direction**, if $\langle \boldsymbol{p}, \nabla f(\boldsymbol{x}_k) \rangle < 0$.

The Cauchy-Schwarz inequality (see Preliminaries, Page 9) gives the bounds

$$-\|\boldsymbol{p}\|_2 \|\nabla f(\boldsymbol{x}_k)\|_2 \leq \langle \boldsymbol{p}, \nabla f(\boldsymbol{x}_k) \rangle \leq \|\boldsymbol{p}\|_2 \|\nabla f(\boldsymbol{x}_k)\|_2.$$

We see that the rate of change is the smallest when the first inequality is an equality, which happens if

$$\boldsymbol{p} = -\alpha \nabla f(\boldsymbol{x}_k)$$

for some $\alpha > 0$.

For a visual interpretation of what it means to be a descent direction, note that the **angle** θ between a vector $\boldsymbol{p}$ and the gradient $\nabla f(\boldsymbol{x})$ at a point $\boldsymbol{x}$ is given by (see Preliminaries, Page 9)

$$\langle \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle = \|\boldsymbol{p}\|_2 \|\nabla f(\boldsymbol{x})\|_2 \cos(\theta).$$

This is negative if the vector $\boldsymbol{p}$ forms and angle greater than $\pi/2$ with the gradient. Recall that the gradient points in the direction of steepest ascent, and is orthogonal to the *level sets*. If you are standing on the slope of a mountain, walking along the level set lines will not change your elevation, the gradient points to the steepest upward direction, and the negative gradient to the steepest descent.

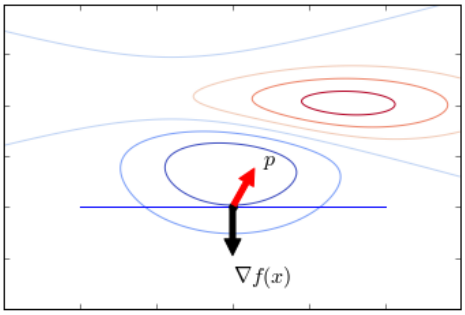

Figure 3.1: A descent direction

Any multiple $\alpha \nabla f(\boldsymbol{x}_k)$ points in the direction of steepest descent, but we have to choose a sensible parameter α to ensure that we make sufficient progress, but at the same time don't overshoot. Ideally, we would choose the value α_k that minimizes $f(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k))$. While finding such a minimizer is in general not easy (see Section 3.2 for alternatives), for quadratic functions in can be given in closed form.

Linear least squares

Consider a function of the form

$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

In Problem Sheet 1 you will show that that the Hessian is symmetric and positive semidefinite, with the gradient given by

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

The method of gradient descent proceeds as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}).$$

To find the best α_k, we compute the minimum of the function

$$\alpha \mapsto f(\boldsymbol{x}_k - \alpha \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b})). \tag{3.2}$$

If we set $\boldsymbol{r}_k := \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) = -\nabla f(\boldsymbol{x}_k)$ and compute the minimum of (3.2) by differentiating, we get the step length

$$\alpha_k = \frac{\boldsymbol{r}_k^\top \boldsymbol{r}_k}{\boldsymbol{r}_k^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k} = \frac{\|\boldsymbol{r}_k\|_2^2}{\|\boldsymbol{A}\boldsymbol{r}_k\|_2^2}.$$

(Verify this!) Note also that when we have $\boldsymbol{r}_k$ and α_k, we can compute the next $\boldsymbol{r}_k$ as

$$\begin{aligned} r_{k+1} &= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}) \\ &= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k)) \\ &= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \alpha_k \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k) = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k. \end{aligned}$$

The gradient descent algorithm for the linear least squares problem proceeds by first computing $\boldsymbol{r}_0 = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0)$, and then at each step

$$\begin{aligned} \alpha_k &= \frac{\boldsymbol{r}_k^\top \boldsymbol{r}_k}{\boldsymbol{r}_k^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k} \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k \\ \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k. \end{aligned}$$

Does this work? How do we know when to stop? It is worth noting that the residual satisfies $\boldsymbol{r} = 0$ if and only if $\boldsymbol{x}$ is a stationary point, in our case, a minimizer. One criteria for stopping could then be to check whether $\|\boldsymbol{r}_k\|_2 \leq \varepsilon$ for some given tolerance $\varepsilon > 0$. One potential problem with this criterion is that the function can become *flat* long before reaching a minimum, so an alternative stopping method would be to stop when the difference between two successive points, $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2$, becomes smaller than some $\varepsilon > 0$.

Example 3.1. We plot the trajectory of gradient descent with the data

$$\boldsymbol{A} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

As can be seen from the plot, we always move in the direction orthogonal to a level set, and stop at a point where we are tangent to a level set.

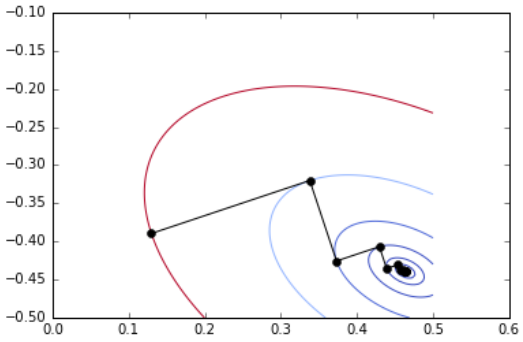

Figure 3.2: Trajectory of gradient descent

3.2 Step length selection

When moving in a descent direction $\boldsymbol{p}_k$ (not necessarily the steepest descent direction),

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

we can only guarantee that the function value will decrease if the step length α_k is small enough. If we move too far in a descent direction, we might even land at a point where f is larger than where we started! It is therefore important to choose a step length that

- is not too small (so that the algorithm does not take too long);
- is not too large (so that we don't end up at a point with larger function value);
- is easy to compute.

An optimal step α for a descent method would be the minimizer of the function

$$\alpha \mapsto \varphi(\alpha) := f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k).$$

In practice, minimizing this function is not always the most efficient thing to do (or even possible). One would rather choose a step length that satisfies some criteria that ensure that the sequence $\boldsymbol{x}_k$ converges to a minimizer $\boldsymbol{x}^*$ under suitable conditions on a function f. One such condition is a **sufficient decrease condition**,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c\alpha \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle =: \ell(\alpha),$$

with $c \in (0, 1)$. Note that $\varphi'(0) = \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle < 0$, because $\boldsymbol{p}_k$ is a descent direction. The function $\ell(\alpha)$ is therefore a line through $f(\boldsymbol{x}_k)$ with a slope $c\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle = c\varphi'(0) > \varphi'(0)$. To see why this condition is necessary, consider the function $f(x) = x^2 - 1$ and the sequence $x_k = \sqrt{1 + 1/k}$ for $k \geq 1$. Clearly, the sequence $f(\boldsymbol{x}_k) = 1/k$ decreases, but fails to converge to the minimizer $f(0) = -1$.

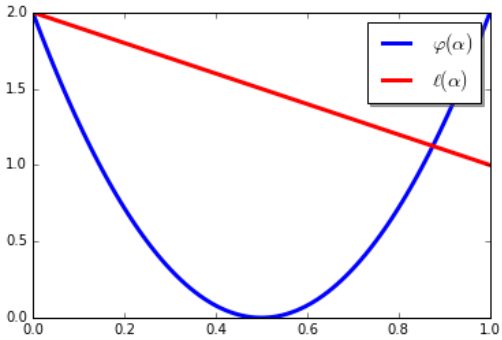

Figure 3.3: The sufficient decrease condition

The sufficient decrease condition (also called Armijo condition) can always be satisfied if α is chosen small enough, but the algorithm may become very slow. It is therefore common to supplement the sufficient descent condition with other criteria that guarantee that sufficient progress is made. Two of the commonly used criteria are:

- the Wolfe conditions, which add a *curvature condition*

  $$\varphi'(\alpha) \geq \tilde{c}\varphi'(0)$$

  for some $\tilde{c} \in (c, 1)$, which gives a lower bound on the slope of the new point;
- the Armijo-Goldstein conditions, which state that a step length α_k should additionally satisfy bound

  $$f(\boldsymbol{x}_k) + (1-c)\alpha_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle \leq f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k), \tag{3.1}$$

  which gives a lower bound on the step size.

Another common approach is **backtracking**: in this method one uses a high initial value of α (for example, $\alpha = 1$), and then decreases it until the sufficient descent condition is satisfied.

Example 3.1. Consider the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ defined by $f(\boldsymbol{x}) = x_1^2 + x_2^2$. The gradient is $\nabla f(\boldsymbol{x}) = 2\boldsymbol{x}$, and the φ function at $\boldsymbol{x}_k = (1,1)^\top$

$$\varphi(\alpha) = f(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)) = 2(1-2\alpha)^2, \quad \varphi'(\alpha) = -8(1-2\alpha).$$

The Armijo-Goldstein conditions (3.1) then state that we can choose α such that

$$2(1-4(1-c)\alpha) \leq 2(1-2\alpha)^2 \leq 2(1-4c\alpha).$$

The optimal step length in this case would be $\alpha = 0.5$.