Exercise 2 – Regression

Introduction to Machine Learning

Hint: Useful libraries

R

```
# Consider the following libraries for this exercise sheet:

library(ggplot2)
library(mlr3verse)
library(mlr3learners)
library(mlr3viz)
library(quantreg)
```

Python

```
# Consider the following libraries for this exercise sheet:

# general
import numpy as np
import pandas as pd
import math
# plots
import matplotlib.pyplot as plt
import seaborn as sns
# sklearn
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
import sklearn.metrics as metrics
```

```
from sklearn.metrics import mean_absolute_error
```

Exercise 1: HRO in coding frameworks

> Learning goals
>
> Translate lecture concepts to code

Throughout the lecture, we will frequently use the `R` package `mlr3`, resp. the `Python` package `sklearn`, and its descendants, providing an integrated ecosystem for all common machine learning tasks. Let's recap the HRO principle and see how it is reflected in either `mlr3` or `sklearn`. An overview of the most important objects and their usage, illustrated with numerous examples, can be found at the `mlr3` book and the `scikit` documentation.

How are the key concepts (i.e., hypothesis space, risk and optimization) you learned about in the lecture videos implemented?

Have a look at`mlr3::tsk("iris")` / `sklearn.datasets.load_iris`. What attributes does this object store?

Instantiate a regression tree learner (`lrn("regr.rpart")` / `DecisionTreeRegressor`). What are the different settings for this learner?

Hint

R

`mlr3::mlr_learners$keys()` shows all available learners.

Python

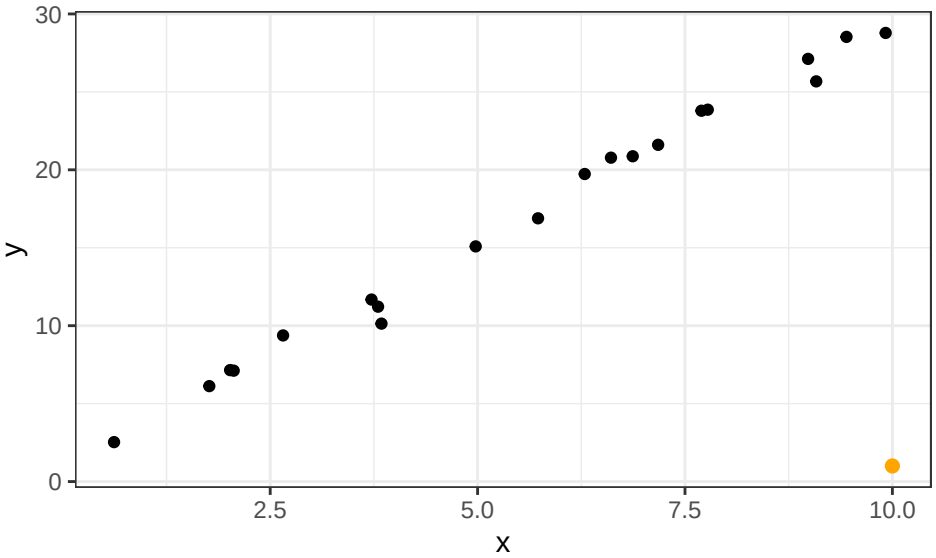
Use `get_params()` to see all available settings.

Exercise 2: Loss functions for regression tasks

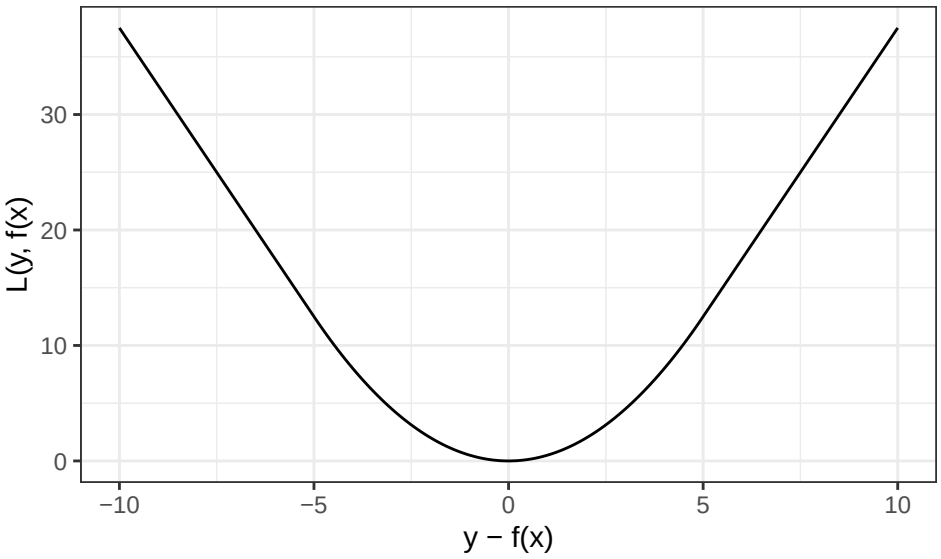
Learning goals

1. Assess how outliers affect models for different loss functions
2. Derive impact of a loss function from visual representation

In this exercise, we will examine loss functions for regression tasks somewhat more in depth.



Consider the above linear regression task. How will the model parameters be affected by adding the new outlier point (orange) if you use $L1$ loss and $L2$ loss, respectively, in the empirical risk? (You do not need to actually compute the parameter values.)



The second plot visualizes another loss function popular in regression tasks, the so-called *Huber loss* (depending on $\epsilon > 0$; here: $\epsilon = 5$). Describe how the Huber loss deals with residuals as compared to $L1$ and $L2$ loss. Can you guess its definition?

Exercise 3: Polynomial regression

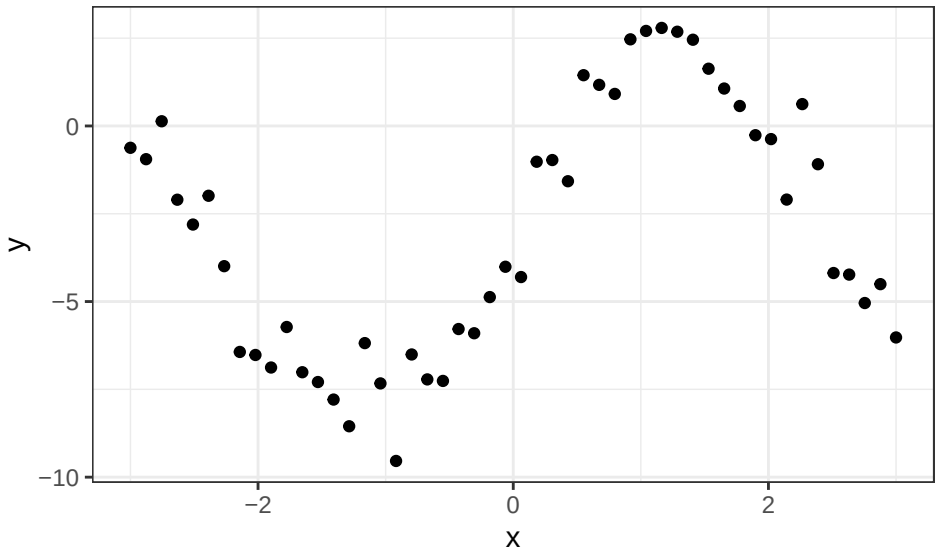
Learning goals

1. Express HRO components for polynomial regression
2. Derive gradient update for optimization
3. Analyze and discuss learner flexibility

Assume the following (noisy) data-generating process from which we have observed 50 realizations:

$$y = -3 + 5 \cdot \sin(0.4\pi x) + \epsilon$$

with $\epsilon \sim \mathcal{N}(0, 1)$.



We decide to model the data with a cubic polynomial (including intercept term). State the corresponding hypothesis space.

State the empirical risk w.r.t. θ for a member of the hypothesis space. Use $L2$ loss and be as explicit as possible.

Only for lecture group A

We can minimize this risk using gradient descent. Derive the gradient of the empirical risk w.r.t θ.

Only for lecture group A

Using the result for the gradient, explain how to update the current parameter $\theta^{[t]}$ in a step of gradient descent.

You will not be able to fit the data perfectly with a cubic polynomial. Describe the advantages and disadvantages that a more flexible model class would have. Would you opt for a more flexible learner?

Exercise 4: Predicting `abalone`

> Learning goals
>
> 1. Implement regression model
> 2. Analyze basic regression fit

We want to predict the age of an abalone using its longest shell measurement and its weight. The `abalone` data can be found here: https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data.

Prepare the data as follows:

R

```r
# Download data
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
abalone <- read.table(url, sep = ",", row.names = NULL)
colnames(abalone) <- c(
  "sex", "longest_shell", "diameter", "height", "whole_weight",
  "shucked_weight", "visceral_weight", "shell_weight", "rings"
)

# Reduce to relevant columns
abalone <- abalone[, c("longest_shell", "whole_weight", "rings")]
```

Python

```python
# load data from url

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
abalone = pd.read_csv(
    url,
```

```
    sep=',',
    names=[
        'sex',
        "longest_shell",
        "diameter",
        "height",
        "whole_weight",
        "shucked_weight",
        "visceral_weight",
        "shell_weight",
        "rings"
    ]
)

abalone = abalone[['longest_shell', 'whole_weight', 'rings']]
print(abalone.head)
```

```
<bound method NDFrame.head of       longest_shell  whole_weight  rings
0             0.455        0.5140     15
1             0.350        0.2255      7
2             0.530        0.6770      9
3             0.440        0.5160     10
4             0.330        0.2050      7
...             ...           ...    ...
4172          0.565        0.8870     11
4173          0.590        0.9660     10
4174          0.600        1.1760      9
4175          0.625        1.0945     10
4176          0.710        1.9485     12

[4177 rows x 3 columns]>
```

Plot `LongestShell` and `WholeWeight` on the x- and y-axis, respectively, and color points according to `Rings`.

Using `mlr3`/`sklearn`, fit a linear regression model to the data.

Compare the fitted and observed targets visually.

R Hint

R

Use `$autoplot()` from `mlr3viz`.

Assess the model's training loss in terms of MAE.

Hint

R

Call `$score()`, which accepts different `mlr_measures`, on the prediction object.

Python

Call `from sklearn.metrics import mean_absolute_error`.