Draft version October 7, 2019
Preprint typeset using LaTeX style emulateapj v. 12/16/11

SURFACE FITTING OF TWO DIMENSIONAL TERRAIN DATA USING LINEAR REGRESSION

Sigbjrn Foss

Draft version October 7, 2019

ABSTRACT

We explore the properties of OLS, Lasso and Ridge regression on a bivariate function with normally distributed noise. The methods are the used to fit a surface using terrain data. Reading of the mean square error and R^2-score for the different regression methods methods show that the OLS regression method provides the best fit for this data set.

Github repository with all source files can be found at this link.

1. INTRODUCTION

Linear regression problems provide a useful starting point for exploring core concepts in supervised learning. Central properties in supervised learning can be expressed analytically in linear regression, providing some intuition regarding their behaviours, which can then be extrapolated to more complicated methods. This paper documents an exploration of regression methods, using the Franke function with normally distributed noise as input data to demonstrate some concepts in linear regression. The regularization methods ridge and lasso are also applied, and compared to the standard OLS. The methods are implemented using cross validation both for testing and training, and the resulting bias, variance, MSE and R^2-scores are analyzed. When the methods are developed sufficiently, they are applied to real terrain data. The methods section provides an outline of the derivation of the three regression methods, and some statistical considerations. The application of these methods to the terrain data is showcased in the results and conslusion sections.

2. METHODS

2.1. *Linear regression*

The digital terrain data is to be fitted by a two variable polynomial. The data set is given in the form of an $n \times n$ matrix, containing the z-values over a grid $\{(x, y)\}$, and is assumed to be on the form

$$z(x, y) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 xy + ... + \beta_P x^p y^p + \varepsilon$$

of order some order (p_x, p_y). For a given order (p_x, p_y), the number of coefficients is $P = (p_x + 1)(p_y + 1)$ due to the cross terms. The points on the surface are then given by

$$\begin{aligned}
\tilde{z}_1 &= \beta_0 + \beta_1 x_1 + \beta_2 y_1 + \ldots + \beta_P x_1^p y_1^p + \varepsilon_1 \\
\tilde{z}_2 &= \beta_0 + \beta_1 x_2 + \beta_2 y_2 + \ldots + \beta_P x_2^p y_2^p + \varepsilon_2 \\
&\vdots \\
z_{n^2} &= \beta_0 + \beta_1 x_{n^2} + \beta_2 y_{n^2} + \ldots + \beta_P x_{n^2}^p y_{n^2}^p + \varepsilon_{n^2}.
\end{aligned}$$

This set of equations is compactly expressed by the matrix vector equation

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

With $\boldsymbol{z} \in \mathbb{R}^N$, with $N = n^2$ and $\boldsymbol{X} \in \mathbb{R}^{N \times P}$. The aim of regression is then to find some function $\tilde{\boldsymbol{z}} = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$ which represents the data $\boldsymbol{z}$ as closely as possible. This is done by defining a cost function which describes the distance between the data points and the fitted points, and minimizing this function w.r.t. the estimators $\tilde{\boldsymbol{\beta}}$. The difference in the three regression methods to be discussed is essentially how the cost function is defined.

2.2. *Ordinary least square (OLS) regression*

The OLS method is derived by defining the cost function as the squared difference

$$Q(\tilde{\boldsymbol{\beta}}) = \sum_{i=0}^{N-1} (z_i - \tilde{z}_i)^2,$$

or in matrix vector notation,

$$Q(\tilde{\boldsymbol{\beta}}) = (\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^T (\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}).$$

To minimize the cost function we set

$$\frac{dQ(\tilde{\boldsymbol{\beta}})}{d\tilde{\boldsymbol{\beta}}} = 0$$

This derivative can be found to be

$$\frac{dQ(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = -2\boldsymbol{X}^T(\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}) = 0,$$

(Hastie, Tibshirami, Friedman, p. 45). If $\boldsymbol{X}^T\boldsymbol{X}$ is invertible, the estimators are given by

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{z}.$$

The matrix inversion restricts this method by excluding design matrices for which $\boldsymbol{X}^T\boldsymbol{X}$ is singular or near singular. However, recognizing the pseudo-inverse in the expression for the estimators,

$$\boldsymbol{X}^+ = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T,$$

the inversion can be replaced by a singular value decompostion, two transpositions and an inversion of a well conditioned, diagonal matrix. The SVD decomposition of $\boldsymbol{X}$ is

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

where $\boldsymbol{U} \in \mathbb{R}^{N \times N}$ and $\boldsymbol{V} \in \mathbb{R}^{P \times P}$ are unitary matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times P}$ is diagonal with entries $\Sigma_{i,i} > 0$ (Hjort-Jensen, page 14). In practice, the $N - P$ last rows of $\boldsymbol{\Sigma}$

and columns of $\boldsymbol{U}$ may be omitted as $\Sigma_{ij} = 0$ for $i \neq j$. Because $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary,

$$\begin{aligned}\boldsymbol{X}^{+} &= \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T}\right)^{+} \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{T} \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}.\end{aligned}$$

This algorithm is more robust at the cost of having higher computational complexity.

2.2.1. *Regularization*

Another approach to avoiding singularity is to add some regularization parameter to the cost function. The ridge and lasso methods are versions of OLS regression, only with such parameter in the cost function. The parameter in the ridge method is $\lambda||\boldsymbol{\beta}||^2$, and for lasso, $\lambda||\boldsymbol{\beta}||_1$. The cost function for each of these parameters are then

$$Q(\tilde{\boldsymbol{\beta}}) = (\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^T(\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}) + \lambda\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}},$$

and

$$Q(\tilde{\boldsymbol{\beta}}) = (\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^T(\boldsymbol{z} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}) + \lambda||\tilde{\boldsymbol{\beta}}||_1,$$

where $||\tilde{\boldsymbol{\beta}}||_1$ is to be understood as the sum of the absolute value of the elements in $\tilde{\boldsymbol{\beta}}$. For the ridge penalty, the regression coefficients can be found to be

$$\tilde{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{z},$$

by the same calculation as for the OLS coefficients. Therefore, λ acts as a shrinkage parameter, where increasing it shrinks the coefficients. It also gives us greater control over the bias and variance of the fit. The discussion in Hastie et. al. page 66, shows how the regression coefficients are penalized according to size. This leads to a dampening effect where, if we choose a larger value of λ, the outlying data points with be weighted less and the variance will be reduced. This of course comes with a cost. When increasing the value of λ, the fit will approach a linear function and the bias increases. As a consequence, implementation of this method requires some amount of finesse in tuning the λ parameter, in contrast to the brute force OLS method. In the lasso method, the regression coefficients will be nonlinear in z_i, so they cannot be expressed in closed form, and must be solved by the implementation of some root finding algorithm. Luckily for us, the Scikit-learn python package includes and efficient function to perform this regression. The behaviour of the results with varying λ is less transparent than in the ridge method, and will not be touched upon in detail here, but the main strength of the lasso is that it can reduce the coefficients for non important output parameters to zero. This amounts to an automatic feature selection integrated in the algorithm itself. It also has a regularization effect similar to the ridge method.

2.3. *Analyzing the regression output*

The aim of regression is to provide some function that not only fits the data well, but can be extrapolated to other, similar data. Two commonly used metrics that measure how well the function is fit to the data are the MSE,

$$\text{MSE} = \frac{1}{N}\sum_{i=0}^{N-1}(z_i - \tilde{z}_i)^2,$$

and the R^2-score,

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}.$$

The MSE is what is minimized in OLS, i.e. the square of the distance of each data point to its fitted point. The R^2-score gives the percentage of the variation in the data set explained by the model found by regression, so a score close to 1 means that the regression function closely fits the data, but it gives no indication of whether the estimation is biased. These two metrics are not sufficient to produce a good model, however. An overfit model will in general have an R^2-score close to one and very low MSE on the data it was fit on, but when the same model is applied to some novel data, it is tuned to fit exactly the data it was trained on, and provide a poor fit to the new data. To rectify this, the data should be split into a test set and a training set, where the parameters are fitted to the training data, but the scores are calculated when the model is applied to the test data. To further improve the results, the data can be fit using resampling methods. This analysis includes the k-fold cross validation resampling technique. The training set is split into k 'folds', then one of the folds is chosen as a test fold. The model is trained on the remaining $k-1$ folds and tested on the test fold by some chosen metric (R2 or MSE for example). This is done choosing each of the folds as a test fold. The metrics are then averaged, and the performance of the model is evaluated. The upside of this method is that the model is tested against parts of the data set which may have attributes which do not appear in the whole set, and the regression translates more nuanced features of the data set. Furthermore, the notion of error in the analysis can be refined. As discussed in section 2.1, the data is assumed to be a polynomial function with normally distributed noise centered at zero with variance σ^2,

$$\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\varepsilon}. \tag{1}$$

The data is fit by one of the regression methods discussed earlier and the fitted function applied to some new data sampled from the same distribution to produce $\tilde{\boldsymbol{y}}$. The squared error of the fit is formulated in terms of expected value as

$$\mathbb{E}\left[(\boldsymbol{y} - \tilde{\boldsymbol{y}})^2\right].$$

This expression can be decomposed through a rather involved algebra exercise. Rewriting the input data as in (1), and adding and subtracting the expected value of the regression estimation,

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{y} - \tilde{\boldsymbol{y}})^2\right] &= \mathbb{E}\left[(\boldsymbol{f} + \boldsymbol{\varepsilon} - \tilde{\boldsymbol{y}})^2\right] \\ &= \mathbb{E}\left[(\boldsymbol{f} + \boldsymbol{\varepsilon} - \tilde{\boldsymbol{y}} + \mathbb{E}[\tilde{\boldsymbol{y}}] - \mathbb{E}[\tilde{\boldsymbol{y}}])^2\right] \\ &= \mathbb{E}\left[((\boldsymbol{f} - \mathbb{E}[\tilde{\boldsymbol{y}}]) + (\mathbb{E}[\tilde{\boldsymbol{y}}] - \tilde{\boldsymbol{y}}) + \boldsymbol{\varepsilon})^2\right] \\ &= \mathbb{E}[(\boldsymbol{f} - \mathbb{E}[\tilde{\boldsymbol{y}}])^2 + (\mathbb{E}[\tilde{\boldsymbol{y}}] - \tilde{\boldsymbol{y}})^2 \\ &\quad + 2(\boldsymbol{f} - \mathbb{E}[\tilde{\boldsymbol{y}}])(\mathbb{E}[\tilde{\boldsymbol{y}}] - \tilde{\boldsymbol{y}}) \\ &\quad + (\boldsymbol{f} - \mathbb{E}[\tilde{\boldsymbol{y}}])\boldsymbol{\varepsilon} + (\mathbb{E}[\tilde{\boldsymbol{y}}] - \tilde{\boldsymbol{y}})\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^2].\end{aligned}$$

Because the noise is centered at zero, the second and third last terms are zero. The third term also equates to zero because $\boldsymbol{f}$ and $\mathbb{E}[\tilde{\boldsymbol{y}}]$ are deterministic,

$$\mathbb{E}[(\boldsymbol{f} - \mathbb{E}[\tilde{\boldsymbol{y}}])(\mathbb{E}[\tilde{\boldsymbol{y}}] - \tilde{\boldsymbol{y}})] = \bar{\boldsymbol{f}}\mathbb{E}[\tilde{\boldsymbol{y}}] - \bar{\boldsymbol{f}}\mathbb{E}[\tilde{\boldsymbol{y}}] + \mathbb{E}^2[\tilde{\boldsymbol{y}}] - \mathbb{E}^2[\tilde{\boldsymbol{y}}].$$

The expected value of $\boldsymbol{\varepsilon}^2$ equals σ^2 since it is centered at zero. The remaining terms are then

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{y}-\tilde{\boldsymbol{y}})^2\right] &= \mathbb{E}[(\boldsymbol{f}-\mathbb{E}[\tilde{\boldsymbol{y}}])^2] + \mathbb{E}[(\mathbb{E}[\tilde{\boldsymbol{y}}]-\tilde{\boldsymbol{y}})^2] + \sigma^2 \\ &= (\bar{\boldsymbol{f}}-\mathbb{E}[\tilde{\boldsymbol{y}}])^2 + \mathbb{E}[(\mathbb{E}[\tilde{\boldsymbol{y}}]-\tilde{\boldsymbol{y}})^2] + \sigma^2 \\ &= \text{bias}^2 + \text{variance} + \sigma^2.\end{aligned}$$

The only noise generated in the system is the normally distributed noise in the input data, and in practice the expectation values can be substituted by the mean over the same variables, i.e.

$$\text{bias} = \bar{\boldsymbol{y}} - \bar{\tilde{\boldsymbol{y}}},$$

$$\text{variance} = \frac{1}{N}\sum_{i=1}^{N}(\bar{\tilde{y}}_i - \tilde{y}_i)^2.$$

The variance of the noise is inherent in the data, and is irreductible. This is the so called bias-variance tradeoff, and it is a foundational restriction on machine learning methods, and the motivation for the ridge and lasso regression methods. Standard OLS provides the fit with the lowest bias, but the variance is potentially very high. The regualrization paramters in lasso and ridge increase the model bias, but decrease the variance, often to the effect of reducing overall error. When performing model selection and opimization, the goal is essentially to optimize the balance of bias vs. variance.

3. RESULTS

3.1. *Ordinary linear regression on the Franke function*

The Franke function is fitted by a polynomial of order $p_x = p_y = 5$ with the OLS SVD algorithm outlined in section 2.1., with a resolution of $n = 50$. The regression is performed with and without train-test-splitting and cross validation, and the resulting MSE and R^2 scores are shown in table 1. The scores are essentially the same, with the difference in MSE only at the order of 10^{-4}. This is possibly a result of the relatively high resolution of the data set, allowing each sample in the k-fold algorithm to mirror the full set. The data is fitted by the same order of polynomial in each case, and the bias and variance are not calculated. Fitting the data with polynomials of different order, some more interesting characteristics become apparent (figure 1). The bias starts off fairly high, and the variance fairly low, but for higher polynomial orders, the bias decreases and the variance increases as expected. The smallest error occurs at $p = 3$, but the bias is still quite high at this point. An optimal fit may possibly be found at $p = 4$, as the bias and variance stabilize at reasonably low values. Figure 2 shows more explicitly the sharp increase in variance with higher order fits.

TABLE 1
MSE AND R^2 SCORES FOR OLS REGRESSION

	MSE	R^2
No splitting	0.011	0.889
Train-test	0.011	0.885
5-fold CV	0.011	0.890

The variance explodes at around $p = 20$, and the bias also increases, but the variance dominates the error. The mean absolute value of the regression coefficients from the OLS method grow exponentially as a function of the order of the fit, as shown in figure 3. Generally, some of the coefficients grow extremely large, while others stay at approximately the same order of magnitude.

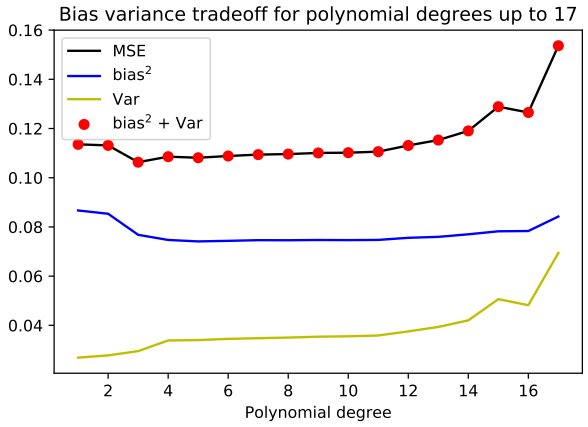

FIG. 1.— Bias, variance and MSE for the OLS method

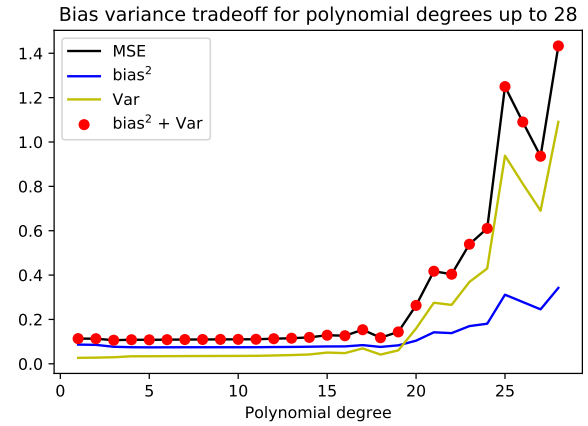

FIG. 2.— Bias, variance and MSE for the OLS method, higher orders

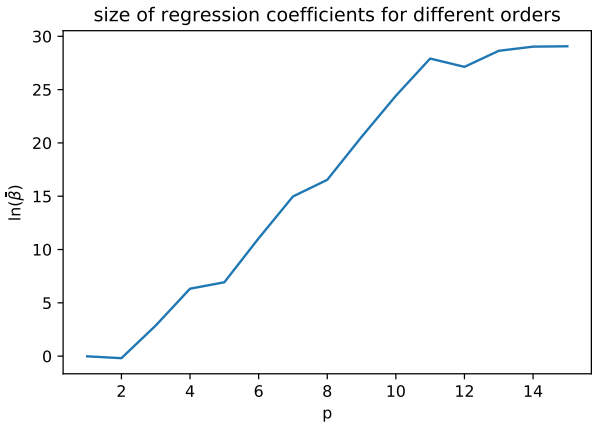

FIG. 3.— Size of OLS regression coefficients

3.2. *Regularized linear regression on the Franke function*

Introducing regularization gives greater control of the behavior of the bias and variance. When the regularization parameter is increased, the bias increases, but the variance decreases (figures 4 and 5). The coefficient size decreases much more steeply with the Lasso method, and most of the coefficients are reduced to zero already at $= 0.002$. With ridge regularization, the coefficients tend slowly to zero.

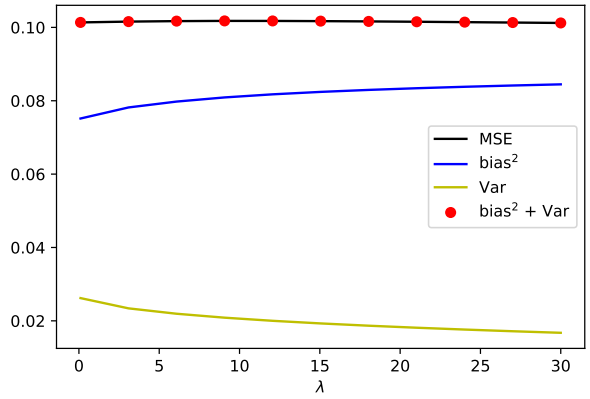

FIG. 4.— Ridge bias-variance tradeoff

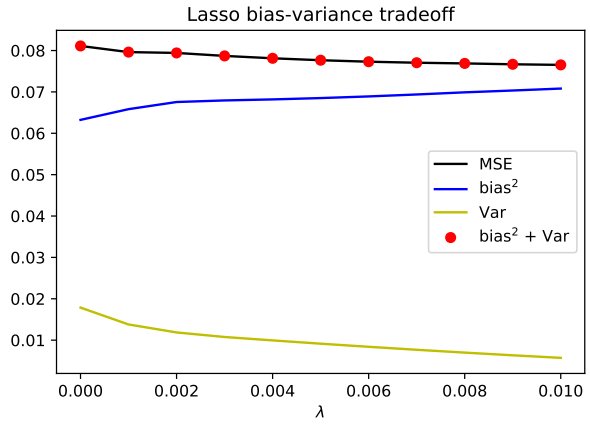

FIG. 5.— Lasso bias-variance tradeoff

3.3. *Introducing real terrain data*

The optimal model for fitting of the surface data can then be found by testing the three regression methods. The model should ideally have a low MSE and R^2-score close to one, ensuring that the fit is good, and have as low bias and variance as possible. The limited computing power of the laptop used for this assignment necessitated the use of a small data set, because the analysis requires the surface to be fit many times. The SRTM data was limited to a 100×100 area and reduced by removing every second pixel, the main features are still retained, although with a low resolution. This was further sectioned into a training and test set. The OLS method was applied first, testing fits with orders up to 30. The bias-variance graph (figure 6) shows that the OLS algorithm only provided fits with extremely high error, especiall for orders 5 to 6. The best OLS fit the lies in the regions shown in figures 7 and 8. Using OLS seems to give

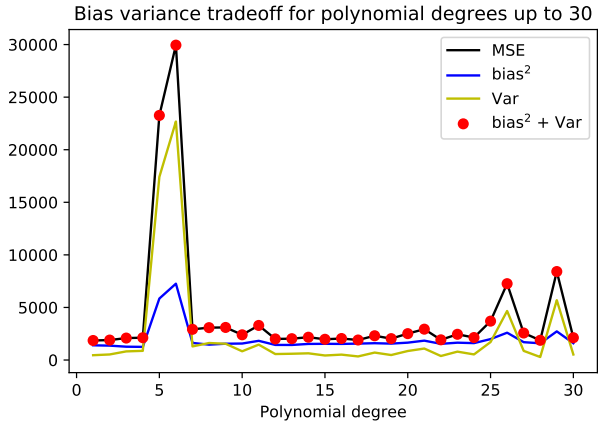

FIG. 6.— OLS bias variance tradeoff for terrain data

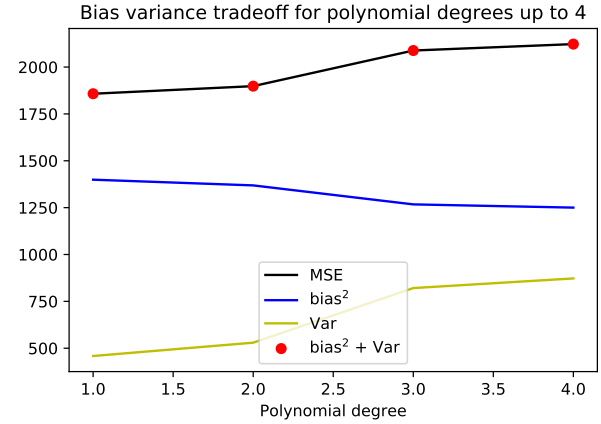

FIG. 7.— OLS bias variance tradeoff for terrain data, $p \leq 4$

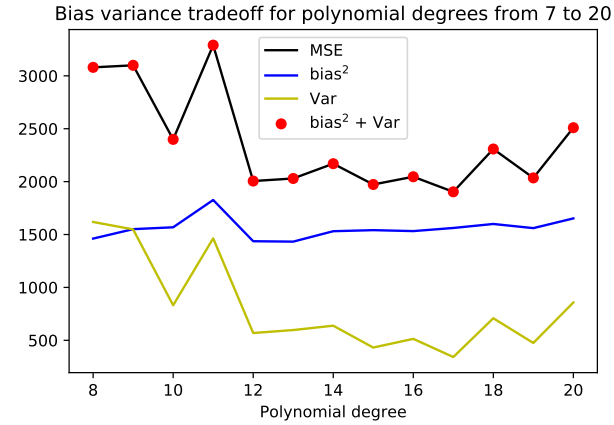

FIG. 8.— OLS bias variance tradeoff for terrain data, $p \in [7, 20]$

the best results for $p = 3$. The bias and variance are relatively balanced, and increasing the order to a value in

the range of 12 to 20 does not give a significant decrease in the error, and is more biased. The ridge method does not seem to improve the situation further (figure 9). The

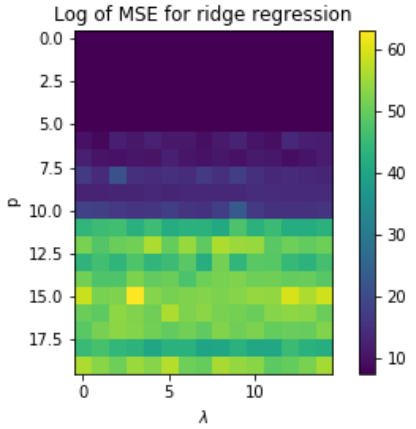

FIG. 9.— MSE for ridge regression

error much more dependent on p than on λ, so the OLS method is still preferred.

4. REFERENCES

1. Hastie, T., Tibshirami, R., Friedman, J. (2009). *The Elements of Statistical Learning* (2. edition). New York: Springer.

2. Hjort-Jensen, M. (2019). *Data Analysis and Machine Learning: Linear Regression and more Advanced Regression Analysis*. Link