Michael Wang

CS 145 HW 1

* 1. a)

**Linear Regression No Normalization**

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Algorithm** | Closed Form | Batch | Stochastic |
| **Training MSE** | 0.08693886675396784 | 0.08697451544846697 | 0.09466830500554894 |
| **Test MSE** | 0.11017540281675801 | 0.11002433396036125 | 0.11562114312148082 |

**Linear Regression No Normalization**

|  |  |
| --- | --- |
| **Learning Algorithm** | **Beta** |
| Closed Form | [ 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] |
| Batch | [ 5.11099398e-01 -1.71061769e-02 -1.27043273e-02 -2.37078209e-02  -7.24456731e-03 -1.69796243e-03 -2.53862153e-02 -2.96251939e-02  -1.54966702e-02 -8.17823665e-03 -7.71887127e-03 2.22711356e-02  1.07905462e-02 -1.79543707e-02 -1.44701960e-02 1.05241534e-03  5.54125251e-03 -1.65166100e-02 1.73492188e-02 -9.82115629e-03  -1.12518543e-02 -2.44141811e-02 1.21291261e-02 -2.23149490e-02  -8.02801246e-03 1.98283266e-02 -1.62212278e-02 1.57604875e-02  5.55577137e-03 2.53532771e-02 -1.79511370e-02 -3.42356978e-02  2.35004039e-02 -1.18551160e-02 -8.24412794e-03 1.12716711e-03  1.07409037e-02 6.06156878e-03 -1.43496961e-02 -7.40972186e-03  -3.60354487e-03 -2.43406736e-02 -1.51292138e-02 -5.61071613e-05  -1.75999565e-02 -5.14268872e-03 -5.92468371e-03 2.30745792e-03  1.85331571e-02 5.98874796e-03 7.61640639e-03 -1.33236301e-02  -1.56650403e-02 1.63894083e-02 -1.87452098e-02 -2.61835529e-02  2.00787228e-02 -2.47418692e-02 6.91430727e-03 -2.54449578e-02  -1.43919710e-02 -1.78931189e-02 -2.34213791e-02 -1.21101406e-02  2.26052698e-02 -1.34394104e-02 7.86038759e-04 -1.40801877e-02  3.00948850e-03 5.37769229e-03 -1.77047769e-02 3.81907443e-03  8.48581785e-03 2.57261588e-02 -1.79649318e-02 -1.99209535e-02  -2.86013003e-02 2.36905631e-02 1.81156267e-03 1.72650620e-02  3.03965343e-02 1.73917701e-02 -2.78469324e-02 1.29744011e-02  2.61354721e-02 -1.46165738e-04 1.74153976e-02 1.20113568e-04  1.37805801e-02 2.25196256e-02 -1.21544862e-02 -1.80534998e-02  1.79496611e-02 -7.14207702e-04 -2.84225146e-02 -1.41093888e-02  -8.20588492e-04 -2.59797606e-02 1.86647924e-02 2.90636596e-02  -1.69292426e-03] |
| Stochastic | [ 5.16874124e-01 -1.73994474e-02 -1.81641313e-02 -2.12884456e-02  -4.75953266e-03 -3.09972939e-03 -2.72079605e-02 -2.95724615e-02  -1.60806499e-02 -1.38911806e-02 -9.95448373e-03 2.17437442e-02  1.20557065e-02 -1.72761424e-02 -7.43595484e-03 1.18032064e-03  3.98070919e-03 -1.40532433e-02 1.80294782e-02 -1.12348413e-02  -3.29107424e-03 -2.01524912e-02 1.86844023e-02 -2.34601531e-02  -6.09835139e-03 2.16269697e-02 -1.76585736e-02 1.71770728e-02  2.39389032e-03 1.97123498e-02 -1.90423295e-02 -3.74976289e-02  2.43422037e-02 -1.07153957e-02 -7.09281177e-03 -3.71436994e-04  9.04385176e-03 2.26488324e-03 -2.10737504e-02 -1.35848909e-02  -7.81006533e-04 -2.47154688e-02 -8.71011393e-03 6.34976908e-03  -1.67865442e-02 -8.39861241e-03 -4.65797361e-03 4.90426956e-03  1.82781663e-02 4.34757051e-03 1.04781697e-02 -1.96031513e-02  -1.74215942e-02 1.47678914e-02 -2.36641404e-02 -2.83718952e-02  1.92052872e-02 -2.57978304e-02 -9.78289612e-07 -2.52323994e-02  -1.08561504e-02 -1.33611611e-02 -2.17646145e-02 -1.45305569e-02  2.13220049e-02 -1.69036369e-02 1.04790709e-03 -1.69607146e-02  -1.89672090e-03 8.78102749e-03 -2.05656512e-02 -1.41630266e-03  2.76405975e-03 3.33139887e-02 -2.03961941e-02 -1.59166299e-02  -3.27595801e-02 2.17220806e-02 2.69167112e-03 2.58178142e-02  3.35668497e-02 1.65598802e-02 -2.48224069e-02 9.90751432e-03  2.65083124e-02 5.29689673e-03 1.38585982e-02 6.93010537e-03  1.35866687e-02 2.02396750e-02 -1.39326081e-02 -2.09075420e-02  2.23205236e-02 -3.72908825e-03 -2.28370435e-02 -1.56714096e-02  -6.18743447e-03 -2.83508756e-02 1.60517651e-02 2.90139403e-02  -2.76416718e-03] |

The closed form solution came out with the best training mean square error with the batch gradient descent solution a very close second and stochastic a bit further away but still within a few percent of the others. This was expected as the closed form solution finds the exact derivative while the batch method only approximates the derivative and the stochastic solution taking even more liberties with its approximation. However, having the best training MSE didn’t necessarily result in the best test MSE with the batch method narrowly beating out the closed form and stochastic once again being a few percent worse than the other two.

* 1. b)

**Linear Regression with Normalization**

|  |  |  |  |
| --- | --- | --- | --- |
| Learning Algorithm | Closed Form | Batch | Stochastic |
| Training MSE | 0.08693886675396784 | 0.0946805372779896 | 0.08698290332528344 |
| Test MSE | 0.11017540281675801 | 0.12472266450738567 | 0.10984667261428331 |

**Linear Regression with Normalization**

|  |  |
| --- | --- |
| **Learning Algorithm** | **Beta** |
| Closed Form | [ 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] |
| Batch | [ 0.52465893 -0.03311216 -0.03351362 -0.04215891 -0.01236286 -0.00634677  -0.0547508 -0.07947043 -0.02851594 -0.00274155 -0.01099968 0.05000511  0.02557767 -0.04496167 -0.01896677 0.00709249 0.00732908 -0.0307751  0.05965278 -0.02259341 -0.01207825 -0.04200014 0.04235027 -0.03129537  -0.01141634 0.04803077 -0.03329509 0.03405127 0.00796672 0.05796462  -0.01669714 -0.0697992 0.05112034 -0.00545846 -0.0295127 -0.00919895  0.04227957 0.04288891 -0.045985 -0.00620281 -0.00745158 -0.05301124  -0.03103887 0.00600219 -0.02876887 -0.01122952 -0.01402166 0.00813152  0.05598156 0.02035506 0.0177971 -0.02680108 -0.03995053 0.05376694  -0.03764486 -0.06814107 0.04140887 -0.05117473 0.0183193 -0.06014669  -0.02886733 -0.05445895 -0.04954092 -0.04016318 0.06827586 -0.02518401  -0.00086127 -0.01705381 0.01106406 0.0424657 -0.02961389 0.01097003  0.01934225 0.07739096 -0.0361859 -0.04098563 -0.05651313 0.07161483  0.0022013 0.04150565 0.09119142 0.05773434 -0.05031867 0.04061636  0.0612851 -0.00129052 0.04762103 0.00839183 0.0511315 0.06678161  -0.02725987 -0.03396703 0.06538834 0.0089678 -0.07106906 -0.01384427  0.0113445 -0.0486249 0.03051124 0.06167238 0.01101253] |
| Stochastic | [ 5.22594704e-01 -3.96686622e-02 -2.99009940e-02 -5.75595053e-02  -1.68649151e-02 -4.96315133e-03 -5.93980912e-02 -6.94347430e-02  -3.68221331e-02 -1.93207529e-02 -1.84293022e-02 5.17990965e-02  2.54087519e-02 -4.17530083e-02 -3.24145924e-02 2.83636233e-03  1.33159689e-02 -3.77196303e-02 4.14264666e-02 -2.25839830e-02  -2.68891656e-02 -5.70336997e-02 2.81857791e-02 -5.19507994e-02  -2.02529155e-02 4.66265576e-02 -3.85395375e-02 3.80187731e-02  1.28583606e-02 5.75060407e-02 -4.26776850e-02 -8.00973883e-02  5.51847972e-02 -2.81813004e-02 -1.90457812e-02 2.06515350e-03  2.58568011e-02 1.38833096e-02 -3.39311568e-02 -1.78816435e-02  -8.50900142e-03 -5.47610136e-02 -3.67978502e-02 -7.65256774e-04  -4.02355957e-02 -1.14052867e-02 -1.38396466e-02 5.21756473e-03  4.48346977e-02 1.47632002e-02 1.77388863e-02 -3.21883233e-02  -3.60739470e-02 3.80445627e-02 -4.47370731e-02 -6.09036421e-02  4.67066016e-02 -5.94848447e-02 1.72709923e-02 -6.00348263e-02  -3.28912967e-02 -4.19522342e-02 -5.47550103e-02 -2.77101833e-02  5.39233535e-02 -3.19914168e-02 2.34288521e-03 -3.19437139e-02  6.08244296e-03 1.23616857e-02 -4.09046364e-02 8.48475315e-03  1.91782383e-02 5.96772895e-02 -4.24862158e-02 -4.72081852e-02  -6.67259238e-02 5.54429767e-02 4.97632729e-03 4.04904482e-02  7.00631462e-02 3.98952373e-02 -6.48474981e-02 3.06243083e-02  6.01592922e-02 1.48311554e-05 4.19223017e-02 -3.69332715e-04  3.33262952e-02 5.22092938e-02 -2.74648978e-02 -4.21294575e-02  4.27882149e-02 -2.73639866e-03 -6.64246823e-02 -3.35845346e-02  -1.89064395e-03 -5.86456191e-02 4.28181004e-02 6.72560371e-02  -2.20730173e-03] |

Applying normalization caused Beta for all three algorithms to increase, with each entry other than B0 being increased by around a factor of two. This is likely because the algorithms with and without normalization are more or less reaching the same conclusion and normalization is merely causing the training data to be reduced by a factor of 2.

1.2)

2 a-b)

**Logistic Regression No Normalization**

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Algorithm** | Batch | Newton | Regularized |
| **Average log(L) 0** | -8445340.615643788 | -inf | -6146889.341514997 |
| **Average log(L) 1000** | -inf | -inf | -inf |
| **Average log(L) 2000** | -inf | -inf | -inf |
| **Average log(L) 3000** | -inf | -inf | -inf |
| **Average log(L) 4000** | -inf | -inf | -inf |
| **Average log(L) 5000** | -inf | -inf | -inf |
| **Average log(L) 6000** | -inf | -inf | -inf |
| **Average log(L) 7000** | -inf | -inf | -inf |
| **Average log(L) 8000** | -inf | -inf | -inf |
| **Average log(L) 9000** | -inf | -inf | -inf |
| **Beta** | [ -990.99118965 11132.77426625 24028.71583476 285.24526439  -72321.40766553 73.93195635] | [4.56000000e+56 7.65300000e+59 2.17390000e+57 6.99008000e+55  2.42614000e+58 3.39826408e+54] | [-5.73703301e+108 5.76451342e+108 -1.90593084e+109 -6.88450664e+107  -3.23313908e+110 -1.46729743e+106] |
| **Train log(L)** | -inf | -inf | -inf |
| **Test Acc** | 0.46520874751491054 | 0.46322067594433397 | 0.5944333996023857 |

**Logistic Regression with Normalization**

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Algorithm** | Batch | Newton | Regularized |
| **Average log(L) 0** | -0.5439602698795735 | -0.4732719507970200 | -0.519254439285961 |
| **Average log(L) 1000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 2000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 3000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 4000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 5000** | -0.4601003753508533 | -0.4601003753508531 | -0.4613772298101670 |
| **Average log(L) 6000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 7000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 8000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Average log(L) 9000** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Beta** | [ 0.27910793 2.15368204 1.23434665 1.32102078 -0.66170494 0.41580785] | [ 0.27910793 2.15368204 1.23434665 1.32102078 -0.66170494 0.41580785] | [ 0.3200224 2.3915878 1.50903132 1.47971911 -0.71934181 0.31093568] |
| **Train log(L)** | -0.4601003753508533 | -0.4601003753508532 | -0.4613772298101670 |
| **Test Acc** | 0.7534791252485089 | 0.7534791252485089 | 0.7594433399602386 |

Batch gradient descent and Newton Raphson produced extremely similar results – without normalization both methods had equally abysmal log likelihoods and test accuracies , and with normalization both had essentially equal Betas, log-likelihoods, and test accuracies of about 75%. The only noticeable difference in the output is that without regularization Newton Raphson’s Beta was much larger than batch gradient descent’s. A possible cause for this may be the fact that when I attempted to take the inverse of the Hessian I was alerted that Python thought the Hessian was zero, so I instead took the inverse of the identity matrix multiplied by 1e-50 which resulted in a very large number On that note, one of the downsides of the Newton-Raphson algorithm is that sometimes it isn’t possible to take the inverse of a matrix. Additionally, I noticed that my computer took significantly longer to run newton-Raphson than it did batch gradient descent although that may be due in part to my implementation. The theoretical benefit of Newton Raphson is that you don’t need to guess a learning rate but that wasn’t an issue in this assignment since the learning rate was given to us and both algorithms arrived at the same answer anyway. It’s possible that Newton Raphson on average approaches the maximum likelihood faster than batch but confirming that would have required me to run the algorithm multiple times under different learning rates. And since each run takes my computer over 10 minutes, doing so would have been prohibitively time consuming.

The derivative of the of the regularized objective function is given by:

In my implementation I used λ = 2 although the results seem to be similar up to λ = 10 (much higher and I encounter overflow errors, much lower and it reduces to non-regularized gradient descent). The resulting test accuracy without normalization was noticeably higher than the other two algorithms without normalization. I suspect this is due to the non-normalized algorithms severely over-fitting the data so having a simplified decision function pushes the accuracy above randomly guessing. With normalization, the training log likelihood is slightly lower than the other two (this may have been the case without normalization, but as they all gave negative infinity that observation could not be made). This was expected as the regularization intentionally deviates from the data to simplify beta. As previously seen, a lower likelihood doesn’t necessarily translate into a lower accuracy as the regularized test accuracy is just slightly better than the other two, again likely due to the other algorithms overfitting the data.