FYS-STK4155 - Applied data analysis and machine learning

Project 1

Even M. Nordhagen

September 12, 2018

- Github repository containing programs and results:
  `https://github.com/evenmn/FYS-STK4155`

Abstract

Do not forget to be specific

Contents

1 Introduction

Should write some motivating words about how much regression is used in different fields etc..

2 Theory

2.1 Regression

A few general words about regression

2.1.1 Ordinary Least Square (OLS)

Image that we have a set of points $\{(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\}$, and we want to fit a p'th order polynomial to them. The most intuitive way would be to find coefficients $\vec{\beta}$ which minimize the error in

$$\begin{aligned} y_0 &= \beta_0 x_0^0 + \beta_1 x_0^1 + \ldots + \beta_p x_0^p + \varepsilon_0 \\ y_1 &= \beta_0 x_1^0 + \beta_1 x_1^1 + \ldots + \beta_p x_1^p + \varepsilon_1 \\ &\vdots \\ y_{n-1} &= \beta_0 x_{n-1}^0 + \beta_1 x_{n-1}^1 + \ldots + \beta_p x_{n-1}^p + \varepsilon_{n-1}, \end{aligned}$$

which for OLS is defines as

$$\text{MSE} = \sum_i \varepsilon_i^2 \tag{1}$$

$$\vec{y} = \hat{X}^T \vec{\beta} + \vec{\varepsilon} \tag{2}$$

Confidence interval of $\hat{\beta}$: Var$(\hat{\beta})$.

2.1.2 Ridge regression

Avoid singularities (OLS breaks down if one of the diagonal elements is zero), add penalty λ.

2.1.3 Lasso regression

2.2 Higher order regression

2.2.1 Terrain

Mention the Franke Function

$$
\begin{aligned}
f(x,y) = {} & \frac{3}{4}\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right) \\
& + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right).
\end{aligned}
$$

2.2.2 Higher order

Although we stick to 2D regression in this project, I add this section for completeness.

2.3 Error analysis

Cost function (loss function)
Different methods to estimate error:

- Absolute error
- Relative error
- Mean square error (MSE)
- R^2 score function

3 Methods

3.1 Resampling techniques

A resampling technique is.. There are plenty of resampling techniques, and we have already went through several of them in this course:

- Validation set approaches
- Leave one out validation
- Jackknife resampling
- K-fold validation
- Bootstrap method
- Blocking method.

For this particular project we have been focusing on the bootstrap and the k-fold validation methods, so here I will cover them only

3.1.1 Bootstrap method

3.1.2 K-fold validation method

3.2 Singular Value Decomposition (SVD)

3.3 Minimization methods

When the interaction term is excluded, we know which α that corresponds to the energy minimum, and it is in principle no need to try different α's. However, sometimes we have no idea where to search for the minimum point, and we need to try various α values to determine the lowest energy. If we do not know where to start searching, this can be a time consuming activity. Would it not be nice if the program could do this for us?

In fact there are multiple techniques for doing this, where the most complicated ones obviously also are the best. Anyway, in this project we will have good initial guesses, and are therefore not in need for the most fancy algorithms.

3.3.1 Gradient Descent

Perhaps the simplest and most intuitive method for finding the minimum is the gradient descent method (GD), which reads

$$\alpha^+ = \alpha - \eta \cdot \frac{d\langle E(\alpha)\rangle}{d\alpha}. \tag{3}$$

where α^+ is the updated α and η is a step size. The idea is that one finds the gradient of the energy with respect to a certain α, and moves in the direction which minimizes the energy. This is repeated until one has found an energy minimum, where the energy minimum is defined as either where $\frac{d\langle E(\alpha)\rangle}{d\alpha}$ is smaller than a given tolerance, or the energy fluctuates around a value are smaller than a tolerance, and thus changes minimally.

To implement equation 2, we need an expression for the derivative of E with respect to alpha:

$$\bar{E}_\alpha = \frac{d\langle E(\alpha)\rangle}{d\alpha}. \tag{4}$$

By using the expression for the expectation value for the energy $\langle E(\alpha)\rangle$ in equation 4

$$\langle E(\alpha)\rangle = \frac{\langle\psi_T(\alpha)|H|\psi_T(\alpha)\rangle}{\langle\psi_T(\alpha)|\psi_T(\alpha)\rangle} \tag{5}$$

and applying the chain rule of differentiation, it can be shown that equation 3 is equal to equation 5

$$\bar{E_\alpha} = 2\left[\langle E_L(\alpha)\frac{\bar{\psi_\alpha}}{\psi_\alpha}\rangle - \langle E_L(\alpha)\rangle\langle\frac{\bar{\psi_\alpha}}{\psi_\alpha}\rangle\right] \tag{6}$$

where

$$\bar{\psi_\alpha} = \frac{d\psi(\alpha)}{d\alpha}. \tag{7}$$

The algorithm of this minimization method is thus as follows:

```
for(max number of iterations with minimizing)

        do M Monte Carlo cycles
        calculate E and dE/dalpha

        Check if dE/dalpha < eps or alpha fluctuation over the last 5 steps
            ↪ is < eps


                if yes, print optimal alpha and break loop
                if no, continue to next iteration
```

4 Code

4.1 Code structure

4.2 Implementation

4.3 Optimalization

5 Results

6 Discussion

7 Conclusion

A Appendix A