Arman Sarjou

A Comparison of Naïve Bayes and Random Forests Algorithms for Speech Accent Recognition using Mel-Frequency Cepstral Coefficients (MFCCs) – Supplementary Materials

Glossary

Stability – Defined in this study as it is in Liu, et al (2017): The variation in prediction over repeated model runs using the same parameter settings and dataset

F1 Score – Calculated as $\frac{2*Precision*Recall}{Precision+Recall} * 100$. This is measured as the balance of precision and recall in a binary classification system.

Precision – Calculated as $\frac{True\ Positive}{True\ Positive+False\ Positives} * 100$. This indicates how good the model is at not giving false positives.

Recall – Calculated as $\frac{True\ Positive}{True\ Positive+False\ Negatives} * 100$. This indicates how good the model is at not giving false negatives.

Accuracy – In this study, Accuracy is calculated as: $(1 - Classification\ Error) * 100$

Classification Error – Calculated by Matlab's *loss* and *resubloss* functions with loss function set to 'Classiferr'. This is a measure of predictive inaccuracy of classification models (Mathworks, 2020). Uses the following equation:

$$L = \sum_{j=1}^{n} w_j I\{\hat{y}_j \neq y_j\}$$

where L is weighted average classification loss, w_j is the weight for observation j, I{x} is the indicator function, y_j is sample class label, $\hat{y}_j$ is class with maximum posterior probability. Sample weights are normalized and sum to 1.

Area Under Curve (AUC) – This is the Area under a Receiver Operator Characteristics Curve. It provides a measure of performance for binary classification systems across various classification thresholds. High AUC is desirable. Has a maximum value of 1.

Receiver Operator Characteristics (ROC) Curve – A graph of True Positive Rate vs False Positive Rate which shows performance of a binary classification system at various classification thresholds.

Standard Deviation – Calculated as: $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}$ where N is number of samples, x_i is sample i and μ is the mean. This is a measure of how spread out the data points are (how much variance there is).

Bayes Theorem – $p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$; where P(X) is a normalization constant required to ensure that the sum of conditional probabilities on the left-hand side equals to 1 over all values of Y (Bishop, 2006). In a Bayesian Statistical approach, probabilities provide quantification of uncertainty (Bishop, 2006).

Posterior Probability – Probability after the value of selection is observed (known as p(Y|X)). The Posterior probability is proportional to the Prior multiplied by the Likelihood Function (Bishop, 2006)

Prior Probability – Probability before value of selection is observed which is used to calculate posterior using Bayes' Theorem (known as p(Y)) (Bishop, 2006)

Kernel Distribution – A Non-Parametric method of estimating the probability density function of a random variable which makes no prior assumptions about the distribution of the data (Bishop, 2006)

Kernel Smoothing Width – A smoothing bandwidth value which controls how smooth the resulting probability density curve will be. A higher smoothing width will result in a smoother curve. (Mathworks, 2020)

Kernel Smoothing Function – A function which determines the shape of the derived probability density function estimate (Mathworks, 2020)

Leaf Size – Number of samples per end node of a decision tree
Number of Splits – The number of branches per decision tree (depth of tree)
Number of Trees – The number of Decision Trees within the RF
Stratified – Sampling which preserves the class distribution of the dataset
Bagging – Also known as bootstrap aggregation is where data is randomly sampled without replacement. This is used to form uncorrelated ensembles of decision trees in RF
Parallel Computing – A way of saving time and computational cost by splitting large computational tasks into smaller tasks that can be run at the same time
K-Fold Cross-Validation – A method of cross-validation where data is split into K number of folds in which the Kth fold is used as a validation set and the other folds of data are used to train a model. Over K number of iterations, models are trained, and cross-validated on data in this way to prevent overfitting. The cross-validation loss is the average loss on the validation set over K number of iterations.
Hold Out Cross-Validation – A method of cross-validation where data is separated randomly using a percentage split where the majority (usually around 80%) of data is used for training and the separated data partition is used for validation. This method is usually used with extremely large datasets.

Intermediate Results

Initially, it was decided that, in order to follow Ma & Fokoue (2014) and carry out a comparison, I would use the method of cross-validation that they used. In this study, they define accuracy as:

$$Accuracy = \frac{\sum 1 - \frac{TP + TN}{N}}{m}$$
$$where\ m = crossvalidation\ size\ (number\ of\ iterations)$$
$$N = number\ of\ samples$$

This meant that they carried out randomly stratified hold-out cross-validation for each run and took the average from all of these runs as the accuracy. When I tried to implement this, it took a very long time as it required training and validation 500 models per hyperparameter combination. When originally attempted, this took 38 hours for Naïve Bayes and could not be completed for Random Forests. Therefore, instead a K-Fold cross-validation was used which was more suitable for data of this size. This was done using 10-Folds in line with Ahmad, et al., (2015) who used 10-Fold cross-validation in a comparison of NB and RF models with a similar sized MFCC dataset.

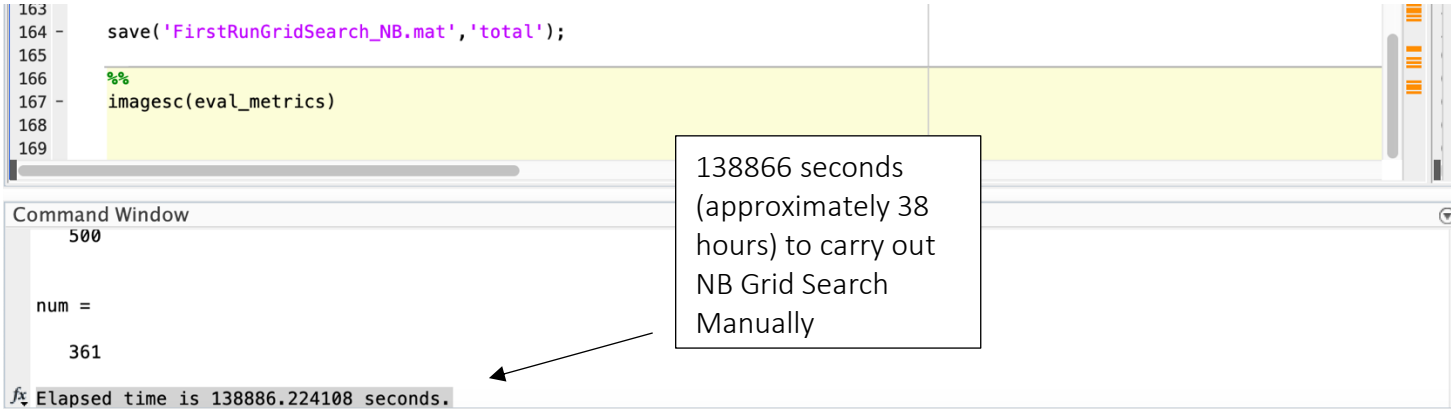

Figure 1: Elapsed time for Naïve Bayes Grid Search using Iterative Hold out Method from Ma and Fokoue (2014)

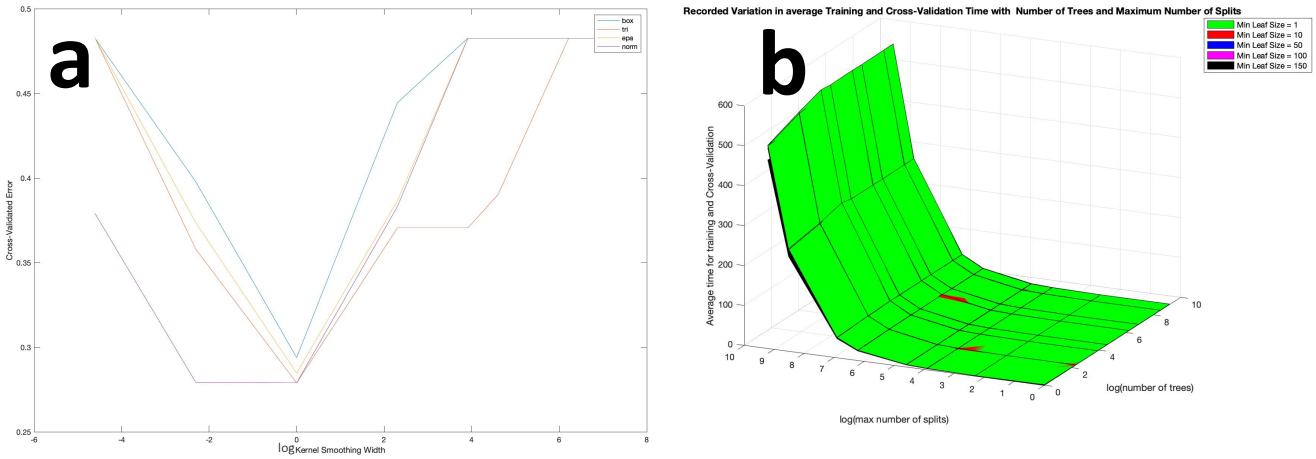

Figure 2: a) Cross-validation error vs log(Kernel Smoothing Width). b) Average time to train and carry out 10-Fold cross-validation vs log(maximum number of splits), log(number of trees).

The 10-Fold cross-validation would involve 50x less computational power as for each grid combination there would be 50x less iterations that would need to be run.
When optimizing for kernel width, initially widths $1*10^{-6}$, $1*10^{-5}$, $1*10^{-4}$, $1*10^{-3}$, $1*10^{-2}$, $1*10^{-1}$, 1, 10, 100, 1000 were tested. This resulted in the graph shown in figure 1b. This indicated that the best range for kernel width would be in the range of 1-100 (as done to produce final models)

When observing the average time taken to carry out cross-validation during the RF grid search, there was a clear relationship between the tree depth (Max Number of Splits) and the time taken. Tree depth is the dominating factor that determines how fast cross-validation can happen during training.

Lastly, the variation in standard deviation of error for the Random Forests grid search should be noted. The relationships are shown in Figure 4 are described in the poster. The axis to these graphs have been log transformed to allow viewing of all results on one graph. Parameter values varied from 1 to 10000.

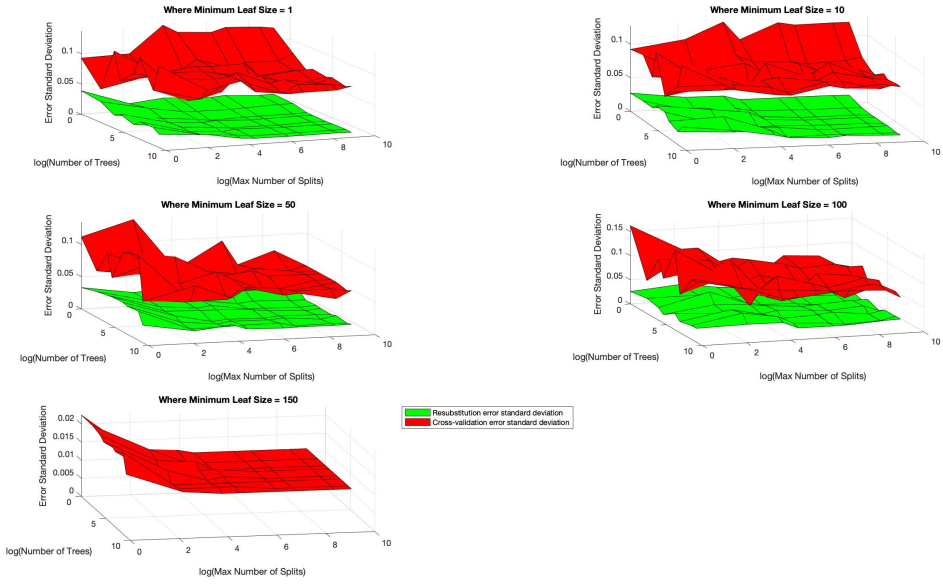

Figure 3: Subplots of Error Standard Deviation vs log(max number of splits), log(number of trees). Each subplot has varying minimum leaf size.

Implementation Details including Main Implementation Choices

The dataset initially contained samples of 329 rows of 12 MFCC values which are labelled by where the English speaker is natively from (EN, FR, IT, GE, ES, US. The distribution of the original dataset meant that US data is over sampled and the dataset in general would not be well balanced. To resolve this, and to be able to compare results with Ma and Fokoue (2014) who use the same dataset, the dataset was relabled to US vs non-US accent. This turned the problem from a multiclass classification task with unbalanced classes to a binary classification system with balanced classes which can be compared with existing literature.

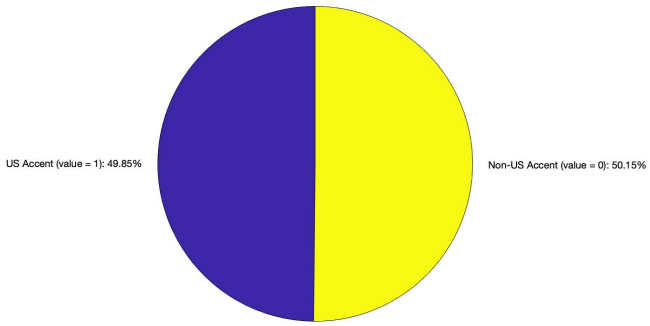

Figure 4: Pie chart of Sample Labels (US vs Non-US Accent)

To understand the dataset, it was important to look at the Pearson and spearman's rank correlation between the predictors. The Pearson's correlation matrix shows little strong correlation between the predictors.

The Spearman's rank describes monotonic relationships between predictors whereas Pearson's describes linear relationships between predictors. It is important to understand the correlation between the predictors as correlation is the normalized covariance which plays a key role in Naïve Bayes as Naïve Bayes assumes that all predictors are independent. If it was seen that there was high correlation between certain predictors, one of the predictors in question could be removed.

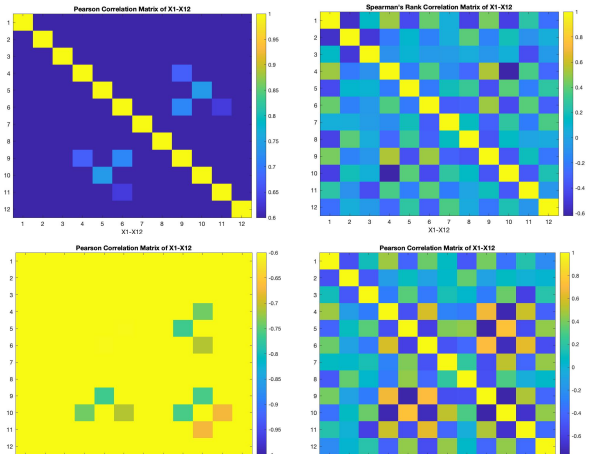

Figure 5: Various Correlation Matrices for the Dataset

Grid search was done manually instead of using Matlab's *Bayesopt* function which carries out Bayesian optimization. This was as Grid search using nested for loops meant full control of the output of each combination of parameters as well as the ability to test any parameters that were wanted. *Bayesopt* on Matlab, provides the most flexible Bayesian Optimisation but wouldn't allow altering of sample prior or to measure fully the impacts of increasing certain parameters.

Fitcnb is the only Naïve Bayes classifier that is available on Matlab but for Random Forests it was decided to use *templateTree* and *fitcensemble* with the 'Bag' parameter to allow bootstrap aggregation. Nested for loops were used instead of CVPartition. Both were tested with but using a nested for loop it would be easier to monitor the performance of each fold and to derive descriptive statistics like standard deviation much easier.

To simulate real world deployment, final models are tested once on the unseen testing data which was held out at the very beginning of the experiment.