Optimization

First order methods: Gradient descent

Learning goals

- LEARNING GOAL 1
- LEARNING GOAL 2

INTRODUCTION

Let f be the height of a mountain depending on the geographic coordinates (x_1, x_2)

$$f : \mathbb{R}^2 \to \mathbb{R}, \quad f(x_1, x_2) = y.$$

Goal: Reach the valley

$$\arg\min f(\mathbf{x})$$

Central idea: Iterative line search algorithms

1. At $\mathbf{x} \in \mathbb{R}^d$ we search for a **descent direction d** along which f decreases
2. Along **d** we go until f is sufficiently reduced (**step size control**).

"Walking down the hill, towards the valley."

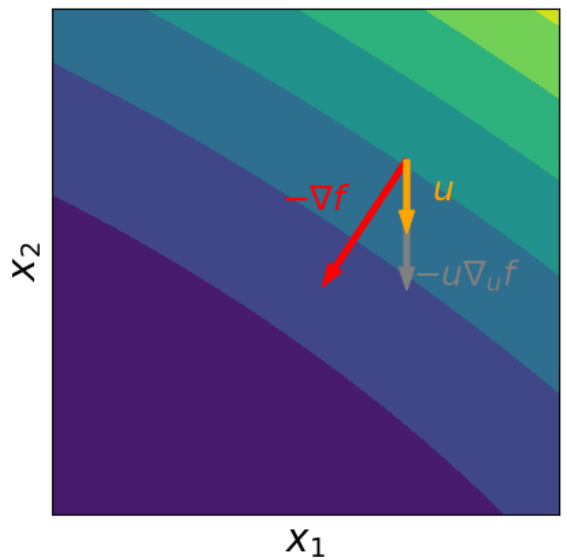
LINE SEARCH FOR SMOOTH FUNCTIONS

In the following, let $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable.

Definition: A vector $\mathbf{d} \in \mathbb{R}^d / \{\mathbf{0}\}$ is called **descent direction** in $\mathbf{x}$ if

$$\nabla_{\mathbf{d}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{d} < 0,$$

i.e., the directional derivative at $\mathbf{x}$ towards $\mathbf{d}$ is negative. (for the terminology, refer to chapter 1 **Mathematical concepts**.)



LINE SEARCH FOR SMOOTH FUNCTIONS

Line search algorithms can be summarized as follows:

Algorithm Line search

1: Choose a starting point $\mathbf{x}^{[0]} \in \mathbb{R}^d$

2: **for** $t = 0, 1, 2, \ldots$ **do**

3: Calculate a descent direction $\mathbf{d}^{[t]}$ for the current $\mathbf{x}^{[t]}$:

$$\nabla_{\mathbf{d}^{[t]}} f(\mathbf{x}^{[t]}) = (\mathbf{d}^{[t]})^\top \nabla f(\mathbf{x}^{[t]}) < \mathbf{0}$$

4: Find a $\alpha^{[t]}$ such that with $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + \alpha^{[t]}\mathbf{d}^{[t]}$ the function value becomes smaller, i.e. $f(\mathbf{x}^{[t+1]}) < f(\mathbf{x}^{[t]})$.

5: If no convergence has been achieved and the maximum number of iterations has not been exceeded, continue with step 2.

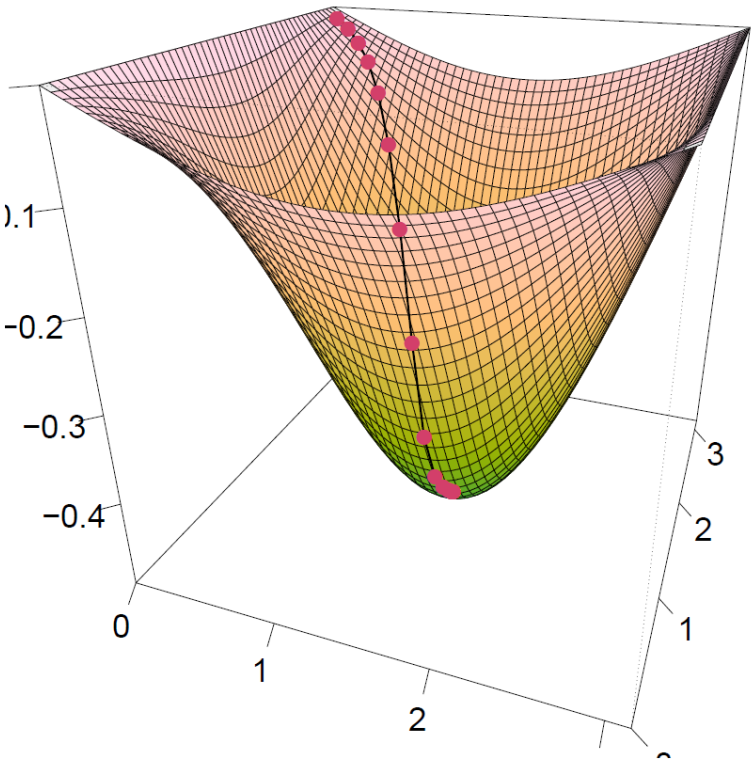
6: **end for**

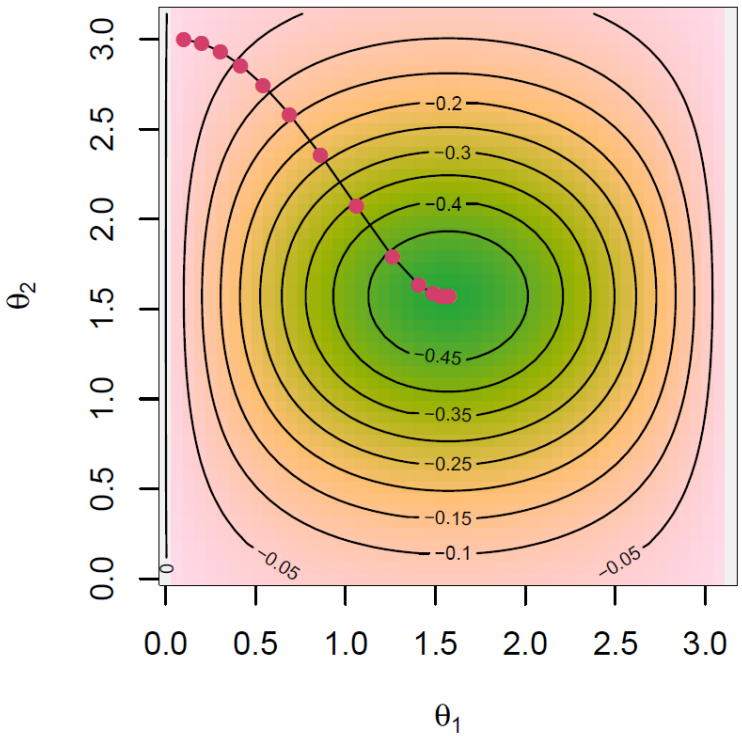
Please note that the terminology is misleading as “line search” on the one hand refers to Step 4 (selecting the step size that decreases $f(\mathbf{x})$) and on the other hand “line search” is the umbrella term for iterative algorithms that are based on finding a local minimum using descent methods.

GRADIENT DESCENT

Line search with the direction of the steepest descent (the negative gradient) is commonly referred to as **gradient descent** (GD).

$$f(x_1, x_2) = -\sin(x_1) \cdot \frac{1}{2\pi} \exp\left((x_2 - \pi/2)^2\right)$$





OPTIMIZING THE LINEAR MODEL WITH GD

Given are n observations $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$. We assume that the unknown connection between $\mathbf{x} \in \mathbb{R}^p$ and $y \in \mathbb{R}$ can be described as $f(\mathbf{x}) = y$.

Aim: Find a good estimate for f give $\mathcal{D}$.

Approach: We restrict f to the hypothesis space of linear functions $\mathcal{H} = f(\mathbf{x} \mid \boldsymbol{\theta}) = \{\boldsymbol{\theta}^\top \mathbf{x}, \boldsymbol{\theta} \in \mathbb{R}^p\}$. We use the quadratic loss as a loss function, and minimize the resulting empirical risk

$$\begin{aligned} \arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)^2 \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2 \end{aligned}$$

OPTIMIZING THE LINEAR MODEL WITH GD

The gradient of $\mathcal{R}(\boldsymbol{\theta})$ is

$$\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \frac{\partial \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{n} 2 \cdot \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right) \mathbf{x}^{(i)}.$$

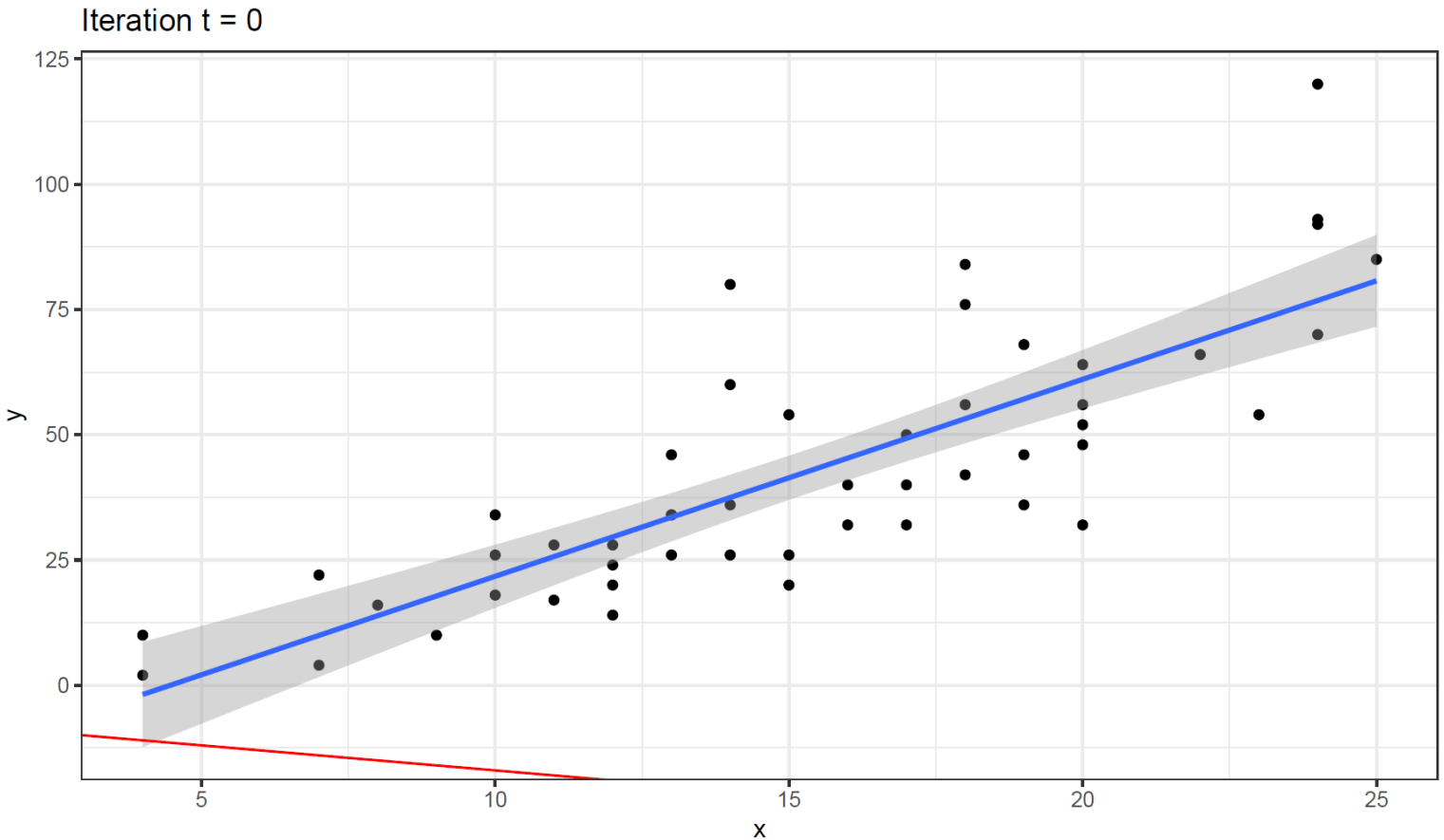
Therefore, an iteration of gradient descent is

$$\boldsymbol{\theta}^{[t+1]} \leftarrow \boldsymbol{\theta}^{[t]} + \alpha^{[t]} \sum_{i=1}^{n} 2 \cdot \left(y^{(i)} - \boldsymbol{\theta}^{[t]\top} \mathbf{x}^{(i)}\right) \mathbf{x}^{(i)}.$$

α determines the length of the step and is called **step size** or, in risk minimization, **learning rate**.

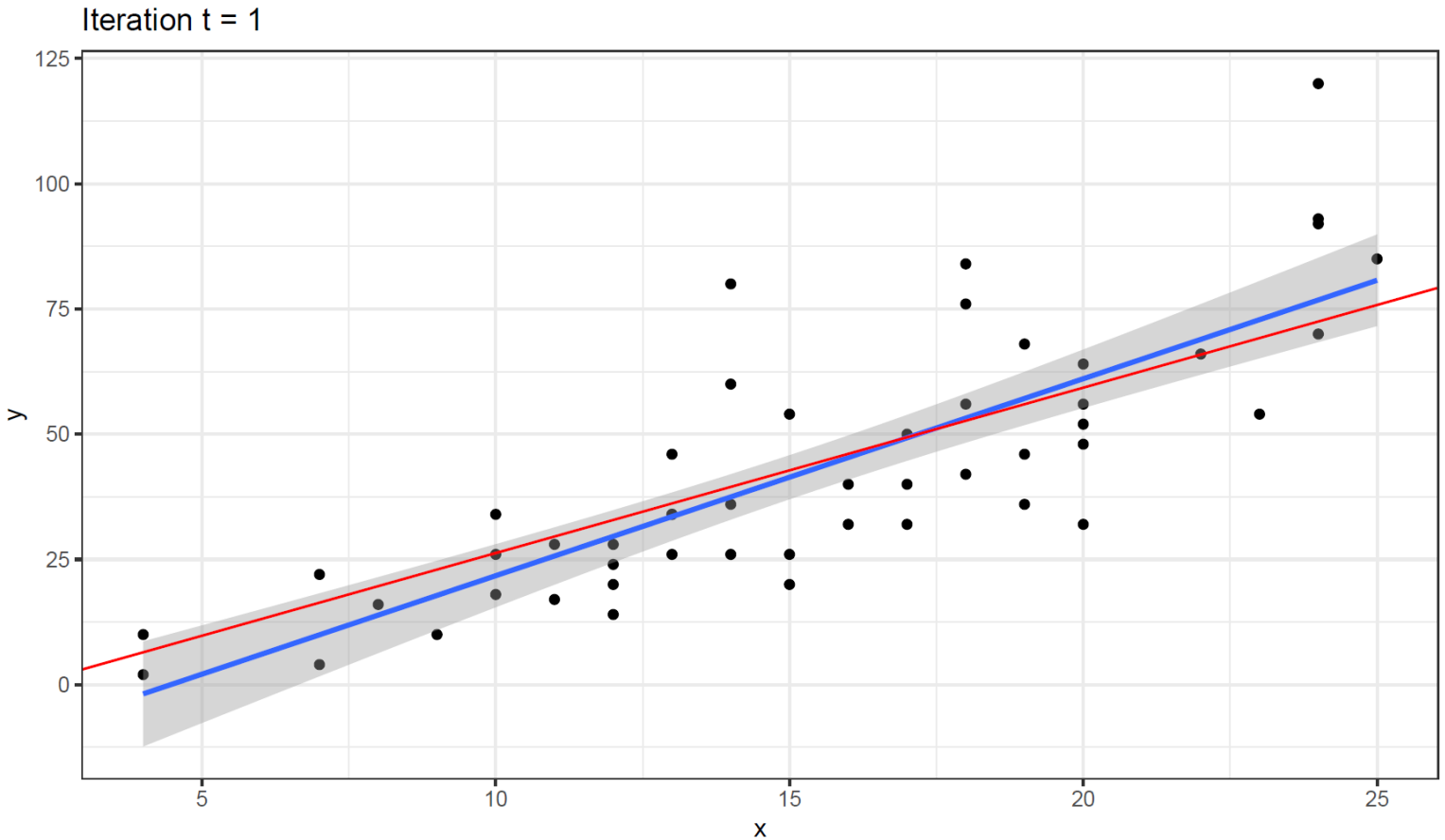
Note: For illustration, we optimize the linear model with quadratic loss with gradient descent, even though a closed-form solutions exists in this case.

OPTIMIZING THE LINEAR MODEL WITH GD



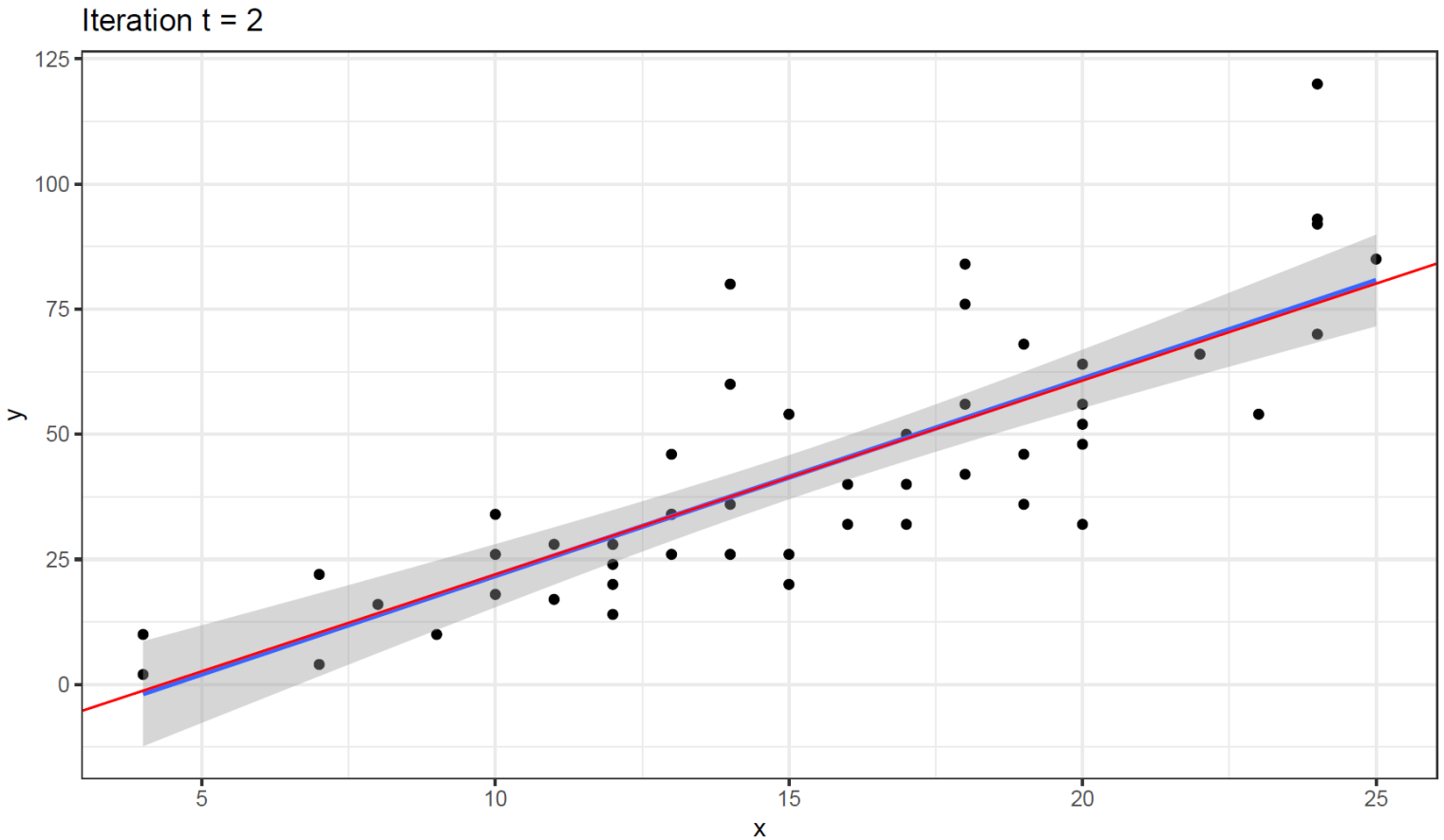
blue: true underlying relationship. red: linear model with parameters estimated by gradient descent.

OPTIMIZING THE LINEAR MODEL WITH GD



blue: true underlying relationship. red: linear model with parameters estimated by gradient descent.

OPTIMIZING THE LINEAR MODEL WITH GD



blue: true underlying relationship. red: linear model with parameters estimated by gradient descent.