Impact Of Training Data On The Performance Of Linear Regression And Logistic Regression

|  |  |  |  |
| --- | --- | --- | --- |
| Saad Tariq Trinity College Dublin Ireland maliksa@tcd.ie | | Yifei Trinity College Dublin Ireland fangy@tcd.ie | Shuqi Wu Trinity College Dublin Ireland wush@tcd.ie |
|  |

# Introduction

Adequate amount of training and testing data is crucial to the performance of a machine learning algorithm. Training data allows to tune our model’s parameters as best as possible, whereas with testing data we measure and judge exactly how well our trained model performs under new data.

In this paper, we investigate the performance of machine learning algorithms as the training data increases. Our goal is to measure the impact that the size of training data has on the various metrics associated with the performance of a machine learning algorithm.

# RELATED WORK

Brian et al. conducted a similar research, where they investigated the effect of increasing data size on bias-variance tradeoff for classification algorithms. The paper found that increasing data set sizes leads to decrease in variance (which leads to overfitting) and has negligible impact on bias. Galdi et al. reviewed and contrasted the different metrics used to evaluate machine learning algorithms, and noted a positive correlation between training data and predictive accuracy of the model.

# methodology

**3.1. Finalizing algorithms that we need to analyze.**

Be began by choosing which algorithms we wanted to test to answer our question. Two important categories of Machine Learning problems are the regression problems and the classification problems. For our experiment, we decided to analyze one algorithm for each of those problems, namely Linear Regression and Logistic Regression.

**3.2. Collecting datasets for algorithms**

The next step involved finding appropriate datasets to test each of the machine learning algorithms. Each of our algorithms was tested under two different datasets: A ‘simple’ dataset, and a ‘complex’ one. For project, ‘complex’ and ‘simple’ are relative terms, and a ‘complex’ dataset would have over 3 times the feature of a simpler one.

Following is the description of the classification datasets:

1. Census Income Data (Simple dataset): This dataset contains annual income of individuals and associated variables. Features include age, years of education, hours worked per week etc., and the last column indicating whether each individual makes over or under $50k per year. This is a multivariate, binary class dataset
2. Dota2 Game Results (Complex dataset): Dota2 is a popular online video game. This dataset contained information related to Dota2 matches and their results. The features include Cluster ID (associated with server location), Game Mode (different variants of matches that can be played), Game Type (e.g. matchmaking based on player rank), and 113 other features, each of which corresponds to a unique character that a player can play with. Finally, the last column of the dataset contains the results of a match (1 indicates team won the match, -1 indicates a loss). This is a multivariate, binary class dataset.

**3.3 Dataset Preprocessing:**

All four of our datasets contained integer type, real valued features and several categorial features, both nominal and ordinal.

We were able to use the integer type and real valued features out of the box and did not do any preprocessing with them. We stripped the dataset of all the categorical, nominal features with multiple values like ‘Country of origin’, ‘Marital status’ etc. For categorical features with binary values (i.e. Sex), we replaced the values ‘Male’ and ‘Female’ with integers 0 and 1. Finally, we removed ordinal categorical features with multiple values. For categorical features, we could have employed schemes likes ‘One Hot’ encoding to be used with Logistic Regression, but we decided not to as our model, possibly due to the nature of the datasets that had those features, exhibited high performance with very few samples and features, so One Hot preprocessing was deemed unnecessary.

With the remainder of the features, we calculated the correlation between them, with the intent of stripping highly correlated variables as algorithms like Linear Regression and Logistic Regression are known have poor performance with highly correlated features.

<ADD PLOTS FOR FEATURE CORRELATION HERE>

**3.4 Performance Evaluation:**

The metrics used to evaluate the performance of Logistic Regression were:

1. Accuracy.
2. Log Loss
3. Area under ROC

For Linear Regression:

1. Accuracy
2. Log Loss
3. R-squared coefficient
4. Mean squared error

To measure the effect of training data on the aforementioned performance metrics, we developed scripts that followed the following general workflow:

1. Randomly select **n** training samples from dataset having **N** instances. The remaining instances (**N**-**n**) serve as testing data.
2. Train the model using the **n** samples
3. Test the model against (**N**-**n**) samples by performing cross validation.
4. Acquire performance metrics.
5. Increase **n** and repeat step #1.

The scripts developed for dataset analysis were capable of performing both Hold Out and KFold cross validation schemes. Due to the large size of the datasets, Hold Out method was mainly used.

# results and discussion

**4.1 Logistic Regression**

Our experiments showed that the impact of training data was more pronounced on the complex dataset than it was on the simple one, for which performance stabilized quickly and no trends could be identified.

For the simple dataset, the performance very quickly improved and stabilized as training data was increased to 1000 instances. Within this interval (0-1000), the accuracy rose from a low of 72% to 80% (**Figure 1**). The log loss fell very sharply from 1.4 to 0.5 (**Figure 2**). Finally, the ROC fluctuated between 0.57 and 0.61 (**Figure 3**) and did not seem to stabilize as training data was increased.

For the complex dataset, a linear trend was observed across all metrics up until 20000 instances of training data. Within this interval, the accuracy rose from 52% to 59% (**Figure 4**), the log loss fell from .85 to .67 (**Figure 5**), and the ROC increased from 0.52 to 0.63 (Figure 6). The performance did not improve further after 20000 instances.

In summary, it appears that training data has a lot stronger impact on performance of Logistic Regression when the dataset is complex and has a lot of features. Whereas for simpler features, training data has minimal impact on performance.

**4.1 Linear Regression**

**##########**

**ADD LINEAR REGRESSION RESULTS HERE**

**##########**

# LIMITATIONS

Due to time constraints, the performance Logistic Regression was not measured against other types of datasets, e.g. multiclass/multilabel datasets.

We also chose one instance of a ‘simple’ dataset and one instance of a ‘complex’ dataset for each type of algorithm. Performing the same experiment against multiple simple and complex datasets may reveal a different trend.

The Logistic Regression model’s performance on the complex dataset was subpar (around 60% predictive accuracy at best), but since our project was only concerned mainly with the *changes* in the performance, we elected to change the dataset/algorithm. For future research, it would be interesting to investigate the change in performance for





