Learning goals:

- Understand the benefits and drawbacks of decision trees compared to the models we've seen so far.
- Learn the algorithm for fitting a decision tree.
- Develop intuition for entropy.

Decision Trees

UC Berkeley Data 100 Summer 2019
Sam Lau

Break!

Fill out Attendance:

http://bit.ly/at-d100

Announcements

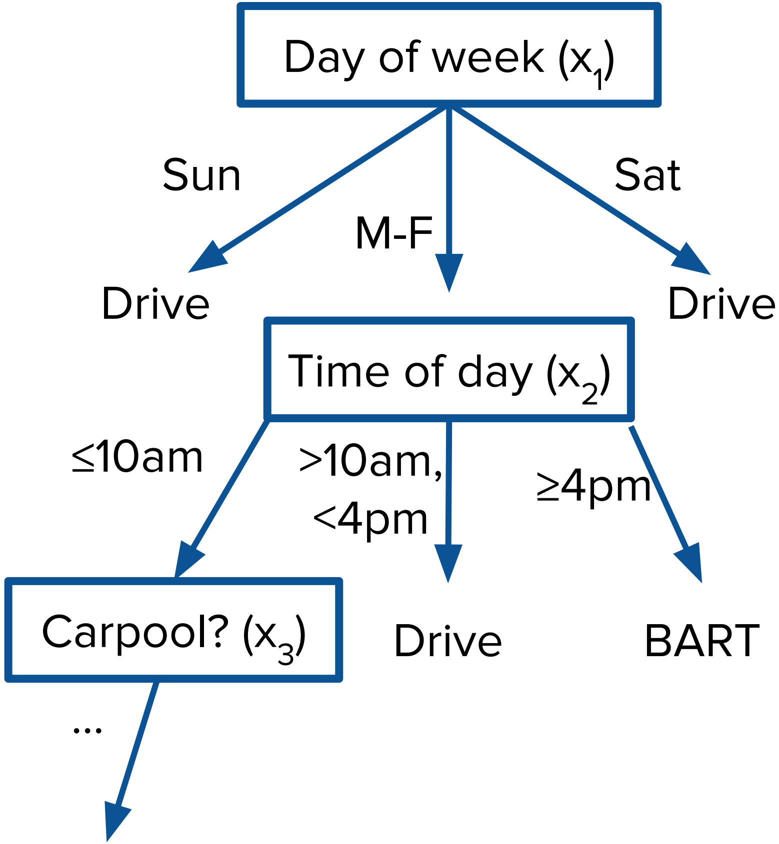
- Project 2 due **today!**
- Project 3 out today, due **Tues**
- Final next Thurs, Aug 15 9:30am-12:30pm in 10 Evans.
  - Can bring **two** handwritten double-sided cheat sheets.
  - Bring **pencil and eraser** (not pen).
  - Will provide midterm reference sheet again.

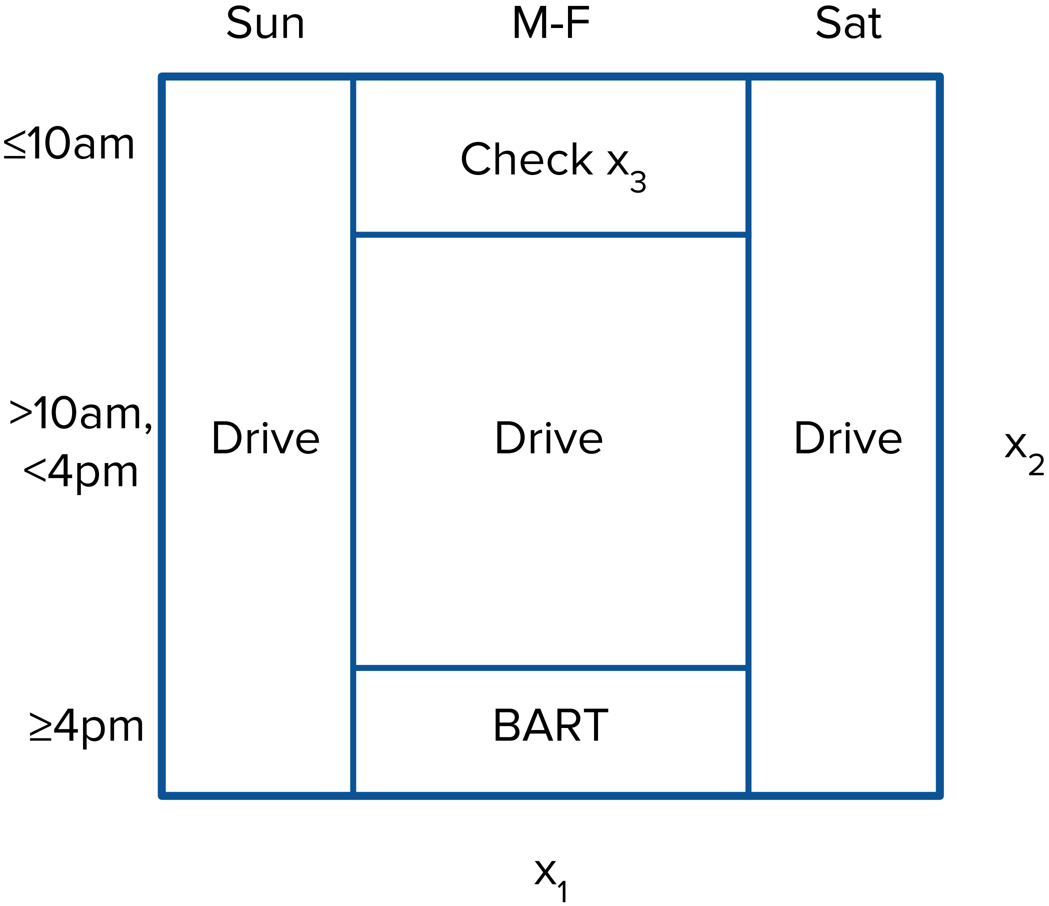
Decision Trees

Decision Trees

- **Decision trees** are a useful nonlinear algorithm used for both classification and regression.
- Intuition: Learn a set of binary rules rather than a set of coefficients for a linear model.
- E.g: Is it faster to drive to SF? Or take the BART?
  - If weekend, drive.
  - If ≤10 am, take BART.
  - If I can use carpool lane, drive.
  - Etc.

Decision Trees, Visualized





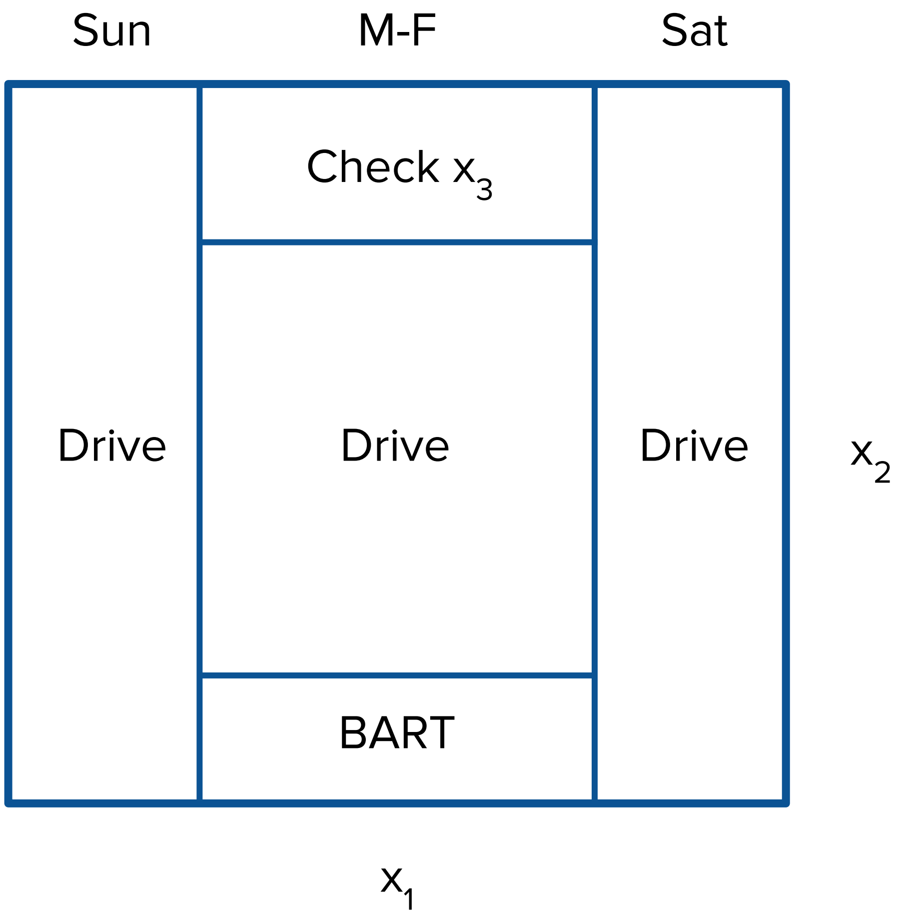
Decision Tree Traits

Works with both numeric and categorical data w/o extra work.

Easier to **interpret** compared to linear/logistic model.

Fits complex, nonlinear boundaries w/o feature eng.

Can use for both regression and (multiclass) classification.

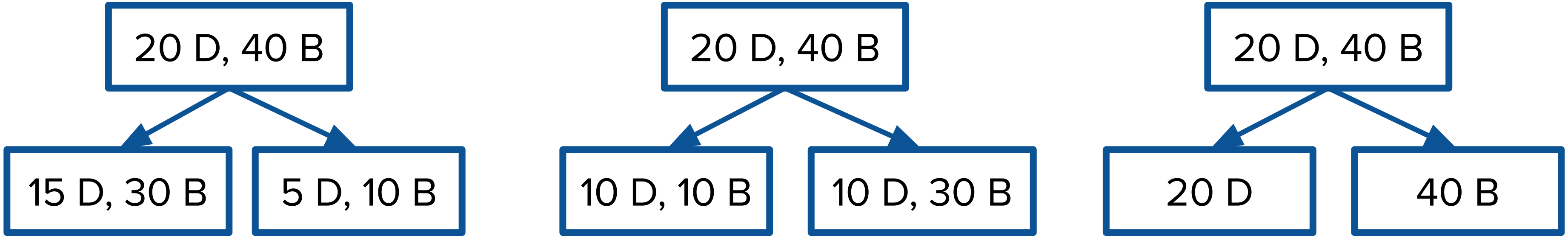


Building a Decision Tree

- Every branch in decision tree splits training data into nodes. A node is **pure** if all points have same labels.
- Start with all training data in one node. Repeat the following until all nodes are pure:
  - Pick best feature j and best split value θ.
    - E.g. j=2 for Hour of Day and θ = 10 (for 10am)
  - Split data into two nodes (one for $x_j < \theta$, one for $x_j \geq \theta$).

How do we find the best split?

- Intuition: Good splits find features that get closer to a partition of the training labels.



Worst split **Good split** **Ideal split**

- As usual, we will define a loss function to minimize.

Entropy

- **Entropy** measures how "disorderly" a node is. Better nodes have low entropy.

$$S(\text{node}) = -\sum_C p_C \log_2 p_C \qquad \text{where } C \text{ is one of possible labels}$$

$$p_C = \text{proportion of points in node with class } C.$$

- IOW: Suppose we pick a random point from a node. Low entropy means we are quite sure what that point will be.

Cross-entropy and entropy are closely related. Both come from information theory, a useful branch of math that examines the resolution of uncertainty.

Practice with Entropy

$$S(\text{node}) = -\sum_C p_C \log_2 p_C \qquad \text{where } C \text{ is one of possible labels}$$

$$p_C = \text{proportion of points in node with class } C.$$

- Check that:
  - For a pure node, S = 0.
  - For a node with an equal number of two labels, S = 1.
  - For a node with k points and k labels, S = $\log_2 k$.
- S = $-(1)(\log_2 1) = 0$.
- S = $-(0.5)(\log_2 0.5) - (0.5)(\log_2 0.5) = 1$.
- S = $-(1/k)(\log_2 1/k) * k = -\log_2 1/k = \log_2 k$.

Loss of a Split

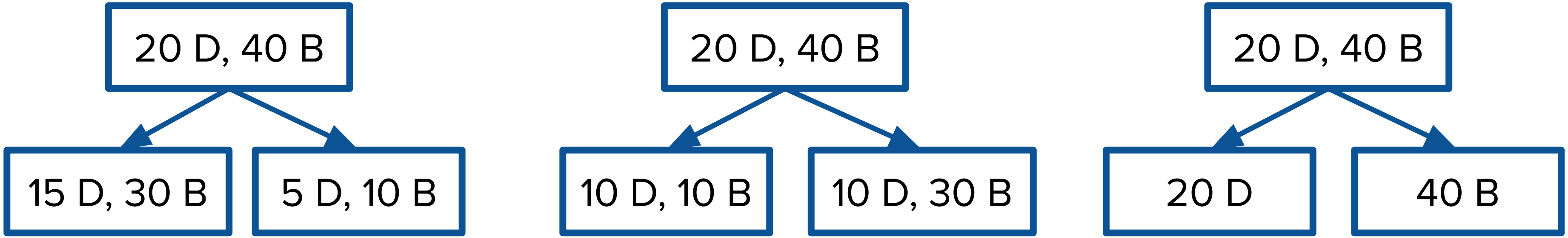
- Parent node N has K points.
- Child nodes N_1 and N_2 have k_1 and k_2 points ($k_1 + k_2 = K$).
- Loss of split = split entropy

  = weighted average entropy of N_1 and N_2:

$$L_{split} = \frac{k_1 S(N_1) + k_2 S(N_2)}{K}$$

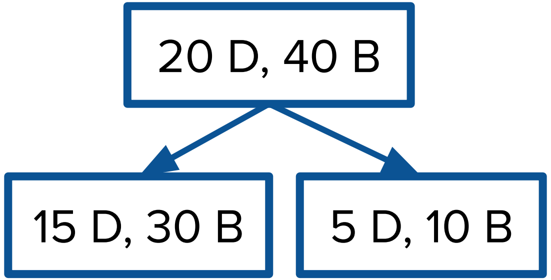
- IOW: Every time we grow tree, compute all possible splits. Then, pick the one that gives the least average entropy.

Practice with Loss of Split



- Find the loss for each of the splits above.
- Then find the information gain: S(N) - entropy of split.

Practice with Loss of Split

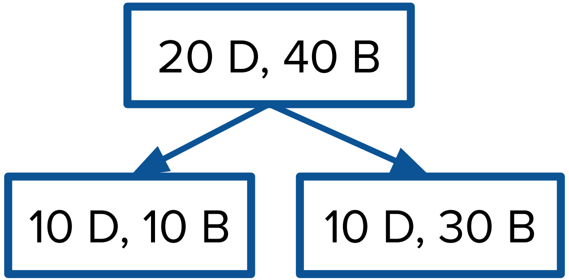


$$S(N_1) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.918$$

$$S(N_2) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.918$$

$$L = (45 * 0.918 + 15 * 0.918) / 60 = 0.918$$

Practice with Loss of Split

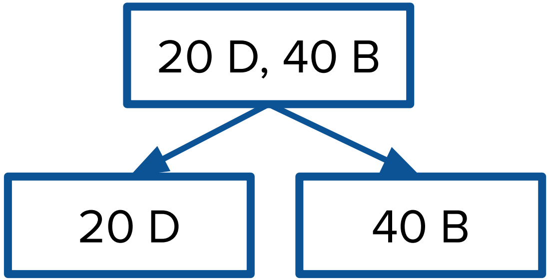


$$S(N_1) = 1$$

$$S(N_2) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = 0.811$$

$$L = (20 * 1 + 40 * 0.811) / 60 = 0.874$$

Practice with Loss of Split

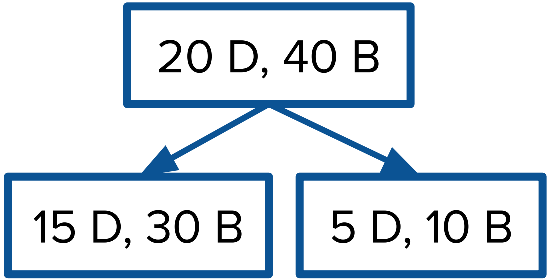


$$S(N_1) = 0$$

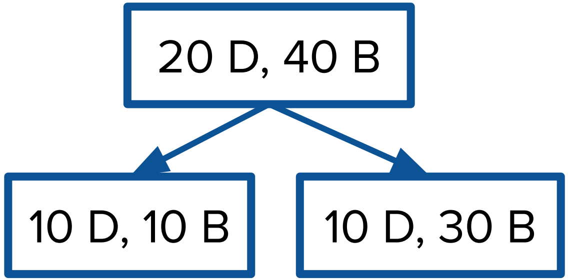
$$S(N_2) = 0$$

$$L = 0$$

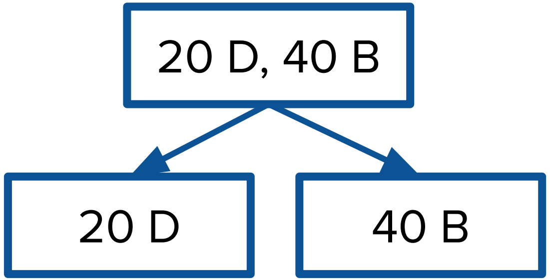
Practice with Loss of Split



`L = 0.918`
`Info gain = 0`



`L = 0.874`
`Info gain = 0.044`



`L = 0`
`Info gain = 0.918`

Intuition check: Why can't we always pick the rightmost split?

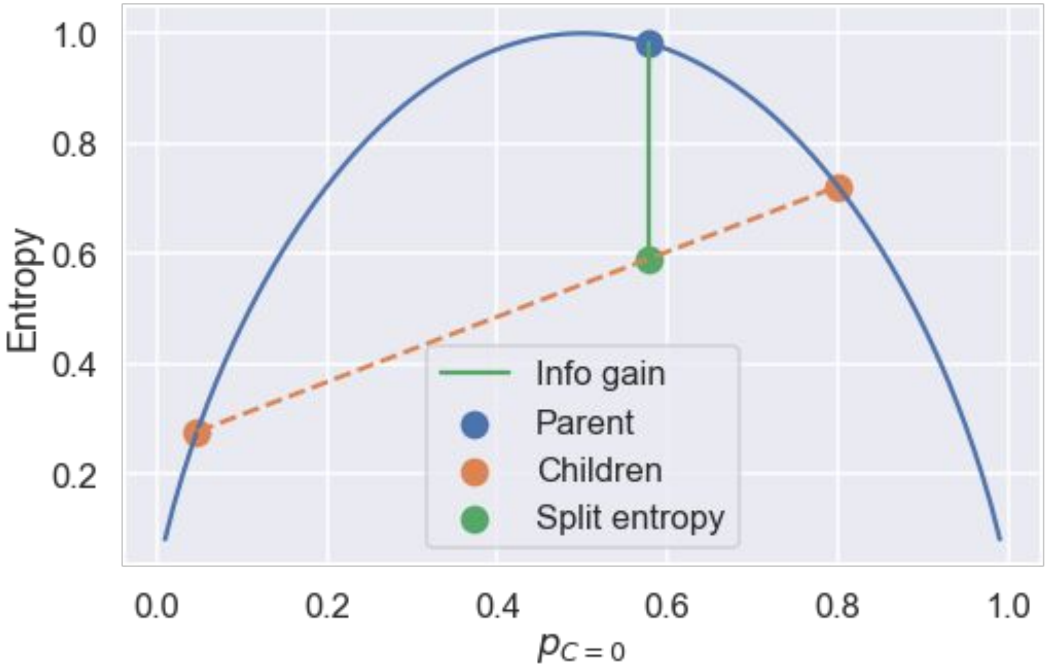
(Demo)

Split Entropy Traits

Info gain for split entropy is always +ve except in two cases:

1. Split puts all points in one child node.
2. Child nodes have same p_C as parent for all C.

This is true for **strictly concave** loss functions.



Good questions to ask your TA: Why won't we ever gain entropy via a split? What does the dotted line represent in the plot? Are there non-strictly concave loss functions?

Fitting Decision Tree

- Start with all training data in one node. Repeat the following until all nodes are pure:
  - Pick an impure node.
  - Find feature j and value θ that minimize loss of split.
  - Split into two child nodes (one for $x_j < \theta$, one for $x_j \geq \theta$).

(Demo)

Decision Problems

A decision tree will always have 100% training accuracy. Why?

Sadly, this means decision trees grossly overfit.

Can approach by: enforcing a max tree depth, pruning tree, don't split if node is too small, etc.

Or, do some clever bootstrapping! We'll save that for tomorrow.