CS60050
Machine Learning

Assignment 1 Report

Problem Statement:

Predict the price of a house based on its area, number of floors in the house, number of bedrooms, number of bathrooms. Since the problem is a prediction based problem, that is, given the features of the house, predict its price, using Regression.

Data:

A Dataset of houses is provided with the features namely, sqft (area of house), floors (number of floors), bedrooms (number of bedrooms), bathroom (number of bathroom) and price of the house. First 80% of the rows are used to train the Regression model. Last 20% of the data was completely unseen during the training and was used to test the performance of the model.

Preprocessing:
Data was scaled down to value 0 to 1 by the transformation of each data point by
(value – min)/(max – min).

PART A (Implementing Linear Regression):

Model Used:

y = f(sqft, floors, bedrooms, bathrooms)
  = A0 + A1(sqft) + A2(floors) + A3(bedrooms) + A4(bathrooms)

Model Parameters A0, A1, A2, A3, A4.

Final parameters obtained: (Without Regularization)
(A0 = 0.03854472, A1 = 0.00283557, A2 = 0.03859827, A3 = 0.01197913, A4 = 0.05465918)

Final parameters obtained: (Without Regularization)
Lambda = 0.001
(A0= 0.0448580 A1= 0.0017947 A2= 0.0299428606 A3= 0.009211102717 A4= 0.037695445)
Lambda = 0.01
(A0= 0.0448581174 A1= 0.0017947351 A2= 0.02994276 A3= 0.0092110602 A4= 0.0376952)
Lambda = 0.1
(A0= 0.04485866 A1= 0.0017946645 A2= 0.029941757 A3= 0.0092106419 A4= 0.03769379)
Lambda = 1
(A0= 0.04489538 A1= 0.001789341 A2= 0.02988058 A3= 0.009193942 A4= 0.03760007)
Lambda = 10
(A0= 0.045167462 A1= 0.0017502805 A2= 0.029422183 A3= 0.0090659880 A4= 0.036904188)
Lambda = 100
(A0= 0.0474684429 A1= 0.0014291341 A2= 0.025364353 A3= 0.0080286184 A4= 0.031129331)

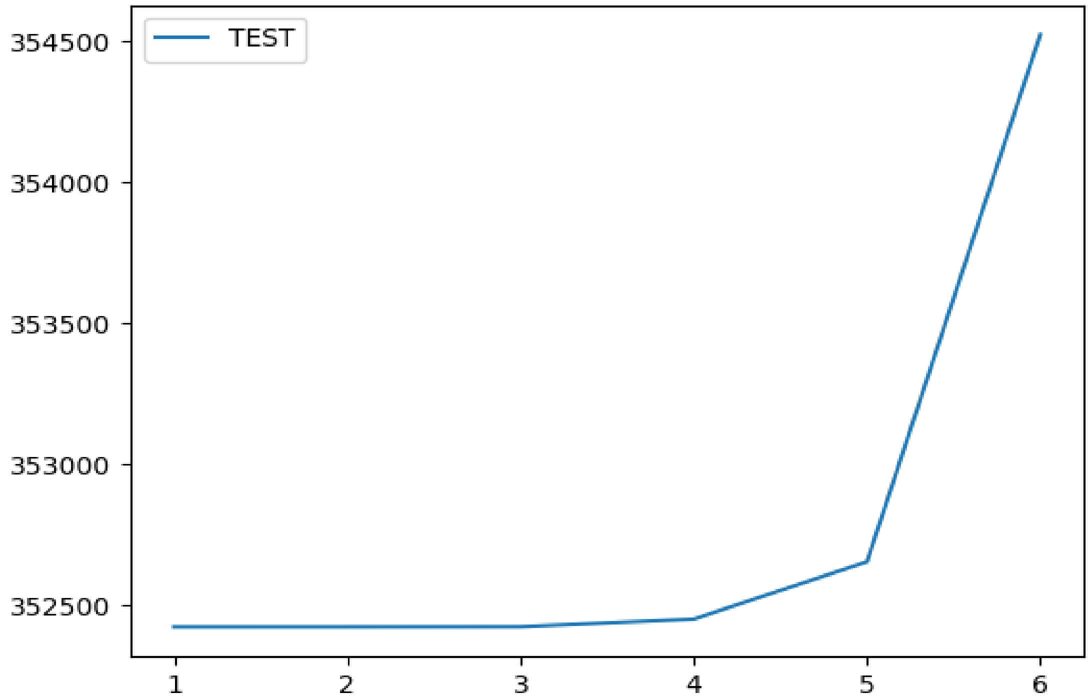

Fig 1. RMSE vs Regularization Weights

Part B(Experimenting with the Optimization Algorithm)

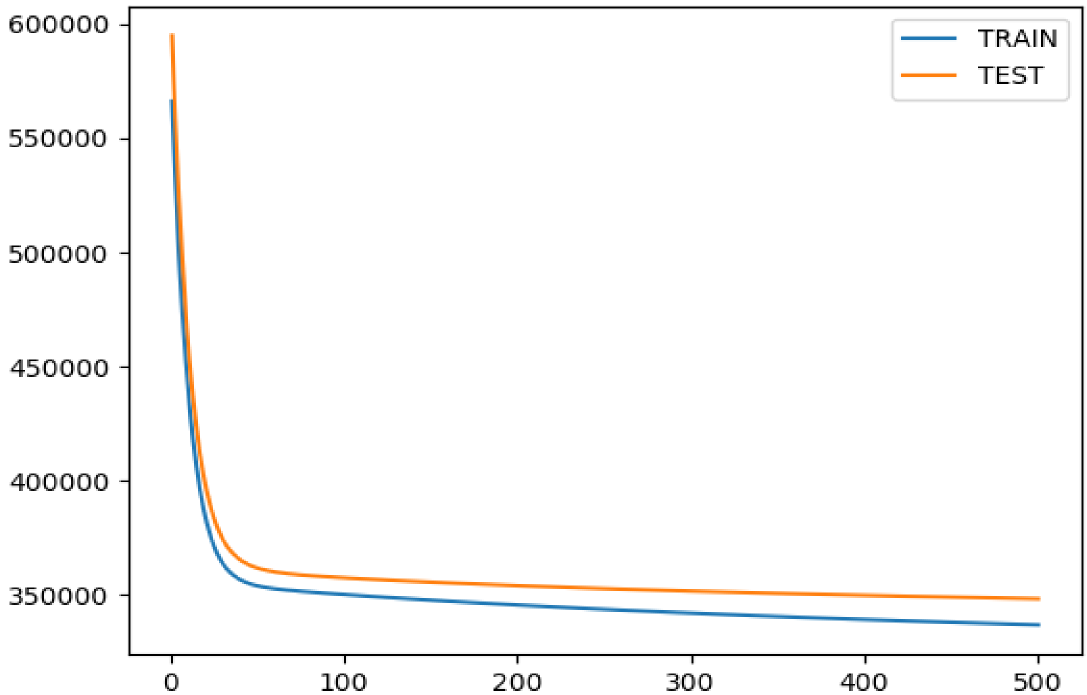

Fig 2. RMSE Training and RMSE Test vs Iteration for Gradient Descent

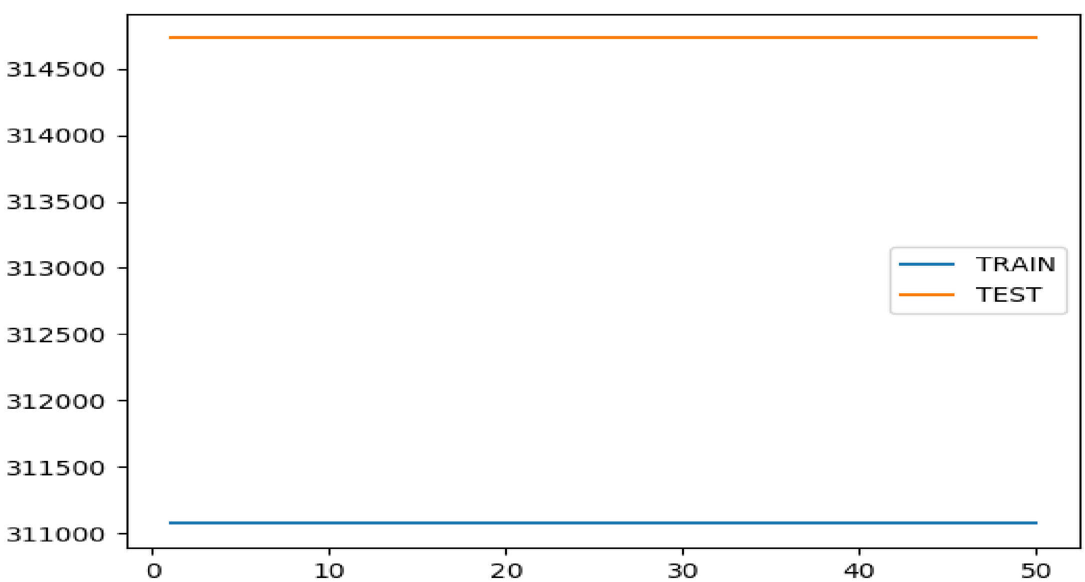

Fig 3. RMSE Training and RMSE Test vs Iteration for IRLS

I would prefer IRLS over Gradient Descent because IRLS yields direct result at one go for the best possible fit however Gradient Descent takes a lot of iterations to converge to the local minima being slower and time consuming. Although IRLS cannot be applied everywhere (specially when the matrix is vulnerable to lose rank however if it can it should be preffered. However, if dataset is huge comptation for IRLS is very high. Again in this case we would prefer gradient descent. Hence in a general case it is better to use Gradient Descent. IRLS has limited application. It works fine in this case.

PART C(Experimenting with Combination of Features)

Model Used:
LINEAR COMBINATION OF PARAMETERS
y = f(sqft, floors, bedrooms, bathrooms)
= A0 + A1(sqft) + A2(floors) + A3(bedrooms) + A4(bathrooms)

Model Parameters A0, A1, A2, A3, A4.

Final parameters obtained: (Without Regularization)(Learning Rate = 0.05)
(A0 = 0.0385447224351 A1 = 0.00283556832544 A2 = 0.0385982733285 A3 = 0.0119791328198 A4 = 0.0546591833259)

Model Used:
QUADRATIC COMBINATION OF PARAMETERS
y = f(sqft, floors, bedrooms, bathrooms)
= A0 + A1(sqft) + A2(floors) + A3(bedrooms) + A4(bathrooms) + A5(sqft)^2 + A6(floors)^2 + A7(bedrooms)^2 + A8(bathrooms)^2

Model Parameters A0, A1, A2, A3, A4, A5, A6, A7, A8.

Final parameters obtained: (Without Regularization)(Learning Rate = 0.05)
(A0 = 0.0369044664706 A1 = 0.00276007080547 A2 = 0.0336166109309 A3 = 0.0114624708445 A4 = 0.0516435356065 A5 = 0.000454961558687 A6 = 0.0128831241677 A7 = 0.00214843817556 A8 = 0.0316672515121)

Model Used:
CUBIC COMBINATION OF PARAMETERS
y = f(sqft, floors, bedrooms, bathrooms)
= A0 + A1(sqft) + A2(floors) + A3(bedrooms) + A4(bathrooms) + A5(sqft)^2 + A6(floors)^2 + A7(bedrooms)^2 + A8(bathrooms)^2 + A9(sqft)^3 + A10(floors)^3 + A11(bedrooms)^3 + A12(bathrooms)^3

Model Parameters A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12.

Final parameters obtained: (Without Regularization)(Learning Rate = 0.05)
(A0 = 0.0367358503231 A1 = 0.00274280440706 A2 = 0.0326556294064 A3 = 0.0113691437282 A4 = 0.0509983700457 A5 = 0.000451538016081 A6 = 0.0122506945226 A7 = 0.00212826612996 A8 = 0.0312416577677 A9 = 0.000195380602729 A10 = 0.0043415135068 A11 = 0.000371100654975 A12 = 0.0179618292502)

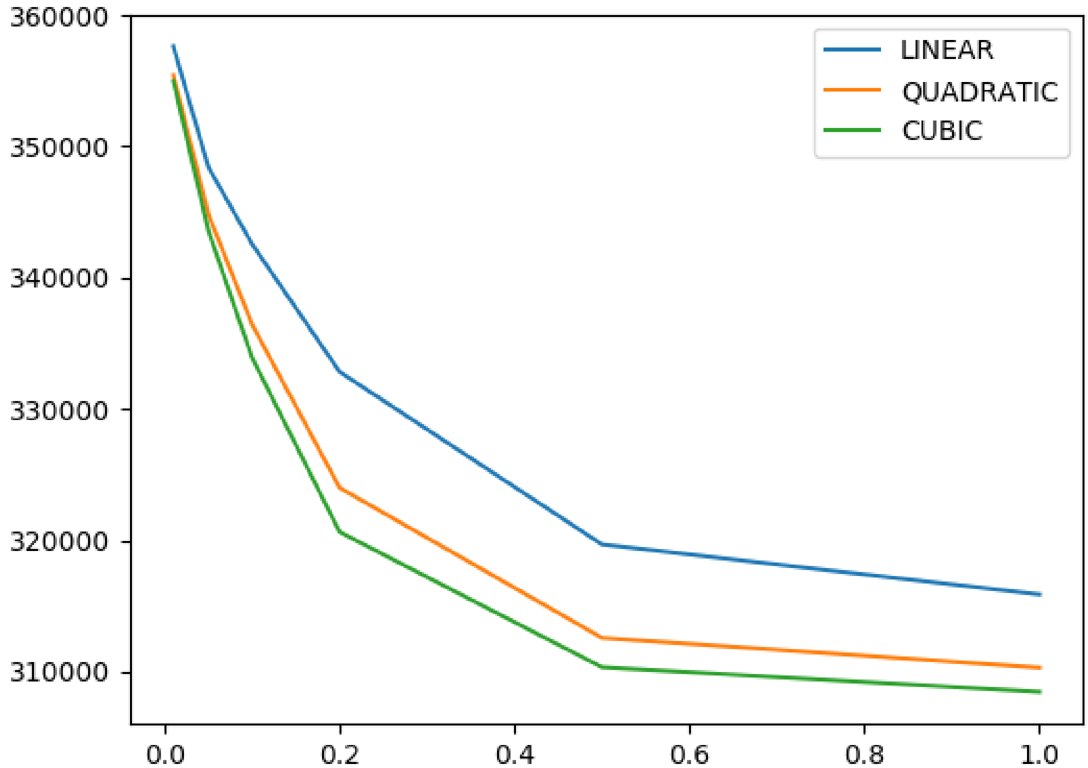

Fig 4: RMSE vs Learning Rate Plot for Linear Quadratic and Cubic Parameter Combinations

PART D(Experimenting with Cost Functions)

I. Mean Absolute Error Cost Function used in Linear Regression keeping all other parts of the algorithm unchanged

[A0 = 0.01445922, A1 = 0.01445922, A2 = 0.01445922, A3 = 0.01445922, A4 = 0.01445922]

II. Mean Squared Error Cost Function used in Linear Regression keeping all other parts of the algorithm unchanged

[A0 = 0.00022856, A1 = 0.01730542, A2 = 0.01903084, A3 = 0.04036871, A4 = 0.20306277]

III. Mean Cubic Error Cost Function used in Linear Regression keeping all other parts of the algorithm unchanged

The algorithm diverges for every value of learning rate taken in the model.

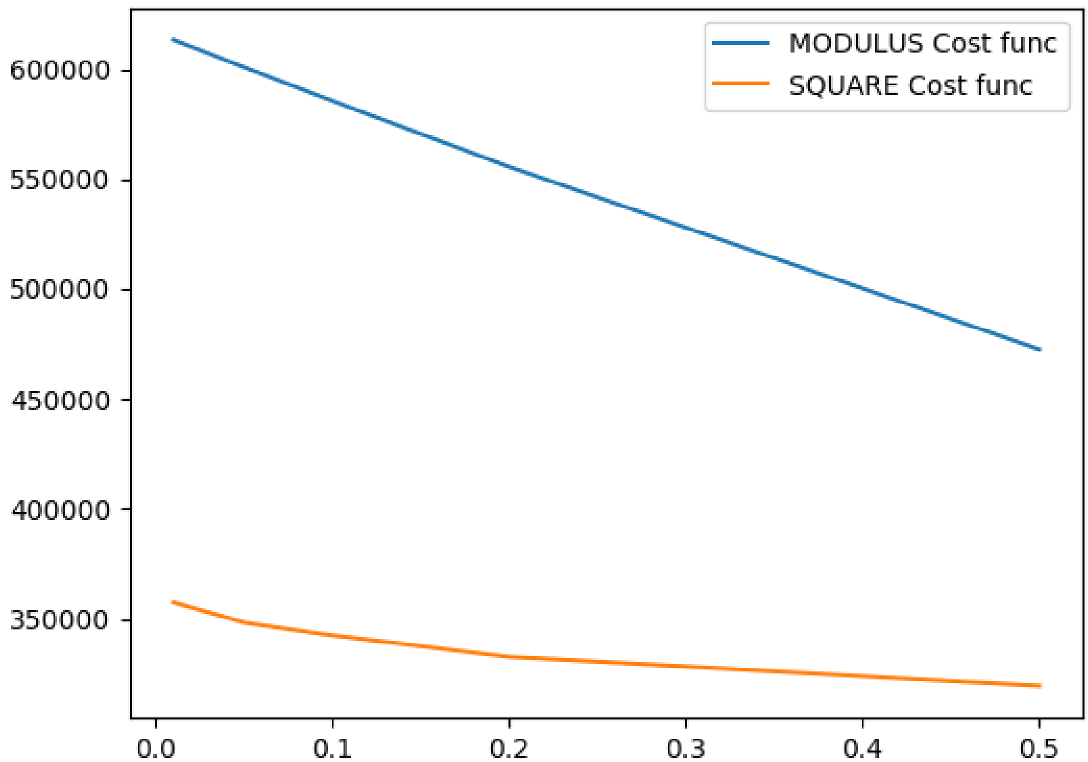

Fig 5. RMSE vs Learning rate for each Cost Function

Since the data is sufficient, I would prefer using the third degree terms as we see the error reduces significant using more features. Given data is limited it is advised to go for the model which is less complex because model with higher complexity leads to overfitting.