Analysis of Linear Regression and LDA

Rajae Faraj, Eric Liu, Orla Mahon

September 28, 2019

Abstract

This project effectuated two separate linear machine learning implementations - logistic regression and linear discriminant analysis - on two distinct tasks: one predicting a wine's quality based on its chemical properties (eg. pH, density), and the other tumor malignancy based on the tumor's various properties (eg. area, radius). We compared the run-time of these two models, as well as the final accuracy averaged over 5-fold cross validation. Both the training and prediction of our models were done on two benchmark datasets of relatively small size (1599 wine-set examples and 699 cancer-set examples). We found that LDA performed better than logistic regression on both datasets, in both run-time and accuracy, and that logistic regression performed best with a learning rate of 0.1. On the wine dataset, increasing the number of gradient-descent iterations (by factors of 10) was found to improve our final accuracy; however, doing the same on the cancer dataset did not - the accuracy there fluctuated between 1 and 100 iterations. We will discuss potential motivations for this phenomenon in the results. Finally, adding additional features (such as quadratic expansions) to the cancer dataset improved its logistic regression accuracy, but adding the same features to the wine dataset did not.

Introduction

In this project, we tackled two different probabilistic approaches to binary classification: logistic regression (LR) and linear discriminant analysis (LDA). Both linear models were applied to two datasets. The wine dataset contained 11 features and 1599 examples, while the cancer dataset consisted of 10 features and 699 examples. Since the sizes of these datasets are significantly small, we were wary of overfitting and of outliers. To prevent the former, we pruned the datasets of any training examples that had missing data and avoided adding complex features and feature interactions. The latter is discussed in the datasets section.

LDA performed better than LR in terms of accuracy and run-time on both datasets. The accuracy for each was as follows (averaged over 5-fold cross validation):

- Wine Dataset
  - LR: 62.63% (with 10,000 gradient descent iterations)

  - LDA: 73.67%

- Cancer Dataset
  - LR: 64.12%, up to 69.68% when removing outliers (with 100 gradient descent iterations)
  - LDA: 95.44%

The two models are different in nature, one of them is a discriminative model (LR), while the other is a generative model (LDA). LDA is generally found to be more practical when classifying training examples into different groups, whereas LR performs better when binarily mapping dependent features to independent ones (1). Moreover, LDA is known to be superior to LR in terms of accuracy and efficiency when all of the LDA's assumptions are fulfilled (2). Thus, the over-performance of LDA is justifiable in this case. We also discovered that a learning rate of 0.1 for LR is the most optimal for both datasets. The accuracy of LR over the wine dataset is thought to be affected by convergence as it improves up to 10,000 iterations, but there is only a marginal increase of 3% accuracy when going from 100 to 10,000 iterations. This confirms that the choice of the learning rate was appropriate since when using an optimal learning rate, gradient descent is guaranteed to converge. As for the cancer dataset, the gradient descent converges too rapidly to make any assumptions about the quality of the learning rate.

Datasets

We worked with two small datasets: a wine dataset and a cancer dataset.

The wine dataset consisted of 1599 training examples, each example having 11 features, and a resultant quality score between 1 and 10. The quality score was mapped to a binary output, where 5 or lower corresponded to a 0-output. There was no missing data in this dataset, so no examples were pruned when training the model.

The cancer dataset consisted of 699 training examples, each example having 10 features, and a resultant output of either 2 (benign) or 4 (malignant), which we mapped to 0 or 1 respectively. Some examples had missing features (denoted by a '?'), so this entire example was pruned from the dataset; there were 16 such examples.

We added additional features under the following structure:

$$X_i = X_{i-k} + X_{i-k}^2, i \in \{k+1, k+2, ..., 2k\}$$

where k is the number of features in the dataset. Thus for each dataset, we doubled the number of features, and each new feature was an expansion $x + x^2$ of the original feature x, moving left to right through the k features. Training on this modified input matrix improved our logistic regression model's accuracy by 5.59% on the cancer dataset, but actually reduced our accuracy on the wine dataset by about 0.5%.

We attempted to clean the data further by removing what we considered "outliers", which we defined as features that were outside of three standard deviations of the mean of that feature's distribution. If a training example had any feature that lay outside of this range, we pruned it. This yielded negative results on the wine dataset, which had 141 total examples pruned, resulting in an accuracy decrease of 6.07%. On the cancer dataset, however, this resulted in an accuracy increase of 5.56% with 53 examples pruned.

When working with datasets of this (small) size, there is always a risk of overfitting; our model could pickup a bias (in the colloquial sense of the word) based on the data that it saw if the set it trained on is not generalized to begin with.

Results

Statistics

The distributions of positive and negative classes for our two datasets is as follows:

	Wine Dataset	**Cancer Dataset**
% Positive Class	53.4709	34.9927
% Negative Class	46.5291	65.0073

The cancer dataset had roughly two thirds of the set be negative-classed samples (benign tumor), whereas the wine dataset was more evenly distributed.

Model Performance

The final accuracy of our models (where 1.0 is 100% accuracy), trained and reporting on the two datasets, is as follows (where the learning rate of logistic regression was 0.1, 5-fold cross validation was used, and quadratic features were added to the cancer dataset):

	Wine Dataset				**Cancer Dataset**			
Iterations	**10**	**100**	**1,000**	**10,000**	**1**	**10**	**100**	**1000**
LR	0.4915	0.5906	0.5950	0.6263	0.6529	0.6529	0.6412	0.6412
LDA	0.7367				0.9544			

The LR model's performance on the wine dataset, as expected, increases with the number of gradient descent iterations. The learning rate of 0.1 yielded the best performance amongst the set tried, and produced lower accuracy than LDA on both datasets as well as a significantly slower runtime. On the cancer set, trainings with 1 and 10 gradient descent iterations were oddly better than those with 100 iterations, which we theorized could've been because the less-trained model resided in a minimum which performed better on this small dataset than the actual minimum it converges to.

The following runtimes assumed 100 iterations for LR, and that 5-fold cross validation was used:
LDA completed 5-fold cross validation on both datasets in a matter of milliseconds, whereas LR with 100 iterations took seconds to terminate. We also tested leave-one-out cross validation ($k = number\_of\_samples$) on both datasets, but for LDA only (due to runtime concerns), and achieved marginal accuracy gains of less than 1%. Overall, LDA greatly outperformed logistic regression in both measures of performance.

	Wine Dataset	**Cancer Dataset**
LR	3.4843 secs	1.3926 secs
LDA	0.04879 secs	0.0097 secs
LDA Leave One Out	4.1943 secs	0.7362 secs

We found the learning rate to not affect the runtime performance of the logistic regression model for either dataset, since our runtime is determined by the number of iterations and not by convergence to a local or global minimum. Had we chosen to not specify the number of iterations, and instead rely on detecting conversion, then the learning rate would surely have affected the runtime performance. However, it did somewhat affect the accuracy of our model. On the wine dataset, our LR model's accuracy dropped considerably at a learning rate of 0.01 and 1, relative to that at 0.1, at roughly 8%. This is likely due to the learning rate being just the right amount to cause the model to fall into a local minimum that is suboptimal compared to the minimum found with a learning rate of 0.1.

Discussion and Conclusion

Through this project we've observed the superior performance, in terms of both accuracy and speed, of linear discriminant analysis over logistic regression with regards to our datasets in question. However, we were able to improve the performance of logistic regression through the addition of more complex features (though this leads to the possibility of overfitting), a higher number of descents, k-fold validation, and more optimally-chosen hyperparameters (eg. the learning rate).

To this last point about hyperparameters, we could have considered an approach similar to simulated annealing, where the learning rate is at first quite high (exploration phase) and then relaxed and lowered (exploitation phase). This method could have better avoided the common pitfall of getting trapped at local minima, due to its initial high learning rate allowing for large jumps in the error-space. Then, once the learning rate is lowered, the model would be able to easily converge on a more-optimal minimum. This would eliminate the ambiguity of what the learning rate should be set to, but leads to more meta-decisions such as how quickly the learning rate should be lowered.

We could similarly remove the specification of the number of descents, and let the model itself halt once it has converged to a minimum or desired accuracy - this would cause increased runtime complexities due to not having a well-defined stopping criterion, but may allow for better-tuned weights.

Lastly, an increased number of folds our k-fold cross validation will likely mark an increased accuracy performance, as seen with LDA (though the increase was marginal). The higher number of folds we do, the more of the dataset we allow our model to interact with (as a training or validation set), as well as the dataset with itself, which would reduce the effect of irregularities or biases within the dataset. However, this would markedly increase the runtime by a factor of k, which for larger datasets is a noticeable problem.

Statement of Contributions

- Rajae Faraj: LDA implementation, writing the report
- Eric Liu: codebase/project setup and structuring, logistic regression implementation, running experiments, writing the report
- Orla Mahon: writing the report

Bibliography

1 - Press S. J, Wilson S. Choosing between logistic regression and discriminant analysis. Journal of the American Statistical Association. 1978;73:699–705.
2 - Hastie, T., Tibshirani, R.,, Friedman, J. (2001). The Elements of Statistical Learning. New York, NY, USA: Springer New York Inc..