



Project high-dimensional regression: Overfitting

- (a) Implement a function in R which computes the ridge regression estimator given the observations of the target and design values and with a general tuning parameter λ for the penalization to be chosen as an input. You shall not use packages that readily include the ridge estimator, but you can use the function [optim](#), or similar ones, to solve optimization problems (similar as for least squares in the course).
- (b) Install the package [glmnet](#) and read the description to use it to compute the lasso estimator.
- (c) We aim to illustrate the overfitting phenomenon for least squares and how regularization can help. Generate data from the polynomial regression model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_p x_i^p + \epsilon_i, \quad 1 \leq i \leq n,$$

with a “sparse” parameter vector $\beta = (0, 0, 2, 0, \dots, 0)^\top$, i.e. the model parameters are set to zero except for the quadratic term. Simulate for the design $x_i \sim \mathcal{U}([-1, 1])$, i.e. uniformly distributed on $[-1, 1]$, and $\epsilon_i \sim \mathcal{N}(0, 1)$ i.i.d. standard normal with $n = 100$. Observations of the response are hence noisy observations of the square function, but this is unknown to the statistician beforehand. Estimate the parameters for $p = 20$ with least squares, the ridge and the lasso estimator, respectively. You can use [cv.glmnet](#) to choose a suitable λ for the lasso and simply take the same λ for the ridge estimator. Compare in a plot the true function and the noisy observations with fits based on the estimated parameters with all three methods. Iterating the procedure shows that the plots can vary from time to time. Therefore, run a Monte Carlo simulation and plot Monte-Carlo averages (and quantiles). You can keep the design fix during all iterations, but re-simulate the noise in each step. Interpret the results.

The project results are discussed in a presentation and a concise report is submitted.