Regression analysis and re-sampling methods

Johan Nereng

Department of Physics, University of Oslo, Norway

Sep 30, 2019

Abstract

1 Introduction

The main aim of this project is to study in more detail various regression methods, including the Ordinary Least Squares (OLS) method, Ridge regression and finally Lasso regression, including use of resampling. Start by fitting polynomials to Franke's function (see Methods).

Having established the model and method, the models are further evaluated using resamling techniques such as cross-validation.

choice of modelling based on insight. Terrain possbily polynomial ? use franke function to test algorithms.

Linear regression models the relationship between the response or target and the predictors linearly. In this project, three different methods are applied to fit the model to the data. Ordinary Least Squares (OLS), Ridge, and Lasso. Since this project assumes that the

Trying to find the relationship between an independent and dependent variables chose approximation ,a..a.s

it is normal to use y for predictors in litteratur,and x0 x1 etc. but for shorthand purposes, x1x2 are called xy, thus z=y in this project.

Project flow: This project starts by introducing

Main findings

2 Methods

2.1 Multiple Linear Regression

Usually, insights into the underlying mechanisms of the origins of a data set, will guide the the choices one makes when trying to model the relationship between the dependent and independent variables. If such insights suggest a linear relationship between response and target, the model describing that relationship should reflect this. When modeling for one response, this means using multiple linear regression, while for more than one response, this would mean using multivariate linear regression. A response y, with approximation $\tilde{y}$, on predictor x with a $m-1$ degree linear approximation may be expressed as

$$z = \sum_{j=0}^{m-1} \beta_j f_j(x) + \epsilon = \tilde{z} + \epsilon \tag{1}$$

, where ϵ is the residual error between the model and the true response [5][p.19]. When the target-predictor relationship can be modeled by power functions, (1) leads to a polynomial fit, ie $f_j(x) = x^{j-1}$. For a data set $\{z_i, x_i\}_{i=0}^{n-1}$, where y_i is the response on predictor x_i, (1) results in n equations. In order to effectively solve for the regression coefficients (β_j), the set of equations may be represented by a the matrix-vector multiplication. For a polynomial of order

$m-1$, $\boldsymbol{\beta} = [\beta_0, \beta_1, ..\beta_{m-1}]^T$, and for $n-1$ data points, $\boldsymbol{z} = [z_0, z_1, ..z_{n-1}]^T$, while $\epsilon = [\epsilon, \epsilon, ..\epsilon n - 1]^T$. Lastly, the $n \times m$ matrix

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{m-1} \\ 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ & & \dots\dots\dots\dots & & \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{m-1} \end{bmatrix} \tag{2}$$

, known as a Vandermonde Matrix [1][p. 147-148], yield the vector matrix product:

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3}$$

An outtake of the methods which may be applied to (3) in order to solve for the regression coefficients are discussed later. However, the Vandermonde Matrix - which is a special case of what is generally known as a design matrix - is limited to a univariate polynomials. If instead the response z_i is on two predictors (x_i and y_i), such as will later be used in this project, a bivariate polynomial approximation of z will result in a design matrix $\boldsymbol{X}$ where row i is on the form $[1, x_i, y_i, x_i^2, y_i^2, x_i y_i ...]$, with a total of $d = \binom{m+2}{m}$ coefficients.

2.1.1 Ordinary Least Squares (OLS)

OLS fits a function by minimizing the sum of the squares of the errors between a model and a data set. This results in the cost function

$$C(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta}) \tag{4}$$

which, if A has full column rank, has a unique solution [1][p.144] which satisfies the normal equations:

$$(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{b} = \boldsymbol{X}^T\boldsymbol{y} \tag{5}$$

Solving this corresponds to finding the minima of the derivative of the cost function with respect to $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^{OLS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{A}^\dagger\boldsymbol{y}. \tag{6}$$

This matrix inversion may then be carried out using for example SVD or LU decomposion. Alternatively, minimizing (4) may be accomplished using orthogonal transformation with QR decomposition, the details of which may be found in chapter one of Uri M. Ascher and Chen Grief's book [1]. The choice between the two approaches hinges on computational stability versus computational speed.

Computational Speed: The two approaches are as mentioned dissimilar in computational cost, measured in floating point operations (flops) - especially in overdetermined cases, where the matrix $\boldsymbol{X} \in \mathbb{R}^{n,m}$, has $m < n$. The normal equation solution uses $mn^2 + (1/3)n^3$ flops [1][p.146], while the QR decomposition requires $2mn^2 - (2/3)n^3$ flops [1][p.155]. Which means that the normal equations method is significantly faster than the QR approach.

Computational Stability: The conditioning of the two approaches, or stability with respect to input perturbation, are measured in conditioning number of the matrix representing the problem, $K(\boldsymbol{X})$. Higher $K(\boldsymbol{X})$ means less stable.

$$K(\boldsymbol{X}) = ||\boldsymbol{X}^{-1}|| \times ||\boldsymbol{X}|| \tag{7}$$

In order to acquire the normal equations, the product of $\boldsymbol{X}$ and it's transpose is used. Hence, it's conditioning number goes as $K(\boldsymbol{X}^T\boldsymbol{X}) = K(\boldsymbol{X})^2$. This squaring of the conditioning number is not required for the QR approach, which means that the QR method has a lower conditioning number, and is thus the more stable method.

2.2 Shrinkage methods

Alterations to the OLS cost function (4) may be done in order to assert control of the bias variance trade off. Specifically, in order to reduce the variability of prediction errors one may observe when fitting complex models. Shrinkage methods introduce a penalty parameter λ, which effectively shrinks the regression coefficients. In this project, two such shrinkage methods are used, Ridge Regression and Lasso Regression.

2.2.1 Ridge Regression

Ridge Regression cost function [4][p.22] minimizes the square of the residual sum with a penalty term proportional to the square of the coefficients:

$$C(\boldsymbol{\beta}) = \frac{1}{n}(\boldsymbol{z} - \boldsymbol{X}^T\boldsymbol{\beta})^T(\boldsymbol{z} - \boldsymbol{X}^T\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \tag{8}$$

Where $\lambda \geq 0$ is the penalty constant. This parameter may decrease variance when compared to OLS, and may therefor also reduce the prediction error of the model [2][p.61-62.]. Originally however, this λ parameter was introduced to address rank deficiency [2][p.64] - which it does, as the method adds a non-zero constant to the diagonal elements.
The cost function (8) results in the following expression for the regression coefficients:

$$\boldsymbol{\beta}^{Ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{z} \tag{9}$$

2.2.2 Lasso Regression

2.3 Re-sampling techniques

Thus, in order to find a well suited λ, it is common to split the data set into training data and a test data. Such a split allows for

. After splitting the data, the training data is then used to find values for the coefficients, and

By doing so, the estimation of the response may be evaluated

, shrink parm. MSE on a split set. test train. evaluate at test. IOT effectively doso, resampling. K-fold - test kfold på stort sett først. BAYESAN og6.6

2.4 Model evaluation

There are, as outlined above, several regression models to chose from. These models all have parameters that impacts their predictions, such as the number of polynomial coefficients and the regularization parameter. As there are many choices, a measurement of how well a regression model predicts the response corresponding to a real data point is needed. One such measurement is the mean square error:

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(z_i - \tilde{z}_i)^2 \tag{10}$$

This is a natural measurement to use, both because it's simple and because the cost function is based on this measurement when solving the normal equations. However, the MSE is data specific. An alternative is using the R^2, which is a normalized measurement:

$$R^2(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = 1 - \frac{\sum_{i=0}^{n-1}(z_i - \tilde{z}_i)^2}{\sum_{i=0}^{n-1}(z_i - \bar{z}_i)^2} \tag{11}$$

, where $\bar{y}$ is the mean value of $\boldsymbol{y}$. As it is independent of scale, R^2 always returns a value between zero and one, making it easy to interpret.
Another way to measure how well the model works is by finding the confidence intervals for the regression coefficients. Following the train of thought from [2][p.47-49]: When using the normal equations for OLS, these may be found by using an analytic expression for the variances of the coefficients

$$Var(\boldsymbol{\beta}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\hat{\sigma}^2 \tag{12}$$

Where if σ is unknown, an unbiased estimate $\hat{\sigma}$ should be used. Having obtained the variance of the regression coefficients, the 95% confidence interval (CI) of β_i may be found by

$$CI_{0.95}(\beta_i) = \beta_i - 19.6 v_j^{1/2}\sigma, \beta_i + 19.6 v_j^{1/2}\sigma \tag{13}$$

Where v_j is the diagonal element of row j of the matrix $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$.

2.4.1 Bias Variance trade-off

For a response $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$, with $\mathbb{E}[\boldsymbol{\epsilon}] = 0$ and $Var[\boldsymbol{\epsilon}] = \sigma^2$, the expected mean square error (MSE) of a regression model fit $\tilde{\boldsymbol{y}}$ from an input $\boldsymbol{x} = \boldsymbol{x}_0$ may be calculated by (see appendix 1):

$$\mathbb{E}\left[(\boldsymbol{y} - \tilde{\boldsymbol{y}})^2\right] = \frac{1}{n}\sum_i (f_i - \mathbb{E}\left[\tilde{\boldsymbol{y}}\right])^2 + \frac{1}{n}\sum_i(\tilde{y}_i - \mathbb{E}\left[\tilde{\boldsymbol{y}}\right])^2 + \sigma^2. \tag{14}$$

2.5 Algorithm implementation

2.5.1 Evaluating the implementation

In order to evaluate the implementation of the OLS algorithm

$Var(\beta) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\sigma^2$.

confidence interval for beta hastie et all p.47

choice of model - regression type - linear etc. polynomials - understanding of the problem at hand. choice of minimization evaluation of betas type of regression method, OLS etc.

2.6 Franke's function

Describe Franke's function.

3 Discussion of results

3.1 Conclusions

References

[1] Uri M. Ascher and Chen Greif. Linear least squares problems. In *A First Course on Numerical Methods*, pages 141–166. Society for Industrial and Applied Mathematics, 2011.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, New York, NY, second edition edition, 2009.

[3] Morten Hjorth-Jensen. Lectures notes in fys-stk4155. data analysis and machine learning: Linear regression and more advanced regression analysis, Sep 13 2019.

[4] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, 2019.

[5] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Adaptive computation and machine learning. MIT Press, Cambridge, 2012.

Appendices

Some Appendix Following [2][p.223] følg https://stats.stackexchange.com/questions/204115/understanding-bias-variance-tradeoff-derivation