Machine Learning: Assignment 1
Report

By Ayush Bansal (15CS30006)

Part (a): Implementing Linear Regression

Model Parameters (for different values of Regularisation Terms):

Regularisation: 0
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.38163370597e-16 (Preferred Model)

Regularisation: 0.001
W: [[0.04290763 0.02836742 0.04498156 0.48086435]]
B: 1.38296931262e-16

Regularisation: 0.005
W: [[0.04297637 0.02958632 0.04616856 0.47772039]]
B: 1.38430491926e-16

Regularisation: 0.01
W: [[0.04305619 0.03106528 0.0476068 0.47387169]]
B: 1.38183918392e-16

Regularisation: 0.02
W: [[0.04319669 0.03388175 0.05033936 0.46643235]]
B: 1.39036651866e-16

Regularisation: 0.05
W: [[0.04348464 0.04132523 0.05751546 0.44596415]]
B: 1.38574326488e-16

Regularisation: 0.1
W: [[0.04362235 0.05104476 0.06675833 0.41687315]]
B: 1.37772962501e-16

Regularisation: 0.2
W: [[0.0431146 0.06373317 0.07847004 0.37178893]]
B: 1.25064151567e-16

Regularisation: 0.3
W: [[0.04207433 0.0709786   0.08478586 0.33794235]]
B: 1.25033329876e-16

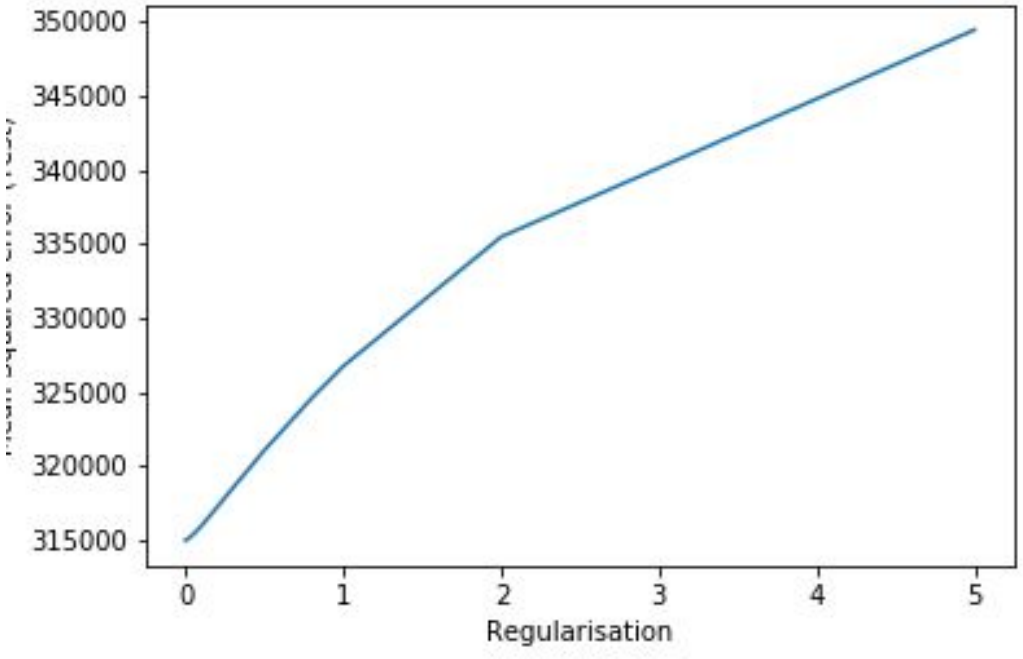
Regularisation: 0.5
W: [[0.03946667 0.07726215 0.08944991 0.28933279]]
B: 1.25320998999e-16

Regularisation: 0.8
W: [[0.03553803 0.07799035 0.0883034   0.24145273]]
B: 1.24457991628e-16

Regularisation: 1.0
W: [[0.03318689 0.07627508 0.0856189   0.21855156]]
B: 1.18118997007e-16

Regularisation: 2.0
W: [[0.02464905 0.063395   0.06974217 0.1508889]]
B: 1.24591552293e-16

Regularisation: 5.0
W: [[0.01375587 0.03865522 0.04189443 0.08010735]]
B: 1.24468265525e-16



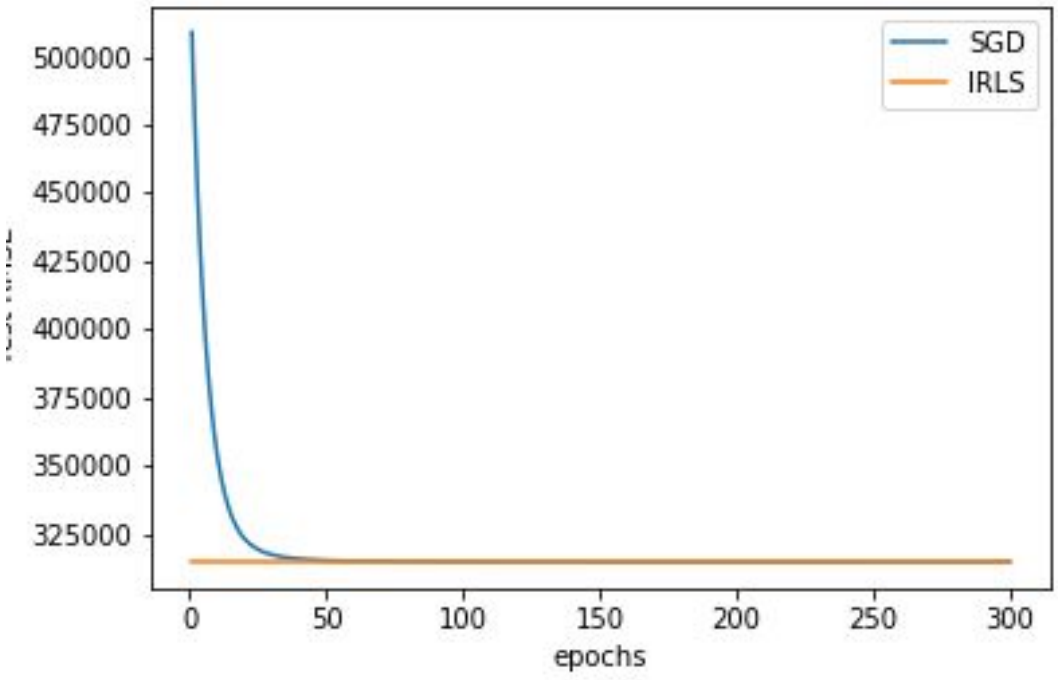
Plot of Test RMSE vs Weightage of Regularisation terms

Part (b): Experimenting with optimization algorithms

Model Parameters (Last Params W_4 corresponds to bias term):

Gradient Descent:
W : [[4.28900544e-02, 2.80593780e-02, 4.46814925e-02,
4.81656579e-01, 2.38139092e-15]]

Iterative Re-weighted Least Square:
W : [[4.28897509e-02, 2.80575827e-02, 4.46796089e-02,
4.81659611e-01, 1.29922501e-16]] (Preferred Model)



Plot of Test RMSE vs No of Iterations

Preferred optimization Algorithm for this Problem : For this problem, we would want to use Iterative re-weighted Least Square Algorithm. As we can see from the RMSE vs no_of_iterations plot, this method gives us the correct result in 1 epoch, i.e., finds the solution to the optimization problem directly.

Part (c): Experimenting with combinations of features

Model Parameters (for different values of Learning rates for different combination of features):

Linear Combination of Features

Learning Rate: 0.001
W: [[0.13023408 0.13590333 0.22919323 0.2583774]]
B: 5.70262942324e-17

Learning Rate: 0.002
W: [[0.05824802 0.1252119 0.14044848 0.32352888]]
B: 8.5284854174e-17

Learning Rate: 0.005
W: [[0.04443523 0.03732991 0.05413245 0.466231]]
B: 1.37413376096e-16

Learning Rate: 0.01
W: [[0.04291138 0.02818548 0.04481385 0.4814436]]
B: 1.38894872083e-16

Learning Rate: 0.02
W: [[0.04288976 0.02805764 0.04467967 0.48165951]]
B: 1.3887021473e-16

Learning Rate: 0.05
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.38142822803e-16

Learning Rate: 0.08
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.38968844144e-16

Learning Rate: 0.1
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.39190760325e-16

Quadratic Combination of Features

Learning Rate: 0.001
W: [[0.0180407 -0.5335147 -0.22125778 0.20548923 -0.06729786 -0.31175639
0.07035724 0.33039539 0.18623357 0.38477806 0.27250281 0.14035198
-0.12611893 0.17145546]]
B: 1.09226527957e-16

Learning Rate: 0.002
W: [[0.44468439 -0.1720388 -0.33131643 0.10648573 -0.04516709 -0.07796143
-0.12445444 -0.1815536 -0.1026485 0.33474652 0.12575811 0.17851654
0.02228346 0.38785292]]
B: 1.43183607763e-16

Learning Rate: 0.005
W: [[0.01232917 0.35860985 0.09880835 -0.12728462 -0.04723773 0.16502405
0.12653775 -0.23388354 -0.09151423 -0.22818411 -0.13437211 -0.04043717
0.19912148 0.60369519]]
B: 9.67071676899e-17

Learning Rate: 0.01
W: [[0.28812432 -0.13241587 -0.12094905 -0.00575959 -0.03356024 0.00440327
-0.19040588 -0.0445494 0.04665895 0.3779656 -0.24882752 0.00141676
-0.05599396 0.70188312]]
B: 8.32062392767e-17

Learning Rate: 0.02
W: [[0.11463857 -0.07834634 -0.0229218 -0.25902779 -0.02277647 0.06093879
-0.08302089 -0.04261534 0.03110439 0.17011196 -0.05674345 -0.01816671
0.01353313 0.78737846]]
B: 6.84118271961e-17 (Preferred Model)

Learning Rate: 0.05
W: [[0.08295 0.11881637 0.04003727 -0.28069776 -0.02170564 0.08550715
-0.04754874 -0.0757716 -0.1510338 0.25430744 -0.18470728 -0.01317022
-0.23751362 1.05125593]]
B: 3.07600484509e-17

Learning Rate: 0.08
W: [[0.09061248 -0.03443621 0.0131175 -0.18646033 -0.02229769 0.09766502
-0.05332516 -0.09032575 -0.00930013 0.43643601 -0.39197318 -0.00392286
-0.40691213 1.19298531]]
B: 2.73039094065e-17

Learning Rate: 0.1
W: [[8.55917646e-02 -1.10201940e-01 4.80618949e-04 -1.43746422e-01
-2.26249013e-02 1.05461282e-01 -4.86466884e-02 -9.77810639e-02
5.22623466e-02 5.13192035e-01 -4.64692746e-01 3.54050406e-04
-4.77596793e-01 1.24132290e+00]]
B: 2.67532285124e-17

Cubic Combination of Features

Learning Rate: 0.001
W: [[0.25977922 0.36657341 -0.11131576 0.0381969 -0.25985316 -0.07308772
0.2562231 -0.10098219 -0.19131803 -0.16865699 0.0459678 0.14289665
0.18502473 -0.07623825 -0.22217098 -0.24805118 0.29546638 -0.08058951
0.03446264 -0.08810948 -0.42985175 0.41026233 0.44246752 0.14679672]]
B: 1.38278643725e-16

Learning Rate: 0.002
W: [[0.05961119 -0.34899338 0.06495774 0.40987408 -0.13222216 0.18839273
-0.24255342 0.16995915 -0.3409491 0.2601234 0.20451579 0.17634073
-0.2887499 -0.0170577 0.23189535 0.2281567 -0.31812565 -0.1606393
0.24222977 0.24186803 -0.17022154 0.16065619 -0.10076056 0.05250802]]
B: 8.81068882784e-17

Learning Rate: 0.005
W: [[0.20448823 -0.04190484 -0.18505977 0.14383748 0.18140436 0.15361629
-0.49019693 0.13178417 0.0779908 -0.01817264 0.19774877 -0.15091538
-0.05862735 -0.202978 0.19502746 -0.03441413 -0.18870501 0.09494492
0.12121597 -0.14077256 0.00222454 -0.00508613 0.25943008 0.3100761]]
B: 1.40602393811e-16

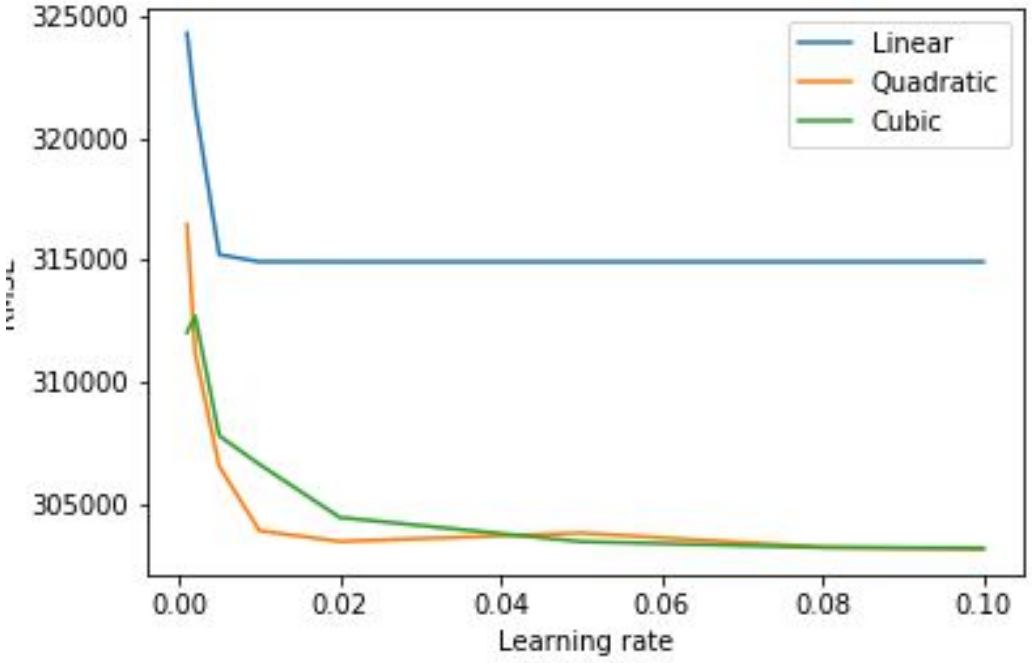
Learning Rate: 0.01
W: [[-0.05147127 -0.0594089 0.06142463 -0.05566401 -0.12751286 -0.10853915
0.05795625 0.14846218 -0.21287204 0.28627803 0.07053615 -0.17585933
0.25707278 -0.11552623 -0.17979441 -0.12904225 0.21411964 0.15952024
-0.18806225 0.32121112 0.15374787 -0.17764675 0.03643236 0.37209585]]
B: 8.04857112774e-17

Learning Rate: 0.02
W: [[0.16055289 -0.05720662 -0.14904614 -0.30668734 0.32068801 -0.03989426
-0.14210411 -0.15672769 0.3097212 -0.15720551 -0.12883233 -0.03402534
-0.11163249 0.00997363 0.06940518 -0.32926514 0.20561817 0.09073615
0.08293523 0.05896887 0.24827002 -0.26333987 0.2787478 0.58345688]]
B: 1.09507416309e-16

Learning Rate: 0.05
W: [[0.09467923 -0.01226297 0.02736467 -0.23263065 -0.14491447 -0.28729036
  0.43071238 -0.02063865 0.02734419 -0.25784749 0.31952031 0.32238703
 -0.38000359 -0.08111688 -0.4772311   0.25064268 0.17149721 0.37872197
 -0.01365768 -0.26231722 -0.01409472 0.00564897 -0.34278159 1.11965793]]
B: 2.93833462156e-17

Learning Rate: 0.08
W: [[8.61407400e-02 -5.95977606e-02   8.43780046e-03 -1.59950512e-01
  -1.01231427e-01 -1.53824636e-01   1.47139532e-01   8.54217737e-02
   1.23082696e-01 -1.99796003e-02 -7.62004443e-05 -7.90054466e-02
   3.01685412e-02 -9.58135169e-02   1.11147893e-01   3.11949623e-01
  -4.08937343e-01 -1.23152314e-01   6.05376973e-01 -4.46322112e-01
   2.63125100e-02 -2.76698608e-02 -4.55965885e-01   1.23146536e+00]]
B: 2.84052711948e-17

Learning Rate: 0.1
W: [[0.08634286 -0.08295425 0.00372578 -0.15731958 -0.09808438 0.05068971
  0.17472819 -0.14961226 -0.20063958 -0.1420684   0.44635844 0.16494375
 -0.21396872 -0.09580331 0.04640279 0.11101337 -0.12892441 0.14869856
  0.34228869 -0.44186306 0.07269725 -0.07355023 -0.45622494 1.22684266]]
B: 2.42977670629e-17



Plot of Test RMSE vs Learning Rate

Preferred Combination of Features : As is evident from the plot of different Learning rates vs Test RMSE, quadratic combination for function seems to work better compared to both Linear and Cubic combinations in general.

Part (d): Experimenting with cost functions

Model Parameters (for different values of Learning rates for different cost functions):

Mean Absolute Error Cost Function

Learning Rate: 0.001
W: [[0.04024118 0.38510575 -0.77155589 0.67098721]]
B: -0.0230899942163

Learning Rate: 0.002
W: [[0.03148942 0.27723138 -0.57435887 0.6333028]]
B: -0.0439315211105

Learning Rate: 0.005
W: [[0.02717743 0.07287505 -0.17757212 0.49783425]]
B: -0.10313880856

Learning Rate: 0.01
W: [[0.03730489 0.02757124 0.00849493 0.36672553]]
B: -0.147496818971

Learning Rate: 0.02
W: [[0.0466238 0.04832415 0.04713456 0.31322957]]
B: -0.155141700405

Learning Rate: 0.05
W: [[0.04796402 0.05251947 0.04997769 0.30625324]]
B: -0.156379410064

Learning Rate: 0.08
W: [[0.04794735 0.05258579 0.04998816 0.30617719]]
B: -0.156418739156

Learning Rate: 0.1
W: [[0.04798617 0.05257884 0.04999912 0.30616746]]
B: -0.156402544824

Mean Squared Error Cost Function

Learning Rate: 0.001
W: [[0.04206521 0.22102294 -0.44839919 0.68898716]]
B: 1.80049433367e-16

Learning Rate: 0.002
W: [[0.0371807 0.08706615 -0.20777818 0.63509884]]
B: 1.9376241502e-16

Learning Rate: 0.005
W: [[0.03851143 0.01145159 -0.00151656 0.53344649]]
B: 1.56500222459e-16

Learning Rate: 0.01
W: [[0.04209164 0.0235177 0.03939552 0.48975644]]
B: 1.38646243769e-16

Learning Rate: 0.02
W: [[0.04287045 0.0279435 0.04455977 0.48185237]]
B: 1.38278438247e-16 (Preferred Model)

Learning Rate: 0.05
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.39190760325e-16

Learning Rate: 0.08
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.39692126512e-16

Learning Rate: 0.1
W: [[0.04288975 0.02805758 0.04467961 0.48165961]]
B: 1.37423649993e-16

Mean cubed Error Cost Function (DOES NOT Converge since negative loss possible)

Learning Rate: 0.001
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.002
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.005
W: [[nan nan nan nan]]
B: nan

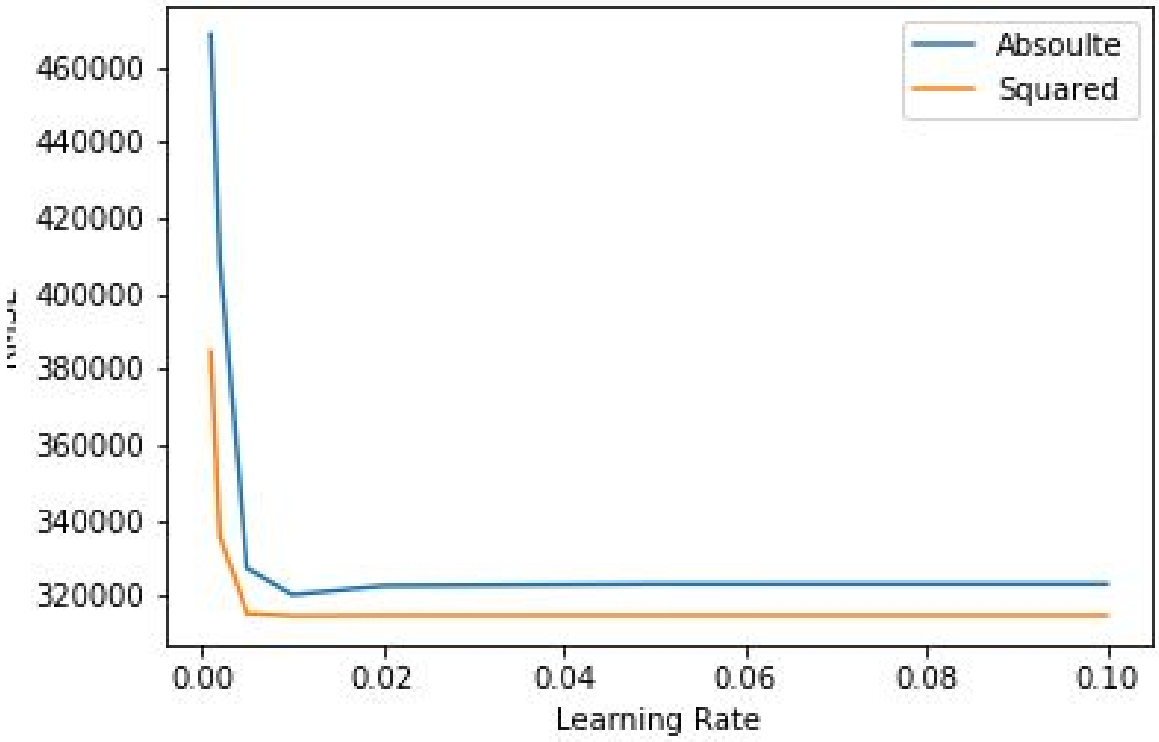
Learning Rate: 0.01
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.02
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.05
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.08
W: [[nan nan nan nan]]
B: nan

Learning Rate: 0.1
W: [[nan nan nan nan]]
B: nan



Plot of Test RMSE vs Learning Rate

Preferred Cost Function : First we see that it is not necessary for the Cubic Cost Function to converge. As, the Cubic Cost can also give Negative values, we find that the gradient values explode and thus we have not been able to plot the graphs for cubic cost function. Among Absolute and Squared Cost Function, we prefer Squared cost function as it gives lower test RMSE value as evident from the graph (test RMSE vs Learning rate).