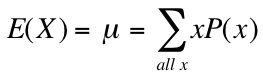
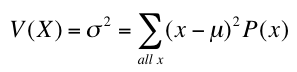
Module 3 – Assessing Accuracy of Linear Regression

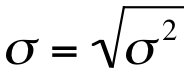
1. Variable – Discrete and Continuous
2. Probability distribution or Density Function
3. Discrete Probability Distribution
   1. X = Outcome eg in rolling of dice 1,2 ..6; in flipping of coin: head or tail
   2. P(X) = probability of X occurring – 1/6 for dice, ½ for coin
   3. x = 1; P(X=x) = 1/6;
   4. Expected value of X,



* 1. Variance



* 1. Standard Deviation

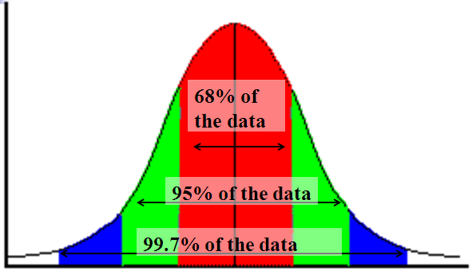


1. Continuous Probability Distribution
   1. Probability Density Function:
      1. f(x) ≥ 0 for all real x



* + 1.  f(x) = 1
    2. P(a ≤ x ≤ b) = f(x) dx
  1. Normal Distribution is defined by mean, μ and standard deviation, σ.
  2. Probability density function of a Normal Distribution



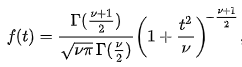
* 1. The 68 – 95 – 99.7 rule:

68% values fall within 1 standard deviation ,

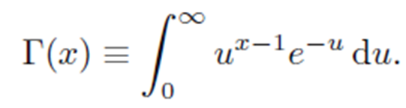
95% values within 2 σ and

99.7 % values within μ ± 3 σ

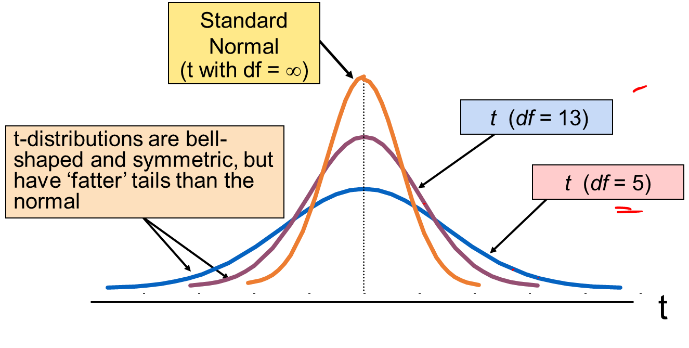
1. Student’s t-distribution
   1. Probability distribution function



* 1. is degrees of freedom and the only variable defining the distribution
  2. and Gamma Function



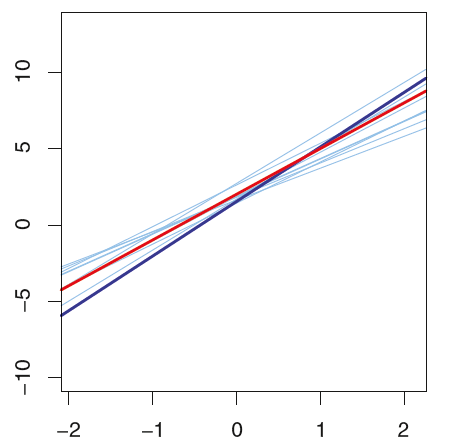
* 1. t-distribution and relation to normal distribution where standard normal distribution has mean, μ = 0, and standard deviation, σ = 1.



1. Unbiased and biased estimators
   1. Standard Error SE(w0) estimates the sampling error
   2. Unbiased estimators:

For a univariate y=w0 + w1x

w0, w1 are considered to be unbiased if expected value of w0 and w1, E(w0) and E(w1) is w0 and w1. So a sample’s w0 and w1 can be different from w0 + w1. But the expected value will be w0 and w1.



* 1. Biased Estimator: In case there is an error in sampling then there can be a consistent difference between calculated w0 and w1 and actual w0 + w1.
  2. Standard Error, SE(w0) =

where s = standard deviation of pred y from y and

n = sample size

* 1. Relationship between dependent and independent variable

To check the significance of the dependent variable in estimating the independent variable, we say that it is significant if the value is sufficiently different from 0 taking into consideration the standard error:

Where

And

Where

1. RSE, Residual Standard Error

RSE =

1. Total Sum of Squares Dispersion of y from mean y

TSS =

1. Residual Sum of Squares, Difference of predicted to actual y

RSS =

1. R square, How much of variation of y is captured by prediction as a ratio

MODULE 4

Multiple Linear Regression

4.1

1. Linear Regression on multiple features/independent variables
   1. Linear regression of the form

y=w0 + w1x1 + w2x2 + w3x3 + …… + wDxD

* 1. Observations are depicted as

( x11, x21, … , xD1, y1 ),

( x12, x22, … , xD2, y2 ),

…

( x1n, x2n, … , xDn, yn )

* 1. The series of simultaneous equations thus formed can be written in **matrix** form as

Y = WX

* 1. For any given set of values for w0 , w1, w2, w3, …… wD the **error term** can be calculated as:

E(w0 , w1, w2, w3, …… wD) =

Note: ½ is put for convenience only as minimizing error term requires first derivative of the square term – derivative of x2 is 2x – and 2 and ½ will cancel.

* 1. As proved for single linear regression, the error function is a **convex function**. Hence there is a **single stationary point which is minima**. Hence first differential equal to zero would give us the required point.

= 0

= 0

……

= 0

* 1. Bringing the w0 , w1, w2, w3, …… wD terms together in the above

…

* 1. In matrix form we get

AW = Y or

* 1. And solving we get, W = A-1Y
  2. Hence we get all w0 , w1, w2, w3, …… wD where Error term is minimum

4.2

1. Gradient Descent Algorithm

In gradient descent algorithm as seen in module 2 we take an initial set of values for w0 , w1, w2, w3, …… wD. Then we reduce error by changing the value of wi by learning rate() times the 1st derivative of wi.

wjk+1 = wik -

where

Repeat until Ek+1 - Ek < 10-20 or the required range.

1. Stochastic Gradient Descent
2. The Algorithm
   1. Arrive at initial w0 to wD. (how?)
   2. Initialize k=1
   3. Repeat to convergence
      1. Randomly shuffle the data set
      2. For n = 1 to N
         1. wk+1 = wk -
         2. k=k+1
3. What happens at wk+1 = wk -

{\displaystyle {\begin{bmatrix}w\_{1}\\w\_{2}\end{bmatrix}}:={\begin{bmatrix}w\_{1}\\w\_{2}\end{bmatrix}}-\eta {\begin{bmatrix}{\frac {\partial }{\partial w\_{1}}}(w\_{1}+w\_{2}x\_{i}-y\_{i})^{2}\\{\frac {\partial }{\partial w\_{2}}}(w\_{1}+w\_{2}x\_{i}-y\_{i})^{2}\end{bmatrix}}={\begin{bmatrix}w\_{1}\\w\_{2}\end{bmatrix}}-\eta {\begin{bmatrix}2(w\_{1}+w\_{2}x\_{i}-y\_{i})\\2x\_{i}(w\_{1}+w\_{2}x\_{i}-y\_{i})\end{bmatrix}}.}

1. Mini Batch Gradient Descent
   1. Divide the data set into T batches
   2. Initialize w
   3. Repeat till convergence:
   4. For t = 1 to T
      1. Do gradient descent on tth batch

4.3

1. Dependency of independent variable on dependent variables

H0 : wD-q+1 = wD-q+2 = wD-q+3 = wD-q+4, …… = wD = 0

H1 : atleast 1 is non zero

F- statistics to test the hypothesis

F =

where RSS0 corresponds to model with D – q parameters

and RSS corresponds to model with all D parameters

If F statistic is sufficiently large and corresponding p-value sufficiently large then null hypothesis does not hold true.

4.4 Best Subset Selection

1. Divide data into training and test data – 2/3 and 1/3. Create model, M0 with 0 features – this will predict independent variable, y, to be mean of all seen values of y. Create models with 1 feature each and pick the best model, M1 with min training error. Next create models with 2 features each and pick the best model M2. And so on till you have one model with all features MD. Now run models M0 to MD on the testing data and choose the model with least error.

(2D models to be created)

4.5

1. Forward Stepwise selection: Create model, M0 with 0 features – this will predict independent variable, y, to be mean of all seen values of y. Take a model with 1 feature each and take the best model, M1. Add 1 additional feature and take the best model, M2 with first selected feature and 1 additional feature. And so on till MD. Now run models M1 – MD on testing data. (One can also add features till there is no significant improvement in model on adding an additional feature?) And choose the model with smallest RSS or largest R2 .

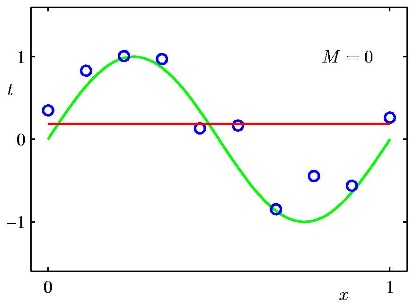
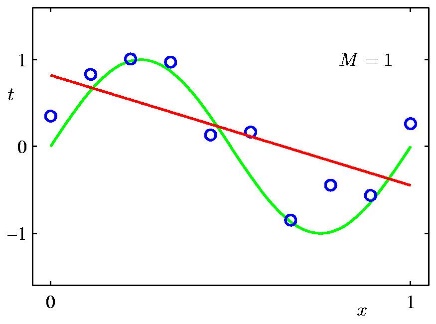
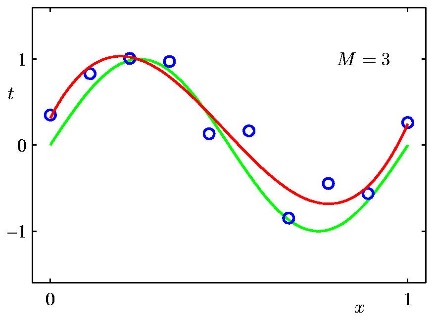
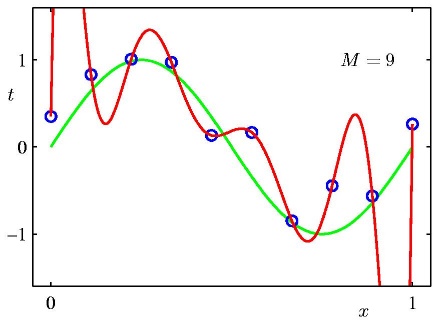
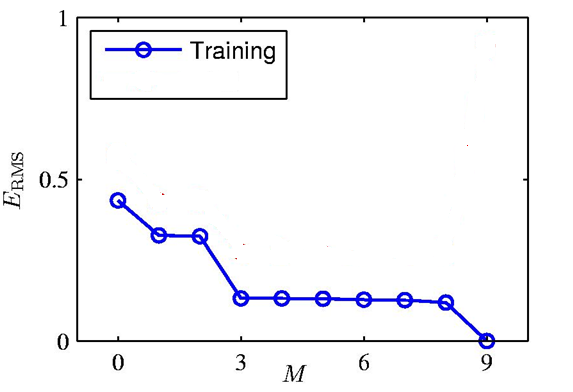
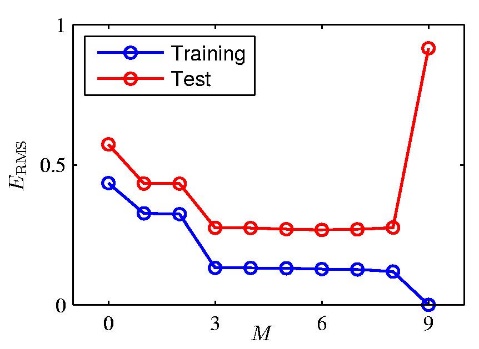
([D \* (D+1)]/2 + 1 models to be created )

1. Backward Stepwise selection: Take a model, MD, with all features. Now create models by removing 1 feature and take the best model, MD-1 . Next create models by removing another feature and take the best model, MD-2 with D-2 features. And so on till M0. Now run models M1 – MD on testing data. And choose the model with smallest RSS or largest R2 .

Module 5

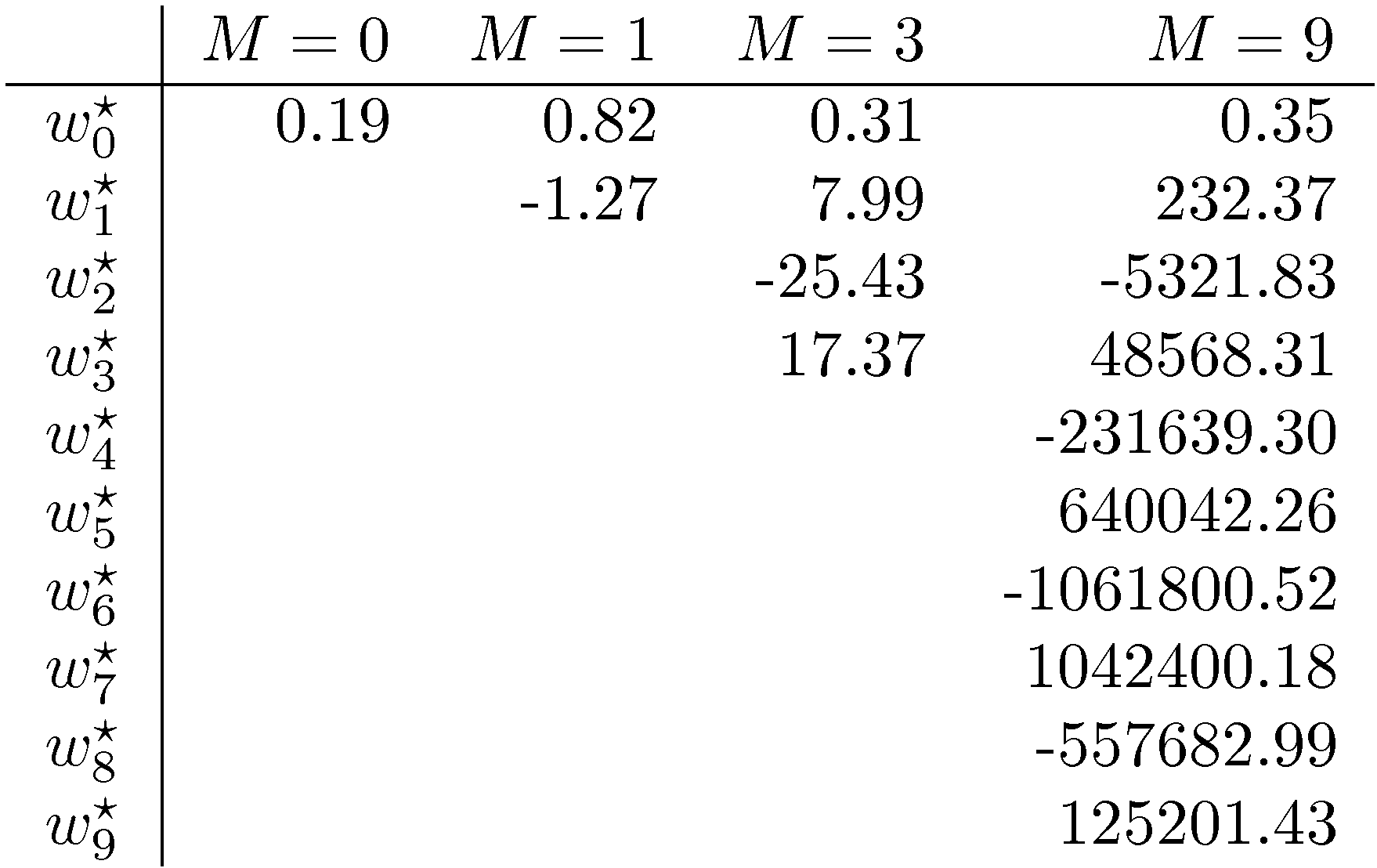
5.1

1. Third order polynomial

5.2

1. Overfitting



In overfitting, the coefficients become quite big.

5.3 Regularization

1. Solving for overfitting
   1. Sufficient data – 10 times parameters is rule of thumb
   2. Put a bound on regression coefficients by regularization
   3. Ridge Regression
      1. Add constraint sum of squares of coefficients is < s => < s or
      2. Minimize

E

* 1. Lasso Regression

Add constraint sum of absolute value of coefficients is < s => < s

* 1. Ridge regression is more popular than lasso

1. Bias – Variance Decomposition
   1. Quality of Fit
      1. Bias – Simplifying assumptions made by model to make the target function easier to learn.
      2. Variance – model gives different results for different training sets.
      3. Underfitting caused by High Bias. A straight line fitted to a polynomial.
      4. Correct Fit – good balance between bias and variance
      5. Overfitting – high variance – training results do not hold good for test data.
   2. Addressing overfitting
      1. Reduce the number of features so that data is 10 times the features. Identify critical features with business knowledge, or use forward or backward selection algorithm
      2. Regularization : Use lasso or ridge regression to put constraint on sum of coefficients