Announcements

Coronavirus – COVID-19

- Take care of yourself and others around you
- Follow CMU and government guidelines
- We're "here" to help in any capacity that we can
- Use tools like zoom to communicate with each other too!

Announcements

Assignments

- HW6 (online)
  - Due Thu 3/26, 10 pm

Final Exam

- Format TBD

Announcements

Office Hours

- Zoom + OHQueue
- See piazza for details

Recitation

- Zoom session during normal recitation time slot
- See piazza for details

Zoom

- Let us know if you have issues
- Recommend turning on video when talking (mute when not talking)

Announcements

Lecture

- Recorded ahead of time and posted on Canvas
- Encouraged to watch during lecture time slot
- Zoom session during lecture time slot to answer any questions (optional)

"Participation" Points

- Polls open until 10 pm (EDT) day of lecture
- "Calamity" option announced in recorded lecture
  - Don't select this calamity option or you'll lose credit for one poll (-1) rather than gaining credit for one poll (+1).
- Participation percent calculated as usual

Introduction to Machine Learning

Decision Trees

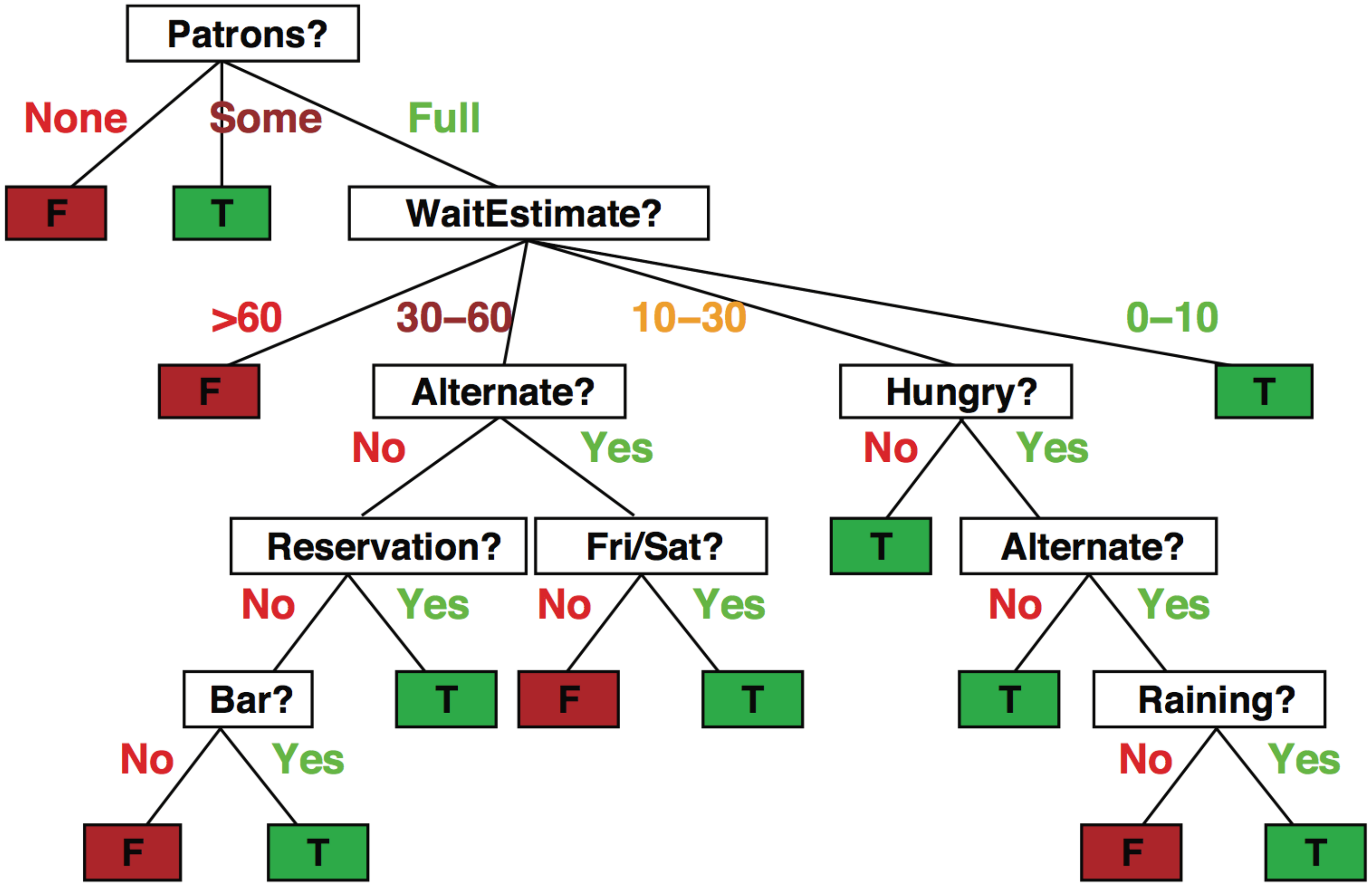
Instructor: Pat Virtue

Decision trees

Popular representation for classifiers

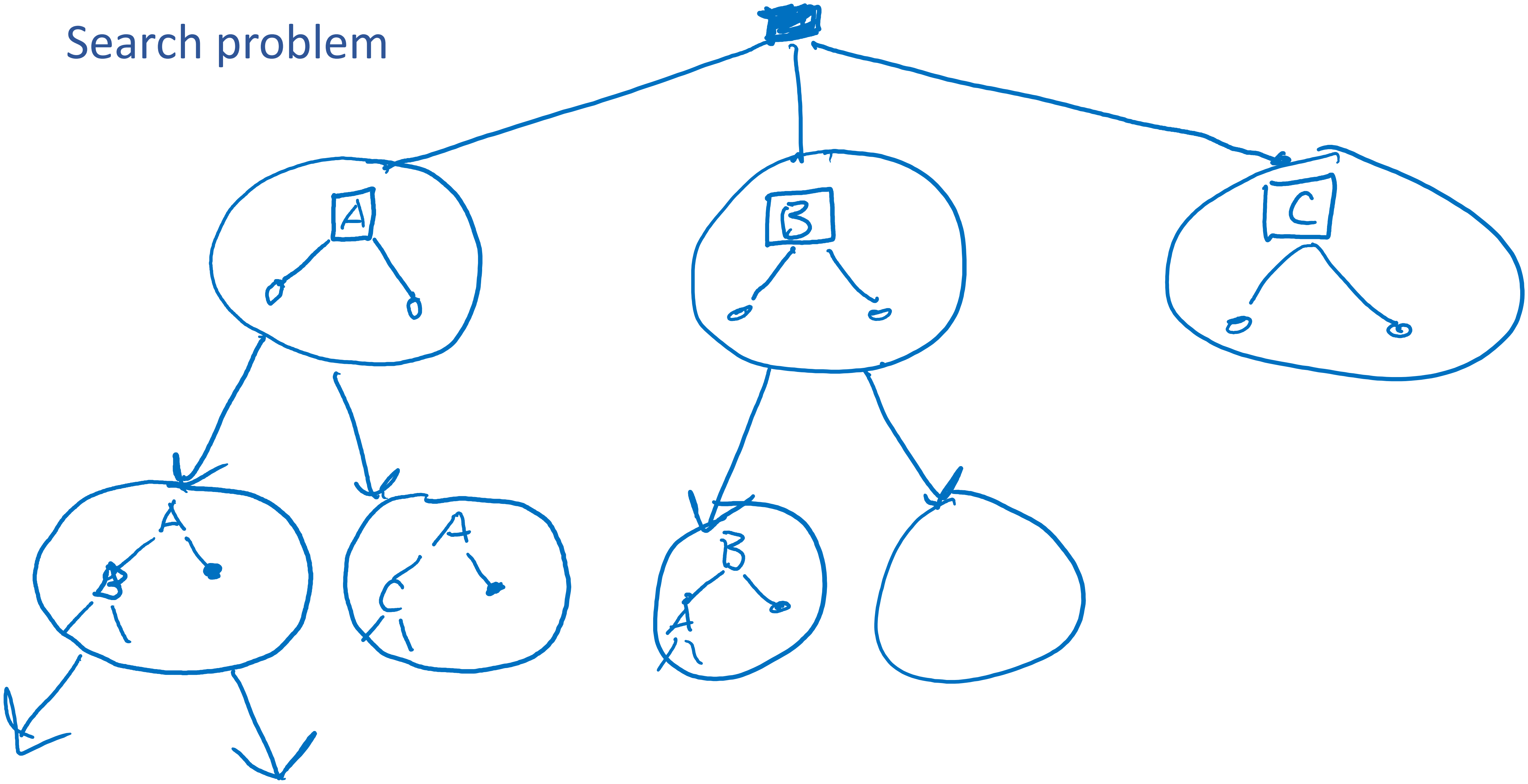
- Even among humans!

I’ve just arrived at a restaurant: should I stay (and wait for a table) or go elsewhere?



Build a decision tree

Search problem



Building a decision tree

```
Function BuildTree(n,A)    // n: samples, A: set of attributes
   If empty(A) or all n(L) are the same
       status = leaf
       class = most common class in n(L)
   else
       status = internal
       a ⇐ bestAttribute(n,A)
       LeftNode = BuildTree(n(a=1), A \ {a})
       RightNode = BuildTree(n(a=0), A \ {a})
    end
end
```

n(L): Labels for samples in this set

Decision: Which attribute?

Recursive calls to create left and right subtrees, n(a=1) is the set of samples in n for which the attribute a is 1

Identifying ‘bestAttribute’

There are many possible ways to select the best attribute for a given set.

- We started with using error rate to select the best attribute
- We will discuss one possible way which is based on information theory.

Previous Lecture Poll 4

Which attribute {A, B} would error rate select for the next split?

1) A
2) B
3) A or B (tie)
4) I don't know

Dataset:

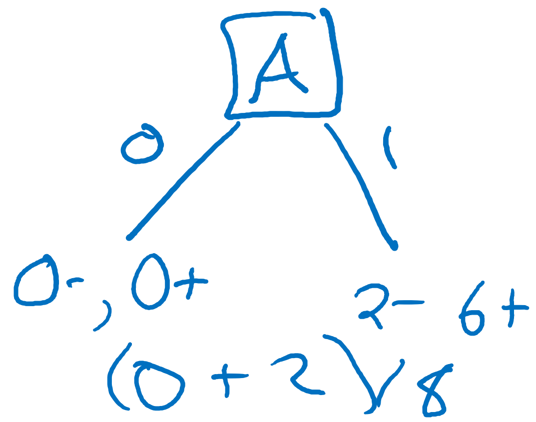
Output Y, Attributes A and B

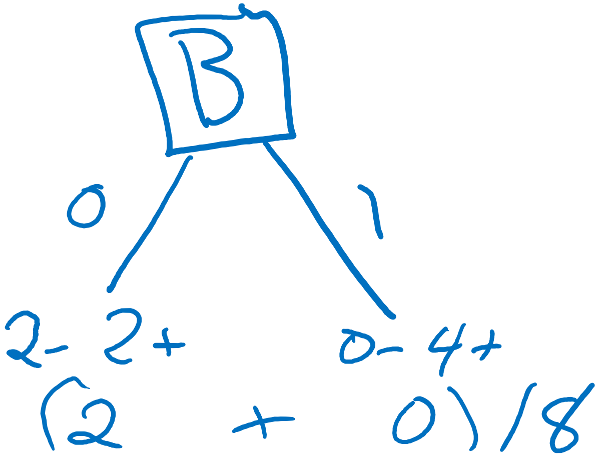
Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Previous Lecture Poll 4

Which attribute {A, B} would error rate select for the next split?

1) A
2) B
3) A or B (tie)
4) I don't know





Dataset:
Output Y, Attributes A and B

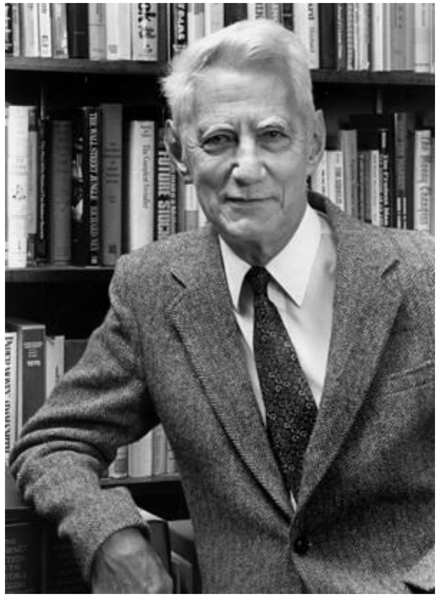
Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Entropy

- Quantifies the amount of uncertainty associated with a specific probability distribution
- The higher the entropy, the less confident we are in the outcome
- Definition

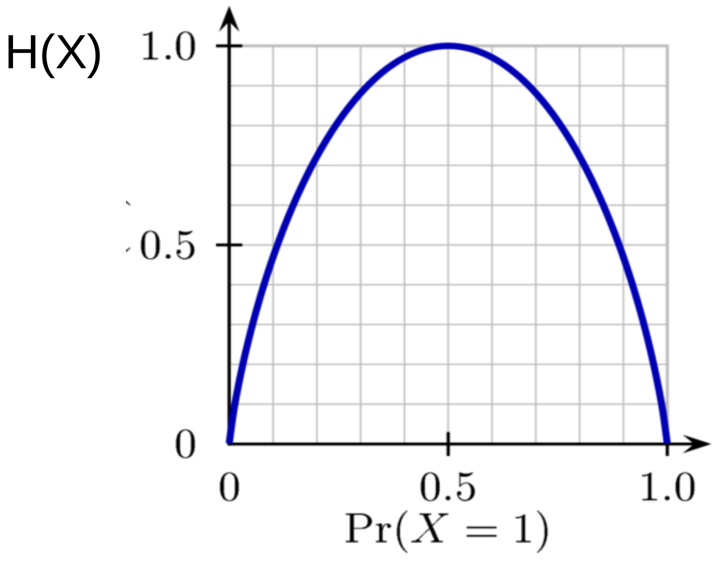
$$H(X) = \sum_x p(X = x) \log_2 \frac{1}{p(X = x)}$$

$$H(X) = -\sum_x p(X = x) \log_2 p(X = x)$$



Claude Shannon (1916 – 2001), most of the work was done in Bell labs

Entropy



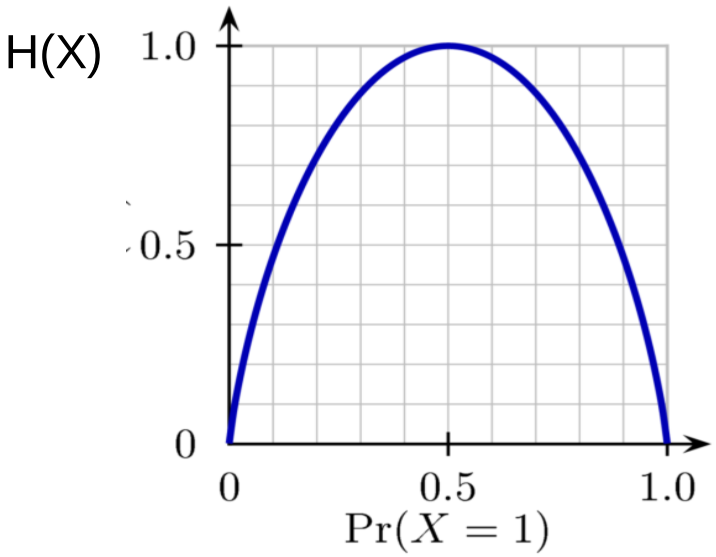
- Definition

$$H(X) = -\sum_{x} p(X = x) \log_2 p(X = x)$$

- So, if P(X=1) = 1 then

- If P(X=1) = .5 then

Entropy



- Definition

$$H(X) = -\sum_{x} p(X = x) \log_2 p(X = x)$$

- So, if P(X=1) = 1 then

$$H(X) = -p(x=1)\log_2 p(X=1) - p(x=0)\log_2 p(X=0)$$
$$= -1\log 1 - 0\log 0 = 0$$

- If P(X=1) = .5 then

$$H(X) = -p(x=1)\log_2 p(X=1) - p(x=0)\log_2 p(X=0)$$
$$= -.5\log_2 .5 - .5\log_2 .5 = -\log_2 .5 = 1$$

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

- For a decision tree, we can use **mutual information** of the output class *Y* and some attribute *X* on which to split **as a splitting criterion**
- Given a dataset *D* of training examples, we can estimate the required probabilities as…

$$P(Y = y) = N_{Y=y}/N$$
$$P(X = x) = N_{X=x}/N$$
$$P(Y = y|X = x) = N_{Y=y,X=x}/N_{X=x}$$

where $N_{Y=y}$ is the number of examples for which $Y = y$ and so on.

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

- **Entropy** measures the **expected # of bits** to code one random draw from X**.**
- For a decision tree, we want to **reduce the entropy of the random variable we are trying to predict**!

Conditional entropy is the expected value of specific conditional entropy $E_{P(X=x)}[H(Y \mid X = x)]$

Informally, we say that **mutual information** is a measure of the following:
If we know X, how much does this reduce our uncertainty about Y?

Piazza Poll 1

Which attribute {A, B} would **mutual information** select for the next split?

1) A
2) B
3) A or B (tie)
4) I don't know

Dataset:
Output Y, Attributes A and B

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y=y) \log_2 P(Y=y)$

Specific Conditional Entropy: $H(Y \mid X=x) = -\sum_{y \in \mathcal{Y}} P(Y=y \mid X=x) \log_2 P(Y=y \mid X=x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X=x) H(Y \mid X=x)$

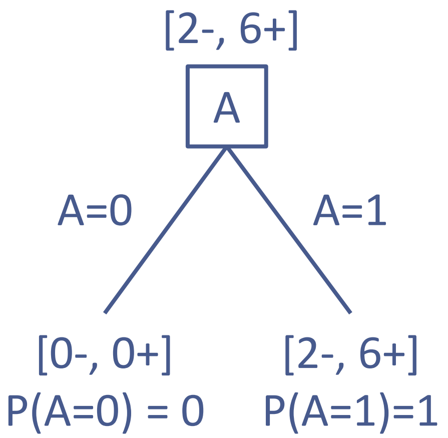
Mutual Information: $I(Y;X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

$$H(Y) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right]$$

$$H(Y \mid A=0) = undefined$$
$$H(Y \mid A=1) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right] = H(Y)$$

$$H(Y \mid A) = P(A=0)H(Y \mid A=0) + P(A=1)H(Y \mid A=1)$$
$$= 0 + H(Y \mid A=1)$$
$$= H(Y)$$
$$I(Y;A) = H(Y) - H(Y \mid A) = 0$$



Slide credit: CMU MLD Matt Gormley

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y=y) \log_2 P(Y=y)$

Specific Conditional Entropy: $H(Y \mid X=x) = -\sum_{y \in \mathcal{Y}} P(Y=y \mid X=x) \log_2 P(Y=y \mid X=x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X=x) H(Y \mid X=x)$

Mutual Information: $I(Y;X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

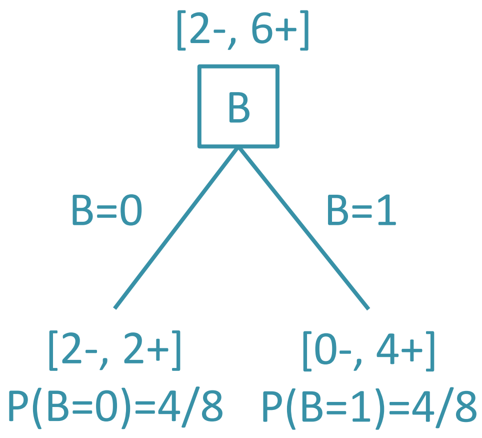
$$H(Y) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right]$$

$$H(Y \mid B=0) = -\left[\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right]$$
$$H(Y \mid B=1) = -[0\log_2 0 + 1\log_2 1] = 0$$

$$H(Y \mid B) = P(B=0)H(Y \mid B=0) + P(B=1)H(Y \mid B=1)$$
$$= \frac{4}{8}H(Y \mid B=0) + \frac{4}{8} \cdot 0$$

$$I(Y;B) = H(Y) - H(Y \mid B) > 0$$

$I(Y;B)$ ends up being greater than $I(Y;A) = 0$, so we split on B



Slide credit: CMU MLD Matt Gormley

How to learn a decision tree

- Top-down induction [ID3]

Main loop:

1. $X \leftarrow$ the "best" decision feature for next *node*
2. Assign X as decision feature for *node*
3. For each value of X, create new descendant of *node* (Discrete features)
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes (steps 1-5) after removing current feature

6. When all features exhausted, assign majority label to the leaf node

How to learn a decision tree

- Top-down induction [ID3, C4.5, C5, …]

Main loop: C4.5

1. $X \leftarrow$ the "best" decision feature for next *node*
2. Assign X as decision feature for *node*
3. For "best" split of X, create new descendants of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes
6. Prune back tree to reduce overfitting
7. Assign majority label to the leaf node

Handling continuous features (C4.5)

Convert continuous features into discrete by setting a threshold.

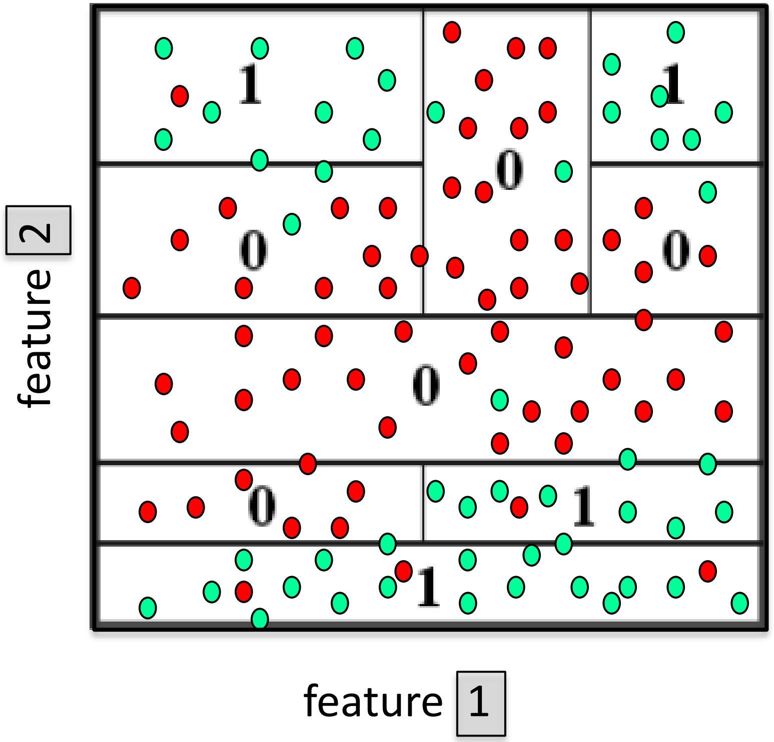
What threshold to pick?

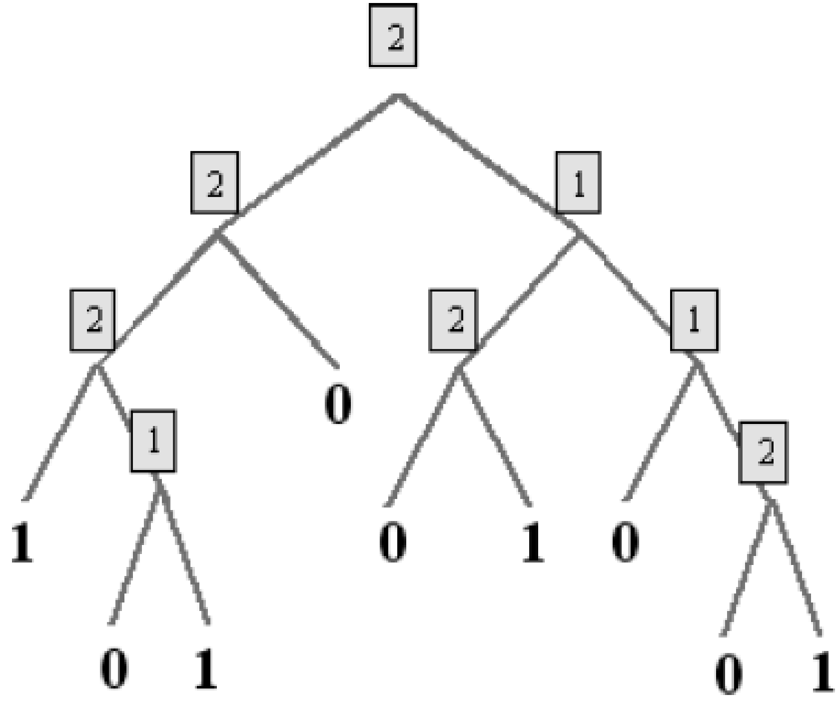
Search for best one as per information gain. Infinitely many??

Don’t need to search over more than ~ n (number of training data),e.g. say X_1 takes values $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}$ in the training set. Then possible thresholds are

$$[x_1^{(1)} + x_1^{(2)}]/2, [x_1^{(2)} + x_1^{(3)}]/2, \dots, [x_1^{(n-1)} + x_1^{(n)}]/2$$

Dyadic decision trees
(split on mid-points of features)





When to Stop?

- Many strategies for picking simpler trees:
  - Pre-pruning
    - Fixed depth (e.g. ID3)
    - Fixed number of leaves
  - Post-pruning
  - Penalize complexity of tree

Penalize Complexity of Tree

- Penalize complex models by introducing cost

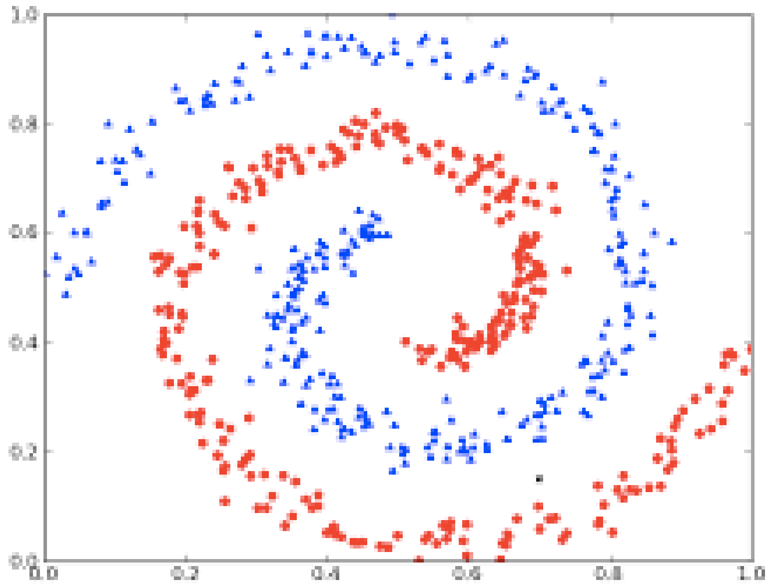
$$\hat{f} = \arg\min_T \left\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} \text{loss}(\hat{f}_T(X_i), Y_i)}_{\text{log likelihood}} + \underbrace{\text{pen}(T)}_{\text{cost}} \right\}$$

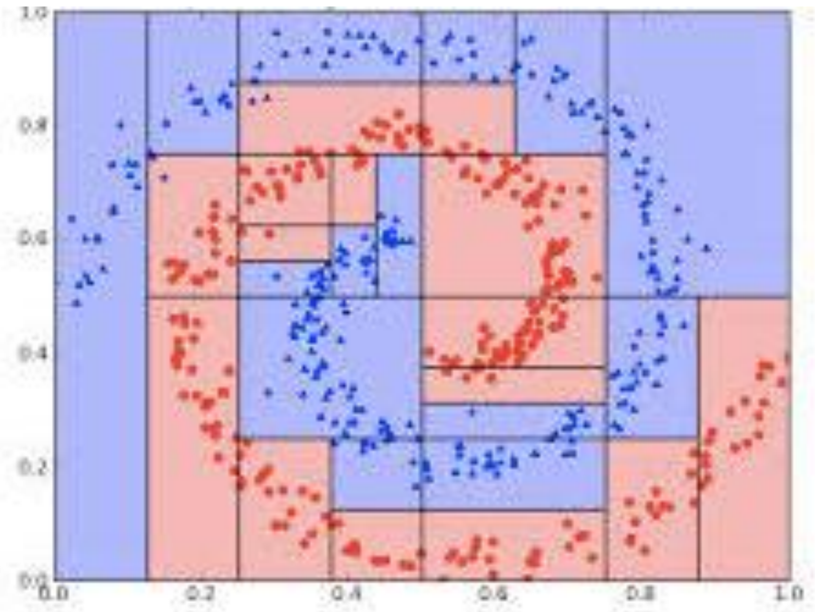
$$\begin{aligned} \text{loss}(\hat{f}_T(X_i), Y_i) &= (\hat{f}_T(X_i) - Y_i)^2 && \text{regression} \\ &= \mathbf{1}_{\hat{f}_T(X_i) \neq Y_i} && \text{classification} \end{aligned}$$

$\text{pen}(T) \propto |T|$ penalize trees with more leaves

CART – optimization can be solved by dynamic programming

Example of 2-feature decision tree classifier



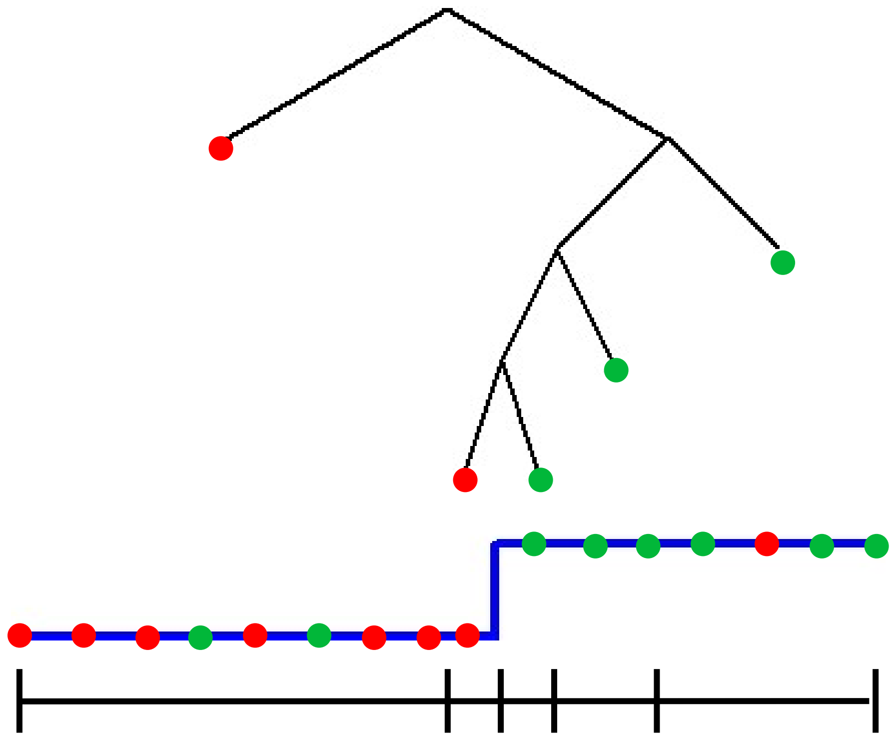


cs.uchicago.edu

How to assign label to each leaf

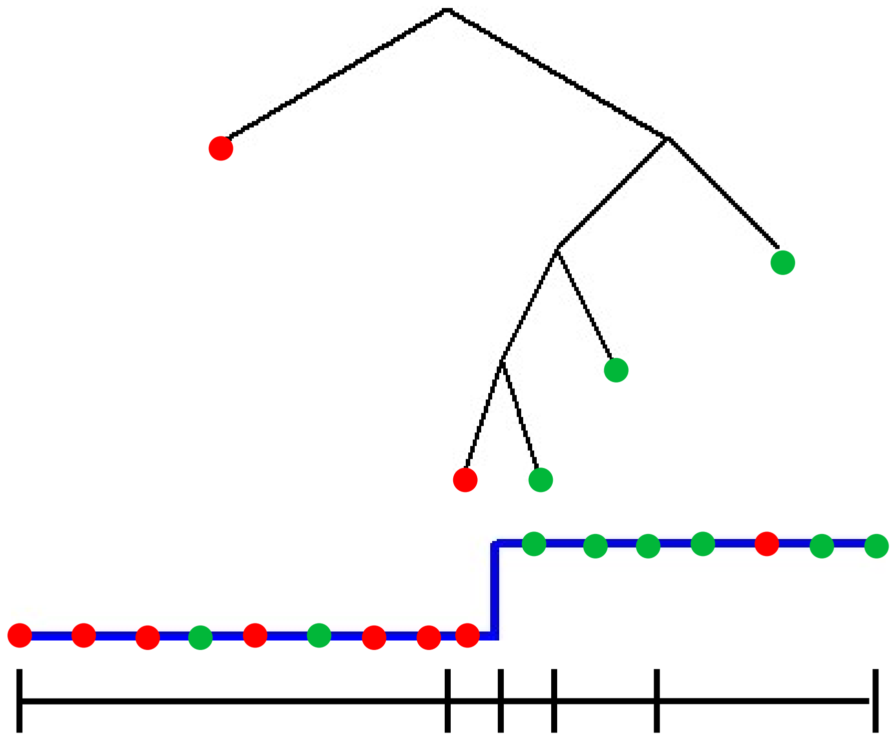
Classification – Majority vote

Regression – ?

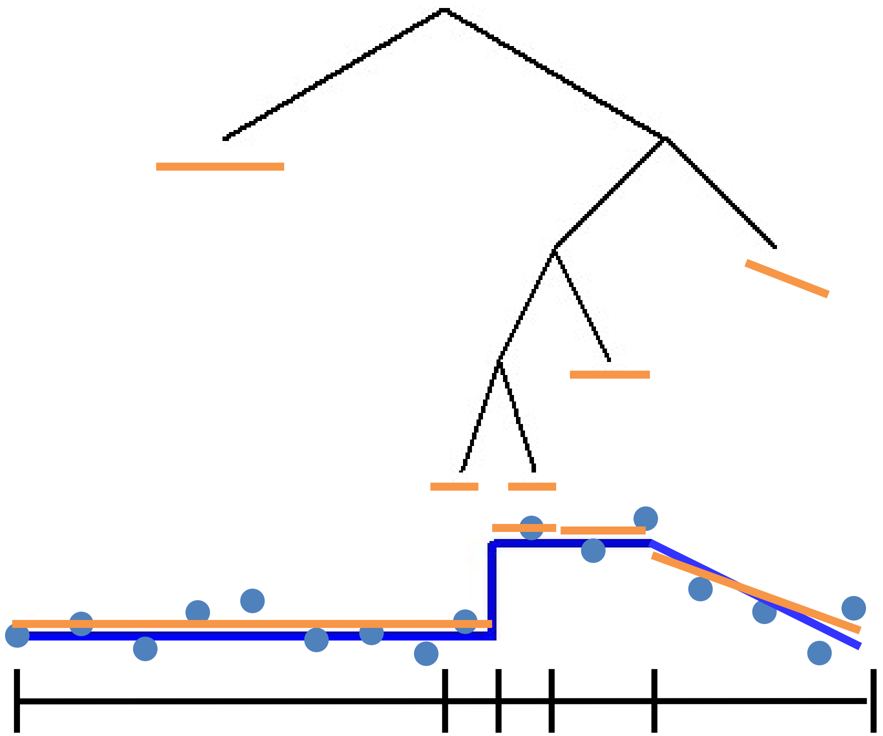


How to assign label to each leaf

Classification – Majority vote



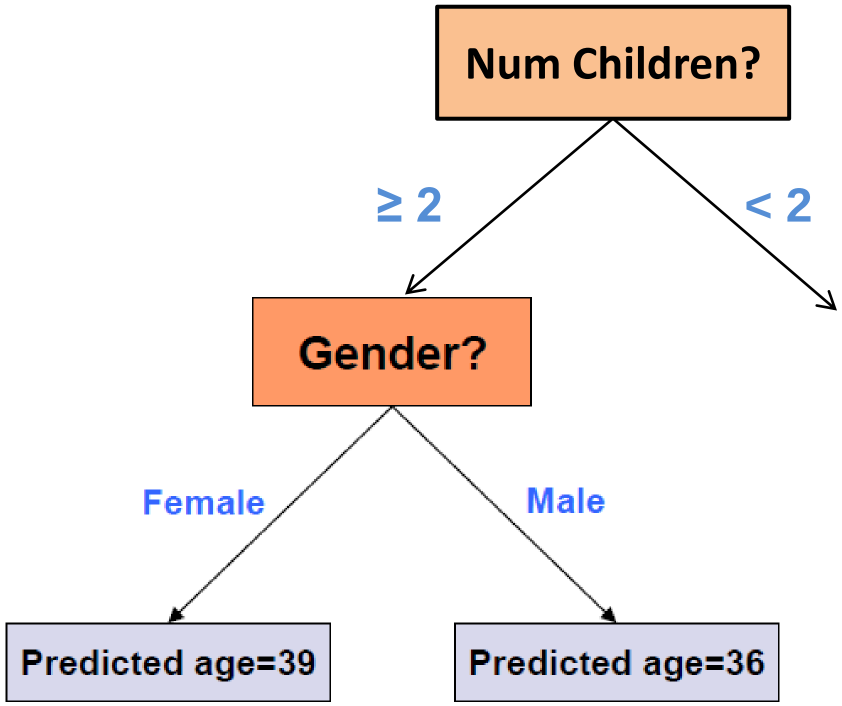
Regression – Constant/Linear/Poly fit



Regression trees

$X^{(1)}$ .... $X^{(p)}$ Y

Gender	Rich?	Num. Children	# travel per yr.	Age
F	No	2	5	38
M	No	0	2	25
M	Yes	1	0	72
:	:	:	:	:



Average (fit a constant) using training data at the leaves

What you should know

- Decision trees are one of the most popular data mining tools
  - Simplicity of design
  - Interpretability
  - Ease of implementation
  - Good performance in practice (for small dimensions)
- Mutual Information (entropy) to select attributes (ID3, C4.5,…)
- Decision trees will overfit!!!
  - Must use tricks to find “simple trees”, e.g.,
    - Pre-Pruning: Fixed depth/Fixed number of leaves
    - Post-Pruning
    - Complexity penalized model selection
- Can be used for classification, regression and density estimation too