Machine Learning and Terrain data

Mathias M. Vege

October 7, 2018

Abstract

None

Contents

1 Introduction

A now-fully emergent field of data analysis, is that of regression and resampling. Included as a subset of the field of machine learning, regression is widely used as tool of prediction. Together with resampling we are offered ways of estimating the error of our models.

One common way to investigate the efficacy of a model is to use the Franke function[1].

2 Methods and theory

Let us start of by summing up notational conventions. We will denote a *case* or *sample* by x_i, where $i = 0, \cdots, n$. This will have a corresponding *response* or *outcome* $y_i = f(x_i)$. We typically want to estimate f by some approximate function $\tilde{f}$. As we shall see, one method is by performing linear regression.

2.1 Linear regression

As mentioned, linear regression is a method of performing a fit of parameters x_i to a given data set y_i. From this we wish to minimize the error between the *response* or real data points y_i, and some approximation $\tilde{y}_i = \tilde{f}(x_i)$. This defines our loss function, which we will seek to minimize. In linear regression this will be by building an *design matrix* built from p *explanatory variables* or *predictors* $\boldsymbol{x_i} = x_i^0 + x_i^1 + \cdots + x_i^p$. The explanatory variables is simply the degrees of the polynomial we seek to approximate f by. So far, we have not mentioned the variables which will be tuned to the model, the *regression parameters* $\beta = (\beta_1, \ldots, \beta_p)^T$. The keen-eyed reader will now notice that the predictors and regression parameters all are of equal number, and may have guessed that they form our linear approximation - this is true:

$$\tilde{y}_i = \beta_0 x_{i0} + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

From this, we get a neat matrix equation,

$$\boldsymbol{y} = \beta \boldsymbol{X} + \boldsymbol{\epsilon} \tag{1}$$

We see that our goal has immediately become to minimize β. The definition of the error vector $\boldsymbol{\epsilon}$ will be given later.

The simplest case of such a linear regression minimization problem is just a simple line[3, ch. 3.1, p. 61]. However, this can be generalized greatly and we end up with the standard go-to method called *Ordinary Least Squares*.

2.1.1 Ordinary Least Squares regression(OLS)

To begin our journey towards a solution of OLS, we start by defining the *loss function* for OLS.

$$\min_{\beta \in \mathbb{R}^p} ||\boldsymbol{X}\beta - \boldsymbol{y}||_2^2 = \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(\boldsymbol{x}_i^T \beta - y_i\right)^2 \tag{2}$$

We see that the goal is to minimize the L_2 norm between the response y_i and the fitted predictors $\boldsymbol{x}_i^T \beta$ [4, ch. 4, p. 21].

To find a solution to the loss function, we can start by observing the dimensions of equation (1),

$$\underbrace{\boldsymbol{y}}_{\mathbb{R}^n} = \underbrace{\boldsymbol{X}}_{\mathbb{R}^{n \times p}} \underbrace{\beta}_{\mathbb{R}^p} \tag{3}$$

We then see that in order to free β, we must multiply by the transposed of $\boldsymbol{X}$ such that we can take the inverse and solve for β.

But, before we can do that we need to define what the optimal β vector is. We start by defining a function $Q(\beta)$ which gives us the squared error of the spread,

$$\begin{aligned} Q(\beta) &= \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2 \\ &= (\boldsymbol{y} - \tilde{\boldsymbol{y}})^T (\boldsymbol{y} - \tilde{\boldsymbol{y}}) \end{aligned}$$

which is the matrix equation

$$Q(\beta) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) \tag{4}$$

This equation is called the cost function[4, p. 12]. The minimum of this function can be found by differentiating by β and setting the result equal to zero.

$$\begin{aligned} \frac{\partial Q(\beta)}{\partial \beta_j} &= \frac{\partial}{\partial \beta_j} \sum_{i=0}^{n-1} (y_i - \beta_0 x_{i0} - \beta_1 x_{i1} - \dots \beta_{p-1} x_{i,p-1})^2 \\ &= -2 \sum_{i=0}^{n-1} x_{ij} (y_i - \beta_0 x_{i0} - \beta_1 x_{i1} - \dots \beta_{p-1} x_{i,p-1}) \end{aligned}$$

where the element x_{ij} is picked out by the differentiation w.r.t. β_j. Performing all of the derivatives leaves us with the matrix equation,

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

which we require to be zero in order to find an optimal β. From this the *residues* or *error vector* $\boldsymbol{\epsilon}$ is defined,

$$\boldsymbol{\epsilon} = \boldsymbol{y} - \tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X}\beta \tag{6}$$

The solution should now be obvious. We need to solve equation (5) for $Q(\beta) = 0$,

$$\boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{X}\beta) = 0.$$

As alluded to in (3), we now need to multiply by the inverse of $\boldsymbol{X}^T\boldsymbol{X}$, which yields after splitting the parenthesis

$$\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{X}\beta)$$

And thus, the β-values is simply

$$\beta = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} \left(\boldsymbol{X}^T\boldsymbol{X}\right) \boldsymbol{y} \tag{7}$$

This solution does not account for errors in the measurements or responses. Say there is an σ_i associated with each measured y_i, we will have to use the χ^2 function as a our function to minimize[2, see notes on regression]. We will have to rescale our solution by $\boldsymbol{X} \to \boldsymbol{A} = \boldsymbol{X}/\boldsymbol{\Sigma}$ where Σ is a diagonal matrix of every uncertainty σ_i.

2.1.2 Ridge regression

2.1.3 Lasso regression

2.2 The Bias-Variance decomposition

It is possible to define variance of the predicted values $\tilde{y}$. If we take the estimator of our cost function,

$$E[Q(\beta)] = E\,[] \tag{8}$$

2.3 R^2 score

2.4 Bootstrapping

2.5 k-fold Cross Validation

3 Implementation

[5]

4 Results

5 Discussion and conclusion

6 Appendix

6.1 The Franke Function

As given in [1],

$$F(x,y) = 0.75\exp\left(-\left(0.25\left(9x-2\right)^2\right) - 0.25\left(\left(9y-2\right)^2\right)\right) \tag{9}$$

$$= 0.75\exp\left(-\frac{1}{49}\left(\left(9x+1\right)^2\right) - 0.1\left(9y+1\right)\right) \tag{10}$$

$$= 0.5\exp\left(-\left(9x-7\right)^2/4.0 - 0.25\left(\left(9y-3\right)^2\right)\right) \tag{11}$$

$$= -0.2\exp\left(-\left(9x-4\right)^2 - \left(9y-7\right)^2\right) \tag{12}$$

References

[1] Richard Franke. A critical comparison of some methods for interpolation of scattered data. Technical report, Monterey, California: Naval Postgraduate School., 1979.

[2] Morten Hjort-Jensen. Data analysis and machine learning: Linear regression and more advanced regression analysis lecture notes. https://compphysics.github.io/MachineLearning/doc/web/course.html, October 2018.

[3] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[4] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. *ArXiv e-prints*, March 2018.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.