Technical Note: How to Rationally Compare the Performances of Different Machine Learning Models?

Terazima Maeda*
Faculty of Engineering, Nagoya University, Nagoya 464, Japan
*Maedater@outlook.com

Abstract: Nowadays, there is a large number of machine learning models that could be used for various areas. However, different research targets are usually sensitive to the type of models. For a specific prediction target, the predictive accuracy of a machine learning model is always dependent to the data feature, data size and the intrinsic relationship between inputs and outputs. Therefore, for a specific data group and a fixed prediction mission, how to rationally compare the predictive accuracy of different machine learning model is a big question. In this brief note, we show how should we compare the performances of different machine models by raising some typical examples.

Keywords: Machine learning, training, testing, comparison.

1. Background

In recent years, machine learning becomes popular again with the boosting concept of “big data” and deep learning techniques [1]. Many areas have started to use machine learning as a promising predictive algorithm to “mine” the intrinsic non-linear relationship between the input and output data [2]. Usually, for a specific data set, people need to compare different machine learning algorithms before deciding which algorithm could provide the best predictive performance. However, different machine learning models have very different algorithmic principles and parameters, which leads to high difficulty to find out the best model.

In this technical note, we use some typical examples to show how we could compare different machine learning model for a given data set. Also, we will propose several suggestions for machine learning beginners.

2. Case Analysis

The factors that should be compared among different machine learning algorithms include 1) training accuracy and 2) training time. In terms of the training accuracy, people need to separate the data set into the training and testing sets. (sometimes, an additional validation set is also useful [3], but not required). People need to use the training set to fit the data, with the principle of the optimization process provided by the used machine learning model. Then the data in the testing set can be used for evaluating the accuracy of the model by acquiring the error and accuracy rates. Since the testing result is completely dependent to the specific selected training data, we can do multiple training with different data components, and then acquire averaged error and accuracy results from their corresponding testing sets. Usually, a strict cross-validation is recommended [4]. However, due to the high computational cost of cross-validation, we can alternatively use a “sensitivity test” to evaluate the model performance after multiple training using randomly shuffled data set. Li and co-workers have found that such a method can help to quickly acquire the reliable results of root mean square error (RMSE) [5] and accuracy rates from different machine models [6–8]. Especially, they discovered that such a comparative method is particularly robust for large data set. Thus, such a fast
model evaluation method can be useful to quickly compare the performances of different machine learning algorithms for the same data set.

In terms of the training time, this is usually dependent to the complexity of the model: more algorithmic parameters would require more training time to converge [9]. However, we may want to keep the training time as short as possible. We strongly suggest that the training time should also be recorded and considered as the factor that determine the best machine learning model.

3. Suggestions

Based on the discussions above, Here, we provide a suggested machine learning training & testing process for comparing different machine learning models for a specific data set:

1. Shuffle and split the data into training and testing sets.
2. Perform model training and testing. Record the training time while the model converges.
3. Record the RMSEs, accuracy rate in the testing set.
4. Split the training and testing data for many more times and repeat Steps 1-3.
5. Calculate the average RMSE, average accuracy rate and average training time for the specific model.
6. Repeat Steps 1-5 for the same machine learning model with different parameters.
7. By comparing the average RMSEs, average accuracy rates and average training times, find out the best parameters in the same machine learning model.
8. Repeat Steps 1-7 for a different machine learning mode.
9. Compare the best average RMSEs, average accuracy rates and average training times among different evaluated machine learning model.

References

5. Wikipedia contributors Root-mean-square deviation.