project 1, FYS-STK4155

Regression and Resampling of Terrain data and the Franke function

A. G. Eriksen

(Dated: October 10, 2020)

The project motivation lies in exploring linear regression methods with regards to simulated and real terrain data.

The Ordinary Least Squares method has been explored for the simulated terrain, both with and without resampling methods.

The resampling methods used are the bootstrap and k-fold Cross validation methods.

The intent was to implement Ridge regression and Lasso regression for the simulated data and then move on to the real terrain, but time ran out and bugs kept popping up. The only data here are from the OLS

CONTENTS

I. INTRODUCTION

The aims and rationales of the project. What we have done in the project. Brief summary of structure of the report.

The aim for the project was to explore regression methods for predicting an assumed functional dependence. To build up and test the methods we used a function called the ”Franke function”[1] to simulate a terrain and provide a definitive functional dependency between the input data and the targets.

The initial regression was made using various polynomial interpolations as the basis, and studying the results of the fit for various models to try and map out error estimates in regards to the complexity of the models.

Once the simplest models were tested we added resampling of the generated data using the bootstrap method to improve and explore the accuracy of the models, exploring the bias- variance trade-off in particular.

We move on to compare with a different, popular resampling method called k-fold Cross- Validation(k-fold CV). This would provide some comparisons with the bootstrap method.

Following this, the intent was to implement a few more regression methods and explore reductions in the variance of the model predictions from penalising specific features of the model, resulting in more stable predictions. Once this was done we could bring the code to actual terrain data, whose assumed functional dependency we could not know, nor really could guarantee and go over the various regression and resampling methods for the predictions. Time ran out and this remains unexplored.

II. METHOD

Theoretical models and technicalities

The basis for the regressions used here, is the Franke function[1]

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

The function is defined for values $x, y \in [0,1]$. To make the simulation more authentic, noise can be added when generating the targets through the function, with a normally distributed noise, ε. Here ε follows a normal distribution of $\mathcal{N}(0,1)$, $\sigma^2 = 1$. Though the strength of the noise can be varied to highlight statistical effects.

The input x and y values are generated as arrays, either through a normal distribution of numbers in the range and then sorted, or just as a range or linearly spaced array with a given amount of elements. Though to get a proper terrain grid, we need to mesh the 2 together. Once this is done, we generally flatten the arrays before working on them to simplify things.

The regression model can be written as a matrix-vector product,

$$\tilde{y} = \mathbf{X}\beta. \tag{2}$$

This is based on an assumption that the target values are dependent on the input variables along a model like

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \varepsilon \tag{3}$$

This allows us to define a cost function dependent on β, it's derivative and the optimal β:[2] [3]

$$C(\boldsymbol{\beta}) = \frac{1}{n}\{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^2\}, \tag{4}$$

$$\frac{\partial C(\boldsymbol{\beta}}{\partial \boldsymbol{\beta}} = 0 = \boldsymbol{X}^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \tag{5}$$

$$\hat{\boldsymbol{\beta}} = \Big(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\Big)^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}. \tag{6}$$

```
**Input:** $\boldsymbol{x}$ and $\boldsymbol{y}$ and polynomial degree $n$
**Output:** Feature matrix X with dimension $mxn$
$n \leftarrow length(\boldsymbol{x})$
$m \leftarrow int((n+1) \cdot (n+2)/2)$
$X \leftarrow matrix.ones(nxm)$
**for** $i \leftarrow 1$ ***to*** $n+1$ **do**
    $q \leftarrow int((i) * (i+1)/2)$
    **for** $k \leftarrow 0$ ***to*** $i+1$ **do**
        $X_{(:,q+k)} \leftarrow x^{i-k} \cdot y^k$
    **end**
**end**
**return** $X$
```

Algorithm 1: make feature matrix X given input $\vec{x}, \vec{y}$ and dimension n

With an expression for the optimal beta, we now have to find a way to build our feature matrix for a polynomial of the n-th degree, to allow for varying model complexity.

with an algorithm for the feature matrix and a set of targets generated by the Franke function we just need to fit the features and predict an output. Following this, we can apply various statistics to compare the model to the targets. These would include

$$MSE(y,\tilde{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2 = \mathbb{E}[(y-\tilde{y})^2], \tag{7}$$

$$R^2(y,\tilde{y}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \mathbb{E}[y])^2}, \tag{8}$$

$$\mathbb{E}[y] = \frac{1}{n}\sum_{i=0}^{n-1} y_i, \tag{9}$$

$$\mathbb{E}[(y-\tilde{y})^2] = ... = \mathbb{E}[(y-\mathbb{E}[\tilde{y}])^2] + \mathbb{E}[(\tilde{y}-\mathbb{E}[\tilde{y}])^2] + \sigma^2, \tag{10}$$

$$\mathbf{Bias}(y,\tilde{y} = \mathbb{E}[(y-\mathbb{E}[\tilde{y}])^2], \tag{11}$$

$$\mathbf{Var}(\tilde{y}) = \mathbb{E}[(\tilde{y}-\mathbb{E}[\tilde{y}])^2], \tag{12}$$

$$MSE(y,\tilde{y}) = \mathbf{Bias}(y,\tilde{y}) + \mathbf{Var}(\tilde{y}) + \sigma^2. \tag{13}$$

Moving on from this, we can begin to refine the data we have somewhat, using resampling methods. The ones we will make use of, are Bootstrap and k-fold Cross Validation.

The essence here, is that we have a limited data set. To compensate for this, methods of selecting which data to run allows us to make the best use of what data we do have.

```
Input: feature matrix, $X$, targets, $y$, and number of bootstraps, $N$
Output: model fits and predictions
$\boldsymbol{\tilde{y}^{fit}} \leftarrow \textbf{array}(N)$
$\boldsymbol{\tilde{y}^{predict}} \leftarrow \textbf{array}(N)$
$\boldsymbol{\beta} \leftarrow \textbf{array}(N)$
for $i \leftarrow 0$ to $N$ do
    Shuffle $X$ and $y$
    Split data into training and test sets
    $\beta \leftarrow OLS(X_{train}, y_{train})$
    $\tilde{y}_i^{fit} \leftarrow \boldsymbol{X_{train}\beta}$
    $\tilde{y}_i^{predict} \leftarrow \boldsymbol{X_{test}\beta}$
end
return $\boldsymbol{\beta}$, $\boldsymbol{\tilde{y}_{fit}}$, $\boldsymbol{\tilde{y}_{predict}}$
```

Algorithm 2: The Bootstrap method of resampling

III. RESULTS

Results of study and discussion of results

IV. CONCLUSION

Beyond discussion, this is actually concluding things. perspectives of study

```
Input: feature matrix, $\boldsymbol{X}$, targets, $\boldsymbol{y}$, and number of folds, $k$
Output: model fits and predictions
$\tilde{y}^{fit} \leftarrow \textbf{array}(k)$
$\tilde{y}^{predict} \leftarrow \textbf{array}(k)$
$\boldsymbol{\beta} \leftarrow \textbf{array}(k)$
Split $k$ folds $\boldsymbol{\mu}, \boldsymbol{\nu} \subset \boldsymbol{X}, \boldsymbol{y}$
for $i \leftarrow 0$ to $k$ do
    $\boldsymbol{X}^{\boldsymbol{test}} \leftarrow \boldsymbol{X}\{\mu_i\}$
    $\boldsymbol{X}^{\boldsymbol{train}} \leftarrow \boldsymbol{X}\{\boldsymbol{\mu} - \mu_i\}$
    $\boldsymbol{y}^{\boldsymbol{test}} \leftarrow \boldsymbol{y}\{\nu_i\}$
    $\boldsymbol{y}^{\boldsymbol{train}} \leftarrow \boldsymbol{y}\{\boldsymbol{\nu} - \nu_i\}$
    $\beta \leftarrow OLS(X_{train}, y_{train})$
    $\tilde{y}_i^{fit} \leftarrow \boldsymbol{X}_{\boldsymbol{train}}\boldsymbol{\beta}$
    $\tilde{y}_i^{predict} \leftarrow \boldsymbol{X}_{\boldsymbol{test}}\boldsymbol{\beta}$
end
return $\boldsymbol{beta}$, $\tilde{\boldsymbol{y}}^{\boldsymbol{fit}}$, $\tilde{\boldsymbol{y}}^{\boldsymbol{predict}}$
```

Algorithm 3: k-fold Cross Validation method of resampling

V. APPENDIX

extra material, e.g. superfluous code, tables and figures not fitting into the text itself.

[1] R. Franke, *A critical comparison of some methods for interpolation of scattered data*, Tech. Rep. (NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 1979).

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction* (Springer Science & Business Media, 2009).

[3] M. Hjorth-Jensen, Lecturenotes fys-stk4155 – applied data analysis and machine learning (2020).