USING NAÏVE BAYES AND RANDOM FORESTS TO PREDICT GAMMA OBSERVATIONS WITH CHERENKOV RADIATION DATA

Divya Balasubramanian & Linh Pham

CITY UNIVERSITY OF LONDON EST 1894

Motivation:

Particle physics and the distinct detection of particles in the Earth's atmosphere is an interesting area of research. On interacting with the Earth's atmosphere, gamma rays produce showers of particles that travel at high speeds. Experimental data including both gamma and hadron particles are produced by the Monte Carlo simulation. The classification of the nature of the particle is undertaken by the MAGIC Telescope and the simulation aims to improve its sensitivity. Research is underway to identify the most appropriate approach to classify the particles from the large amount of background showers produced which also contains hadrons[3]. We decided to test this problem using two of the strongest classifiers, Random Forests and Naïve Bayes.

Data Description & Analysis: The data was generated by a Monte Carlo program to simulate the observation of high energy gamma particles in a ground-based Cherenkov gamma telescope. High energy gammas can be classified by parameters, extracted after pre-processing images of air showers into elliptical clusters. There are 19,020 data points, divided into 10 numerical features, as described statistically in Table 1. Some of the features showed high level of correlation (Figure 1), but a robust classifier should still be able to work. The binary dependent variable contains two classes: gamma (class 1) and hadron (class 0)[3]. All variables were standardised using z-score before training to reduce the effect of varying scales of the input variables (Table 1), especially as kernel densities were used to train Naïve Bayes models. The data has a bias towards the gamma class (64.8% of total data). Hence, a minority oversampling method called ADASYN was applied to the training data to generate 3,326 new data points for the hadron class, resulting in a more balanced dataset[8].

Hypothesis: Research carried out by Bock et al, found Random Forests to be the most powerful classifier measured by ROC, even when compared with more complex algorithms such as neural networks, to distinguish between gamma (signal) and hadron (background) events in the MAGIC dataset[3]. Another study by Wainer, comparing 14 classification algorithms on 115 binary datasets, including MAGIC, ranked Random Forests as a top classifier while Naïve Bayes as one of the worst, measured by error rates[14]. Based on this literature, we expect Random Forests to outperform Naïve Bayes in terms of accuracy and ROC. Although, we expect that Random Forests will be computationally more expensive than Naïve Bayes [13].

Data Analysis – Before Normalization & Data Balancing

	Mean	St_Deviation	Skewness	Minimum	Maximum
flength	53.25	42.36	2.01	4.28	334.18
fwidth	22.18	18.35	3.37	0.00	256.38
fsize	2.83	0.47	0.88	1.94	5.32
fconc	0.38	0.18	0.49	0.01	0.89
fconc1	0.21	0.11	0.69	0.00	0.68
fasym	-4.33	59.21	-1.05	-457.92	575.24
fm3long	10.55	51.00	-1.12	-331.78	238.32
fm3trans	0.25	20.83	0.12	-205.89	179.85
falpha	27.65	26.10	0.85	0.00	90.00
fdist	193.82	74.73	0.23	1.28	495.56

Table 1. Descriptive statistics

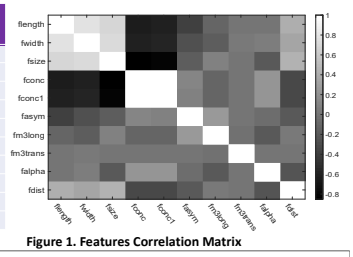

Figure 1. Features Correlation Matrix

Choice of Training and Evaluation Methodology: Data was randomly shuffled before splitting into 70% train set and 30% held-out test set. ADASYN was applied to balance the training data. The grid search for both methods was run on training data both before and after applying ADASYN. Each model was trained on 3-fold cross validation partitions due to the long processing time to train Random Forests. These folds were fixed to ensure comparability between different models. The best model was selected based on the lowest mean classification error. Finally, best models were applied to the held-out test set and comparisons between Naïve Bayes and Random Forests were made using not just accuracy but also other metrics such as ROC and F1.

Naïve Bayes Overview: Naïve Bayes is a simple yet powerful classifier which uses Bayesian probability to calculate the posterior probability of a class given values of the feature variables. The algorithm makes a key assumption that all feature variables are independent of each other. Based on this assumption, the posterior is proportional to the product of the prior and the likelihoods – both of which can be estimated from the training data or assumed a distribution such as Gaussian or Binomial. Finally, Maximum a Posteriori decision rule is applied such that the class with the highest product of the prior and likelihood is chosen. **Advantages:** The conditional independence assumption makes it easier to calculate the posterior, resulting in shorter training time. Thus, the model is helpful when the dimension of the input space is high. Despite this strong assumption, the classifier may still work well because the decision boundaries can be insensitive to the details in the class conditional densities[2]. **Disadvantages:** However when the variables are related, the conditional independence assumption may also lead to wrong calculations of the posterior distribution. As a result, the model might perform worse than more sophisticated models including Random Forests[8]. If one of the feature variables has a likelihood of zero, the Naïve Bayes algorithm might fail to run due to how it calculates the posterior. To deal with this problem, a technique called Laplace smoothing can be used.

Random Forests Overview: Random Forests is an ensemble method[5] that is a collection of decision trees used for classification. The algorithm performs bootstrapping to create a random subset of the predictors, with replacement, alongside recording observations that are not included ('out of bag' observations) for each decision split. The oobError is the error rate for the classification of out of bag observations[5]. The number of trees, minimum leaf size and number of predictors are the hyperparameters that influence the classification and hence need to be considered for tuning to find the optimum values. **Advantages:** Random Forests overcome the problem of overfitting and there is no necessity of pruning trees. Individual trees are parallelizable thereby making them very efficient. They are interpretable and non-parametric for various data types [1]. Random forests are a reliable algorithm when working with large datasets including those with missing values. They work well on both categorical as well as numeric data.

Disadvantages: As per some research, it is computationally expensive in memory and CPU time[6]. Particularly with increased number of trees, the training time significantly increases. This is true in our research as well due to which we chose 3-fold cross validation to reduce the CPU time. They are not easy to interpret as compared to decision trees.

Naïve Bayes - Parameters and Experimental Results:

Naïve Bayes - Cross Validation Results - After Normalization & Data Balancing

- **allNormal**: models for whom all input variables follow Gaussian distributions
- **allKernel**: models for whom all input variables follow kernel distributions
- **mixed**: models for whom kernel distributions are applied only to input variables that have skewness outside the [-1,1] range[10]. The other input variables follow Gaussian distributions.

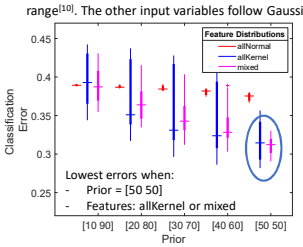

Figure 2. Naïve Bayes – Box plots of error rates for different distribution types for input variables across all prior assumptions (fold-level results)

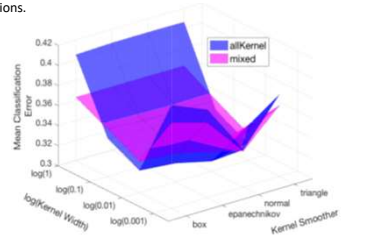

Figure 3. Mean Error Surface Plot for the impact of kernel width and kernel smoother for models that use Kernel distributions

The grid search for the Naïve Bayes model used the following parameters:

- five different priors for the binary class to reflect different degrees of bias towards gamma i.e. [10 90] means class 0 (hadron) has a prior of 10% and class 1 (gamma) prior of 90%.
- different distribution types for the input variables – allNormal, allKernel and mixed [10] (See explanation for each type above).
- Finally, when kernel distribution was used, 4 different kernel width options were tried in the range of (0.001, 0.01, 0.1 and 1), this range was chosen based on a trial run using the 'Hyperparameter Optimization' function in Matlab.

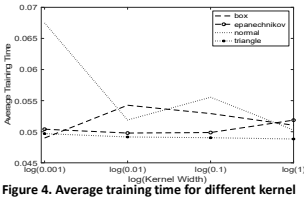

Figure 4. Average training time for different kernel smoothers across all kernel widths (the average is taken between allKernel and mixed kernel models)

Results: Models with priors closer to uniform distribution performed better during training (Fig.2) as the bias towards one class would result in unnecessarily high posterior for that class, consistent with findings by Rish et al[12]. Cross validation results also reinforce that non-parametric kernel densities give better estimates and lower training errors for inputs than Gaussian densities (*allNormal*) when input variables are not regular or are multimodal[15]. The choice of kernel smoothers did not have noticeable impact on classification errors because the final distribution is an average of all the kernels (Fig.3). However, kernel widths, which control how "smooth" each kernel curve is, play a deciding role in the final distribution and the likelihood of each input variable[15]. Fig 3. shows that kernel width at 0.01 gave the best accuracies, while larger bandwidths failed to describe the true distribution of inputs and produced worse results. All models took less than 0.3s to train and kernel widths did not affect the training speed (Fig.4). The best model from cross validation had a mean error of 28.29%, using a uniform prior distribution [50 50] with "triangle" kernel estimates for all input variables and a kernel width of 0.01. When trained with the imbalanced data set, the best model had mean error of 21.05%, using a uniform distribution for the class priors, "normal" kernels for all inputs and a kernel width of 0.1.

Random Forests – Parameters and Experimental Results:

We ran a grid search on the following parameters:

- Number of trees - It is considered in the range of 1 to 100 with an incremental change by 10 trees in each run.
- Minimum leaf size - It is considered in the range of 1 to 10 with an incremental change of 1 leaf in each run.
- Number of predictors – It is considered in the range of 1 to 10 as we have 10 features in the dataset.

The grid search was run twice, once for the normalised imbalanced dataset and once for the balanced dataset.

Results:

- The best model in the grid search performed on the balanced dataset used 80 trees, 1 minimum leaf and 1 predictor parameter with a classification error of 12.55%.
- Figure 5 shows that the training time increases significantly with increase in number of trees. This is expected as the forest becomes denser and thereby causing more decision splits to be undertaken. Furthermore, bootstrapping higher number of trees in each run can also affect the computational time.
- Figure 6 shows that the out of bag error (oob error) reduces gradually with increasing number of trees.
- It is seen from Figure 7 , 8 and 9, that mean classification error reduces with the number of trees, increases with number of leaves and marginally increases with number of predictors, respectively. An increase in the number of trees ideally enhances the chances of more number of samples to be accounted for in the randomized sampling.
- While the variance in the classification error is minimally different between the balanced and imbalanced dataset, imbalanced dataset performs slightly better. This is due to a bias towards the majority class of the dataset thereby causing the data to be mislabelled during validation [11].
- The best model from cross validation on the imbalanced dataset had an accuracy of 87.54% with 70 trees, 1 minimum leaf and 2 predictors.

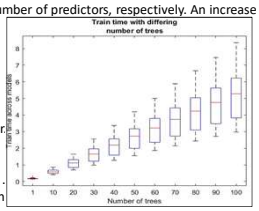

Figure 5: Training time based on number of trees considered in training

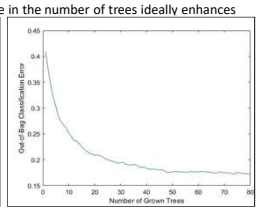

Figure 6: Out of bag classification error based on number of trees considered in training

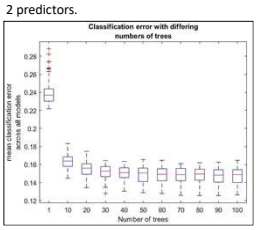

Figure 7: Mean classification error based on number of trees considered in training

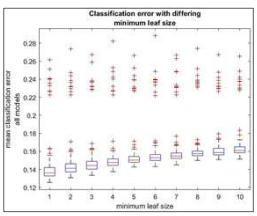

Figure 8: Mean classification error based on minimum leaves considered in training

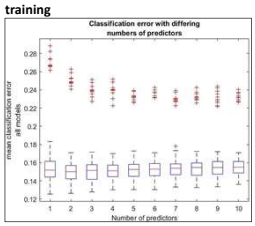

Figure 9: Mean classification error based on number of predictors considered in training

Analysis and Evaluation of Results: Consistent with our hypothesis, the best model for Random Forest (RF) outperfo--rmed best model for Naïve Bayes (NB) in both cross-validation and testing across all three performance measures (Accuracy, F1 and ROC curves). Accuracy and F1 are not the best measures of performance for the gamma/hadron segregation problem because the cost of classifying a background event (hadron) as signal (gamma) (False Positives) is much higher than the cost of classifying a signal as background (False Negatives)[4]. Hence, they should be analysed in conjunction with ROC curves. Figure 12 shows that for Accuracy, RF achieved 87.35% and 85.26% in cross-validation and testing respectively; NB achieved 71.71% and 75.73%. For F1, RF achieved 87.85% and 88.12%, while NB got 73.98% and 80.38%. The ROC curves in Figure 13 show that the True Positive Rates (TPR) for RF are higher than that for NB within the acceptable thresholds for False Positive Rates (FPR) from 0 to 0.2, as suggested by Boch et al[3].

Although RF outperformed NB, the algorithm performed slightly worse in testing than during cross-validation (Figure 12), suggesting RF overfitted the training data and was penalised by reduced generalisability. This result is a surprise as one of the benefits of RF as a classifier is reducing variance while keeping the bias low thanks to the use of the ensemble of trees[7]. However, according to Hastie et al, if the depth of the trees is too rich it is likely to overfit the data. The depth of the tree is controlled by the minimum leaf size, which is the minimum number of observations in each leaf. Smaller leaf sizes mean deeper trees, which is the case as the best RF model has a minimum leaf size of only 1.

What NB gained in simplicity and computational efficiency - all NB models took less than 0.3s to train, the algorithm lost out in term of performance (Figure 5). In this particular case, it might be because some feature variables are highly correlated (Figure 1), violating the conditional independence assumption. On the other hand, our study showed that NB had a higher level of generalisability than RF, as shown by a better accuracy in testing than in cross-validation (Figure 12). This is because the simplicity of the NB algorithm prevents it from overfitting to its training data.

For both NB and RF the value of Negative Predictive Value (NPV) were much higher in cross validation than in testing. For NB, NPV fell from 72.45% to 64.72%. For RF, NPV fell from 86.57% to 76.79% (Figure 12). These results came from the oversampling of the minority negative class to balance the training data, which made both algorithms much better at classifying the negative class (hadron) during training. However during testing, both algorithms would have a tendency to over-classify the negative class, leading to higher values for False Negatives (FN) and lower values for NPV in testing.

Naïve Bayes & Random Forests : Results for applying best models (trained on *balanced* data) on the held-out test set

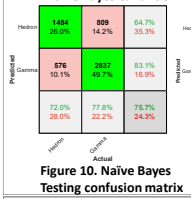

Figure 10. Naïve Bayes Testing confusion matrix

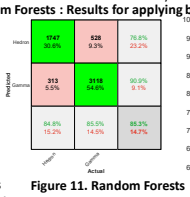

Figure 11. Random Forests Testing confusion matrix

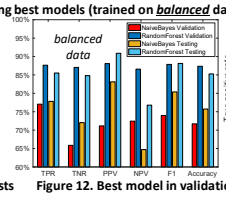

Figure 12. Best model in validation & testing (balanced data)

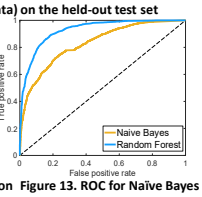

Figure 13. ROC for Naïve Bayes and Random Forests - Testing

Best models (trained on *imbalanced* data)

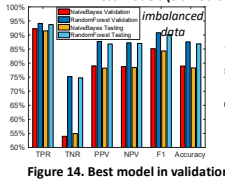

Figure 14. Best model in validation & testing (imbalanced data)

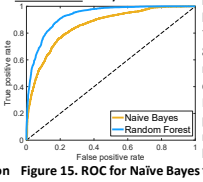

Figure 15. ROC for Naïve Bayes and Random Forests - Testing

The results of training on imbalanced data were similar to those on balanced data in terms of RF outperformance over NB. However, training on the biased data increased the performance measures for both models, more so for NB (Accuracy from 75.73% to 78.29% in testing) than RF (Accuracy from 85.26% to 86.86%) (Figure 14). This increase came from the models getting very good at classifying the majority class (gamma), as shown by the sharp rise in True Positive Rate (TPR). However, both models got worse at predicting the minority class (hadron), as shown by much lower TNRs. NB also lost its generalisability as it became too reliant on the posteriors calculated from the biased data. Both models also exhibited an improvement in ROC curves with NB more so than RF.

Lessons Learned and Future Work: Our study reinforced the observation from previous research[3],[14] that RF is a superior classifier for high-energy astrophysical events. However, as the data in this study is simulated and not real data, generalisation of the results for real data is not guaranteed. It is also important to notice the tendency of RF to overfit the training data given special conditions with the inputs. NB did not perform as well but might be a more scalable solution given its high bias low variance nature. Based on these lessons, we concluded that further studies could experiment with the following strategies to improve performance of the models. To reduce overfitting for RF, we could rank feature importance and include only those that are relevant, as suggested by Hastie et al[7]. Dimensionality reduction can help reduce the feature correlation issue (Table 1) to improve the performance of NB. Finally, works by Boch et al[3] suggested binning samples into different levels of energy (variable *fsize*) to improve performance.

[1] Ali,J., Khan, R., Ahmad, N., Maqsood, I., (2012), "Random Forests and Decision Trees", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, ISSN (Online): 1694-0814
[2] Bishop, C.M. 2006, *Pattern recognition and machine learning*, Springer, New York.
[3] Bock, R.K., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jiřina, M., Klaschka, J., Kotrč, E., Savický, P., Towers, S., Vaiciulis, A. & Wittek, W. 2004, "Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope", *Nuclear Inst. and Methods in Physics Research, A*, vol. 516, no. 2, pp. 511-528.
[4] Bock, R.K. (2007) UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope]. Major Atmospheric Gamma Imaging Cherenkov Telescope project (MAGIC)
[5] Breiman, L. 2001, "Using Iterated Bagging to Debias Regressions", Machine Learning, vol. 45, no. 3, pp. 261-277.
[6] Chen, B., Sheridan, R., Hormak, V., & Voigt, J. (2012). Comparison of Random Forest and Pipeline Pilot Naive Bayes in Prospective QSAR Predictions. Journal Of Chemical Information And Modeling, 52(3), 792-803. doi: 10.1021/ci200615h
[7] Hastie, T., Tibshirani, R. & Friedman, J. H. (Jerome H.) 2017, *The elements of statistical learning: data mining, inference, and prediction*, Second edn, Springer, New York
[8] He, H., Bai, Y., Garcia, E.A. & Li, S. 2008, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", IEEE, , pp. 1322.
[9] Janitza, S. & Hornung, R. 2018, "On the overestimation of random forest's out-of-bag error", PloS one, vol. 13, no. 8, pp. e0201904
[10] Manning, C. D., P. Raghavan, and M. Schütze. *Introduction to Information Retrieval*, NY: Cambridge University Press, 2008
[11] Mellor, A., Boukir, S., Haywood, A. & Jones, S. 2015, "Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin", ISPRS Journal of Photogrammetry and Remote Sensing, vol. 105, pp. 155-168
[12] Rish, I., Hellerstein, J., and Jayram, T. 2001, "An analysis of data characteristics that affect naive Bayes performance". Technical Report RC21993, IBM T.J. Watson Research Center
[13] Singh, A., N., M. and Lakshmigianthan, R. (2017). Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms. International Journal of Advanced Computer Science and Applications, 8(12).
[14] Wainer, J. 2016, "Comparison of 14 different families of classification algorithms on 115 binary datasets".
[15] Zhang, W., Zhang, Z., Chao, H. & Tseng, F. 2018, "Kernel mixture model for probability density estimation in Bayesian classifiers", *Data Mining and Knowledge Discovery*, vol. 32, no. 3, pp. 675-707