Lecture 3

Most modern optimization methods are iterative: they generate a sequence of points $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots$ in $\mathbb{R}^d$ 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.

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 with $\|\boldsymbol{p}\|_2 = 1$ 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 $\alpha = 0$,

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

also known as the *directional derivative* of f 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 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

$$\boldsymbol{p} = -\frac{\nabla f(\boldsymbol{x}_k)}{\|\nabla f(\boldsymbol{x}_k)\|_2}.$$

In making a step $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$, the part of $\boldsymbol{p}_k$ that is of interest is only the direction, not the size: the latter can be adjusted using the step length parameter α_k. We can therefore choose $\boldsymbol{p}_k$ as in (3.1) as the direction of steepest descent.

Step length selection

The step length can then be chosen as the minimizer of the function

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

In practice minimizing this function is not always the most efficient (or even possible) thing to do. 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 set of conditions are the Armijo-Goldstein conditions, which state that a step length α should satisfy

$$\varphi(0) + (1-c)\cdot\alpha\cdot\varphi'(0) \leq \varphi(\alpha) \leq \varphi(0) + c\cdot\alpha\cdot\varphi'(0). \tag{3.2}$$

for a constant $c \in (0, 1/2)$ (typically of order 10^{-4}). Note that

$$\varphi(0) = f(\boldsymbol{x}_k), \quad \varphi(\alpha) = f(\boldsymbol{x}_k - \alpha\nabla f(\boldsymbol{x}_k)), \quad \varphi'(0) = -\|\nabla f(\boldsymbol{x}_k)\|_2^2,$$

so that the inequalities (3.2) can be written equivalently (after some rearranging) as

$$c\alpha\|\nabla f(\boldsymbol{x}_k)\|_2^2 \leq f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k - \alpha\nabla f(\boldsymbol{x}_k)) \leq (1-c)\alpha\|\nabla f(\boldsymbol{x}_k)\|_2^2$$

We explain these inequalities.

1. The right bound in (3.2) is a *sufficient decrease condition*: it ensures that $f(\boldsymbol{x}_{k+1})$ not only decreases, but decreases enough to converge to a local minimum. 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 $f(0) = -1$.

2. As the right bound can always be satisfied when α is small enough, the left-hand side is there to ensure that the step-length is not too short. A popular alternative is to replace the left-hand side by the *curvature condition* $\varphi'(\alpha) \geq \tilde{c}\varphi'(0)$ for some $\tilde{c} \in (c, 1)$, leading to what is know as the Wolfe conditions, but we will not discuss these at this point.

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 (3.2) then state that we can choose α such that

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

For the choice $c = 1/4$, the valid interval is part of the x-axis delimited by the vertical lines in Figure 3.1. The optimal step length in this case would be $\alpha = 0.5$.

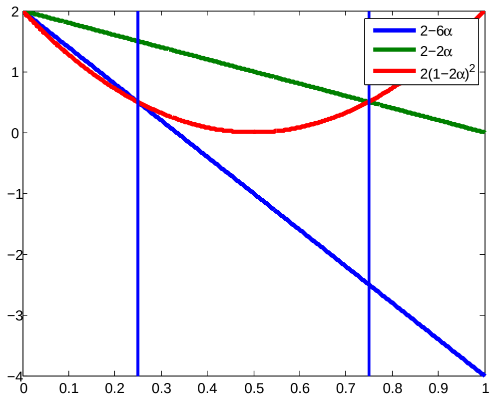

Figure 3.1: Choosing a step length.

Linear least squares

An important special case is when the function has the form

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

Recall from Problem (1.5) 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

$$\alpha \mapsto f(\boldsymbol{x}+\alpha\boldsymbol{A}^\top(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x})). \tag{3.3}$$

If we set $\boldsymbol{r} := \boldsymbol{A}^\top(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x})$ and compute the minimum of (3.3) by differentiating, we get the step length

$$\alpha = \frac{\boldsymbol{r}^\top\boldsymbol{r}}{\boldsymbol{r}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{r}}.$$

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 $\varepsilon > 0$.

Example 3.2. We test this method with the linear regression problem from Lecture 1, where we determinded the relationship $Y = \beta_0 + \beta_1 X$ of adult mass to basal metabolic rate in mammals. In this example, the matrix $\boldsymbol{A}$ is the 2×573 matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{573} \end{pmatrix},$$

where the x_i represent the mass of mammal i, and the vector $\boldsymbol{b}$ consists of the metabolic rate parameters. The 2-vector $\boldsymbol{x}$ represents the two values β_0 and β_1. A naive MATLAB code for gradient descent looks as follows.

```
function xout = graddesc(A,b,x,tol)
    r = A'*(b-A*x);
    while norm(r,2)>tol
        Ar = A*r;
        alpha = r'*r/(Ar'*Ar);
        x = x+alpha*r;
        r = r-alpha*A'*Ar;
    end
    xout = x;
end
```

The result is the same as when using the MATLAB solver or CVX, $\beta_0 = 1.36$, $\beta_1 = 0.70$.

In the next lecture we will introduce the concept of rate of convergence and analyse the rate of convergence of gradient descent.