Neural Shrubs - Combining Decision Trees with Neural Networks

Kyle A. Caudle

Joint work with: Randy Hoover, Aaron Alphonsus

Department of Mathematics and Computer Science
South Dakota School of Mines & Technology

05 February, 2019

2019 SDSU Data Science Symposium

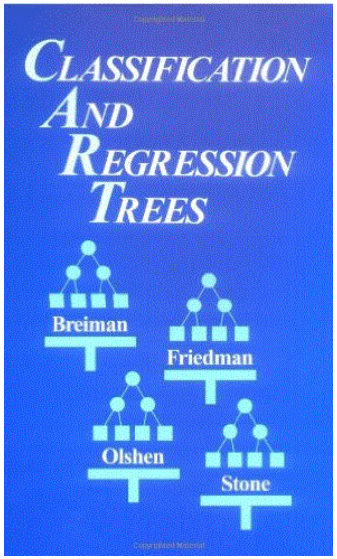
Outline

1. Introduction
2. Building a Binary Decision Tree
3. Neural Shrubs
4. Demonstration 1 (MNIST)
5. Demonstration 2 (SensIT)
6. Concluding Remarks/Open Questions

2019 SDSU
Data Science
Symposium

Background (Classification and Regression Trees)

- Binary decision trees were developed by social scientists in the early 60's. The use of trees in regression can be traced to the Automatic Interaction Detection program (AID) developed by Morgan and Sonquist at the University of Michigan.
- Leo Breiman and Jerome Friedman began working on trees independently in the early 70's and then teamed up with Stone and Olshen to publish what most would consider the bible of the subject "Classification and Regression Trees" in 1984.



2019 SDSU Data Science Symposium

Introduction

- The classical approach is to search over all variables and all possible split values for each variable and determine which one gives you the largest decrease in impurity.
- Suppose there are k features, the i^{th} historical observation would be represented as $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{ki}, y_i)$.
- The matrix of all historical observations is denoted as $\mathbf{X}$.

2019 SDSU Data Science Symposium

Introduction

- The partitions, $R_1(d, c)$ and $R_2(d, c)$, can be defined as follows:

$$R_1(d, c) = \{\mathbf{x}_i \in \mathbf{X} | x_{id} \leq c\}, R_2(d, c) = \{\mathbf{x}_i \in \mathbf{X} | x_{id} > c\}.$$

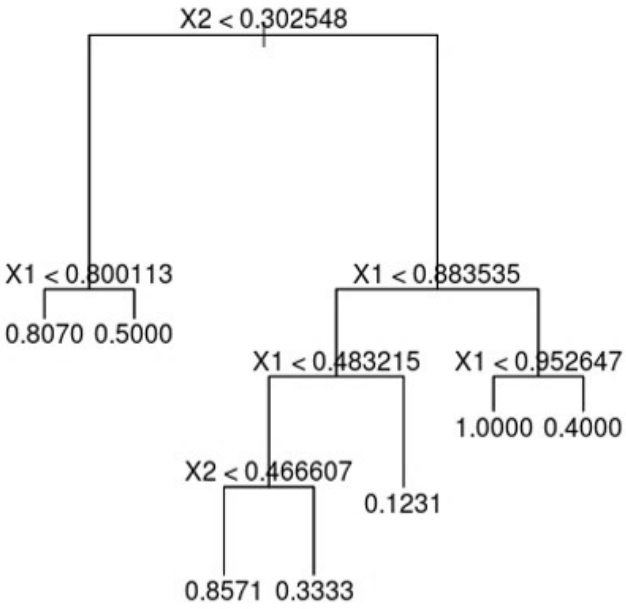
- We can then define the optimal partitions as (Regression Tree):

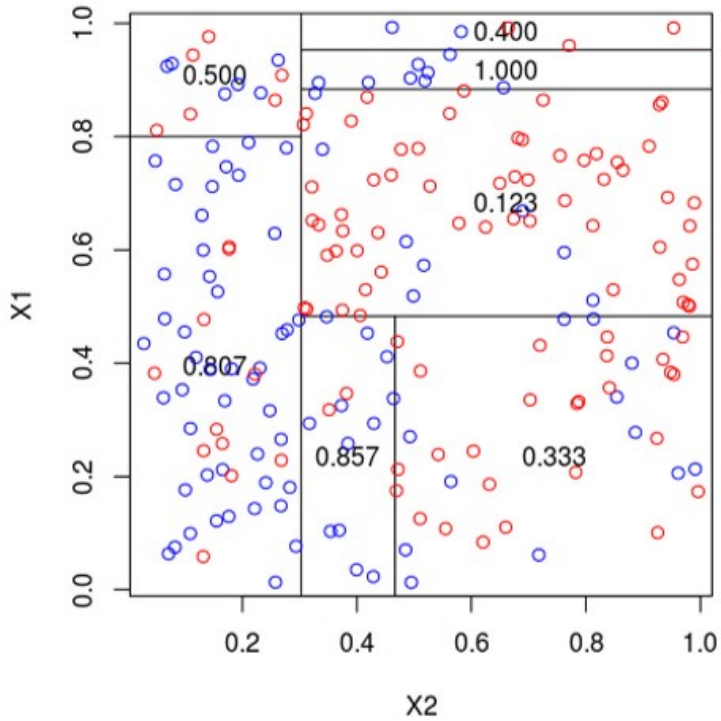
$$\arg \min_{d,c} \left[\sum_{i:\mathbf{x}_i \in R_1(d,c)} (y_i - \bar{y}_1)^2 + \sum_{i:\mathbf{x}_i \in R_2(d,c)} (y_i - \bar{y}_2)^2 \right],$$

  where $\bar{y}_1$ and $\bar{y}_2$ are the means of the response variables for the two partitions.
- For classification trees we minimize the Gini Index instead of the sum of squares: $G = \sum_{i=1}^{J} p_i(1 - p_i)$ where p_i is the proportion of correctly classified instances in class i.

Motivation

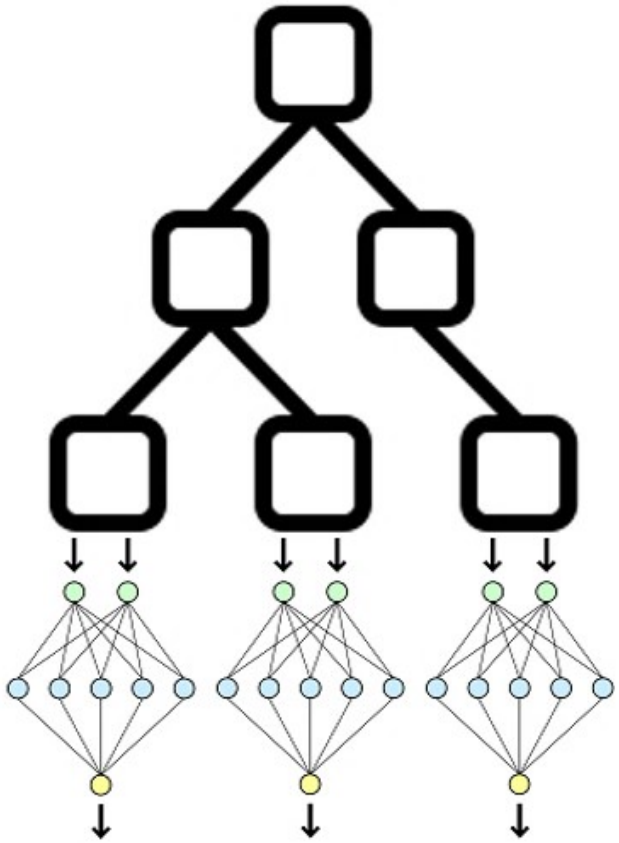
- After fitting the correct size tree, the response value is found by averaging the responses at the leaf node (regression tree) or by using the majority vote (classification tree).





Neural Shrubs

- Create a decision tree by partitioning the training space.
- Standard decision tree methodology is used which includes pruning.
- At each node, we train a neural network.



2019 SDSU Data Science Symposium

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY A LEGACY OF EXCELLENCE

Neural Shrubs

Advantages

- Improved accuracy versus a standard decision tree
- Shorter training time as compared to a standard neural network
- The tree exists to aid in interpretability

Disadvantages

- Large dataset is required so that leaf nodes have enough data to build neural network
- Neural shrubs have a lower classification rate than a standard neural network

2019 SDSU Data Science Symposium

Demonstration 1: MNIST

- Handwritten digits 0-9, grayscale pixels of handwritten images - strange for a classification tree. More suitable for neural network
- 60,000 training samples, 10,000 test samples, 780 attributes
- Accuracy: Decision Tree = 81.78%, Neural Shrub = 86.76%, Neural Network = 98.03%
- Training Time: Neural Shrub = 144.25s (Decision Tree) + 179.95s (Longest time for NN at node) = 324.2s
- Training Time: Neural Network = 351.9s

2019 SDSU
Data Science
Symposium

Demonstration 2: SensIT

- Sensory information used to determine vehicle classification
- 78,823 training samples, 19,705 test samples, 50 attributes
- Accuracy: Decision Tree = 66.73%, Neural Shrub = 72.53%, Neural Network = 73.85%
- Training Time: Neural Shrub = 33.09s (Decision Tree) + 46.88s (Longest time for NN at node) = 79.97s
- Training Time: Neural Network = 119.89s

2019 SDSU
Data Science
Symposium

Concluding Remarks/Open Questions

- How much training data is needed at each node?
- At what node accuracy would it not be necessary to train a neural network?
- Correct pruning? Should we under fit the tree slightly? Should we over fit it slightly?
- Topology. If things are ”topologically similar” we would have numerous leaf nodes which ”vote” for the same category. Can we then combine similar leaf nodes into one neural network?

2019 SDSU
Data Science
Symposium