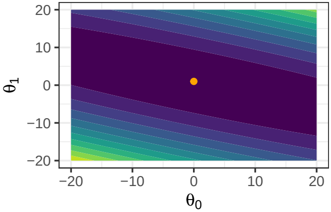
Optimization

Unconstrained problems



Learning goals

- Definition
- Practical examples

DEFINITION: OPTIMIZATION PROBLEM

$$\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$$

with objective function

$$f: \ \mathcal{S} \to \mathbb{R}.$$

The problem is called

- **unconstrained**, if the domain $\mathcal{S}$ is not restricted:

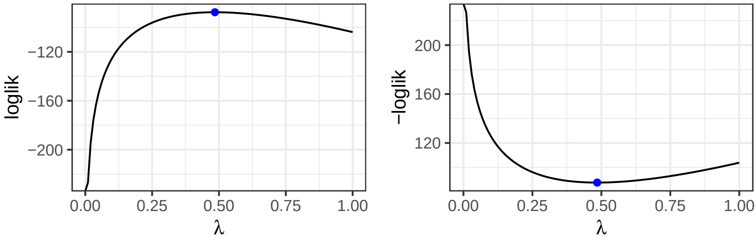
$$\mathcal{S} = \mathbb{R}^d$$

- **smooth** if f is at least $\in \mathcal{C}^1$
- **univariate** if $d = 1$, and **multivariate** if $d > 1$.

NOTE: A CONVENTION IN OPTIMIZATION

W.l.o.g., we always **minimize** functions f.

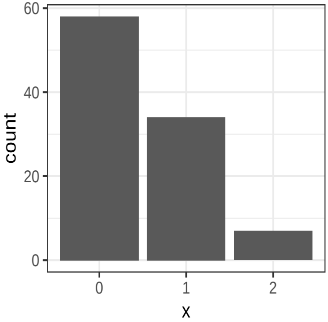
Maximization results from minimizing $-f$.



Poisson example: Maximizing the log-likelihood (left) is equivalent to minimizing the negative log-likelihood (right).

EXAMPLE 1.1: MAXIMUM LIKELIHOOD ESTIMATION: POISSON DISTRIBUTION

$\mathcal{D} = \left(x^{(1)}, ..., x^{(n)}\right)$ is sampled i.i.d. from density $f(x \mid \boldsymbol{\theta})$. We want to find λ which makes the observed data most likely.



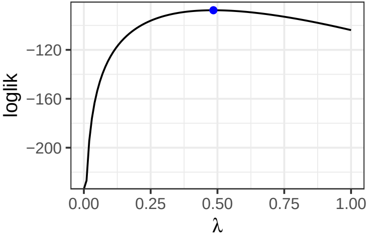
Example: Histogram of a sample drawn from a Poisson distribution
$f(k \mid \lambda) := \mathbb{P}(x = k) = \frac{\lambda^k \cdot \exp(-\lambda)}{k!}$.

EXAMPLE 1.1: MAXIMUM LIKELIHOOD ESTIMATION: POISSON DISTRIBUTION

We operationalize this as **maximizing** the log-likelihood function (or equivalently: minimizing the negative log-likelihood) with respect to λ:

$$\begin{aligned}\hat{\lambda} &= \text{arg min}_\lambda \; -\ell(\lambda, \mathcal{D}) = \text{arg min}_\lambda - \log \mathcal{L}(\lambda, \mathcal{D}) = \text{arg min}_\lambda - \log \prod_{i=1}^{n} f\left(\mathbf{x}^{(i)} \mid \lambda\right) \\ &= \text{arg min}_\lambda - \sum_{i=1}^{n} f\left(x^{(i)} \mid \lambda\right) = \text{arg min}_\lambda \sum_{i=1}^{n} \frac{-\lambda^{\mathbf{x}^{(i)}} \cdot \exp(-\lambda)}{\mathbf{x}^{(i)}!}\end{aligned}$$

EXAMPLE 1.1: MAXIMUM LIKELIHOOD ESTIMATION: POISSON DISTRIBUTION



Example: The log-likelihood of a Poisson distribution for data example above. The objective function is univariate and differentiable, and the domain is unconstrained.

EXAMPLE 1.2: MAXIMUM LIKELIHOOD ESTIMATION: NORMAL DISTRIBUTION

Density: $f(\mathbf{x} \mid \mu, \sigma) = \frac{1}{\sqrt{2\mu\sigma^2}} \; exp(\frac{-(\mathbf{x}-\mu)^2}{2\sigma^2})$

Since we want to have an univariate, unconstrained optimization problem, we set $\sigma = 1$ and estimate only μ, which is $\in \mathbb{R}$ and therefore unconstrained.

Likelihood: $\mathcal{L}(\mu, \sigma^2 \mid \mathbf{x}^{(i)}) = \sum_{i=1}^n f(\mathbf{x}^{(i)}) = (2\pi\sigma^2)^{-n/2} exp^{(-\frac{1}{2\sigma^2})} \sum_{i=1}^n (\mathbf{x}^{(i)} - \mu)^2$

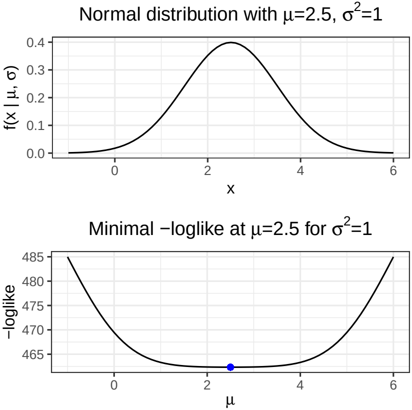
MLE$_\mu$: $\hat{\mu} = \text{arg min}_\mu - \ell(\mu, \sigma, \mathcal{D}) = \text{arg min}_\mu - \log \mathcal{L}(\mu, \sigma, \mathcal{D}) =$

$\text{arg min} - \log \left(\prod_{i=1}^n f(\mathbf{x}^{(i)} \mid \mu, \sigma) \right) \; = \; \text{arg min} \sum_{i=1}^n f(\mathbf{x}^{(i)} \mid \mu, \sigma) \; =$

$\text{arg min} \; \frac{n \log(2\pi\sigma^2)}{2} + \frac{1}{2\sigma^2} \sum_{i=1}^n (\mathbf{x}^{(i)} - \mu)^2$

$\implies \; \partial\mu \; f(\mu, \sigma^2) = \frac{1}{\sigma^2} \sum_{i=1}^n (\mathbf{x}^{(i)} - \mu) = 0 \; \implies \; \hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^{(i)}$

EXAMPLE 1.2: MAXIMUM LIKELIHOOD ESTIMATION: NORMAL DISTRIBUTION

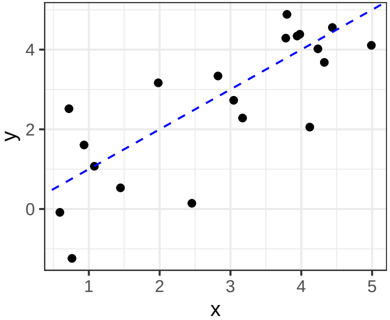


If we wanted to estimate σ as well, we would now have a multi- (/bi-) variate constrained optimization problem, since $\sigma > 0$. We will cover this problem type later in this lecture.

EXAMPLE 2: NORMAL REGRESSION

Assume a dataset $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$ generated according to

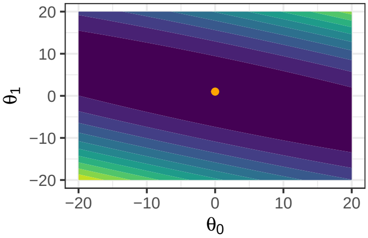
$$y^{(i)} = \boldsymbol{\theta}^\top \mathbf{x}^{(i)} + \epsilon^{(i)}, \qquad \epsilon^{(i)} \overset{iid}{\sim} \mathcal{N}(0, 1).$$



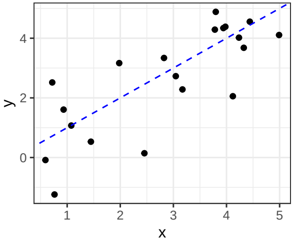
EXAMPLE 2: NORMAL LINEAR REGRESSION

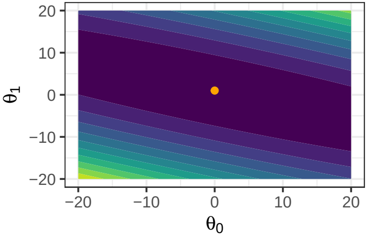
In normal linear regression the goal is to find a vector $\boldsymbol{\theta}$ which minimizes the sum of squared errors (SSE):

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2$$



EXAMPLE 2: NORMAL REGRESSION





- The problem is multivariate, smooth, and unconstrained
- Since the problem is a quadratic form, we easily obtain a geometric interpretation of the problem
- The problem has a closed-form solution, which is given by $\boldsymbol{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y}$, where $\mathbf{X}$ is the design matrix

EXAMPLE 3: RISK MIN. IN MACHINE LEARNING

- $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$ denotes a dataset where $f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$ is a model, parameterized by $\boldsymbol{\theta}$ (e.g. linear model).
- Let $L\left(y, f(\mathbf{x})\right)$ be the point-wise loss function which measures the error of a prediction $f(\mathbf{x})$ compared to the true output y.
- We want to find the model which minimizes the **empirical risk**

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right).$$

Formulate without θ and then explain why we usually parameterize the hypothesis space.

RISK MINIMIZATION IN MACHINE LEARNING

Machine learning consists of three components:

$$\textbf{Machine Learning} = \underbrace{\textbf{Hypothesis Space + Risk}}_{\text{Formulating the optimization problem}} + \underbrace{\textbf{Optimization}}_{\text{Solving it}}$$

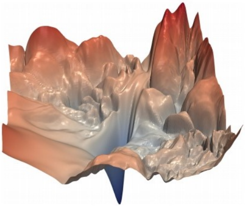
- **Hypothesis Space:** Define (and restrict!) what kind of model f can be learned from the data.
- **Risk:** Define the risk function $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ that quantifies how well a specific model performs on a given data set via a suitable loss function L.
- **Optimization:** Solve the resulting optimization problem through optimizing the risk $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ over the hypothesis space.

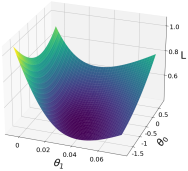
RISK MINIMIZATION IN MACHINE LEARNING

The (computational) complexity of the optimization problem

$$\arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$$

and hence the choice of the numerical optimization algorithm is influenced by the model structure and the choice of the loss function:, i.e., smoothness, convexity.





Loss landscapes of ML problems.

Left: ResNet-56, right: Logistic regression with cross-entropy loss

Source: `https://arxiv.org/pdf/1712.09913.pdf`