

MATH 829: Introduction to Data Mining and  
Analysis  
Decision trees

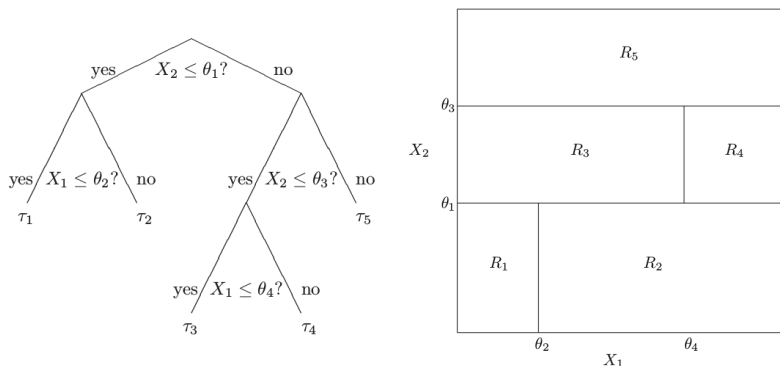
Dominique Guillot

Departments of Mathematical Sciences  
University of Delaware

April 6, 2016

## Tree-based methods:

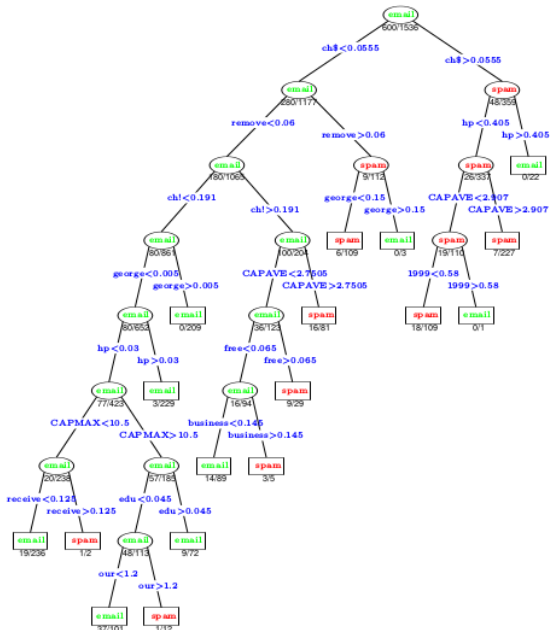
- Partition the feature space into a set of rectangles.
- Fit a simple model (e.g. a constant) in each rectangle.
- Conceptually simple yet powerful.



Izenman, 2013, Figure 9.1.

# Example: spam data

ESL, Figure 9.5.



## **Advantages:**

- Often mimics human decision-making process (e.g. doctor examining patient).
- Very easy to explain and interpret.
- Can handle both regression and classification problems.

## **Advantages:**

- Often mimics human decision-making process (e.g. doctor examining patient).
- Very easy to explain and interpret.
- Can handle both regression and classification problems.

## **Disadvantage:**

- Basic implementation is generally not competitive compared to other methods.

## Advantages:

- Often mimics human decision-making process (e.g. doctor examining patient).
- Very easy to explain and interpret.
- Can handle both regression and classification problems.

## Disadvantage:

- Basic implementation is generally not competitive compared to other methods.
- However, by **aggregating many decision trees** and using other variants, one can improve the performance significantly.
- Such techniques lead to state-of-the-art models.

## Advantages:

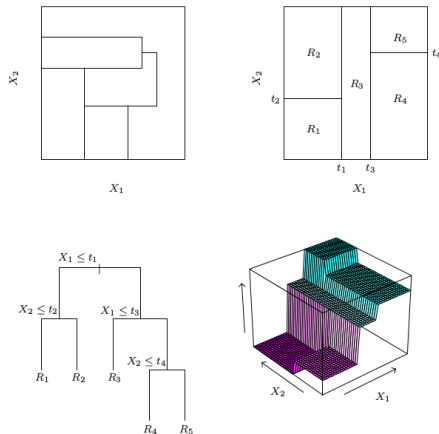
- Often mimics human decision-making process (e.g. doctor examining patient).
- Very easy to explain and interpret.
- Can handle both regression and classification problems.

## Disadvantage:

- Basic implementation is generally not competitive compared to other methods.
- However, by **aggregating many decision trees** and using other variants, one can improve the performance significantly.
- Such techniques lead to state-of-the-art models.
- However, in doing so, one loses the easy **interpretability** of decision trees.

# Binary decision trees

To simplify, we will only consider **binary** decision trees.



ESL, Figure 9.2.

Top Left: Not binary. Top Right: binary.

Bottom Left: Tree corresponding to Top Right partition. Bottom Right: Prediction surface.



# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

Suppose we have a partition of  $\mathbb{R}^p$  into  $M$  regions  $R_1, \dots, R_m$ .

# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

Suppose we have a partition of  $\mathbb{R}^p$  into  $M$  regions  $R_1, \dots, R_m$ .

We predict the response using a constant on each  $R_i$ :

$$f(x) = \sum_{i=1}^m c_i \cdot \mathbf{1}_{x \in R_i}.$$

# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

Suppose we have a partition of  $\mathbb{R}^p$  into  $M$  regions  $R_1, \dots, R_m$ .

We predict the response using a constant on each  $R_i$ :

$$f(x) = \sum_{i=1}^m c_i \cdot \mathbf{1}_{x \in R_i}.$$

In order to minimize  $\sum_{i=1}^n (y_i - f(x_i))^2$ , one needs to choose:

$$\hat{c}_i = \text{ave}(y_j : x_j \in R_i).$$

# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

Suppose we have a partition of  $\mathbb{R}^p$  into  $M$  regions  $R_1, \dots, R_m$ .

We predict the response using a constant on each  $R_i$ :

$$f(x) = \sum_{i=1}^m c_i \cdot \mathbf{1}_{x \in R_i}.$$

In order to minimize  $\sum_{i=1}^n (y_i - f(x_i))^2$ , one needs to choose:

$$\hat{c}_i = \text{ave}(y_j : x_j \in R_i).$$

How do we determine the regions  $R_i$ , i.e., how do we “grow” the tree?

# How to grow a decision tree?

## Regression tree:

- Data:  $y \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{n \times p}$ .
- Each observation:  $(y_i, x_i) \in \mathbb{R}^{p+1}$ ,  $i = 1, \dots, n$ .

Suppose we have a partition of  $\mathbb{R}^p$  into  $M$  regions  $R_1, \dots, R_m$ .

We predict the response using a constant on each  $R_i$ :

$$f(x) = \sum_{i=1}^m c_i \cdot \mathbf{1}_{x \in R_i}.$$

In order to minimize  $\sum_{i=1}^n (y_i - f(x_i))^2$ , one needs to choose:

$$\hat{c}_i = \text{ave}(y_j : x_j \in R_i).$$

How do we determine the regions  $R_i$ , i.e., how do we “grow” the tree?

We need to decide:

- 1 Which variable to split.
- 2 Where to split that variable.

# Growing a tree

- Finding a (globally) optimal tree is generally computationally infeasible.
- We use a greedy algorithm.

# Growing a tree

- Finding a (globally) optimal tree is generally computationally infeasible.
- We use a greedy algorithm.

Consider a splitting variable  $j \in \{1, \dots, p\}$  and splitting point  $s \in \mathbb{R}$ .



# Growing a tree

- Finding a (globally) optimal tree is generally computationally infeasible.
- We use a greedy algorithm.

Consider a splitting variable  $j \in \{1, \dots, p\}$  and splitting point  $s \in \mathbb{R}$ .

Define the two half-planes:

$$R_1(j, s) := \{x \in \mathbb{R}^p : x_j \leq s\}, \quad R_2(j, s) := \{x \in \mathbb{R}^p : x_j > s\}.$$

# Growing a tree

- Finding a (globally) optimal tree is generally computationally infeasible.
- We use a greedy algorithm.

Consider a splitting variable  $j \in \{1, \dots, p\}$  and splitting point  $s \in \mathbb{R}$ .

Define the two half-planes:

$$R_1(j, s) := \{x \in \mathbb{R}^p : x_j \leq s\}, \quad R_2(j, s) := \{x \in \mathbb{R}^p : x_j > s\}.$$

We choose  $j, s$  to minimize

$$\min_{j, s} \left[ \min_{c_1 \in \mathbb{R}} \sum_{x_i \in R_1(j, s)} (y_i - c_1)^2 + \min_{c_2 \in \mathbb{R}} \sum_{x_i \in R_2(j, s)} (y_i - c_2)^2 \right].$$

- The determination of the splitting point  $s$  can be done very quickly.

# Growing a tree

- Finding a (globally) optimal tree is generally computationally infeasible.
- We use a greedy algorithm.

Consider a splitting variable  $j \in \{1, \dots, p\}$  and splitting point  $s \in \mathbb{R}$ .

Define the two half-planes:

$$R_1(j, s) := \{x \in \mathbb{R}^p : x_j \leq s\}, \quad R_2(j, s) := \{x \in \mathbb{R}^p : x_j > s\}.$$

We choose  $j, s$  to minimize

$$\min_{j,s} \left[ \min_{c_1 \in \mathbb{R}} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2 \in \mathbb{R}} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right].$$

- The determination of the splitting point  $s$  can be done very quickly.
- Hence, determining the best pair  $(j, s)$  is **feasible**.

**Repeat the same process to each block.**

## Stopping rules and pruning

- Generally, the process is stopped for a given region when there are **less than** 5 observations in that region.

# Stopping rules and pruning

- Generally, the process is stopped for a given region when there are **less than** 5 observations in that region.

## **Problem with previous methodology:**

- Likely to **overfit** the data.
- Can lead to poor prediction error.

# Stopping rules and pruning

- Generally, the process is stopped for a given region when there are **less than** 5 observations in that region.

## **Problem with previous methodology:**

- Likely to **overfit** the data.
- Can lead to poor prediction error.

**Pruning the tree.** Strategy: Grow a large tree (overfits), and then prune it (better).

# Stopping rules and pruning

- Generally, the process is stopped for a given region when there are **less than 5** observations in that region.

## Problem with previous methodology:

- Likely to **overfit** the data.
- Can lead to poor prediction error.

**Pruning the tree.** Strategy: Grow a large tree (overfits), and then prune it (better).

### • **Weakest link pruning:**

(a.k.a cost complexity pruning)

Let  $T \subset T_0$  be a **subtree** of  $T_0$  with  $|T|$  **terminal nodes**. For  $\alpha > 0$ , define:

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha \cdot |T|.$$



# Stopping rules and pruning

- Generally, the process is stopped for a given region when there are **less than 5** observations in that region.

## Problem with previous methodology:

- Likely to **overfit** the data.
- Can lead to poor prediction error.

**Pruning the tree.** Strategy: Grow a large tree (overfits), and then prune it (better).

### • **Weakest link pruning:**

(a.k.a cost complexity pruning)

Let  $T \subset T_0$  be a **subtree** of  $T_0$  with  $|T|$  **terminal nodes**. For  $\alpha > 0$ , define:

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha \cdot |T|.$$



**Pick a subtree minimizing  $C_\alpha(T)$ .**



Pick a subtree  $T \subset T_0$  minimizing:

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha \cdot |T|.$$

(Here,  $\hat{y}_{R_m}$  = average response for observations in  $R_m$ .)

Pick a subtree  $T \subset T_0$  minimizing:

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha \cdot |T|.$$

(Here,  $\hat{y}_{R_m}$  = average response for observations in  $R_m$ .)

- $\alpha$  is a **tuning parameter**.
- Trade-off between fit of the model, and tree complexity.
- Choose  $\alpha$  using cross-validation.

Pick a subtree  $T \subset T_0$  minimizing:

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha \cdot |T|.$$

(Here,  $\hat{y}_{R_m}$  = average response for observations in  $R_m$ .)

- $\alpha$  is a **tuning parameter**.
- Trade-off between fit of the model, and tree complexity.
- Choose  $\alpha$  using cross-validation.

Once  $\alpha$  has been chosen by CV, use whole dataset to find the tree corresponding to that value.

- So far, we discussed **regression** trees (continuous output).

# Classification trees

- So far, we discussed **regression** trees (continuous output).
- We can easily modify the methodology to predict a *categorical* output.

# Classification trees

- So far, we discussed **regression** trees (continuous output).
- We can easily modify the methodology to predict a *categorical* output.
- We only need to modify our *splitting and pruning criteria*.

- So far, we discussed **regression** trees (continuous output).
  - We can easily modify the methodology to predict a *categorical* output.
  - We only need to modify our *splitting and pruning criteria*.
- For continuous variables, we picked a constant in each box  $R_i$  to minimize the sum of squares in that region:

$$\min_{c \in \mathbb{R}} \sum_{x_i \in R_i} (y_i - c)^2.$$

- So far, we discussed **regression** trees (continuous output).
  - We can easily modify the methodology to predict a *categorical* output.
  - We only need to modify our *splitting and pruning criteria*.
- For continuous variables, we picked a constant in each box  $R_i$  to minimize the sum of squares in that region:

$$\min_{c \in \mathbb{R}} \sum_{x_i \in R_i} (y_i - c)^2.$$

As a result, we choose:

$$\hat{c}_i = \frac{1}{N_i} \sum_{x_k \in R_i} y_k,$$

where  $N_i$  denotes the number of observations in  $R_i$ .



Similarly, when the output is categorical, we can count the proportion of class  $k$  observations in node  $i$ :

$$\hat{p}_{ik} = \frac{1}{N_i} \sum_{x_l \in R_i} \mathbf{1}_{y_l \in R_i}.$$

## Classification trees (cont.)

Similarly, when the output is categorical, we can count the proportion of class  $k$  observations in node  $i$ :

$$\hat{p}_{ik} = \frac{1}{N_i} \sum_{x_l \in R_i} \mathbf{1}_{y_l \in R_i}.$$

We then classify the observations in node  $i$  using a **majority vote**:

$$k(i) := \operatorname{argmax}_k \hat{p}_{ik}.$$

Similarly, when the output is categorical, we can count the proportion of class  $k$  observations in node  $i$ :

$$\hat{p}_{ik} = \frac{1}{N_i} \sum_{x_l \in R_i} \mathbf{1}_{y_l \in R_i}.$$

We then classify the observations in node  $i$  using a **majority vote**:

$$k(i) := \operatorname{argmax}_k \hat{p}_{ik}.$$

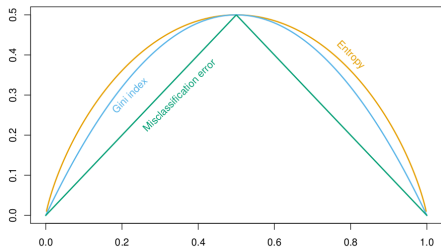
Different measures are commonly used to determine how good a given partition is (and how to split a given partition):

- 1 **Misclassification error:**  $\frac{1}{N_i} \sum_{x_l \in R_i} \mathbf{1}_{y_l \neq k(i)} = 1 - \hat{p}_{i,k(i)}$ .
- 2 **Gini index:**  $\sum_{k \neq k'} \hat{p}_{ik} \hat{p}_{ik'} = \sum_{k=1}^K \hat{p}_{ik} (1 - \hat{p}_{ik})$ .
- 3 **Cross-entropy (or deviance):**  $-\sum_{k=1}^K \hat{p}_{ik} \log \hat{p}_{ik}$ .

# Classification trees (cont.)

With two classes and a proportion of  $0 < p < 1$  observations in the second class, we have (exercise):

Measure	Value
Misclassification error	$1 - \max(p, 1 - p)$
Gini index	$2p(1 - p)$
Cross-entropy	$-p \log p - (1 - p) \log(1 - p)$



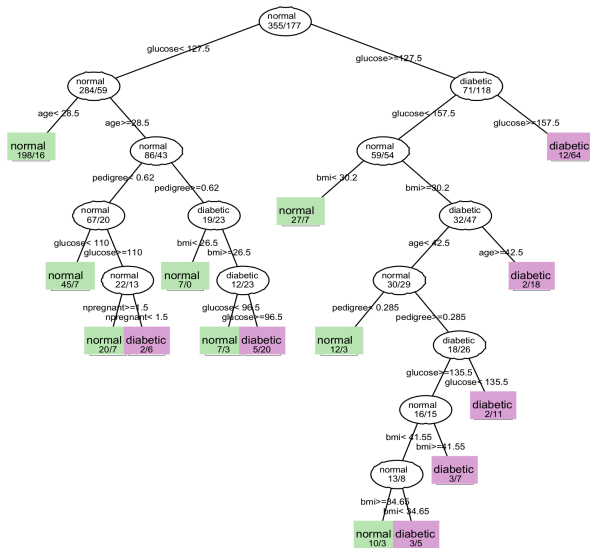
ESL, Figure 9.3.

# Case study: Pima Indians Diabetes (Izenman, 2013)

- Pima Indian (nativa American) population lives near Phoenix, Arizona.
- The diversion of the water and the introduction of non-native diet had devastating effects on the health of the people. They have the highest prevalence of type 2 diabetes in the world, much more than is observed in other U.S. populations. They have been the subject of intensive study of diabetes. <sup>1</sup>
- Patients listed in the dataset are females at least 21 years old of Pima Indian heritage.
- 8 input variables (e.g. number of times pregnant, body mass index, plasma glucose concentration, etc.).

<sup>1</sup>Wikipedia

# Case study (cont.)



Classification tree for the Pima Indians diabetes data. Impurity measure = Gini index. (Izenman, Figure 9.5.)