Visualizing Trees of a Random Forest Model
CPSC 547 Project Proposal

Ken Lau
ken.lau@stat.ubc.ca

October 31, 2014

1 Domain and Task Description

It is usually not difficult to visualize an entire classification tree in a single image without overwhelming the user with visual clutter. Interpretation becomes a problem when we are required to visualize thousands of trees at once. Random forests have become a very common data mining algorithm due to advantages of high accuracy and ability to handle large amounts of input variables. A random forest generates thousands of classification trees through bootstrap and randomized subset feature selection, and makes a prediction from the input based on the majority vote of all the trees. Interpretation becomes very difficult since it is infeasible to analyze every single tree individually. However, a single tree is easy to interpret and contains a lot of useful information. For example, the label of an interior node represents the input variable used in partitioning the feature space into binary groups that optimizes prediction of the same target value. The leaf nodes correspond to the target values with the most occurrences within the particular group. This project will focus on the construction of a visualization system that accommodates the analysis of thousands of trees generated by random forests.

2 Data

We will be working with signal data coming from 6 different modulations. These represent the classes we want to predict. These modulations consist of on-off shift keying (OOK), binary phase shift keying (BPSK), offset quadrature phase shift keying (OQPSK), binary frequency shift keying at a carrier frequency of 868.3 MHz (BFSKA), binary frequency shift keying at 868.95 MHz (BFSKB), and binary frequency shift keying between 868.03 MHz and 868.66 MHz (BFSKR2). These modulations appear as leaf nodes in the tree. Furthermore, we extract a total of 144 quantitative features (input variables) for each received signal. These features appear as labels of the tree. Each classification tree generated from the random forest is stored as a tree structure in a JSON file. Figure 1 presents the structure of a single classification tree. We will analyze a collection of around one thousand binary trees.

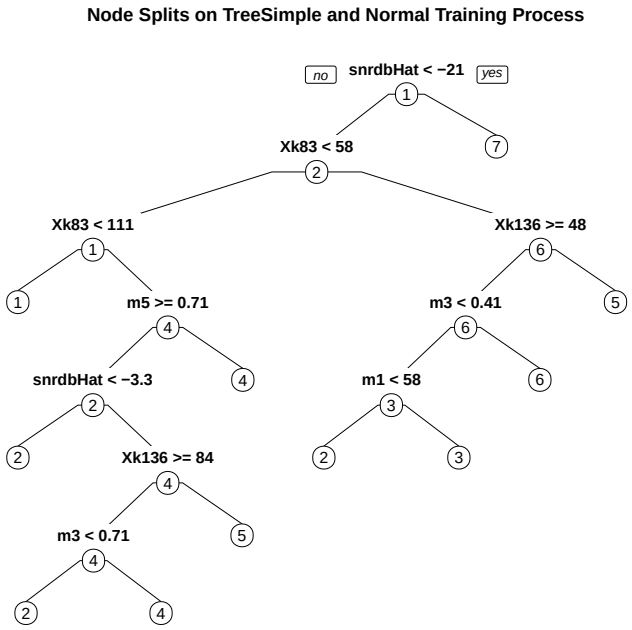


Figure 1: A single classification tree. Labels above each interior node represents split-point for a particular input variable. Leaf nodes represent modulation prediction labeled as 1 to 6.

3 Personal Expertise

This project should tie in with my MSc. project. I have been working with tree-based methods on signal modulation data mostly during the summer. I have some experience in applications of random forests. However, this will be my first time in implementing an application for visualizations.

4 Proposed Infovis Solution

This project aims to construct a consensus tree using the collection of trees generated by a random forest, and computing a weighted average of majority votes and information gain. The consensus tree would provide a useful summary of all the trees built from the random forest. The visualization would allow users to analyze and interpret input variables that are used across all the trees, thereby providing guidance in model selection and detection of

useful interactions of variables. The consensus tree would be grown horizontally from left to right. Each child node would be ordered using spatial encoding from top to bottom based on the weighted average of majority votes and information gain. We could collapse or expand nodes by clicking on them. The selection of one node would highlight all other nodes with the same input variable. Hovering over a node would display number of appearances and average information gain across all trees with the same structure up to corresponding depth. Majority vote and information gain of each input variable would also be computed across all generated trees. Dot plots with scores ordered by spatial encoding from highest score (top) to lowest score (bottom).

Idiom	**Proposed Vis Solution**
What: Data	Collection of Trees.
What: Derived	Consensus Tree computed by weighted score between majority votes and info gain based on all the trees.
What: Derived	Marginal majority vote and information gain values for every subset of consensus tree.
What: Derived	Summary statistics of average majority vote and information gain for each input variable
How: Encode	Consensus tree horizontal layout with collapsible nodes. Selection highlighting of same nodes, and hovering over nodes to display detailed information.
How: Encode	Dot plot with spatially encoded summary statistics for each input variable.

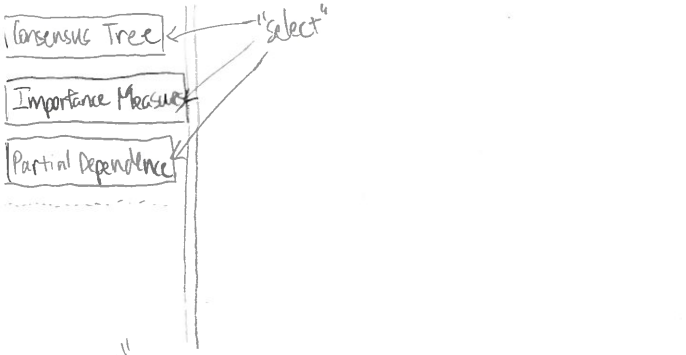
Table 1: Abstractions for proposed vis solution

5 Scenario of Use and Illustrations

From the left panel of the main screen, the user selects one of consensus tree, importance measures, or partial dependence options to analyze. For example, if the consensus tree option is selected, then a sankey diagram displaying the consensus tree will be brought forth for the user to analyze. The user can navigate the whole tree by using the sliders on the right and bottom of the screen. If the user clicks on a node, then all corresponding nodes with the same input variable will be highlighted with red. Moreover, hovering over a node displays detailed information on marginal majority vote and information gain score. If the user selects importance measure option from

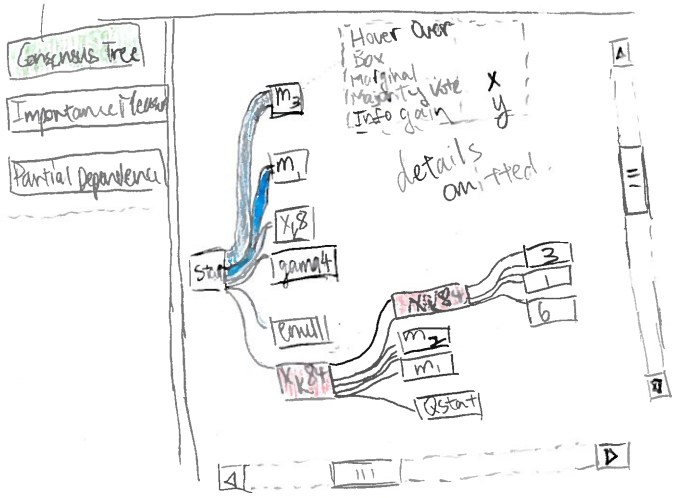
the left panel, then dot plots with scores from majority vote and information gain across all nodes will be brought forth to the screen. Selecting an input variable from one of the dot plots will highlight all corresponding input variables from the other dot plots. If the user selects partial dependence from the left panel, then partial dependence plots for pairs of input variables will be brought forth. The user can select 2 input variables for each of the axis as well as the response class of interest.

Main Screen:



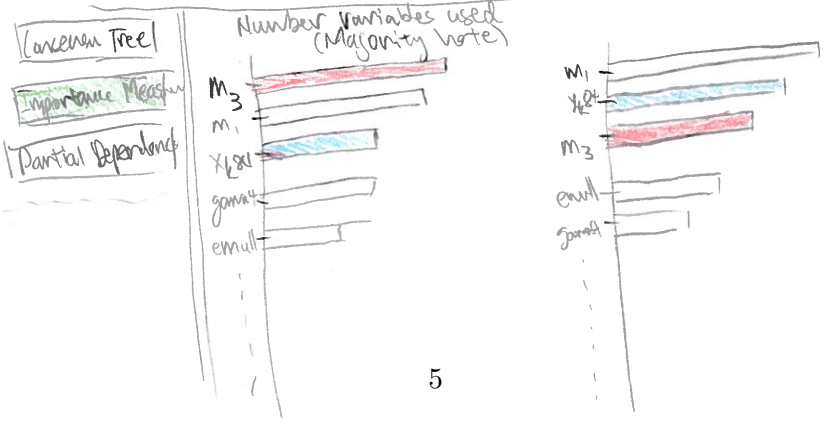
"selected"

Consensus Tree:

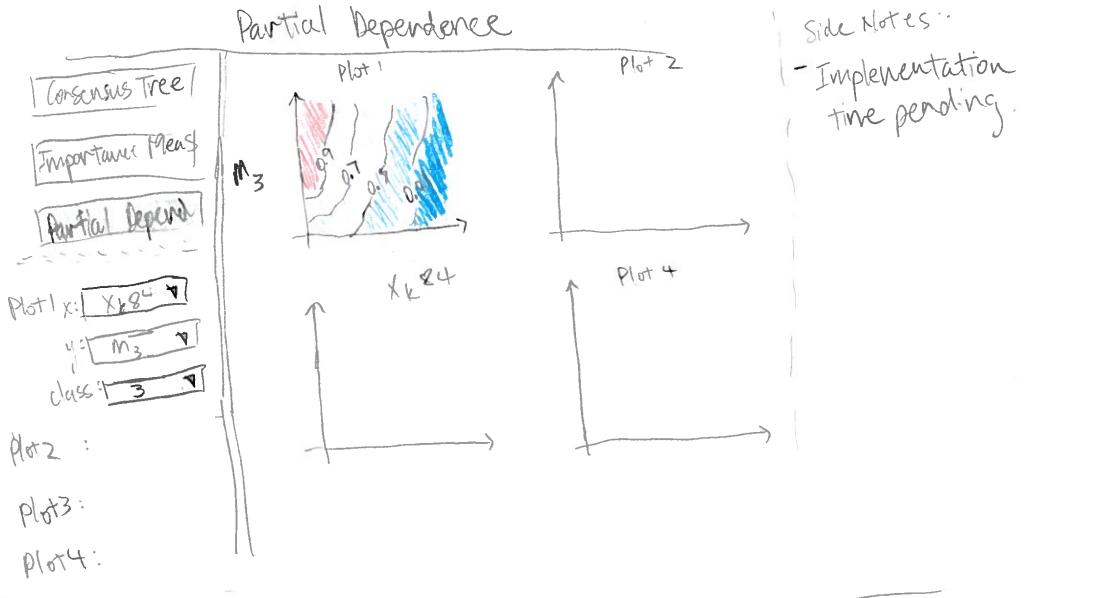


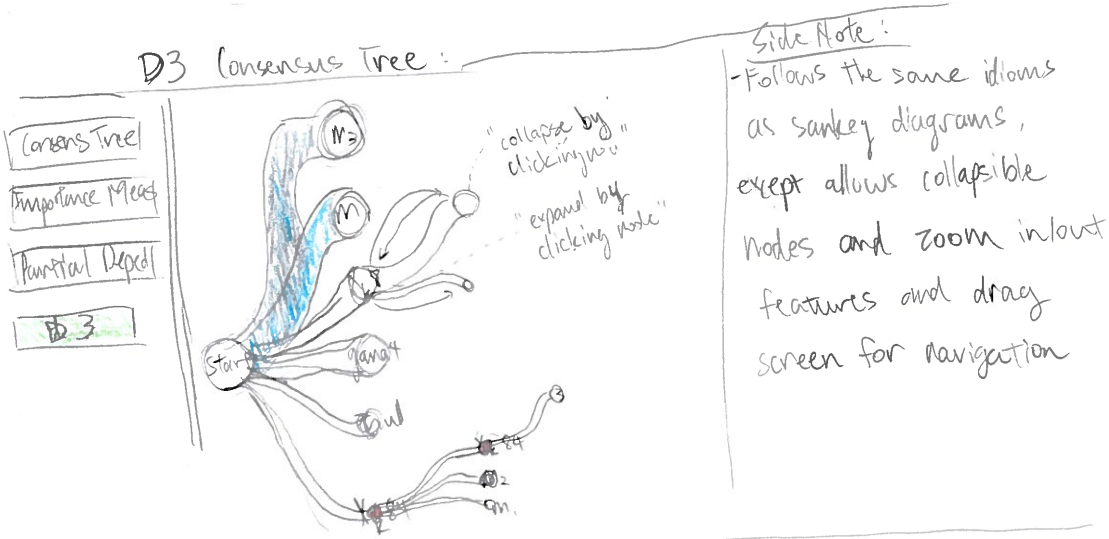
Side Note:
- 1, 2, ... 6 are predicted classes, leaf nodes
- start is starting node that doesn't do anything but connects to majority root nodes
- Nodes are ordered at each depth by weighted score.
- majority vote encoded with width of edge.
- information gain encoded with colour of edge.
- selecting a node colours red

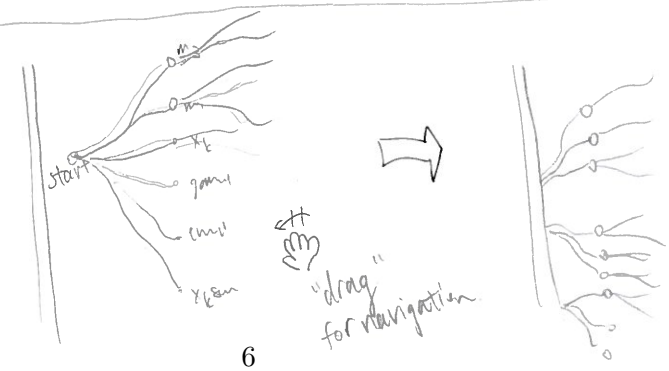
Dot Plots



- selection highlights in red corresponding variables.


Partial Dependence
Consensus Tree
Importance Measf
Partial Depend
Plot 1 x:
y:
class:
Plot 2 :
Plot 3:
Plot 4:
Plot 1
Plot 2
Plot 4
Side Notes:
- Implementation time pending.


D3 Consensus Tree:
Consens Tree
Importance Measf
Partial Depend
D3
"collapse by clicking node"
"expand by clicking node"
Start
Side Note:
- Follows the same idioms as sankey diagrams, except allows collapsible nodes and zoom in/out features and drag screen for navigation


Start
"drag" for navigation

6 Implementation Approach

The computations to construct the consensus tree will be carried out in R or Python. The final consensus tree with all the data at each node will be stored in a JSON file. I'm planning on using sankey diagrams in R along with shiny to display the consensus tree. The other option is d3, which allows more flexibility such as collapsible and linked selection of nodes along with other features. The dot plots will be done with ggplot2 in R.

7 Milestones and Schedule

- **Nov. 5^{th}** - Build a small random forest example with 100 trees and 5 input variables. Extract each tree with info at each node in clean JSON format.
- **Nov. 7^{th}** - Design weighted combination of majority vote and info gain to construct consensus tree.
- **Nov. 12^{th}** - Construct JSON format of consensus tree.
- **Nov. 14^{th}** - Status update. Continue to construct JSON format of consensus tree.
- **Nov. 21^{st}** - Start constructing shiny app. Build sankey diagram using consensus tree.
- **Dec. 1^{st}** - Complete sankey diagrams and dot plots in shiny app. Begin Scaling to large random forest.
- **Dec. 5^{th}** - Finish Scaling to large random forest.
- **Dec. 7^{th}** - Attempt to build consensus tree with d3.
- **Dec. 9^{th}** - Begin write-up and presentation.
- **Dec. 15^{th}** - Complete write-up.

8 Previous Work

The most common method to interpret the random forest model is to analyze a partial dependence plot [3]. It displays the effects of one input variable on the response accounting for the average effects of all other input variables.

Analysis of interaction effects is possible if we encode 2 input variables along the x and y axis. Contour lines are drawn to depict levels corresponding to response value. A problem with partial dependence plots is the limitation of low-dimensional views. In this project, we will analyze up to 144 features. Analyzing every pair of variables would be time consuming. An extension to partial dependence plots are CARTscans which captures the bootstrap calibration [4]. Another method to visualize and interpret random forests is through a 3D visualization of the trees [2]. Information for every classification tree is first extracted and mapped to an intermediate structure. Spatial positions of every node within the trees are then calculated and depicted on the screen.

References

[1] T. Munzner (2014). *Visualization Analysis and Design*. A K Peters Visualization Series. CRC Press.

[2] M. Yang, H. Xu, D. Zhu, H. Chen (2012). *Visualizing the Random Foroest by 3D Techniques*. Communications in Computer and Information Science Volume 312, 2012, pp 639-645.

[3] T. Hastie, R. Tibshirani, and J. Friedman (2001). *Elements of Statistical Learning*. Springer Series in Statistics Springer.

[4] M. Nason, S. Emerson, and M. LeBlanc (2004). *CARTscans: A Tool for Visualizing Complex Models*. Journal of Computational and Graphical Statistics, Volume 13, Number 4, Pages 119.