Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Fuzzy Forests: A New WGCNA Based Random Forest Algorithm for Correlated, High-Dimensional Data

Daniel Conn
UCLA SPH, Biostatistics

Tuck Ngun

Christina Ramirez
UCLA SPH, Biostatistics

Abstract

In this paper we introduce fuzzy forests, a new machine learning algorithm for ranking the importance of features in high-dimensional classification and regression problems. Fuzzy forests is specifically designed to provide unbiased rankings of variable importance in the presence of correlated features. Fuzzy forests uses Weighted Gene Coexpressionn Network Analysis (WGCNA) to detect groups of highly correlated features. Unbiased rankings are obtained by fitting separate random forests on each module. We also introduce our implementation of fuzzy forests in the R package, **fuzzyforests**.

Keywords: Random Forests, WGCNA, machine learning,R.

1. Introduction

The problem of identifying important features in the presence of correlation has been an area of intense research within the statistics and machine learning community. Biological applications have, in particular, spurred the development of high dimensional feature selection methods. While model based feature selection algorithms such as the LASSO or SCAD may efficiently detect important features in presence of correlation (Raskutti *et al.* 2010), this efficiency comes at the cost of making parametric assumptions that may not hold in practice.

Random forest variable importance measures (VIMs) offer a nonparametric alternative to model based feature selection algorithms (Breiman 2001). Random forests is a popular ensemble based machine learning algorithm. While random forest VIMs have demonstrated the ability to accurately capture the true importance of features in settings where the features are independent, it is well-known that random forests VIMs are biased when features are correlated with one another (Strobl *et al.* (2007); Strobl *et al.* (2008); Nicodemus and Malley

(2009)).

Fuzzy forests cope with correlated features by taking a piecewise approach. We partition the set of features into distinct modules such that the correlation within each module is high and the correlation between groups is low. We then use an iterative feature selection algorithm to select the most important features from each. A random forest is then fit to the features that have survived this first round. A final iterative random forest, combining features from all modules, selects the top k features.

There are a variety of algorithms for partitioning features into distinct, high correlation modules. In this regard, WGCNA is our method of choice. WGCNA is a rigorous framework for detecting correlation networks (Zhang and Horvath 2005). Although it was first developed to detect modules of highly correlated genes, it has found application in a variety of biological contexts. The R package **WGCNA** is a robust, computationally efficient, and well-documented implementation of the WGCNA framework. We expect that researchers already familiar with **WGCNA** will easily adopt the fuzzy forests algorithm and we expect that newcomers to WGCNA will be able to make good use of **WGCNA**'s fine documentation and tutorials.

The article is organized as follows. In section 2 of this article, we briefly review the random forests, WGCNA, and introduce the fuzzy forests algorithm. In section 3, we introduce the R package **fuzzyforest**. In section 4, we provide a heuristic proof that under the right assumptions, the VIMs obtained by fitting a separate random forest on each module are asymptotically unbiased. In section 5, we conduct simulations to comparing fuzzy forests to both random forests and conditional inference forests (CIFs). We demonstrate that fuzzy forests has performance comparable to that of CIF's although at much lower computational cost. In section 6, we use fuzzy forests to determine which biological factors are important in determining how well an HIV patient copes with the virus. Section 7 ends the article with a discussion and summary of our results.

2. Variable Importance Measures and the Fuzzy Forests Algorithm

2.1. Variable Importance Measures

In this section, we introduce basic notation and define variable importance measures. We assume that our data comes in the form of n independently and identically distributed iid. pairs $(X, Y) \sim G$. Here X is p dimensional feature vector and Y is a scalar outcome. In the case of both classification and regression we are interested in modeling the conditional mean of Y given a feature vector X_i. We denote this conditional mean alternatively as $E[Y|X]$ or $f(X)$ and we assume that $Y|X$ has distribution $f(X) + \epsilon$, where the ϵ are iid with variance σ^2. In the regression setting, Y is unrestricted. In classification, Y is restricted to take the value 0 or 1.

If the goal is to predict a new outcome Y based off of measurements, X, a good estimate of $f(X)$ is all that is required. We are interested in more than prediction. We are interested in understanding how $f(X)$ changes as function of particular features. Let $X = (X_1, \ldots, X_p)$ so that X_i corresponds to the ith feature. If the value of $f(X)$ varies widely according to a particular value of X_i, X_i is, in some sense, an "important" in determinant of the outcome, Y.

If p is low dimensional ($p = 1, 2$), we can simply plot our estimate of $f(X)$ to understand

how it varies as function of X. If p is moderate or large, $f(X)$ is difficult to interpret. It is most common in this case, to assume $f(X)$ has a specific parametric form so that $f_\beta(X)$ is known up to a finite dimensional parameter β. In the case of linear regression, β is a vector of regression coefficients and we can measure the importance of one feature versus another feature by examining the absolute magnitude of their corresponding coefficents.

If the parametric model $f_\beta(X)$ is a close approximation to $f(X)$, it is possible that interpretations based off of $\hat{f}_{\hat{\beta}}(X)$ will not be misleading. Likewise, if $f_\beta(X)$ is a poor approximation of $f(X)$, the resulting interpretation will be misleading. The parametric approximation $f_\beta(X)$ may be inadequate for a variety of reasons. This may occur if important features are not observed. Even if all appropriate features are measured, $f_\beta(X)$ may fail to capture important interactions between features. In the case of linear regression, the true $f(X)$ may be nonlinear in such a way that that the best linear approximation fails to capture.

Permutation VIMs provide a means of summarizing the importance of individual features without making parametric assumptions. We define the permutation VIM of variable X_i as

$$VIM(X_i) = E(f(X_1, \ldots, X_i, \ldots, X_p) - f(X_1, \ldots, \tilde{X}_i, \ldots, X_p))^2. \tag{1}$$

Here, X_i and $\tilde{X}_i$ are iid with distribution G_{X_i} where G_{X_i} is the marginal distribution of X_i. This form of the VIM is given in a slightly different form in (Gregorutti *et al.* 2013) and (Zhu *et al.* 2012). These authors also discuss conditions under which the estimate of the permutation VIM derived from random forests is consistent.

2.2. An Introduction to Random Forests

Random forests is a popular ensemble method that has been applied in the setting of both classification and regression. The random forests algorithm works by combining the predictions of an ensemble of classification and regression trees. Each tree is grown on a separate bootstrap sample of the data. The number of trees grown in this manner is denoted as *ntree*. Each regression tree is highly unstable and gives highly variable predictions. Averaging multiple trees over many bootstrap samples leads to more stable estimates of $f(X)$. The algorithm described thus far is known as bagging (bootstrap-aggregating). This algorithm is a special case of random forests.

A further element of randomness is introduced by random forests. Before a node in the tree split a subset of features is chosen at random. Of these randomly chosen features, the feature with the highest marginal importance is used to split the node.

Call the kth decision tree $\hat{f}_k(X_i)$. In the case of regression trees, $\hat{f}(X_i) = \frac{1}{ntree} \sum_{k=1}^{ntree} \hat{f}_k(X_i)$. In the case of classification

References

Breiman L (2001). "Random Forests." *Machine learning*, **45**(1), 5–32.

Gregorutti B, Michel B, Saint-Pierre P (2013). "Correlation and Variable Importance in Random Forests." *arXiv preprint arXiv:1310.5726*.

Nicodemus KK, Malley JD (2009). "Predictor correlation impacts machine learning algorithms: implications for genomic studies." *Bioinformatics*, **25**(15), 1884–1890.

Raskutti G, Wainwright MJ, Yu B (2010). "Restricted Eigenvalue Properties for Correlated Gaussian Designs." *The Journal of Machine Learning Research*, **11**, 2241–2259.

Strobl C, Boulesteix AL, Kneib T, Augustin T, Zeileis A (2008). "Conditional Variable Importance for Random Forests." *BMC bioinformatics*, **9**(1), 307.

Strobl C, Boulesteix AL, Zeileis A, Hothorn T (2007). "Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution." *BMC bioinformatics*, **8**(1), 25.

Zhang B, Horvath S (2005). "A General Framework for Weighted Gene Co-Expression Network Analysis." *Statistical applications in genetics and molecular biology*, **4**(1).

Zhu R, Zeng D, Kosorok MR (2012). "Reinforcement Learning Trees."

Affiliation:

Daniel Conn
Department of Biostatistics
UCLA School of Public Health
Los Angeles, United States of America
E-mail: djconn17@gmail.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd