Kevin Chuang
704769121

UCLA CS 145 Homework #1

DUE DATE: Friday 01/25/2019 11:59 pm

1.1 In LinearRegression\linearRegression.py, fill in the missing lines in the python code to estimate β, using (1) closed-form solution, (2) batch gradient descent, and (3) stochastic gradient descent.

a. Report the learned weights and MSE (Mean Square Error) in the test dataset for each version, are they the same and why?

Closed form solution –

Weights: [5.17285600e-01 -1.70173005e-02 -1.25229040e-02 -2.37364105e-02
-7.26850224e-03 -1.75015833e-03 -2.54105104e-02 -2.95826147e-02
-1.54205779e-02 -8.11041550e-03 -7.73115897e-03 2.23326880e-02
1.08377404e-02 -1.78034252e-02 -1.42799326e-02 1.10863872e-03
5.58962376e-03 -1.66481000e-02 1.74175345e-02 -9.79321083e-03
-1.13160627e-02 -2.43621012e-02 1.22042094e-02 -2.22008980e-02
-8.08868427e-03 1.98280275e-02 -1.62549170e-02 1.57163255e-02
5.55093555e-03 2.52723067e-02 -1.79696813e-02 -3.42412589e-02
2.33967228e-02 -1.18951150e-02 -8.29832518e-03 1.08683008e-03
1.07503176e-02 5.89595929e-03 -1.42884432e-02 -7.60366278e-03
-3.59068468e-03 -2.43039502e-02 -1.50352102e-02 -4.91648480e-05
-1.75975159e-02 -5.12186137e-03 -6.03505757e-03 2.11963730e-03
1.84672144e-02 5.97564034e-03 7.70482473e-03 -1.32971032e-02
-1.56211468e-02 1.64262479e-02 -1.87298040e-02 -2.62080745e-02
1.98841713e-02 -2.47382511e-02 7.11668306e-03 -2.56090472e-02
-1.43803106e-02 -1.78350545e-02 -2.34158378e-02 -1.21549137e-02
2.26194590e-02 -1.35242391e-02 8.88066425e-04 -1.42204055e-02
2.99114634e-03 5.22524532e-03 -1.79063948e-02 3.83684473e-03
8.33356729e-03 2.56888081e-02 -1.80756710e-02 -1.99695440e-02
-2.86138337e-02 2.35867028e-02 1.90433998e-03 1.72159943e-02
3.03296234e-02 1.74398815e-02 -2.78753941e-02 1.30140929e-02
2.60430914e-02 -2.59504768e-04 1.74699574e-02 3.43722771e-05
1.37552942e-02 2.24646356e-02 -1.22617221e-02 -1.82281224e-02
1.80041301e-02 -7.43819418e-04 -2.84486814e-02 -1.42173525e-02
-9.10220722e-04 -2.59410878e-02 1.86651575e-02 2.90379883e-02
-1.63292879e-03]

Training MSE: 0.08693886675396784

Test MSE: 0.11017540281675801

Batch Gradient Descent –

Weights: [5.14824311e-01 -1.70525890e-02 -1.25950406e-02 -2.37249829e-02
-7.25899807e-03 -1.72938959e-03 -2.54007919e-02 -2.95995935e-02
-1.54508292e-02 -8.13738578e-03 -7.72625767e-03 2.23082522e-02
1.08188951e-02 -1.78636181e-02 -1.43557111e-02 1.08614156e-03
5.57045518e-03 -1.65957303e-02 1.73903919e-02 -9.80427797e-03
-1.12904142e-02 -2.43827724e-02 1.21743702e-02 -2.22463584e-02
-8.06452850e-03 1.98282038e-02 -1.62416432e-02 1.57339496e-02

5.55277390e-03 2.53045906e-02 -1.79621727e-02 -3.42391694e-02
2.34379421e-02 -1.18792273e-02 -8.27672664e-03 1.10288282e-03
1.07466698e-02 5.96184413e-03 -1.43127672e-02 -7.52650594e-03
-3.59574268e-03 -2.43185239e-02 -1.50726889e-02 -5.20519856e-05
-1.75985270e-02 -5.13013077e-03 -5.99119563e-03 2.19442243e-03
1.84934062e-02 5.98090380e-03 7.66959322e-03 -1.33077812e-02
-1.56385906e-02 1.64114383e-02 -1.87359880e-02 -2.61982357e-02
1.99616009e-02 -2.47396951e-02 7.03622870e-03 -2.55437151e-02
-1.43848862e-02 -1.78581567e-02 -2.34180801e-02 -1.21370004e-02
2.26137932e-02 -1.34905209e-02 8.47435924e-04 -1.41645869e-02
2.99836525e-03 5.28598294e-03 -1.78261363e-02 3.82982119e-03
8.39417990e-03 2.57035705e-02 -1.80316056e-02 -1.99502271e-02
-2.86088900e-02 2.36280781e-02 1.86740201e-03 1.72356683e-02
3.03562494e-02 1.74206900e-02 -2.78640708e-02 1.29982873e-02
2.60797528e-02 -2.14424633e-04 1.74482921e-02 6.84566377e-05
1.37653783e-02 2.24865406e-02 -1.22189825e-02 -1.81585846e-02
1.79823849e-02 -7.31985766e-04 -2.84383243e-02 -1.41744713e-02
-8.74501021e-04 -2.59565361e-02 1.86650478e-02 2.90483835e-02
-1.65687129e-03]

Training MSE: 0.08694450998349146
Test MSE: 0.11010514764399844

Stochastic Gradient Descent –

Weights: [5.15341618e-01 -2.25689633e-02 -1.00734887e-02 -3.57915439e-02
-1.62619169e-02 -5.73525024e-03 -1.84913278e-02 -3.08619141e-02
-1.19475970e-02 -4.64640374e-03 4.34054298e-03 9.83209764e-03
1.55000381e-02 -8.54433524e-03 -1.20722985e-02 1.62956747e-02
1.62333107e-02 -1.51628805e-02 1.48911451e-02 -6.08237672e-03
-2.66945715e-03 -2.61133933e-02 2.61822788e-02 -2.35189663e-02
-1.09237190e-02 7.27720648e-03 -1.37220179e-02 1.73500078e-02
6.90445801e-03 3.06967751e-02 -1.56972054e-02 -3.74409649e-02
2.37871474e-02 -2.06817709e-02 -9.47683990e-03 1.08624142e-02
1.28985775e-02 6.70709729e-03 -1.25795291e-02 -4.27129107e-03
3.81630379e-03 -2.51788223e-02 -1.06221621e-02 -1.02344891e-03
-2.99018833e-02 -7.11194451e-03 -2.62044859e-03 -3.15121565e-03
-2.27387675e-03 1.53205208e-04 1.30116767e-02 -1.38623941e-02
-9.13819898e-03 1.27513627e-02 -2.33757040e-02 -2.00535031e-02
2.52557763e-02 -1.95119921e-02 -2.21262232e-03 -1.66892268e-02
-7.28843096e-03 -2.55652789e-02 -2.45580401e-02 -2.20555947e-02
2.41017778e-02 -1.15788754e-03 -7.02426362e-03 -1.81287395e-02
1.72317633e-02 1.18320293e-02 -2.69948629e-02 1.45318548e-03
1.04077002e-02 2.72154660e-02 -1.48128730e-02 -2.88881255e-02
-3.97705962e-02 2.35506196e-02 -6.56990165e-03 2.43811997e-02
1.79283400e-02 1.20192941e-02 -2.22563648e-02 1.23459558e-02
3.22326882e-02 4.52101322e-03 9.67573825e-03 -1.21147813e-02
2.23946419e-02 3.28215136e-02 -3.36202161e-03 -2.19222438e-02
1.04236685e-02 7.24550764e-03 -2.92965793e-02 -9.47358776e-03
-3.57403766e-03 -2.52868733e-02 2.75592583e-02 3.00569243e-02
4.49331073e-03]

Training MSE: 0.11550688999756466
Test MSE: 0.13698967667231002

These scores are all different from one another because they all have different forms of analyzing the data. The closed form solution in particular has a lower MSE because it is more accurate but is slower than its counterparts. The batch gradient computes the gradient using the whole dataset, which works best with smooth, convex datasets and also has a lower MSE, around the same as the closed form. Stochastic gradient descent computes the gradient using a single sample and updates its weight immediately, which is faster but less reliable and is therefore less accurate.

b. Apply z-score normalization for each feature x and report whether the normalization affect β and MSE (Mean Square Error) in the test dataset, for all three versions of the algorithm, and why?

Closed Form Solution:

Weights: [5.23000000e-01 -3.95099505e-02 -3.01401932e-02 -5.71438644e-02
-1.72769796e-02 -4.13700127e-03 -5.86318630e-02 -6.89027284e-02
-3.56331805e-02 -1.87845537e-02 -1.82888714e-02 5.29276130e-02
2.53519018e-02 -4.15812928e-02 -3.30193382e-02 2.65867992e-03
1.34068950e-02 -3.88013327e-02 4.11038867e-02 -2.32239983e-02
-2.68494719e-02 -5.67582270e-02 2.85948574e-02 -5.22058491e-02
-1.94232592e-02 4.61988692e-02 -3.87491283e-02 3.82055256e-02
1.27021593e-02 5.82271850e-02 -4.20937718e-02 -8.05582038e-02
5.50688227e-02 -2.88202457e-02 -1.94706479e-02 2.58596756e-03
2.55048685e-02 1.39991237e-02 -3.38312079e-02 -1.80218433e-02
-8.42135902e-03 -5.61252496e-02 -3.60939866e-02 -1.12787490e-04
-4.02969672e-02 -1.20851201e-02 -1.41809480e-02 5.11770552e-03
4.48842190e-02 1.42864924e-02 1.79066117e-02 -3.08841654e-02
-3.67139837e-02 3.83560781e-02 -4.47435146e-02 -6.08180754e-02
4.69774181e-02 -5.86346690e-02 1.62361334e-02 -6.06942237e-02
-3.38205570e-02 -4.24317897e-02 -5.46648364e-02 -2.89378305e-02
5.33687506e-02 -3.17462303e-02 2.12826319e-03 -3.26837546e-02
6.84819052e-03 1.25455103e-02 -4.09640271e-02 8.88512549e-03
1.94883628e-02 6.04797247e-02 -4.23185183e-02 -4.76582979e-02
-6.69833777e-02 5.66019062e-02 4.63178581e-03 4.13664903e-02
7.10828556e-02 4.08986579e-02 -6.46605942e-02 3.05062530e-02
6.11970818e-02 -6.13118531e-04 4.12093831e-02 8.04511196e-05
3.21203863e-02 5.30651849e-02 -2.83935172e-02 -4.22856651e-02
4.23271015e-02 -1.72635991e-03 -6.75124152e-02 -3.30151234e-02
-2.14687553e-03 -6.00152621e-02 4.30059659e-02 6.79904935e-02
-3.84367853e-03]

Training MSE: 0.08693886675396784
Test MSE: 0.11017540281675804

Batch Gradient Descent:

Weights: [0.5221988 -0.04344821 -0.020052 -0.03202882 -0.00942539 -0.00958477
-0.05100594 -0.07814625 -0.02468184 -0.00383811 -0.01841218 0.06314746
0.03502357 -0.06489762 -0.00701628 0.00201649 0.0039649 -0.0336909

0.05290236 -0.02800661 -0.02293835 -0.01890248 0.0522669 -0.03169305
-0.01248181 0.04603531 -0.03964667 0.03309901 -0.00147732 0.05528857
-0.00918427 -0.06623162 0.06339622 -0.00279317 -0.02173697 -0.01194116
0.04728667 0.05215912 -0.03675415 -0.00411022 -0.00494184 -0.06572891
-0.02916318 -0.00115719 -0.02504948 -0.00838832 -0.01424886 0.00927116
0.06165106 0.0344313 0.00340947 -0.03861151 -0.03429401 0.05444678
-0.03648966 -0.07758121 0.04931701 -0.05937431 0.00511714 -0.04982145
-0.02267689 -0.03685046 -0.03586382 -0.03760134 0.04470351 -0.02949978
-0.01750957 -0.02087519 -0.00208616 0.03169706 -0.02046628 0.02066906
0.01589688 0.06732093 -0.02925956 -0.04101369 -0.0367213 0.07458453
0.00557526 0.05902488 0.07860344 0.06258449 -0.04097813 0.0432094
0.06675249 0.00171656 0.05439298 0.02013676 0.06560942 0.07235056
-0.03002108 -0.04302025 0.05582054 0.01828226 -0.06563223 -0.00650908
0.02221043 -0.04258052 0.02979258 0.05446052 0.01130241]
Training MSE: 0.10025481775580267
Test MSE: 0.13297560346035608

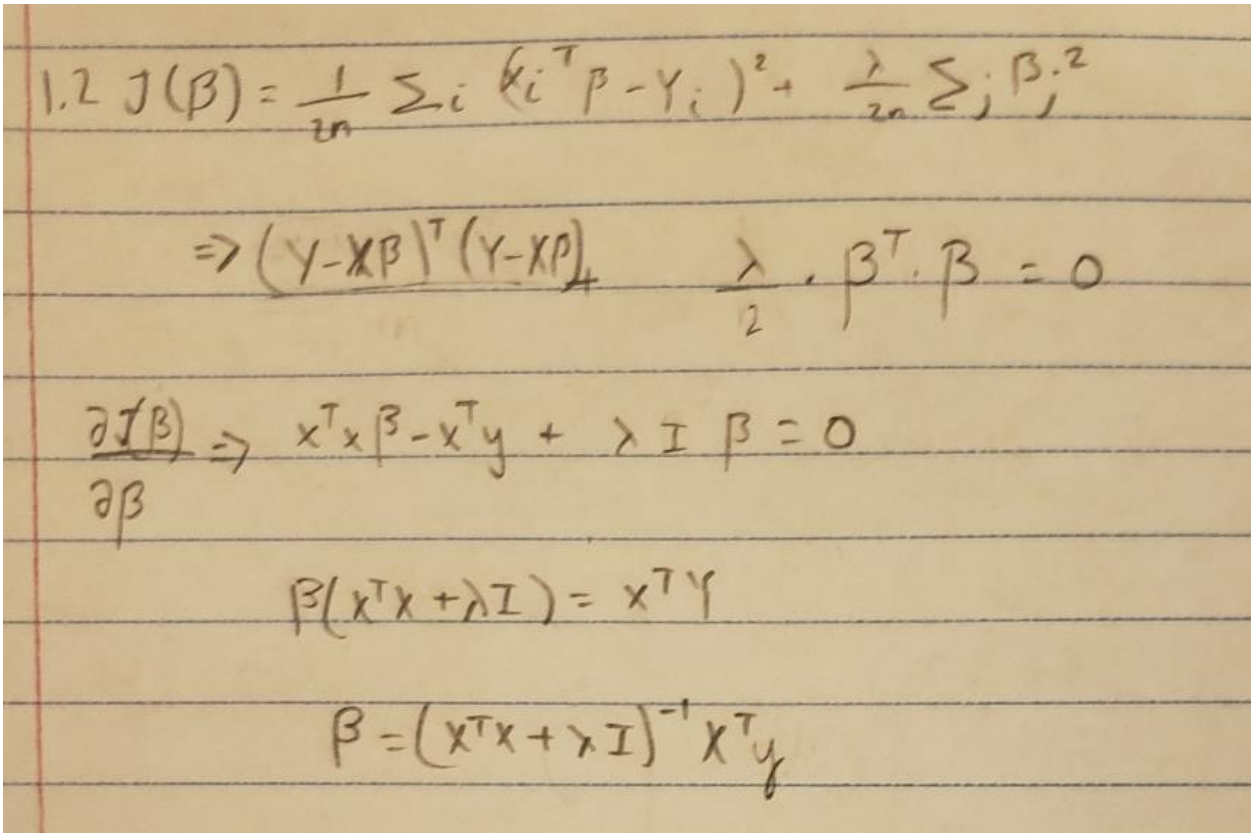
Stochastic Gradient Descent:

Weights: [5.25274244e-01 -4.06578136e-02 -2.84689746e-02 -5.08220058e-02
-1.65883238e-02 -2.92893122e-03 -5.84400808e-02 -6.97276620e-02
-3.22765956e-02 -1.93056337e-02 -1.72620044e-02 5.34340987e-02
2.22838217e-02 -3.93793796e-02 -3.17065213e-02 2.86847109e-03
1.49444184e-02 -3.61184014e-02 3.88164451e-02 -2.33314231e-02
-2.56412865e-02 -5.60699410e-02 2.77019953e-02 -4.99862092e-02
-2.53207443e-02 4.62021908e-02 -3.72478875e-02 3.64626175e-02
1.25535751e-02 5.72841086e-02 -4.47506127e-02 -7.72075412e-02
5.51120022e-02 -3.38245524e-02 -2.33041016e-02 1.68733568e-03
2.40472447e-02 1.54035272e-02 -3.61466588e-02 -1.68619394e-02
-7.81457088e-03 -5.77660444e-02 -3.56323207e-02 -1.31816093e-04
-3.99007862e-02 -1.07579932e-02 -1.81472255e-02 2.90771976e-03
4.23647838e-02 9.65222007e-03 1.62450874e-02 -3.40186324e-02
-3.63832101e-02 3.87905004e-02 -4.21926661e-02 -5.67719056e-02
4.54805418e-02 -5.46170991e-02 1.37838471e-02 -6.12274811e-02
-3.09185846e-02 -4.33756223e-02 -5.27416548e-02 -2.91111047e-02
5.44575516e-02 -3.34711708e-02 3.28501344e-03 -2.98781693e-02
7.56246100e-03 1.01107233e-02 -4.36098717e-02 8.49000792e-03
1.94856523e-02 6.30176516e-02 -4.27905397e-02 -5.00755642e-02
-6.48424365e-02 5.69142660e-02 2.53904459e-03 4.41079079e-02
7.08743880e-02 4.19670322e-02 -6.44864227e-02 3.21903161e-02
5.84584666e-02 -5.18410314e-03 4.25799287e-02 -2.73565782e-03
3.07576052e-02 4.78800718e-02 -2.75799810e-02 -4.19308432e-02
4.16516200e-02 6.39721038e-04 -6.61975444e-02 -3.31121624e-02
-3.33167673e-03 -5.90840923e-02 4.07295811e-02 6.77194054e-02
-4.92511898e-03]
Training MSE: 0.08749104508921064
Test MSE: 0.10980500097130835

Normalization is useful for stochastic gradient descent but is less useful for batch gradient descent. Closed form solution it is just as useful. This is probably because once you normalize the data in batch gradient, the clusters aren't as close together as they used to be and it takes more

iterations for it to converge. For stochastic gradient, the normalization probably helps get rid of the jumpiness between points and allows it to converge faster.

1.2 Ridge regression is to add an l2 regularization term to the original mean square error cost function in linear regression: $J(\boldsymbol{\beta}) = \frac{1}{2n}\sum_i \left(\boldsymbol{x}_i^T\boldsymbol{\beta} - y_i\right)^2 + \frac{\lambda}{2n}\sum_j \beta_j^2$, where $\lambda \geq 0$ is a trade-off between the two items. Please derive the closed-form solution for $\boldsymbol{\beta}$.



2. Logistic Regression and Model Selection

In LogisticRegression\LogisticRegression.py, fill in the missing lines in the python code to estimate β, using (1) batch gradient descent and (2) Newton Raphson method.

a. Report the learned weights and accuracy (Mean Square Error) in the test dataset for each version, are they the same and why? Discuss the pros and cons of the two methods.

Batch Gradient Descent (No normalization):
[-600.31754041 3577.3131062 14391.55580723 169.69136279
-43534.93137465 43.95106657]
Training avgLogL: -0.08522180497579555
Test accuracy: 0.45129224652087474

Batch Gradient Descent (normalization):
[0.27910793 2.15368204 1.23434665 1.32102078 -0.66170494 0.41580785]
Training avgLogL: -0.018246818624085535
Test accuracy: 0.7534791252485089

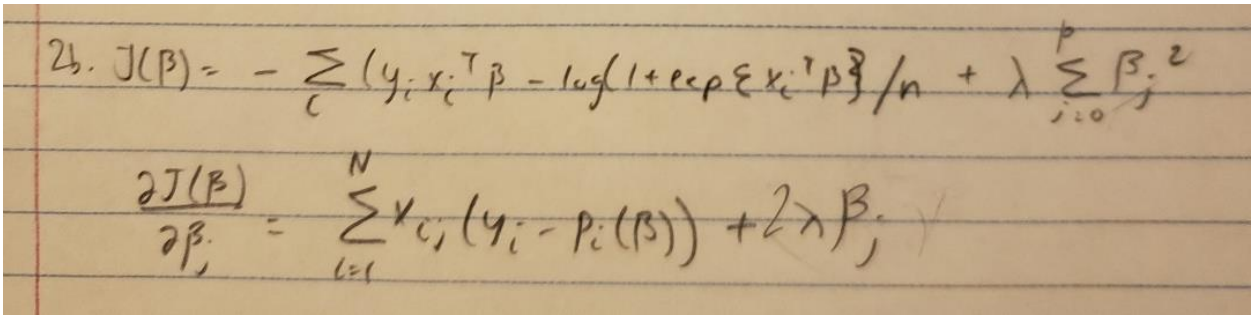
Newton Raphson Method (No normalization):
[4.56000080e+06 7.65300000e+09 2.17390002e+07 6.99008275e+05
2.42614001e+08 3.39835029e+04]
Training avgLogL: -0.1491471805599462
Test accuracy: 0.46322067594433397

Newton Raphson Method (Normalization):
[0.25385658 2.0480294 1.18403548 1.27476403 -0.64516048 0.41505823]
Training avgLogL: -0.020673603748248295
Test accuracy: 0.7475149105367793

b. Similar to linear regression, regularization can be added to logistic regression. Consider the new objective function as:

$$J(\beta) = -\sum_i (y_i \boldsymbol{x}_i^T \beta - log(1 + exp\{\boldsymbol{x}_i^T \beta\}))/n + \lambda \sum_{j=0}^{p} \beta_j^2$$

where n is the total number of data points in the training dataset and p is the dimensionality of attributes. Please compute its first derivative $\frac{\partial J(\beta)}{\partial \beta_j}$ and implement a regularized batch gradient descent algorithm accordingly. Discuss how the regularization is affecting the training loss (in terms of average log likelihood) and test accuracy based on the experimental results.



Regularized Batch Gradient (no normalization):
Weights: [-1.71173328e+01 4.00945598e+03 -6.37131023e+01 -2.07820492e+00
-9.67187560e+02 -5.98691811e-02]
Training avgLogL: -0.1491471805599462
Test accuracy: 0.46322067594433397

Regularized Batch Gradient (normalization):
Weights: [1.89424084e+237 1.03928689e+238 6.95483354e+237 6.07280460e+237
-2.24013568e+237 -3.43431855e+235]
Training avgLogL: 0.06738275633090975
Test accuracy: 0.7495029821073559

Based on these results, we can see that this regularization did not make a huge difference in

affecting the test accuracy but the training average log likelihood has increased. My hypothesis is that because the regularization is here to smooth out the curve even more, it takes longer to converge and is therefore a little less accurate than it without the regularization.