ML Algorithms from Scratch

Brief Overview

Using data from the Titanic dataset (titanic_project.csv file), we will explore the C++ scratch implementations of logistic regression and Naïve Bayes to predict whether passengers survived. Logistic regression will use sex as its only predictor. Naïve Bayes will use age, pclass, and sex as its predictors.

Training of the logistic regression model involves computing weight coefficients while training of the Naïve Bayes model involves computing A-priori probabilities in Bayes' theorem. Comparisons of the two algorithms will be based on training time along with the following test metrics: accuracy, sensitivity, and specificity.

Code Outputs

Logistic Regression

```
Opening file titanic_project.csv.
Total number of observations (rows): 1046

Number of rows for training: 800
Train time: 177067000 nanoseconds
Coefficients: w0 =  0.999878; w1 =  -2.41087

Number of rows for testing: 246
Test metrics: Accuracy = 0.784553; sensitivity = 0.695652; specificity = 0.862595
```

Naïve Bayes

```
Read 800 train rows and 246 test rows.

Survived
====================
A-priori: 0.39
pclass: Mean = 1.90385; variance = 0.727934
sex: Mean = 0.320513; variance = 0.217784
age: Mean = 28.8261; variance = 208.485

Died
====================
A-priori: 0.61
pclass: Mean = 2.43033; variance = 0.589408
sex: Mean = 0.840164; variance = 0.134288
age: Mean = 30.4182; variance = 204.732

Train time: 1179700 nanoseconds

Test metrics: Accuracy = 0.784553; sensitivity = 0.695652; specificity = 0.862595
```

Analysis of Algorithm Results

Both logistic regression and Naïve Bayes produced the same test metrics results. The models both had an accuracy of ~79%, a sensitivity of ~70%, and a specificity of ~87%. The percentages for all the test metrics are above 50%, which indicates that both the models can correctly predict the survival of passengers fairly well.

However, training run times of the two models were quite different, with logistic regression clocking in at 177067000 nanoseconds and Naïve Bayes at 1179700 nanoseconds. For this exploration, Naïve Bayes runs more than 100 times faster than logistic regression! Perhaps adjusting the learning rate and number of iterations to compute the weight coefficients could improve the training run time for logistic regression, but it is likely that it still would not perform any faster than Naïve Bayes.

Generative Classifiers versus Discriminative Classifiers

Generative and discriminative models are similar in that both can be used for classification and regression. Both can accept training inputs and learn to distinguish between various classes based on features of a dataset. Nevertheless, the methods through which the two types of models learn about the dataset are quite different from each other.

Generative models such as Naïve Bayes attempt to learn the function that was used to generate the target variable values from each training observation's features. On the other hand, discriminative models such as logistic regression simply look for a boundary that can be used to separate various classes of data based on their features. Generative models can synthesize new outputs, while discriminative models simply predict a probability based on input observations. The cited article provides an excellent illustration at the end describing the general approach each type of model would take to determining what language a person speaks: "the generative approach is to learn each language and determine which language fits into the speech, and the discriminative approach is to determine the linguistic differences without learning any language at all [1]."

Reproducible Research in Machine Learning

The phrase “reproducible research in machine learning” refers to the recreation of a machine learning workflow to reach the same conclusions as the original work [2]. Reproducibility helps researchers to confirm research findings and get inspiration from each other’s works. Reproducible research is important because it ensures that results are not only correct, but also “ensures transparency and gives us confidence in understanding exactly what was done [3].” If we are unable to understand why an algorithm works the way it does, it makes it difficult to investigate for further improvements or implementing it for real world applications.

Reproducibility can be implemented by having a standard checklist that brings attention to information that could be missing or possible software bugs that could impact the performance of trained models [3] [4]. Such a checklist would include how training data is documented, how features are selected and generated, how models were trained, and how the software environment for training the models were set-up. Following a thorough and standardized checklist helps to create detailed documentations of machine learning pipelines that can allow other researchers to precisely replicate and evaluate the results to that of the original.

References

[1] P. by Daniel, “Generative vs. Discriminative Models,” *Forum of Artificial Intelligence in Medicine (FAIM)*, 01-Nov-2021. [Online]. Available: https://faimglobal.org/generative-vs-discriminative-models/.

[2] P. by Daniel, “Generative vs. Discriminative Models,” *Forum of Artificial Intelligence in Medicine (FAIM)*, 01-Nov-2021. [Online]. Available: https://faimglobal.org/generative-vs-discriminative-models/. [Accessed: 02-Oct-2022].

[3] Z. Ding, “5 - Reproducibility,” *Machine Learning Blog | ML@CMU*, 24-Aug-2020. [Online]. Available: https://blog.ml.cmu.edu/2020/08/31/5-reproducibility/.

[4] E. Rivera-Landos, F. Khomh and A. Nikanjam, "The Challenge of Reproducible ML: An Empirical Study on The Impact of Bugs," 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), 2021, pp. 1079-1088, doi: 10.1109/QRS54544.2021.00116.