Data

The data set chosen includes 336 observations of cellular localizations of E.coli proteins each with eight attributes that describe each observation. Of the eight attributes, seven hold either binary or discrete values with the eighth attribute as a text name describing the E.coli's sequence pattern.[i] The sequence pattern name was removed from the dataset before applying any statistical learning methods since the value does not have predictive capacity when determining the cellular site of a particular E.coli protein.

The target variable from the dataset is the cellular site of an E.coli protein. The target variable (class) can hold one of eight values which are listed below:

cp (cytoplasm) – 143 observations

im (inner membrane without signal sequence) – 77 observations

pp (perisplasm) – 52 observations

imU (inner membrane, uncleavable signal sequence) – 35 observations

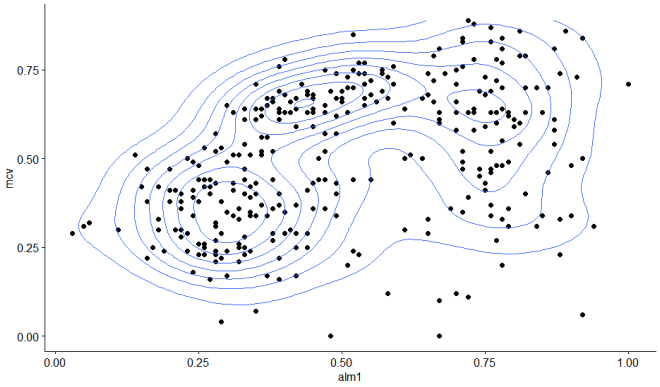
om (outer membrane) – 20 observations

omL (outer membrane lipoprotein) – 5 observations

imL (inner membrane lipoprotein) – 2 observations

imS (inner membrane, cleavable signal sequence) – 2 observations

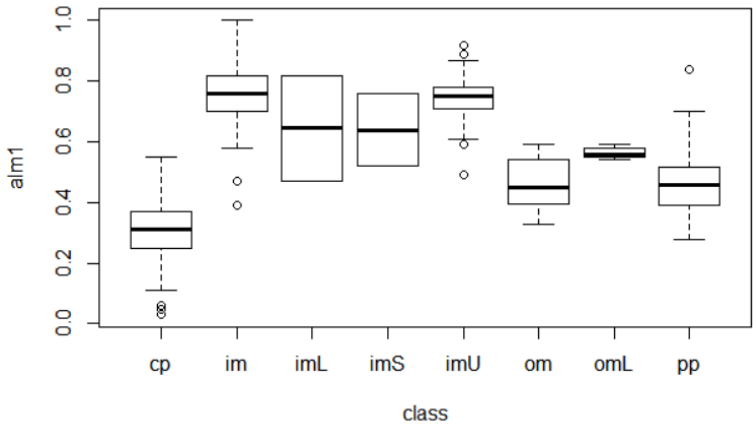
Below is a density plot of the data measured between the two strongest predictor variables, alm1 and mcv, two scores given to proteins based on their sequence and regions.



Looking at the density plot above it is important to keep in mind that although there seem to be three distinct "clusters", there are eight different classes that the data is to be classified by, not three. This is concerning since a lot of overlap could possibly lead to misclassification especially when plotted against the two strongest variables (determined by the importance() function from the randomForest library[ii]). However, by using classification trees and the other five independent variables it is likely that this will not have a very large negative impact on the model's accuracy.

To extend this idea further it may be useful to consider the boxplots of all the different independent predictor variables and how they have an effect on the classes of the dataset. Below, the eight different classes are plotted against the alm1 independent variable. Here we can see that the two classes with the highest number of observations, cp and im with 143/336 and 77/336 observations respectively are

prominently distinguished by the alm1 variable. This will have a good result on our overall missclassification error rate as the majority of observations fall between these two classes and they are easily identified by the strong predictor variable, alm1. Although not shown, a similar concept holds for the relationship between classes and the mcv independent variable.



Problem

Developing a model that is accurate for classifying cellular sites of E.coli protein observations can have a significant impact on research costs. If a model is accurate enough it will decrease the need for a trained human eye to classify each observation into one of the eight classes listed above. In this particular use case, the measurement of each independent variable will still need to be conducted by a researcher, however being able to automate the classification process could both decrease human error as well as research costs for labs or medical centers performing such observations.

Techniques

In order to classify each specific E.coli protein into one of eight cellular sites, classification trees were used. After the construction of an initial classification tree, cross validation was used to determine how many splits would produce the most accurate result. However, performing the cross validation showed that the number of splits that produced the smallest deviance actually used all splits with no pruning required. After recoding the results from a standard classification tree, bagging, random forests and boosting algorithms were applied to see if there would be an increase in the model's accuracy.

Bagging attempts to increase the predictive capacity of a model by splitting the training data into subsets and building a tree for each new subset. The final output is an average of the results from each tree. In classification, instead of an average the classification with the highest frequency is the output of the model.[iii]

Random forests, similar to bagging, produces many trees from which their results are averaged to produce an output. However, during the process of creating the trees only certain predictor variables are used such that succeeding trees cannot use those predictor variables in the same fashion. As a result, the average outcome is calculated from many *uncorrelated* trees. For example, if a dataset has one very strong predictor variable, trees will be split on that variable each time and therefore produce similar trees. However, using random forests this pattern is prevented which reduces variance and in theory should produce a more accurate result.[iv]

Using boosting many trees are built sequentially by using information from the previous trees to decrease variance in the model's output. The aim here is to reduce the residuals in each tree moving forward to produce a more accurate model.[v]

All methods used for classification tree building were imported from the tree[vi] and randomForest[vii] libraries in R version 3.5.1.[viii]

Results

After building a classification tree and applying bagging, boosting and random forests, the misclassification rates for each procedure were as follows:

> Classification tree misclassification rate: 0.1071
> Classification tree with bagging misclassification rate: 0.0655
> Classification tree with random forests misclassification rate: 0.1250
> Classification tree with boosting misclassification rate: 0.1428

Therefore, by introducing a bagging algorithm the misclassification rate of E.coli cellular sites was decreased from 10.71% to 6.55%. However, the introduction of random forests and boosting did not decrease the model's misclassification error rate but instead increased it from the output of the original classification tree.

Conclusion

Using classification trees with the application of a bagging algorithm produced a model that was able to classify cellular sites of E.coli proteins with a misclassification rate of 6.55%. This particular use case of classification trees with boosting, bagging and random forests shows that a classification tree model's accuracy does not always improve after the implementation of boosting or random forest algorithms. However, these results are promising because they imply that producing an accurate model to determine E.coli protein cellular sites is possible which could not only save labor and capital resources in research labs but could also encourage increased research activity in related fields through the use of automation in the classification process.

---

[i] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[ii] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22

[iii] James, G. (2013). An introduction to statistical learning : with applications in R. New York, NY: Springer.

[iv] Ibid.

[v] Ibid.

[vi] Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

[vii] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22

[viii] R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.