CS 6140: Machine Learning **Spring 2018**
College of Computer and Information Science
Northeastern University
Lecture 1 **January, 8**
Instructor: Bilal Ahmed **Scribe: Bilal Ahmed & Virgil Pavlu**

Regression Trees

So far we have looked at decision trees within the context of classification. However, decision trees can also be used for modeling regression trees, in which case they are known as regression trees. Since the output/response variable is continuous for regression problems we cannot use node impurity (and consquently information gain) to guide the tree growing process. Since the goal of regression is to predict values that are as close as possible to the labels we will use the sum of squared errors as our error function.

1 Learning Regression Trees

Consider, that the data consists of tuples of the form (x_i, y_i), where $1 \leq i \leq N$, and $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$. Our goal is to find the splitting variables and the split points, such that with each split our error gets minimized. Finding the optimal tree, which corresponds to finding the optimal set of binary partitions of our input feature space is an NP-complete problem. Therefore we will formulate a greedy algorithm to grow the regression tree as we did for classification trees. For clarity, we will assume that all input features are continuous, and we are only modeling binary splits.

Lets say that for an internal node m, χ_m is the set of datapoints that reach the node. The value output by the node i.e., the prediction, if the node was to become a leaf is given by the average of the response variable y.

Estimate a predicted value per tree node

$$g_m = \frac{\sum_{t \in \chi_m} y_t}{|\chi_m|}$$

where $|\chi_m|$ is the total number of datapoints at node m.

The resulting sum of squared errors at the node is calculated as:

$$E_m = \sum_{t \in \chi_m} (y_t - g_m)^2$$

1.1 Choosing the next split:

Choose the split that realizes the maximum drop in error. Assume, that we are considering to split on feature X_k, with the resulting children nodes χ_{m1} and χ_{m2}. χ_{m1} would correspond to all instances such that $X_k \leq s$ and χ_{m2} contains all instances that have $X_k > s$, where s is the splitting

value. Then,

$$g_{mj} = \frac{\sum_{t \in \chi_{mj}} y_t}{|\chi_{mj}|}$$

$$E'_m(X_k) = \sum_j \frac{|\chi_{mj}|}{|\chi_m|} \sum_{t \in \chi_{mj}} (y_t - g_{mj})^2$$

where $j \in \{1, 2\}$

We shall choose X such that the drop in error i.e., $E_m - E'_m(X_k)$ is minimized.

To find the optimal splitting value we can utilize the strategy of finding the optimal binary split for continuous features i.e., look at the mid-pont of sorted feature values and select the splitting values that produces the maximum reduction in error.

Event though we have only looked at binary splitting and assumed that the input features are continuous, regression trees can also handle discrete features, in which case we can still opt for either binary or multiway splits.

1.2 Stopping Criterion

We can stop splitting a node if the drop in error is not significant. Even though this sounds like a good strategy, it is short sighted. Since we have a greedy strategy for growing the tree, a seemingly ineffective split might produce a very informative split belowo it.

The most effective strategy is to create a very large tree e.g., stop the splitting only when a minimum threshold on the number of instances (such as 5 or even 1) is reached. This tree will surely overfit to the training data, and we would need to prune it using a validation set. During the pruning stage, every pair of leaves with a common parent is evaluated using the validation data. The goal is to determine whether the sum of squared errors would be smaller by removing the two nodes and making their parent a leaf. This is repeated until pruning no longer improves the error on the validation set.

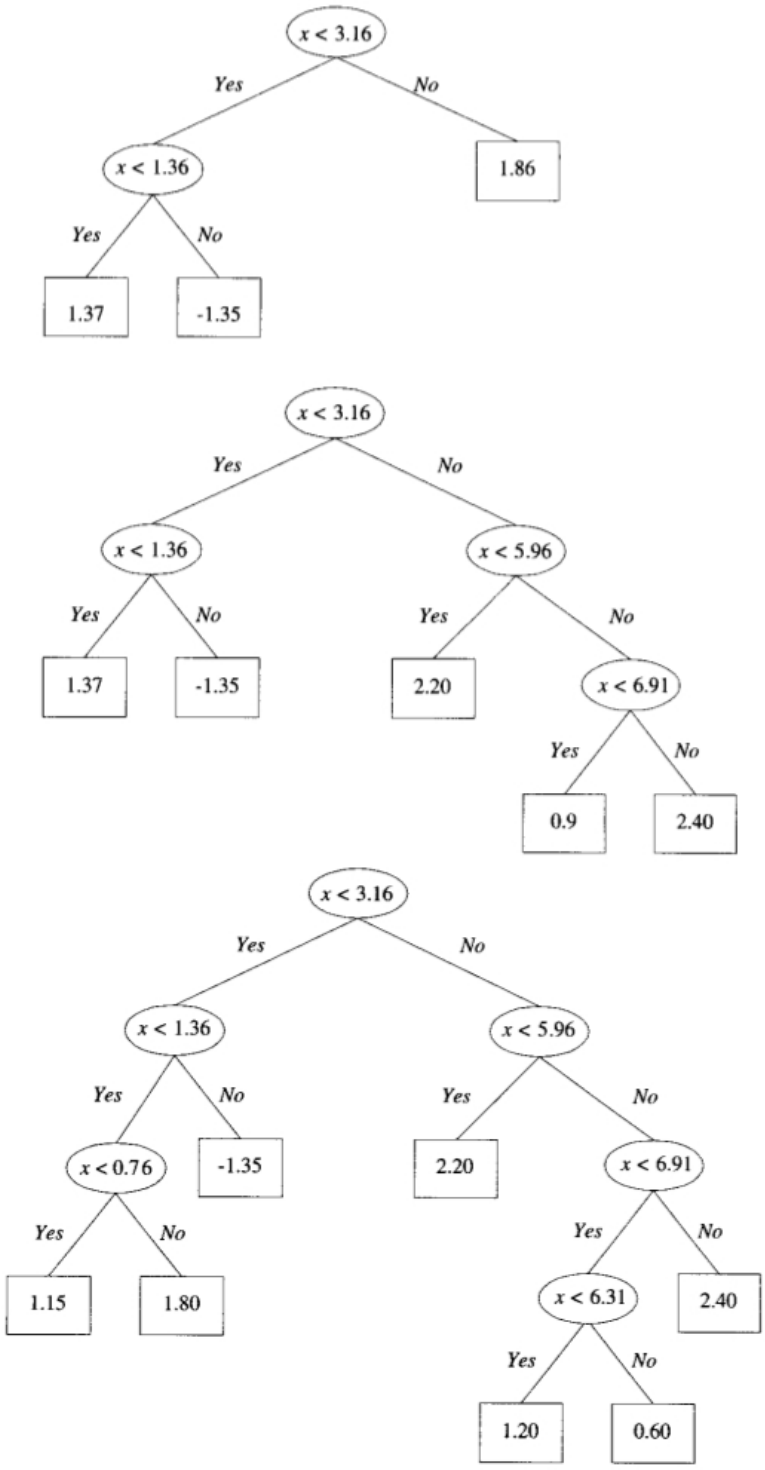

Figure 1: Regression tree

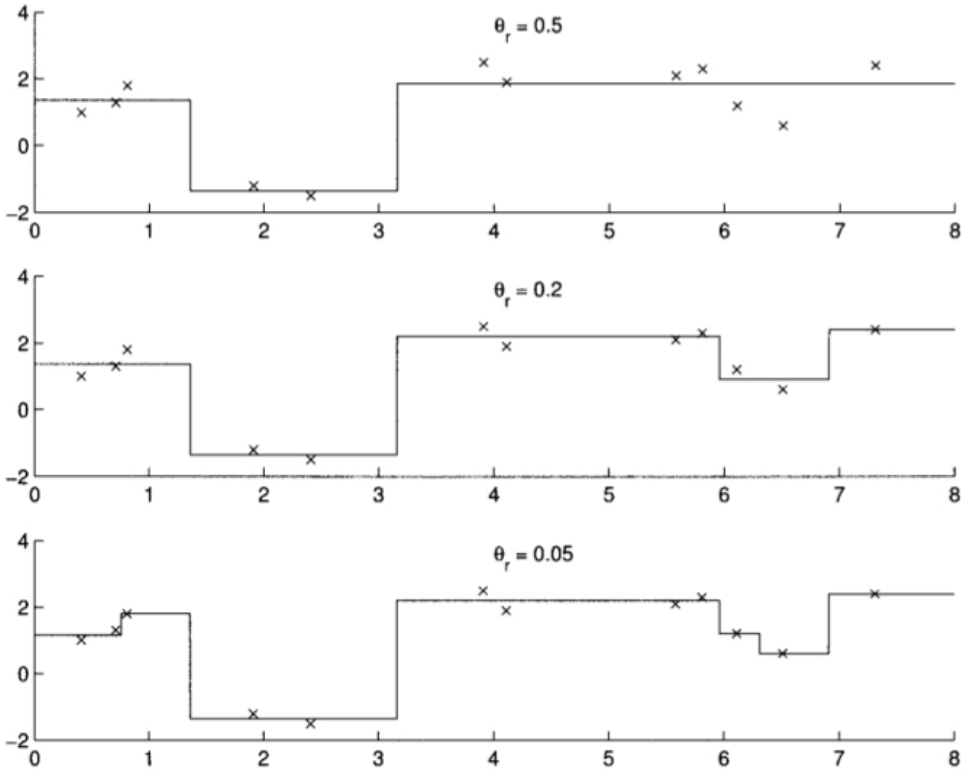

Figure 2: Regression fit