CIS 419/519: Homework 3

{Yongxin Guo}

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: Jiangzhu Heng, Yihang Xu

1 Logistic Regression

1.1 Implementation

The codes are successfully implemented and pass all the autograder tests

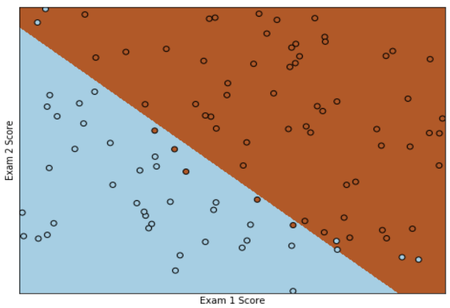
1.2 Test Implementation

The codes are successfully implemented and pass all the autograder tests

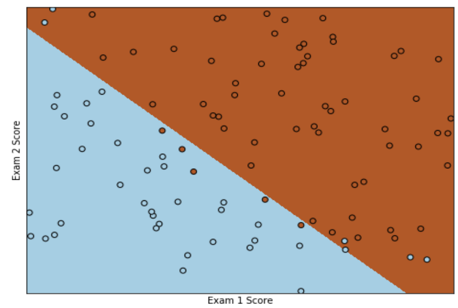
1.3 Analysis

The parameter setup is: alpha = 0.01, epsilon=0.0001 with lambda varying among 1e-08, 1, 3, and 15.

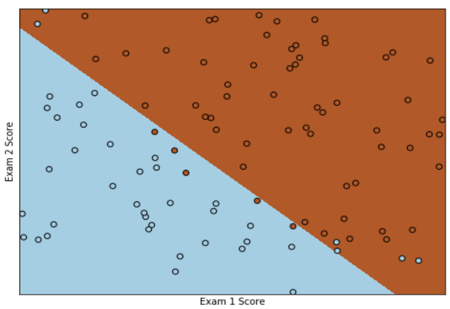
Summary: As shown in the following figures, lambda serves as the penalty for the over-fitting systems, the system with a higher lambda may perform well in the real test data, however, if the lambda is too large then the cost it took for the systems will become very large as seen in the figures when the lambda is tuned up the linear decision boundary shifts toward the left-corner. Under the same lambda, L1 norm may have a larger impact to the systems than L2 norm does. As can be observed when lambda = 15, L1 creates more penalty to the theta values and drive it to a slower number and result in a more deviated decision boundary as compared to one under L2 norm.



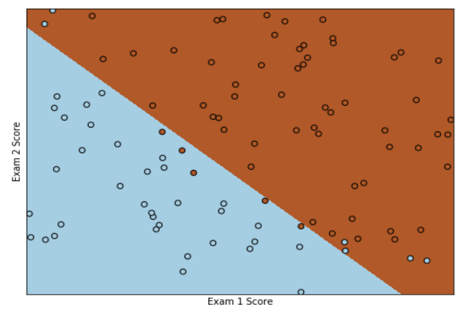
(a) Lambda = 1e-08 under L1



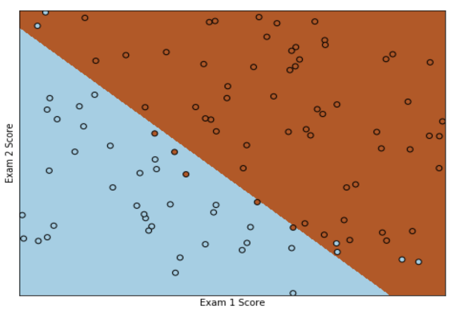
(b) Lambda = 1e-08 under L2



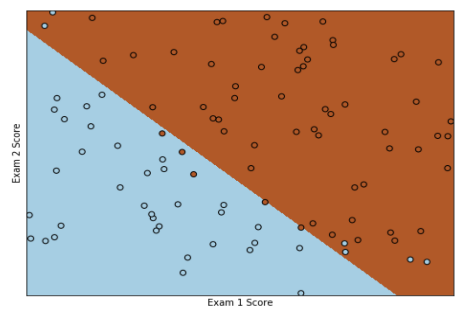
(c) Lambda = 1 under L1



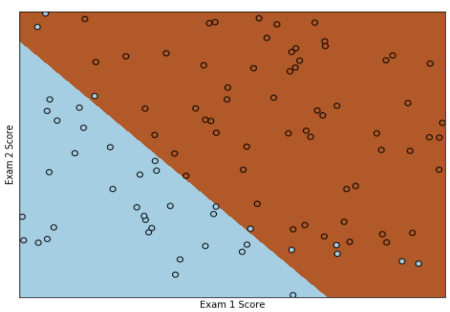
(d) Lambda = 1 under L2



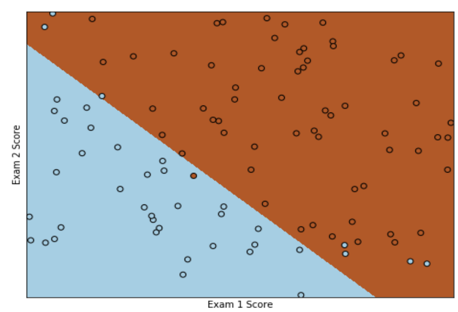
(e) Lambda = 3 under L1



(f) Lambda = 3 under L2



(g) Lambda = 15 under L1

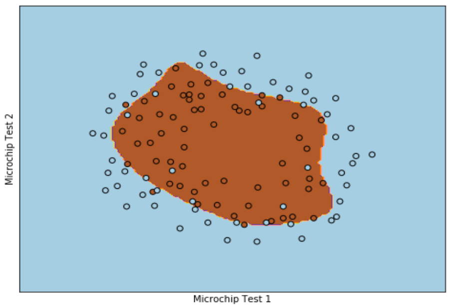


(h) Lambda = 15 under L2

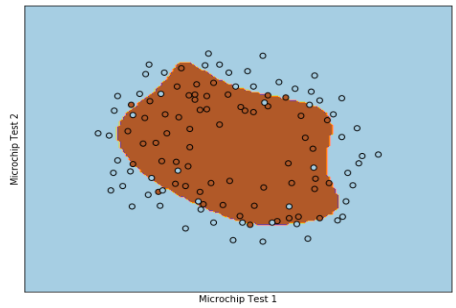
Figure 1: Logistic Regression with various L1 and L2 regularization parameters

1.4 Learning Nonlinear Decision Surface

The parameter setup is: alpha = 0.01, Lambda = 0.0001, epsilon=0.001



(a) Nonlinear Decision Surface with Lambda = 1e-04 under L1



(b) Nonlinear Decision Surface with Lambda = 1e-04 under L2

Figure 2: Logistic Regression for Nonlinear Decision Surface

2 Comparing Algorithms

2.1 Logistic Regression Adagrad

The adagrad has been successfully implemented the similar plots for the nonlinear decision surface as shown above. (all Autograder test passed)

2.2 Comparing Algorithms

For Breast Cancer Wisconsin data: The experiment setup is set to alpha = 0.05, epsilon = 0.006, Xi = 1e-05 (only applied fro adagrad) The following comparison is made between normal logistic gradient descent and logistic adagrad descent for both L1 and L2 regularization with varying lambda parameters among 1e-08, 1e-04 and 1e-02. The performance is evaluated via cross validation of 5 folds and 3 repeats with each repeat followed the shuffling. The performance matrix used here consist of average accuracy across all the trials, so-called cvScores, and the average time elapsed for all the trials, so-called timeScores.
The following tables show the comparison for L1 regularization
L1 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	1.8585 s	0.8803 s
cvScore	0.9625	0.9636

Table 1: lambda = 1e-08 under L1

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	1.7977 s	0.9018 s
cvScore	0.9636	0.9666

Table 2: lambda = 1e-04 under L1

lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	1.5061 s	0.7962 s
cvScore	0.9631	0.9631

Table 3: lambda = 1e-02 under L1

L2 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	1.3582 s	0.5088 s
cvScore	0.9631	0.9607

Table 4: lambda = 1e-08 under L2

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	1.3491 s	0.5623 s
cvScore	0.9625	0.9678

Table 5: lambda = 1e-04 under L2

lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	NaN (not converged)	0.4950 s
cvScore	NaN (not converged)	0.9689

Table 6: lambda = 1e-02 under L2

For Retinopathy data: The experiment setup is set to alpha = 0.001, epsilon = 0.0006, Xi = 1e-05 (only applied fro adagrad) The following comparison is made between normal logistic gradient descent and logistic adagrad descent for both L1 and L2 regularization with varying lambda parameters among 1e-08, 1e-04 and 1e-02. The performance is evaluated via cross validation of 5 folds and 3 repeats with each repeat followed the shuffling. The performance matrix used here consist of average accuracy across all the trials, so-called cvScores, and the average time elapsed for all the trials, so-called timeScores.
The following tables show the comparison for L1 regularization
L1 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	9.5059 s	0.9123 s
cvScore	0.7402	0.5326

Table 7: lambda = 1e-08 under L1

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	9.2885 s	0.8431 s
cvScore	0.7413	0.4853

Table 8: lambda = 1e-04 under L1

lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	9.3279 s	0.8668 s
cvScore	0.7416	0.4987

Table 9: lambda = 1e-02 under L1

L2 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	7.6981 s	0.5697 s
cvScore	0.7405	0.5044

Table 10: lambda = 1e-08 under L2

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	7.7918 s	0.5663 s
cvScore	0.7411	0.5077

Table 11: lambda = 1e-04 under L2

lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	7.0688 s	0.5992 s
cvScore	0.7387	0.5331

Table 12: lambda = 1e-02 under L2

For diabetes data: The experiment setup is set to alpha = 0.001, epsilon = 0.0001, Xi = 1e-05 (only applied fro adagrad) The following comparison is made between normal logistic gradient descent and logistic adagrad descent for both L1 and L2 regularization with varying lambda parameters among 1e-08, 1e-04 and

1e-02. The performance is evaluated via cross validation of 5 folds and 3 repeats with each repeat followed the shuffling. The performance matrix used here consist of average accuracy across all the trials, so-called cvScores, and the average time elapsed for all the trials, so-called timeScores.
The following tables show the comparison for L1 regularization

L1 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	0.0825 s	0.3174 s
cvScore	0.7708	0.5551

Table 13: lambda = 1e-08 under L1

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	0.0819 s	0.4115 s
cvScore	0.7708	0.4796

Table 14: lambda = 1e-04 under L1

lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	0.0786 s	0.3606 s
cvScore	0.7708	0.5008

Table 15: lambda = 1e-02 under L1

L2 Regularization:

lambda = 1e-08:

	Logistic grad	Logistic Adagrad
timeScore	0.0732 s	0.2965 s
cvScore	0.7708	0.4908

Table 16: lambda = 1e-08 under L2

lambda = 1e-04:

	Logistic grad	Logistic Adagrad
timeScore	0.0739 s	0.2675 s
cvScore	0.7708	0.4831

Table 17: lambda = 1e-04 under L2

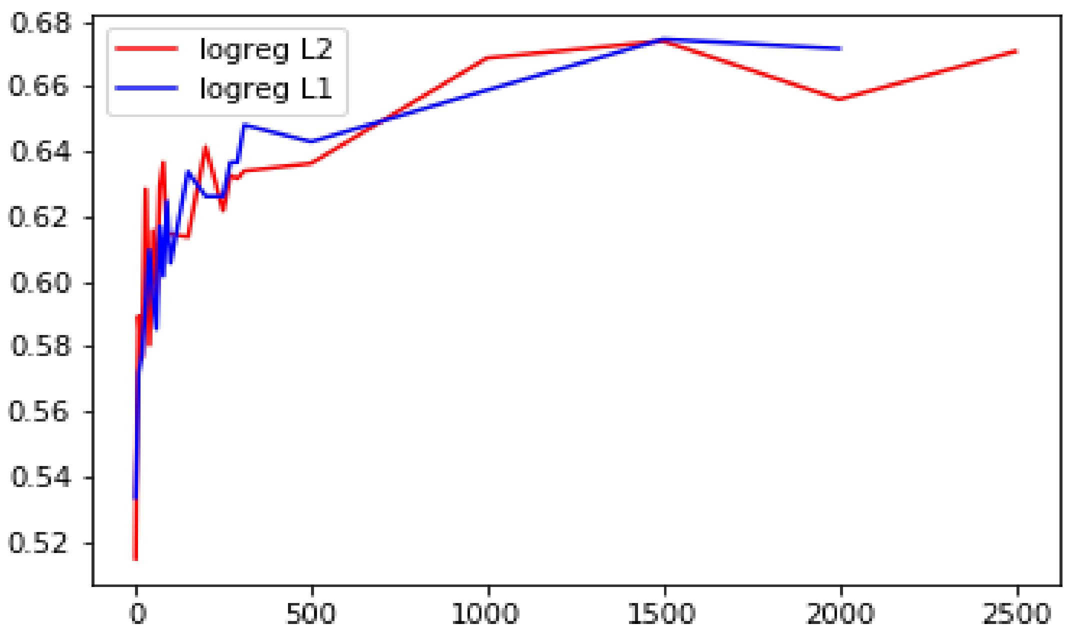
lambda = 1e-02:

	Logistic grad	Logistic Adagrad
timeScore	0.0751 s	0.2461 s
cvScore	0.7708	0.4913

Table 18: lambda = 1e-02 under L2

Summary: It can be clearly seen from the performance matrix tabulated above that adagrad has a much faster converge rate than the normal logistic regression. As the lambda tuned up, the accuracy may drop slightly. For example, in the first dataset, when the lambda = 1e-02, the normal logistic regression won't even converge as it did for the lower lambda, but the adagrad can still converge can give a pretty good prediction within a very small amount of time. It is also found that, when the epsilon, the convergence criteria, become more strict(smaller), the normal logistic regression will have trouble or spend lots of time converging toward it or just not converge. However, the adagrad can still converge with a more strict criteria within a small period of time. Same phenomenon also observed when tuning up the alpha, the step size, for both algorithms. The normal logistic regression are prone to diverge when facing a larger step size, however, adagrad is very insensitive to the large step size since its step size is always adaptive changing in runtime.

2.3 Understanding Regularization and Adagrad



(a) Learning Curve

Learning Curve

The above learning curve shows the progression of logreg under L2 and L1. It can be seen from the figure that under L1 and L2 the learning curve for the logreg is pretty similar. L2 kinda precedes L1 but it eventually starts fluctuating.