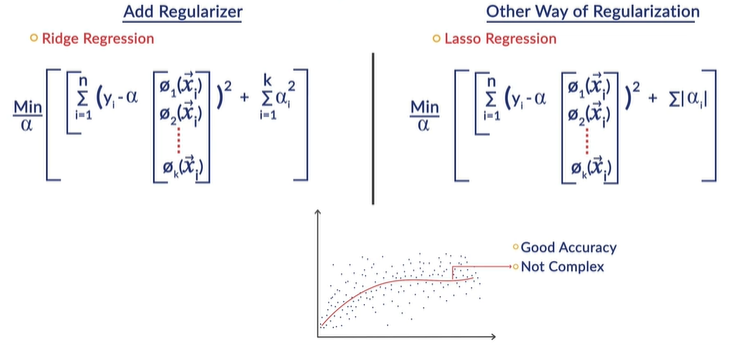
# **Regularized Regression**

A predictive model has to be as simple as possible, but not simpler. There is an important relationship between the complexity of a model and its usefulness in a learning context because of the following reasons:

1. Simpler models are usually more generic and are more widely applicable (i.e., they are generalisable).
2. Simpler models require fewer training samples for effective training than the more complex ones.

**Regularisation** is a process used to create an optimally complex model, i.e., a model that is as simple as possible while performing well on the training data. Through regularisation, one tries to strike the delicate balance between keeping the model simple and yet not making it too naive for any use.



What the regularized technique does is it treats the sum of the squares of the coefficients as a measure of the model complexity. While we are trying to bring the error down by minimizing the cumulative sum of the squares we don’t want alpha (coefficients) which is the one characterizes the model to be too complex. So to balance the error term i.e. sum of the squares of the differences between the predicted value and the observed value (features) we add another term to it which is simply sum of squares of the coefficients. If the algorithm is trying to find an alpha just because the error needs to come down it tries to make the alpha too complex my objective function in this case will not allow the algorithm to do that because I also have the additional sum of squares of coefficients.

Another form of regularized notion is to add the absolute values of each coefficients instead of sum of the squares.

# Ridge and Lasso Regression - I

The sum of squares of coefficients also called regularization coefficient is controlled by lambda which is a hyperparameter. The λ is the hyperparameter in the objective function that we minimize for regularized regression. The λ*= 0* implies that there is no regularization and thus a high chance of overfitting.

|  |  |
| --- | --- |
|  | Here the final model which is the coefficient vector α (alpha) that the algorithm would produce is clearly depend on the λ value that you have chosen before you ran the regression algorithm. Because the regression algorithm minimizes the error + λ terms of sum of the squares of the coefficients. |

|  |  |
| --- | --- |
|  |  |

Whenever we talk about inverse of a matrix, we need make sure the matrix is invertible. Matrices who determinant is zero they are called Singular Matrix. Not every matrix is invertible. So, whenever we create a formula like (XTX)-1 we have to guarantee that the matrix we are trying to inverse (here X) is actually invertible. In this case there is no such guarantee because the data set is what it is. From the given data set we are going to create features and later we are going to come up with X, so what it (XTX) is not invertible? then this formula cannot be applied. However, when we regularize and use the modified formula (XTX + λI), then this becomes far more likely to be invertible.

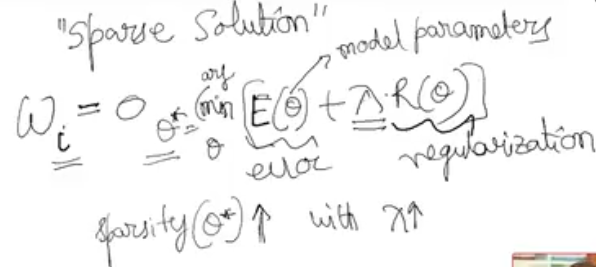
|  |  |
| --- | --- |
|  | In case of Lasso, it is computationally far more intensive than Ridge regression. Mathematically its not clean as you need to use iterative procedure to arrive at the optimum. Ridge regression almost always has a matrix representation for the solution, while lasso requires iterations to get to the final solution. |

Lasso trims down the coefficients of redundant variables to zero and, thus, indirectly performs variable selection as well. Ridge, on the other hand, reduces the coefficients to arbitrarily low values, though not zero. The coefficient of redundant feature essentially become zero.

To know the mathematics behind Ridge and Lasso please go through the optional content. However, [we have put together an optional session here](https://learn.upgrad.com/v/course/376/session/53245/segment/293363) to explain the math involved in these topics. We **highly recommend**that you go through this session to get a solid understanding of the math behind these concepts. Specifically, in the optional session, you will study:

* Contour plots: Visualizing the error and regularization terms, and
* How the lasso results in sparse solutions.

|  |  |
| --- | --- |
|  | where the w 's are the model parameters (coefficients) and λ is the regularization hyperparameter. R(w) in case of Ridge Regression is given as  ∑w2i while for Lasso it is given as  ∑||w|| . |



Lasso regression results in a **sparse solution -** it means many of the model coefficients automatically become exactly zero, wi=0.

If θ∗ is the best model that we end up getting, which is given as:  **θ∗=argmin[E(θ)+λR(θ)]**

The ***difference between min and argmin*** is min produces minimum value of this expression [E(θ)+λR(θ)] across all values of θ, but for argmin we are looking at the value of θ for which this expression becomes minimum.

Then Sparsity(θ∗) increases with increase in λ, where Sparsity(θ∗) of a model is defined by the number of parameters in θ∗ that are exactly equal to zero. The significance of the Sparsity is that Lasso tells us which feature to be used and which not be. So we pick-up all non-zero coefficients in our model building. In case Ridge, we often don’t know what is the minimal set of features we should choose to build optimal model.

We often have a large number of features in real-world problems, but we want the model to be able to pick up only the most useful ones (since we do not want unnecessarily complex models). Since lasso regularisation produces sparse solutions, it automatically performs feature selection.

# Comprehension: Error and Regularization Contours

We saw that both ridge and lasso techniques regularize the coefficients by reducing their magnitude. In other words, they cause **shrinkage** of the coefficients. Ridge and lasso perform shrinkage differently. Specifically, lasso shrinks some of the coefficients to zero, thus performing **feature selection.**Let's try to understand how this happens.

Let's say that we need to find the optional values of two model parameters (coefficients), α1 and α2. We know that the **regularized cost function** has two components: the **error term** and the **regularization term**. The figure below is a schematic illustration of the error (red lines) and regularization terms (green dotted curves for ridge regression and solid blue lines for lasso regression) contours. The two axes represent α1 and α2, respectively.

|  |  |
| --- | --- |
|  | A **contour** of a function f(α) is a trace (locus) of the points that satisfy the equation f(α)=c, for some constant c. The figure shows contours for various values of c: the 'inner' contours represent lower values of the error/regularization terms. You want to find the coefficients that **minimize both the error term and the regularization term.** The zoomed in picture of one of the crossings between an error contour and a regularization contour in the figure illustrates that at every crossing, one could move along the arrow (black arrow in the zoomed in image) shown to keep the error term same and reduce the regularization term, giving a better solution, i.e., the sum of the coefficients reduces while keeping the error constant. |

Thus, the key observation here is that at the optimum solution for α (the place where the sum of the error and regularisation terms is minimum), the corresponding regularisation contour and the error contour must 'touch' each other tangentially and not 'cross'.

The 'blue stars' highlight the touch points between the error contours and the lasso regularisation contours. The 'green stars' highlight the touch points between the error contours and the ridge regularisation terms. The picture illustrates the fact that because of the 'corners' in the lasso contours (unlike in ridge regression), the touch points are more likely to be on one or more of the axes. This implies that the other **coefficients become zero.** Hence, lasso regression also serves as a feature selection method, whereas ridge regression does not.

**Cost Function in Regularised Regression**

A cost function in a regularised regression model has two terms: the error term and the regularisation term. The objective of the learning algorithm is to find the coefficients alpha such that:

**The sum of the error term and the regularisation term is minimised.**

**Feedback :**

*The objective function to minimise is the sum of the error and the regularisation terms.*

**Graphical Interpretation of Ridge and Lasso Regression**

The red contours (in the image above) depict the graph of the error term. Select the correct order of the values of error for the (red) contours. The innermost one is labelled as 3, then 2, then 1.

**1>2>3**

**Feedback :**

*The outer contour represents the highest error value as described in the comprehension.*

**Optimal Values of the Coefficients**

In the contour plot shown, the axes represent the coefficients of the model α1 and α2. The contours represent how the error term and the regularisation term vary with the coefficients.

What kind of a model will you get if both the coefficients have extremely low values, i.e., both the coefficients are close to zero?

A simple model that is likely to underfit the data

**Feedback :**

*When both the coefficients are close to zero, you are somewhere near the origin, where the error contours are high but the regularisation contours are very low. Such a model is likely to underfit the data, as the error is quite high (though it is 'simple', as the coefficients are small).*

**how optimal hyper parameter value is chosen**

can someone explain how  optimal hyper parameter value is chosen from the graph that we draw between Negative Mean Absolute Error and the range of hyper parameters in python. Why Negative Mean Absolute Error is chosen  and why not any other parameters ? what is the significance of the Negative Mean Absolute Error that has on the hyper parameter value that we choose to build our model.

* The optimal hyperparameter is chosen at the point where both train and test scores are better(at peak) in the graph.
* Negative mean Absolute error is used to tune a model or cross validate. MAE(Mean absolute Error) is an error metric - implies lower the better. Negative is opposite of it. Example:  -1.3 is better than 2.7
* If multioutput parameter is ‘raw\_values’, then mean absolute error is returned for each output separately. If multioutput is ‘uniform\_average’ or an ndarray of weights, then the weighted average of all output errors is returned.
* References:
* <https://stackoverflow.com/questions/48244219/is-sklearn-metrics-mean-squared-error-the-larger-the-better-negated>
* <https://stackoverflow.com/questions/55786121/what-is-the-negative-mean-absolute-error-in-sklearns-svm-library>
* <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html#sklearn.metrics.mean_absolute_error>

# Model Selection Criteria

Feature selection strategies are another way of regularizing the model; here, rather than modifying the cost function to make the resultant model simpler, they explicitly choose a subset of features to simplify the model.

Let's quickly recollect the terminology used in the lecture. If ^yi is the predicted value and yi is the actual value, then **RSS** is given as follows: ∑ni=1(^yi−yi)2  is the difference between the actual values and the predicted values. The (^yi−yi) is often called the **residue.**Also, the total sum of squares, TSS, is written as the sum of the explained sum of squares (ESS) and RSS, i.e., TSS=ESS+RSS.

|  |  |
| --- | --- |
|  | To a large extend the model must be able to explain much of these variances.  How do I measure how much of these total variances has my model being able to effectively explain?  That is simple, instead of using yi in the formula (which is the ground truth) we an use the model prediction, i.e. exactly what is called ESS  ESS = ∑ni=1(^yi−y bar)2  . If ^y is closely associated with yi then ESS is close to TSS. In other words that means the model is able to explain the total sum of squares TSS to a large extend. Whatever left out (the features) which model could not explain properly comes under RSS. |

If the value of the ESS is close to the TSS, then the model is able to explain the total variance in y to a large extent. On the other hand, if the ESS is much smaller than the TSS (and thus, RSS is large), then the model is not able to explain much of the variance in y.

|  |  |  |
| --- | --- | --- |
|  | Essentially the points (features aka residuals) for a linear regression can be explained by Gaussian aka normal distribution. In other words the likelihood of a point appearing is a Gaussian,. Its high if you are close to mean of the Gaussian and as you move away from the line the probability of occurrences of such points are going to get exponentially smaller.  Under this probabilistic model (left side exponential fn) the least square regression is a maximum likelihood estimate. If I want to maximize the probability, I need to minimize the pdf (probability density function) | |
|  | | In reality ESS always <= TSS because a good model will never be able to explain all the variables. So the larger (closer to 1) the R2 is better the model. R2 is always between 0-1.  Hence, a model is said to be a good model when it is able to explain most of the TSS; the larger the R2, the better the model. But the problem with using R2 is that it increases with the number of features in the model and does **not have any regularisation**mechanism to keep a check on the complexity. | |

So, in practice, we use the**adjusted R-squared.** Adjusted R-squared penalizes the model for having a large number of variables, d. When d increases, the term (n-d-1) decreases, which increases the term RSS/(n−d−1), and consequently, the adjusted R-squared will decrease.Essentially what adjusted R2 is saying here look you have used more features (may be more than required/necessary), hence I’m going to penalize you. It tries to do some regularization behind the scene to penalize the model.

The reason for using (n-1) is to make the estimates unbiased in oppose to bias. Using n the estimates might become certainly biased. The mathematics is too detail and beyond the scope of data science.

Let’s discuss two new model selection criteria, which are not specific to regression but are used in a variety of problems: **Akaike information criterion (AIC)**and **Bayesian information criterion (BIC).**

|  |  |
| --- | --- |
|  | Here, p is the model likelihood, given the model how likely such a dataset D (entire dataset) to be appeared if the underlying system follow the model of this kind M.  To summarise, AIC is defined as follows: AIC=−lnP(D|M)+d  Here, **d** is the number of parameters in the model, **D** is the entire data set, **M** is the model, and lnP(D|M) is the model likelihood |

|  |  |
| --- | --- |
|  |  |

BIC works well when 'n' is large (i.e., for large data sets); if the data set is small, then BIC can be a very reliable estimate.

# Comprehension: Model Selection Parameters

**Feature Selection using VIF and P-Value**

A regression exercise is performed on yy using the variables x1, x2 and x3. The VIF and the p-value after the first iteration are shown below:

|  |  |  |
| --- | --- | --- |
| Variable | VIF | p-value |
| x1 | 4.2 | 0.001 |
| x2 | 3.1 | 0.09 |
| x3 | 1.8 | 0.04 |

Which variable would you remove and reiterate?

x2

**Feedback :**

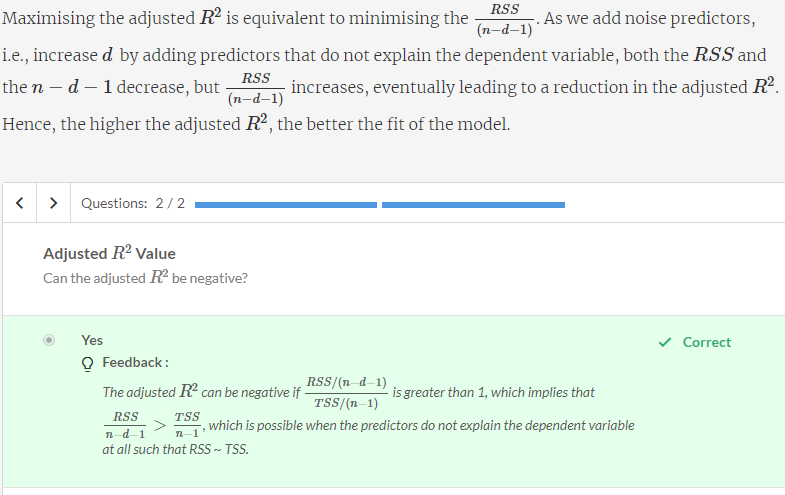
*It has a VIF value greater than 3, and the significance is also greater than 0.05. Hence, it should be eliminated first.*

While creating the best model for any problem statement, we end up choosing from a set of models that would give us the least test error. Hence, the test error, and not only the training error, needs to be estimated in order to select the best model. This can be done in the following two ways:

1. Use metrics that take into account both the model fit and its simplicity. They penalise the model for being too complex (i.e., for overfitting) and, consequently, more representative of the unseen ‘test error’. Some examples of such metrics are adjusted R2, AIC and BIC.
2. Estimate the test error via a validation set or a cross-validation approach.  
   In the validation set approach, we find the test error by training the model on a training set and fitting on an unseen validation set, while the in n-fold cross-validation approach, we take the mean of errors generated by training the model on all folds except the kth fold and testing the model on the kth fold, where k varies from 1 to n.

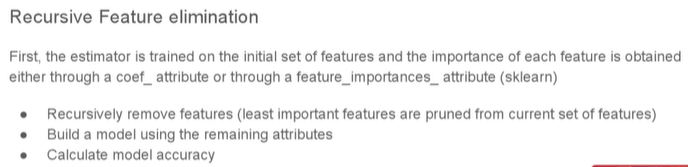
So far, you have understood that MSE of the training might not be a good estimate of the test error. The aforementioned parameters are a manipulation of the RSS (residual sum of squares), wherein a **penalty term** is introduced to compensate for the increase in complexity due to the increase in the number of predictors.

|  |  |
| --- | --- |
|  | AIC and BIC are defined for models fitted by maximum likelihood estimator. Also, maximum likelihood and least square are the same thing for linear models if the errors have a Gaussian distribution. You can refer to additional readings for the expression of the aforementioned parameters in terms of the maximum likelihood estimator.  You can see that as we increase the number of predictors, d, the penalty term in Cp, AIC and BIC increases, while the adjusted R2 decreases. Hence, the lower the value of  Cp, AIC and BIC, the better is the fit of the model. |



# Comprehension: Features' Subset Selection - Best Subset Selection

Recall that you used RFE (Recursive Feature Elimination) while building linear regression models. RFE is one of the many feature selection techniques used in regression.



We have studied the different parameters that can be used to compare the fit of the models. Now, we will look at the different methods of choosing the best set of predictors that shall give the least test error.

As discussed earlier, features' subset selection can be performed using two different methods:

1. Best Subset Selection
2. Stepwise Selection

A brief explanation of the Best Subset Selection algorithm (run on a dataset with p features) is as follows (please refer to the image below): You start with d=0 features, i.e. a null model M0 with no features. Now, as you increase d, you consider every model that has all combinations of d features and select a model which results in the least RSS (or largest R2). This gives you a model Md with d features. Continue this iteration by increasing the value of d by one till you reach d=p and find the models M0, M1, M2,.....,Mp.

Out of all these models M0, M1, M2,.....,Mp, select the best one, as measured by a measure such as Cp, AIC, BIC, Adjusted R2 or mean cross-validated error. The following flowchart should help us understand the algorithm. Here,

* *d*- a counter
* *p* - total number of predictors

|  |  |
| --- | --- |
| https://cdn.upgrad.com/UpGrad/temp/c965609d-9f31-4ac1-a3e4-83631e673b11/Best+Subset+Selection.png |  |

**Best Subset Selection: Count of Models**

How much is (10 2)?

**45**

**Feedback :**

*The number of ways of choosing 2 apples from 10*

**Best Subset Selection: Count of Models**

Suppose there are 14 predictors to build a model. How many models can be built using 0 predictors, 1 predictor, 2 predictors and 3 predictors?

**1, 14, 91 and 364 respectively**

**Feedback :**

*Look at the previous question and figure out what*(10 2)*means*

**Best Subset Selection: Count of Models**

How many total models can be built using 10 predictors?

210

**Feedback :**

Try to generalise the above formula as a summation from 1 to 10 for the number of predictors. What is ∑nk=0(n k)? Think of binomial expansion of (x+y)n.

We can see that the total number of models that need to be analysed for Best Subset Selection is 2p where p is the total number of predictors. If we have 20 predictors, the total number of models is 220 = 1048576 which is over a million. Hence, it becomes computationally infeasible to perform best subset selection for number of predictors greater than 40.

# Comprehension: Features' Subset Selection - Stepwise Selection

Stepwise selection can be performed in the following two ways:

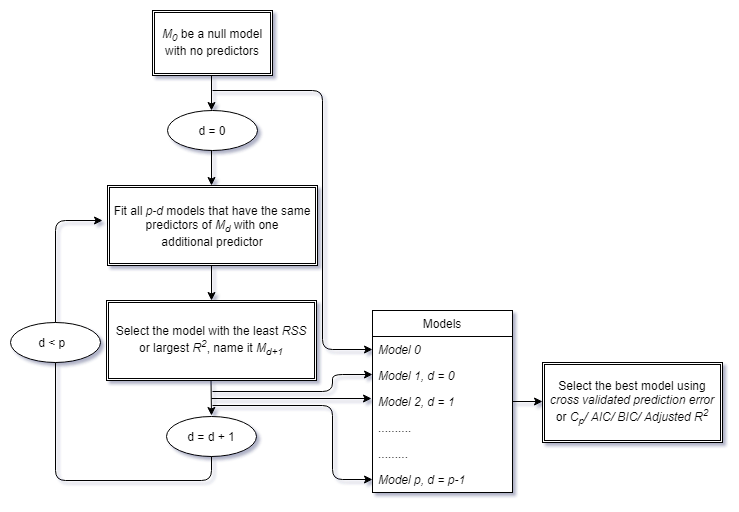
**Forward Stepwise Selection**

A brief explanation of the forward selection algorithm (run on a data set with p features) is as follows (Please refer to the flow chart below):

You start with d=0 features, i.e., a null model M0 with no features. Now, of the (p-d) remaining features, you identify one additional feature that when added to the model Md results in the least RSS (or the largest R2). This gives you a model Md+1 with one additional feature. Continue this iteration by increasing the value of d by one, until you reach d = p and find the models M0, M1, M2,.....,Mp.

Of all these models M0, M1, M2,.....,Mp, select the best one, as measured by Cp, AIC, BIC, adjusted R2 or mean cross-validated error.

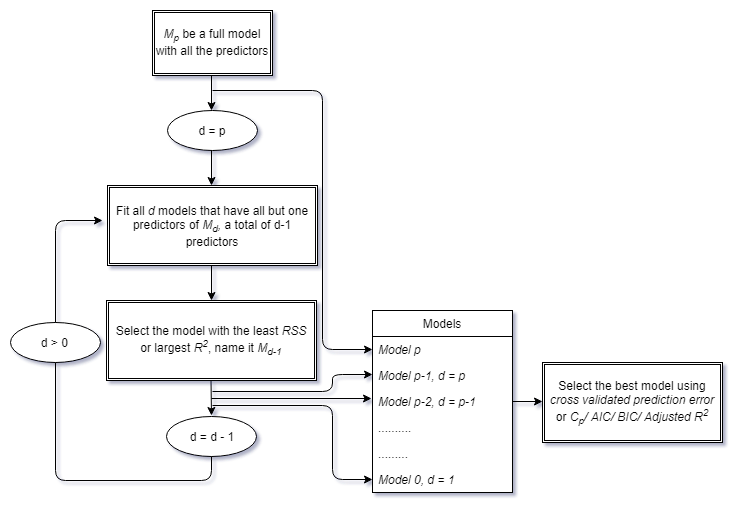
The following flow chart should help you understand the algorithm:



**Backward Stepwise Selection**

The backward selection algorithm is the opposite of the forward one; here, rather than starting with d=0 features and adding a feature in each iteration, you start with d = p features (a model Mp, with all the features as predictors) and remove a feature (the one that minimises the error (or maximises R2)) in every iteration and find the models Mp, Mp−1, Mp−2,............., M0.

Finally, of the models Mp, Mp−1, Mp−2,............., M0, you choose the best one by looking at their  Cp, AIC, BIC, adjusted R2 or mean cross-validated error.

The following flow chart should help you understand the algorithm:

**Stepwise Selection: Count of Models**

In forward stepwise selection, how many models do we fit at each step as we increase 'd'? (The total number of predictors is p.)

P0,Pd,Pd+1,....

**Feedback :**

*Look at the flow chart for forward stepwise selection.*

**Correct**



p−1,p−2,......,2,1

In backward stepwise selection, how many models do we fit at each step as we decrease 'd'? (The total number of predictors is p.)

p,p−1,p−2,......,2,1

**Feedback :**

*Look at the flow chart for backward stepwise selection.*

**Correct**



p−1,p−2,......,2,1

**Feedback :**

*Look at the flow chart for backward stepwise selection.*

**Incorrect**

How many models do we need to fit for forward stepwise selection?

1 + p(p+1)/2

**Feedback :**

*Good that you considered the null model.*

We can see that the total number of models that need to be analysed for forward stepwise selection is 1 + p(p+1)/2, where p is the total number of predictors. It is the same for backward stepwise selection.

So, if the number of predictors is 40, there are just 821 models that need to analysed, which is significantly less than 240. In this way, it is better than best subset selection, but it comes with its own limitations.

Stepwise selection does not ensure that we have chosen the best model. If we start off forward stepwise selection with predictor X1, then the best model with two predictors becomes X1 and X2, as X2 is the next best predictor. But the best model with two predictors may be X2 and X3. This can happen with backward stepwise selection too.

Also, forward stepwise selection can be applied in cases where n<p, where n is the number of observations. However, backward stepwise selection cannot be applied here, as a full model cannot be fit when n<p.

**Method Selection**

Now, suppose we are performing regression using 120 predictors on a data set of 80 observations. Which of the following methods can be used here?



Best subset selection



**Forward stepwise selection**

**Feedback :**

*When n < p and the number of predictors > 40, we can only use forward stepwise selection.*

**Correct**



Backward stepwise selection

Suppose we are regressing an independent variable y on 18 predictors on a data set with 400 observations. Which method will be able to give us the best model, i.e., one with the lowest test error?

**Best subset selection**

**Feedback :**

*Best subset selection can get the best model, as it tries each and every combination, and here, the number of predictors is also less than 40.*

**Correct**



Forward stepwise selection



Backward stepwise selection

# Common Interview Questions

**Feature Engineering**

Which of the following is an example of a linear combination to create new features?

3x + 7yz



**3x + 6siny**

**Feedback :**

*Note that the second term, 6siny, is not a linear combination*

**Incorrect**



3x + 16x^2



-18z

**Correct**

**Feature Selection**

Which of the following methods can be used for feature selection?

**Best Subset Selection**

**Correct**



**Stepwise Selection**

**Correct**



**Lasso Regression**

**Correct**



Ridge Regression

**Best Subset Selection**

Given a dataset with 8 features, how many models do you need to run, so as to arrive at the best subset for the purpose of modelling?

256 i.e. 28

**Error metrics**

What is the advantage of using measures such as Cp, AIC, BIC, Adjusted R2 or mean cross-validated error?

**They penalise the model for being too complex**

**Correct**



They penalise the model for being too simple

**Regularisation**

Which of the ones are correct pairs?

**Ridge Regression, Uses the sum of the square of coefficients as the regularisation term.**

**Correct**



Lasso Regression, Uses the sum of the square of coefficients as the regularisation term.



Ridge Regression, Uses the sum of the absolute value of coefficients as the regularisation term.



**Lasso Regression, Uses the sum of the absolute value of coefficients as the regularisation term.**

**Hyperparameter**

What is the hyperparameter applicable in regularized regression?

Top of Form

**Lambda**

**Feedback :**

*It is the weight given to the sum of the absolute values of the coefficients in case of lasso regression and the sum of the squares of coefficients in case of ridge regression in the overall cost function.*

**Correct**



Alpha

Bottom of Form

**Ridge vs Lasso**

**Lasso regression can be used for feature selection while ridge cannot be, why should we even consider using ridge?**

Though lasso regression can be used for feature selection while ridge regression cannot be, it comes at a huge computational cost. Since it does not convert into a nice invertible function, it is to be solved using an iterative process which has significantly more computational requirements compared to ridge regression which demands a simple tweak to the simple linear regression solution and can be converted to an invertible matrix and can thus be solved using matrix operations and thus has significantly lower computational costs associated with it.

**Generalised Regression**

**Consider a situation where the target variable is not linearly related to a predictor variable, can you create a linear regression model for the same? What exactly does ‘linear’ in linear regression mean?**

The ‘linear’ in linear regression does not stand for the relation between the target variable and the predictor variable. In fact, it stands for the coefficients of the predictor terms in the linear regression solution. It is entirely logical to have a linear regression solution as z= a\*sinx + b\*cosy, however, it will not be a linear regression model if the relation can be defined as y=2a\*y\*x^\*sinx + y^2\*x^2\*sinxy.

**Best Subset Selection**

Explain the process of Best Subset Selection.

*In the Best Subset Selection algorithm, we start with 0 features, i.e. a null model M0 with no features. Now, as we increase the number of features, we consider every model that has all combinations of a certain number of features. and select a model which results in the least RSS (or largest R2). This gives us a model Md with d features. We continue this iteration by increasing the value of d by one till you reach d is equal to the number of features in the dataset and find the models M0, M1, M2,....., Mp. Out of all these models M0, M1, M2,....., Mp, select the best one, as measured by measures such as Cp, AIC, BIC, Adjusted R2 or mean cross-validated error.*

**Backward step-wise selection not applicable in n<p case**

It is mentioned that "backward stepwise selection cannot be applied here, as a full model cannot be fit when n<p." I assume, n is the number of datapoints (i.e., number of rows in the dataset). In that case, did not understand why backward stepwise selection is not applicable.

Also didn't understand why this is not a constraint for foward stepwise selection. Could you explain ?

Please check this out, there are some good reasons as to why this happens (not only in BIC but in general): <https://discuss.analyticsvidhya.com/t/why-backward-selection-can-not-be-used-when-n-p/8161>

You can do forward selection whether n < p or n > p. You can read about this here: <https://gerardnico.com/data_mining/stepwise_regression>

**Ridge regression almost always has a matrix representation for the solution, while lasso requires iterations to get to the final solution. what does this mean?**

Please check these discussions out to understand, how the solution for ridge can be expressed in matrix form:

<https://stats.stackexchange.com/questions/69205/how-to-derive-the-ridge-regression-solution>

<https://stats.stackexchange.com/questions/191705/why-are-solution-to-ridge-regression-always-expressed-using-matrix-notation>

For LASSO, the reason I think it takes iterations is that LASSO does "feature selection" as well (by zeroing out some features from your dataset) hence, it has to go in iterations to find out which among the possible set of features are not useful and can be shrunk to give a sparse representation.

**Can we use lasso regression with PCA technique to make model more stable?**

**PCA**

PCA offers dimensionality reduction by reducing the multicoliniearity. The major drawback is interpretation PC's is tough wrt original features.

**Lasso Regression**

Lasso Regression itself will perform variable/feature selection and regularization. Lasso regression performs L1 regularization, which can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models.

***If your data has lot many features and main objective is not to interpret the original features, its fine to use Lasso Regression with PCA technique.***

***If your data has few features, then only Lasso Regression can be used.***

Happy Learning!

In addition please go through the following link.

<https://stats.stackexchange.com/questions/386385/can-i-use-pca-after-lasso-variable-selection>

**Lasso regression can be used for feature selection while it also does regularization. Ridge cannot be be used for feature selection** why should we even consider using ridge then ? Just to save the computational power ?

When you have less number of features, you can use ridge and yes ridge is computationally more efficient. If you want to reduce the dimensionality, you can use a combination of PCA and Ridge.

# Graded Questions: Ridge and Lasso Regression

**Multicollinearity**

Why is multicollinearity a problem in linear regression? Select the correct option.

The least squares solution is undefined.



**Singularity of the matrices that need to be inverted for the least squares solution causes instability/breakdown of the algorithm.**

**Feedback :**

*Multicollinearity means that some of the attributes (features) are linearly dependent. This means that in a feature matrix, one or more columns can be written as a linear combination of the other features. As a result, the determinant of the matrix will become 0 causing the matrix to be singular.*

**Correct**



The least squares solution has large errors.

**Matrix**

What is the order of the matrix that needs to be inverted for the least squares solution on a data set having n datapoints, each with d explanatory attributes?

n X d



d X d

**Feedback :**

*The order of [Transpose(X)] and X are d X n and n X d, respectively. Thus, the order of the matrix [Transpose(X)]X is d X d.*

**Correct**



n X n



**d X n**

**Feedback :**

*The matrix being inverted is [Transpose(X)]X. What is the order of X? From this, find the order of [Transpose(X)]X.*

Ridge regression is a technique for analysing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large; so, they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard test errors.

Consider a data set (split into training and test) on which you build a ridge regression model. Assume that only the raw attributes have been used and no new features have been developed for model building.

Based on the information above, answer the following questions.

**Regularisation**

As λ increases from 0 to infinity, select the correct option that describes the pattern of the residual sum of squares (RSS) of the training data set.

**Increases steadily**

*Differentiating the cost function with lambda = 0 gives the values of the coefficients that minimise the RSS. Again, putting λ = infinity gives us a constant model with the maximum RSS. Thus, the RSS steadily increases with the variation of lambda.*

As λ increases from 0 to infinity, select the correct option that describes the pattern of the variance of the model.

Decreases steadily

*When λ=0, the alphas have their least square estimate values. The actual estimates heavily depend on the training data, and hence, the variance is high. As we increase λ, the alphas start decreasing and the model becomes simpler. In the limiting case of λ approaching infinity, all betas reduce to zero, and the model predicts a constant and has no variance.*

As λ increases from 0 to infinity, select the correct option that describes the pattern of the (squared) bias of the model.

Increases steadily

**Feedback :**

*When λ=0, alphas have their least-square estimate values and, hence, have the least bias. As λ increases, alphas start reducing towards zero, the model fits less accurately to the training data, and hence, the bias increases. In the limiting case of λ approaching infinity, the model predicts a constant, and hence, bias is maximum.*

**Comprehension: Income Prediction**

You are hired as a consultant by a bank to predict the income of the HH (household) customers of the bank.

You have a data set of a few of the HH customers, in which the dependent variable is the income and there are 86 derived features with which you are going to build the model.

Now, based on this, answer the questions that follow.

**Reasons for High Variance**

After using linear regression, we find that the bias is very low, while the variance is very high. What are the possible reasons for this? Select all that apply.

Overfitting

**Feedback :**

Overfitting leads to low bias and high variance.

**Correct**You missed this!



**Multicollinearity**

**Feedback :**

Multicollinearity causes the model to not understand the trend and pattern correctly, thereby leading to high variance. Bias tends to be low.

**Correct**



Outliers

**Feedback :**

Since the model is trained on outliers, it has not captured the actual trend of the data, thereby leading to high variance. Bias can be low if it has explained even the outliers correctly.

**Correct**You missed this!

**Hyperparameter Value**

You decide to use regularisation to tackle the problem above, i.e., you use ridge and lasso regression. What will happen if you use a very large value of the hyperparameter λ?

**The test error will be high.**

**Feedback :**

Even though the variance will be very low, the test error will be high, as the model will not have captured the behaviour of the data correctly.

**Correct**



**Ridge will lead to some of the coefficients to be very close to 0.**

**Feedback :**

Ridge leads to the shrinkage of the coefficients.

**Correct**



Ridge will cause some of the coefficients to be 0.



**Lasso will cause some of the coefficients to be 0.**

**Feedback :**

Lasso will cause some of the coefficients to be 0.

**Correct**

**Model Selection**

You find the cross-validation error as follows:

Full model with all variables: 109324

Ridge regression (at optimal λ with all variables): 58469

Lasso regression (at optimal λ and 18 variables): 61367

Which model will you use?

**Lasso regression**

**Feedback :**

*Lasso has reduced complexity by variable selection, and the error is also comparable to that in ridge regression. Thus, it has less complexity and will perform better than ridge regression.*

**Parameters Testing**

Which of the following models should be used to predict on the test set?

|  |  |
| --- | --- |
| Model | BIC |
| M1 | 186.1 |
| M2 | 193.6 |
| M3 | 188.9 |

M1

**Feedback :**

*The lower the*BIC*, the less will the test error be.*

Which of the following models should be used to predict on the test set?

|  |  |
| --- | --- |
| Model | Adjusted R2 |
| M1 | 0.72 |
| M2 | 0.67 |
| M3 | 0.83 |

M3

**Feedback :**

*The higher the adjusted*R2*, the less will the test error be.*