COMP-551 Mini Project 1

Kaicheng Wu
kaicheng.wu@mail.mcgill.ca
Mathieu-Joseph Magri
mathieu-joseph.magri@mail.mcgill.ca
Mohammad Sami Nur Islam
mohammad.sami.islam@mail.mcgill.ca

School of Computer Science
McGill University
Canada
February 9th 2023

1 Abstract

In this assignment, our main goal was to familiarize ourselves with the two most commonly used machine learning models, Linear Regression and Logistic Regression, and learn how these two models work. We investigated the performance of Linear Regression on an energy efficiency dataset and the performance of Logistic Regression on a qualitative bankruptcy dataset. Both of these datasets were procured from the UCI data repository. Various experiments were performed on these models, including varying training data sizes, batch sizes, learning rates, regularization parameters, and momentum. The changes in convergence time and test error of the model were noted for these experiments and presented in plots. One of the most important observations we made during the experiments was how varying the parameters influenced our models in many different ways whether that be in terms of convergence speed and performance.

2 Introduction

In summary, there were a total of three project tasks that needed to be run on the two data sets.

The first task for both datasets was to acquire, preprocess, and analyze the data. This subtask included dealing with any missing or malformed features and data. Finally, the last part of this task was to simply compute basic statistics on both datasets.

For the second task, implementing the machine learning models needed to be done. More specifically, there was a need to implement analytical linear regression for the first dataset as well as implementing logistic regression with gradient descent for the second dataset. It was also necessary to implement mini-batch stochastic gradient descent for both models. To implement these models, the recommended approach given in the assignment handout was followed, hence fit and predict functions were created to aid with the goal of implementing the models.

For the third task, many experiments were run on each of the datasets with the models. To run these experiments, we split each dataset into a training set and a test set. The test set was used to determine the performance of our models after training each model with their training sets. The performance of each model was evaluated using the corresponding cost functions for the classification and regression tasks. The six requested experiments as well as a few extra experiments for both datasets, were also done.

Moreover, the first dataset was an energy efficiency dataset characterizing different buildings. The source of this dataset is Angeliki Xifara, and it was processed by Athanasios Tsanas for the Oxford Centre for Industrial and Applied Mathematics. Two papers have used this dataset: *Accurate quantitative estimation of the energy performance of residential buildings using statistical machine learning tools* by A. Tsanas, A. Xifara and *Accurate telemonitoring of Parkinson's disease symptom severity using nonlinear speech signal processing and statistical machine learning* by A. Tsanas. For the second dataset, the data was used to predict bankruptcies. The sources for this dataset are A.Martin, J.Uthayakumar, and M.Nadarajan from the Sri Manakula Vinayagar Engineering College and Pondicherry University. *The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms* by Myoung-Jong Kim and Ingoo Han has used this dataset.

All in all, as a group, we determined that the most important findings for this project were the fact that varying batch sizes in mini-batch stochastic gradient descent had a considerable impact on convergence time during linear regression. The same could not be concluded for logistic regression. Increasing the training size increased the performance of both models, but only up to a certain point, indicating a chance of the models overfitting with larger datasets. Regularization also reduced the performance of both models indicating that it's not required since it could be making the models too simple.

3 Datasets

The first dataset (Energy Efficiency) is multivariate with 768 instances, 8 attributes, and 2 responses. The 8 attributes (Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, and Glazing Area Distribution) are used to predict the 2 targets (Heating Load, Cooling Load). The second dataset (Qualitative Bankruptcy) is another multivariate dataset with a total of 250 instances and 7 attributes (Industrial Risk, Management Risk, Financial Flexibility, Credibility, Competitiveness, Operating Risk, and Class). The Class attribute has two distinct categories (B-Bankruptcy, NB-Non-Bankruptcy), while all other attributes have 3 (P-Positive, A-Average, N-negative).

Cleaning the data was rather straightforward. Indeed, both datasets were not missing any values and had no outliers. Therefore, we kept all the records. It is important to note that the second dataset is qualitative. Therefore one-hot encoding was performed for the model to work with the different classes of each feature. This was performed in place of nominal encoding to prevent some parameters from being given more weight than others, which could lead to overfitting or incorrect results. For dataset 1, basic statistics were calculated, such as the count for each value, mean of all values, standard deviation for all values, the minimum value, the maximum value, and 25th, 50th, and 75th percentile values of the dataset. Additionally, a correlations heat map and histograms and box plots for each feature were implemented. For the second dataset, we calculated the count for all features, the number of unique values used to describe each feature, the most frequent value for each feature, and that frequency count. Histograms for each feature were also implemented.

Figure 1: Correlation Heat Map for Dataset 1

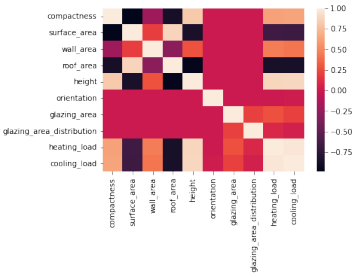
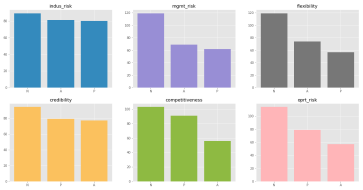


Figure 2: Classification of 6 attributes in Dataset 2



As can be seen from the heatmap in Figure 1, the target values for the linear regression model are not always perfectly correlated with the different features. This could contribute towards the cost function being non-convex.

Figure 3: Classification for the Class attribute of Dataset 2

In terms of ethical concerns with the datasets, there seems to be none.

4 Results

4.1 Linear Regression

When using an 80/20 train-test split, linear regression has an error rate of 9.565 for the training set, while it has an error rate of 11.021 for the test set.

The heights feature has the largest weight indicating that it has the most effect on the output from the model, while the glazing_area_distribution and orientation have the smallest weights.

Figure 4: Feature-Weight pairs for the Heating Load

	feature	weights
0	compactness	6.267066
1	surface_area	-7.475111
2	wall_area	8.820811
3	roof_area	12.220429
4	height	7.216421
5	orientation	-0.036268
6	glazing_area	2.681740
7	glazing_area_distribution	0.327199
8	bias	22.260036

Figure 5: Feature-Weight pairs for the Cooling Load

	feature	weights
0	compactness	-3.989484
1	surface_area	-1.696152
2	wall_area	1.138766
3	roof_area	-2.203959
4	height	8.265669
5	orientation	-0.015688
6	glazing_area	2.703635
7	glazing_area_distribution	0.377926
8	bias	22.175369

As the size of the training data increases, the model has more data about the relationship between the features and the target parameters, which leads to better estimation of the parameter weights and, ultimately, better generalization. However, after around 60-70 percent of the training size, the test error seems to go up again, indicating overfitting.

Figure 6: Performance of Training and Test Set for Heating Load

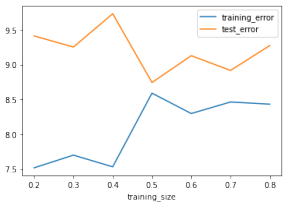
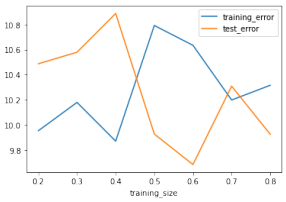


Figure 7: Performance of Training and Test Set for Cooling Load



Our model converges faster with smaller batch sizes, and the fastest convergence seems to be for a batch size of 8. The batch gradient descent with a batch size of 768 performs the worst. The number of epochs has been truncated to make this more evident in the graph. We are also taking a log of the cost to make the changes more apparent.

Figure 8: Varying Batch Sizes for Heating Load

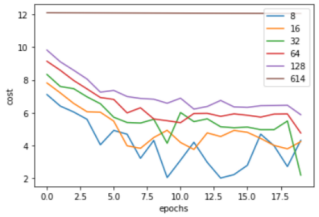
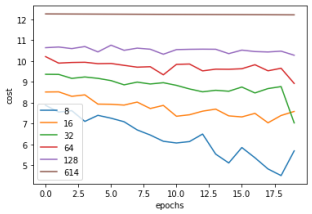


Figure 9: Varying Batch Sizes for Cooling Load



As we increase the learning rate, the mean squared error decreases in both cooling and heating load predictions. In fact, the graphs have the same shape in both cases. This indicates that the algorithm is progressing rapidly toward finding the optimal solution. This is because the learning rate determines how much the weights are updated in each iteration. A higher learning rate ensures that convergence to the optimal solution happens more quickly.

Figure 10: Varying Learning Rates for Heating Load

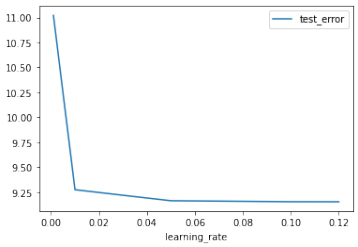
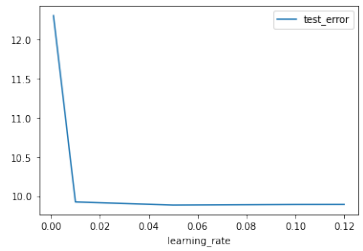


Figure 11: Varying Learning Rates for Cooling Load



4.2 Logistic Regression

When using an 80/20 train-test split, fully batched logistic regression performs at an accuracy of 0.705 for the training set and at an accuracy of 0.720 for the test set when using the same 80/20 split.

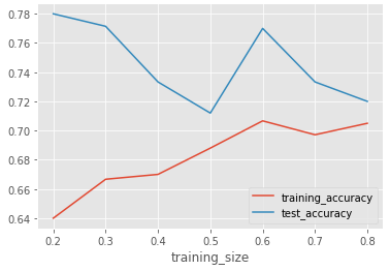
For feature weights, as we can see from the graph below, credibility_P and flexibility_P seem to have the most influence on determining the NB or B status of an instance. All other feature-weights pairs can be viewed below.

	feature	weights
0	indus_risk_A	0.006129
1	indus_risk_P	0.042567
2	indus_risk_N	0.069922
3	mgmt_risk_A	0.025085
4	mgmt_risk_P	0.058662
5	mgmt_risk_N	0.034871
6	flexibility_A	-0.004038
7	flexibility_P	0.125524
8	flexibility_N	-0.002867
9	credibility_A	0.007507
10	credibility_P	0.132223
11	credibility_N	-0.021111
12	competitiveness_A	-0.172329
13	competitiveness_B	0.538926
14	competitiveness_N	-0.247978
15	oprt_risk_A	0.085087
16	oprt_risk_B	0.021657
17	oprt_risk_N	0.011875
18	bias	0.118619

Figure 12: Feature-Weight pairs for Dataset 2

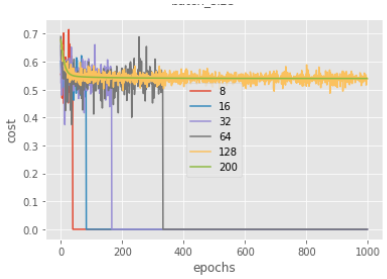
The accuracy of our logistic regression model fluctuates a lot as the training size grows. This behavior may have been caused by overfitting, similar to what we encountered with our linear regression model.

Figure 13: Performance of Training and Test Set for Dataset 2



In the case of logistic regression, the fluctuations in the cost vs. epoch graphs (Figure 14) could be attributed to the cost function being non-convex, meaning that the gradient descent algorithm jumps from local minima to local minima and cannot find the global minima. Due to these fluctuations, it is almost impossible to determine what batch size would be the best for the models.

Figure 14: Varying Batch Sizes for Dataset 2



As the learning rate increases, the accuracy of our model also increases; however, it sharply falls off. This could be due to the fact that the weights of features may be updated in such a manner that the gradient descent algorithm is stuck in local minima and isn't able to climb out of them. The accuracy begins to climb again however, indicating the algorithm is able to find more optimal solutions with larger learning rates.

Figure 15: Varying Learning Rates for Dataset 2

All in all, for the logistic regression model, when training size increases, the training accuracy increases, while the test accuracy has fluctuating behavior. When we modify batch sizes, it is almost impossible to determine the best possible batch size due to the fluctuating results as well. When the learning rate increases, the accuracy first goes through an increase and soon enters a plateau to then increase once again.

4.3 Analytical Linear Regression VS. Mini-Batch Stochastic Gradient Descent based linear regression

The mean squared error in analytical linear regression was higher than the one in mini-batch stochastic gradient descent. In the initial run of mini-batch gradient descent with default hyper-parameters, the error was 10.57, and 12.30, while for analytical linear regression, it was 13.48 and 13.81, respectively for predicting heating and cooling load. One explanation for this could be that the target variables are not always perfectly linear with the input features. This means the analytical linear regression could overfit to the noise and lead to the larger mean square error.

4.4 Extra Experiments

We also implemented regularization in linear regression. We noticed that as the regularization parameter increased beyond 0.001, the test error increased above 100, indicating that regularization is unsuitable for this linear regression model. This holds true for both heating load and cooling load predictions.

We also added momentum and tested different beta values. We observed that the convergence of the gradient descent algorithm was quicker with larger beta values/decay rates. This could be explained by the fact that we are updating the parameters in larger steps by using previous gradient values.

For logistic regression, we tried feeding the model with an array of different regularization parameters. We observed that apart from a sharp fluctuation in the beginning, the accuracy is steadily decreasing as the regularization parameter increases. We believe this decrease in accuracy is due to the model becoming too simple and not being able to capture the underlying relationships in the data. Further, we also tried giving the model different momentum beta values. We observed that as the decay rate increases, the accuracy of the model decreases before finally hitting a plateau. We believe this behavior is caused by the fact that a higher decay rate causes the model to converge faster to a local minimum and be unable to climb out of it.

All graphs for these extra experiments can be found in the associated code for each model.

5 Discussion and Conclusions

In conclusion, we learned about how various hyper-parameters, such as data sizes, batch sizes, and learning rates, affect the performance and accuracy of regression models. Furthermore, we also compared the convergence speeds of models with these different hyper-parameters.

In the future, it would be interesting to run even more experiments on both these models and see how they perform. For example, we can analyze the effects of transforming the data with non-linear bases, and try different evaluation metrics for classification such as ROC curve (receiver operating characteristic curve).

6 Statement of Contributions

The workload was distributed evenly across all 3 team members. Linear Regression was handled by Mohammad Sami Nur Islam, while Logistic Regression was handled by Kaicheng Wu, Mathieu-Joseph Magri, and Mohammad Sami Nur Islam. Finally, all team members contributed to the report.

7 Bibliography

A. Tsanas, A. Xifara, 'Accurate quantitative estimation of the energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567, 2012

A. Tsanas, 'Accurate telemonitoring of Parkinson's disease symptom severity using nonlinear speech signal processing and statistical machine learning', D.Phil. thesis, University of Oxford, 2012

Kim, M.J., Han, I., 'The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms'. Expert Syst. Appl. 25, 637–646, 2003.