Project Report

Horatiu Luci

October 2020

1 Ordinary Least Squares Model

The Ordinary Least Squares (OLS) model was implemented with a Python3 function (estimate_coef) (2) that takes the feature matrix X and the response vector y as input, intercepts it (adds a column of 1s in the beginning in order to have a nonzero y intercept) and returns a vector of weights.

When fitting the model, we want to find the optimal weights, thus the SSE (sum squared error) needs to be minimum with regard to our weights w:

$$SSE = \sum_{i=1}^{n} (y - Xw)^T (y - Xw)$$

So, in order to get the minimum of this function with regards to w we need to differentiate with respect to w, the equation becomes:

$$\frac{\partial SSE}{\partial w} = -2X^T y + 2X^T Xw => X^T Xw = X^T y$$

$$=> w = (X^T X)^{-1} X^T y$$

Therefore, these are the weights that minimise the sum squared error.

The estimate_coef function returns a vector of 11 weights $w = (X * X_t)^{-1} * X_t * y$, where:

- X = intercepted feature matrix
- X_t = Transposed intercepted feature matrix
- X^{-1} = Inverse of a matrix
- y = response vector

This model is intended to use with data that is correlated, generally speaking, the more features - the more accurate is the model. However, if data is not correlated, there might appear singularities or over-fitting.

The following weights are outputted by the OLS Model (test size of 0.3):

Features	intercept	age	sex	**bmi**	bp	**s1**	**s2**	s3	s4	**s5**	s6
OLS	151	29.25	-261.7	**546.29**	388.4	**-901.9**	**506.76**	121.1	288	**659**	41.37

In this model, it can be seen that **bmi**, **s1**, **s2** and **s5** are having a considerable impact on the progression of diabetes, as all of them have high estimated coefficient values.

The R^2 coefficient of determination of this model after 100 runs is roughly between 0.3 and 0.5 and is greatly improved after intercepting the training data (This was a point of failure in earlier models that I tried and skipped the interception step). This indicates that the model is relatively accurate, this score is generally considered acceptable in the medical field (1)

2 Lasso Model

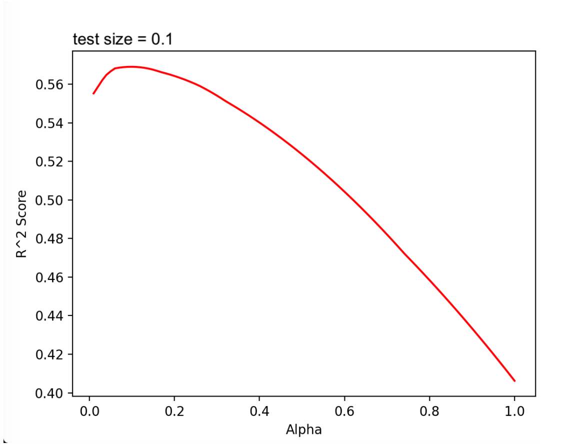
The Lasso model was built with the alpha meta-parameter of 0.5 and using the training set. The following weights were outputted:

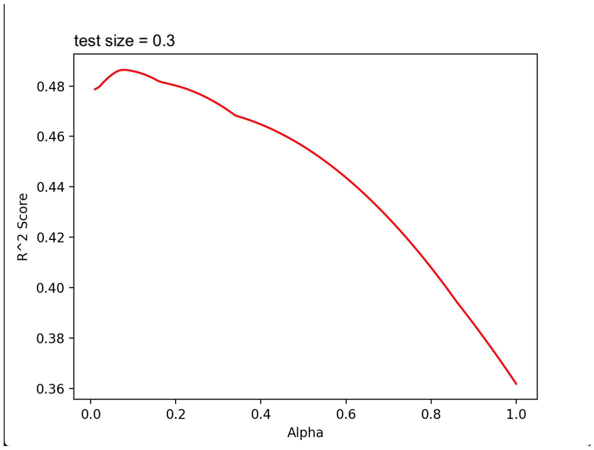
Features	intercept	age	sex	**bmi**	bp	**s1**	**s2**	s3	s4	**s5**	s6
OLS	151	29.25	-261.7	**546.29**	388.4	**-901.9**	**506.76**	121.1	288	**659**	41.37
Lasso	0	0	-0	**527.78**	193.13	**-0**	**-0**	-107.8	0	**289.36**	0

It can be seen that this model sees way less impact from the features **s1**, **s2** and **s5**.Therefore we can say that these features have little impact on improving the model and only make it more complex. In such a case the whole OLS model becomes less reliable than the Lasso model as the coefficient estimates have high variance, therefore the OLS model is more **overfitting** than the Lasso model. The R^2 coefficient of determination of this model after 100 runs is roughly between 0.45 and 0.55, a bit more than the OLS model.

3 Comparing Models of Increasing Complexity

The hyper-parameter alpha is representing regularization strength. A high value of alpha imposes a high penalty on the sum of magnitudes of coefficient estimates, can force more coefficients to zero and can cause under fitting. A number of 100 lasso models was trained (with alpha values between 0.01 and 1) and the R^2 score was outputted for each model in part.





For both plots we can see the Lasso regression uses L1 regularization to force more and more coefficients to be exactly zero (those features are completely ignored by the model and R^2 score decreases), as alpha increases. Higher values of alpha force more coefficients to zero and can cause under-fitting, while values of alpha that are lower, will lead to fewer non-zero features and can cause over-fitting.

4 Choosing the final Model

As it can be seen in the plots, as alpha increases from 0.01 to 1, the R^2 score is increasing until alpha=0.06 for test size = 0.1 & alpha=0.07 for test size = 0.3 and then it starts decreasing. We could also see in the plots that the more training data we have, the higher the overall R^2 score can get (test-size 0.1 vs 0.3). Therefore we can say the models are more and more under-fitting as alpha is approaching 1. And we can have over-fitting as we add more and more training data.

As to choose the best alpha for the Lasso model, we can also see from the plots that the **best R^2 scores are achieved by the models with an alpha in 0.06 range** regardless of the test size.

Compared to the OLS model, the Lasso model can perform slightly better or arguably worse, depending on alpha (grid search can be used to find the optimal alpha). Very low values of alpha will cause the model to resemble linear regression, at the same time very big values will allow correlated data to be treated as not relevant, choosing a proper alpha is vital in regarding Lasso as the better fitting model for this task.

References

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4678365/

[2] https://github.com/horatiuluci/ml-projects/blob/main/1/project1.py