Decision Tree

GU 4241/GR 5241

Statistical Machine Learning

Xiaofei Shi

Tasks

Input → Regressor → Predict real number

Input → Classifier → Predict category

Input → Density Estimator → Probability

Columbia University

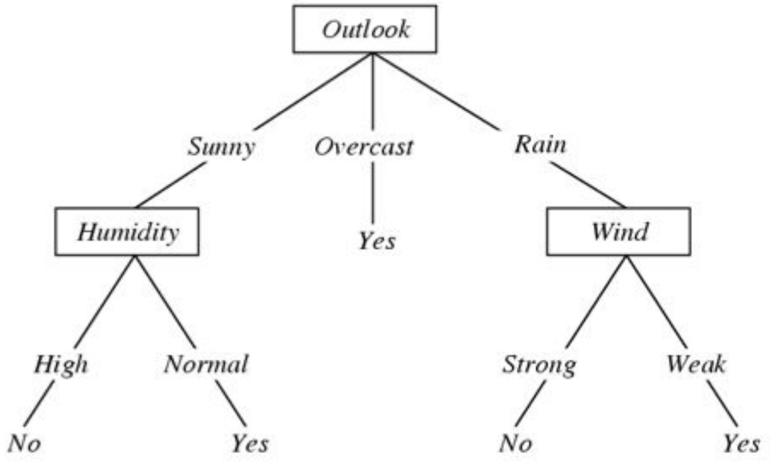
Types of classifiers

- Discriminative classifiers:
  - Directly estimate a decision rule/boundary
  - e.g. decision tree, SVM
- Instance based classifiers:
  - Use observation directly
  - e.g. K nearest neighborhood
- Generative classifiers:
  - Build a generative statistical model
  - e.g. Bayesian Network

COLUMBIA UNIVERSITY

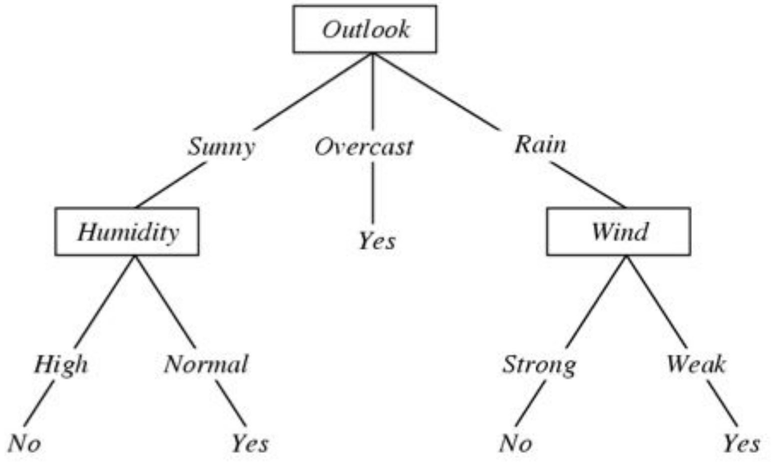
How to make a decision...

- A decision tree for whether to play tennis
- Given a decision tree, how do we assign a label to a test point



How to make a decision...

- A decision tree for whether to play tennis
- Each internal node: test one feature
- Each branch from a node: choose one value for the feature considered
- Each leaf node: prediction for the label



Prediction

If given a decision tree:

- What function does a decision tree represent
- Given a decision tree, how do we assign label to a test point

Now the real problem is:

- How do we learn a decision tree from training data?

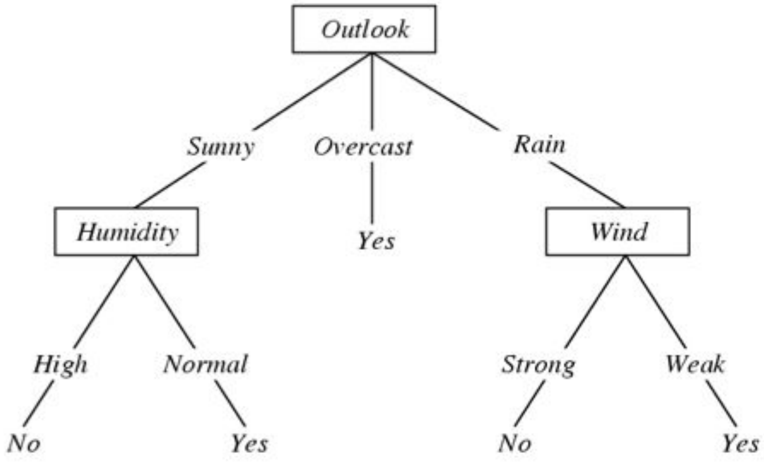
COLUMBIA UNIVERSITY

How to learn and build a decision tree

Top-down induction: (ID3)

Main loop:

1. $X \leftarrow$ the $\boxed{\text{“best”}}$ 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



After all features exhausted, assign majority label to each leaf node.

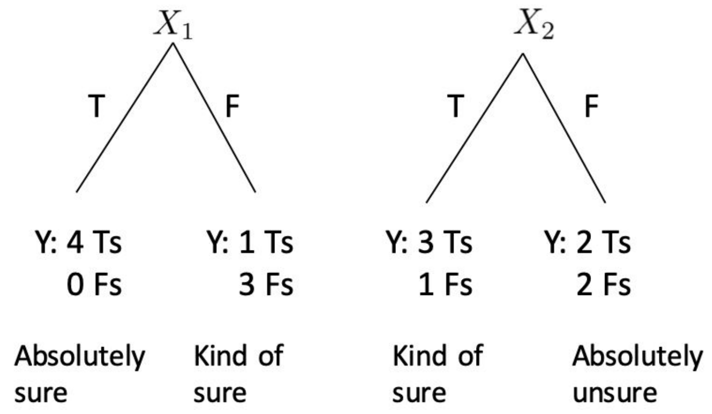
Which feature is considered as “best”

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Columbia University

Which feature is considered as "best"

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F



Good split if we are more certain about classification after split!

COLUMBIA UNIVERSITY

Which feature is considered as "best"

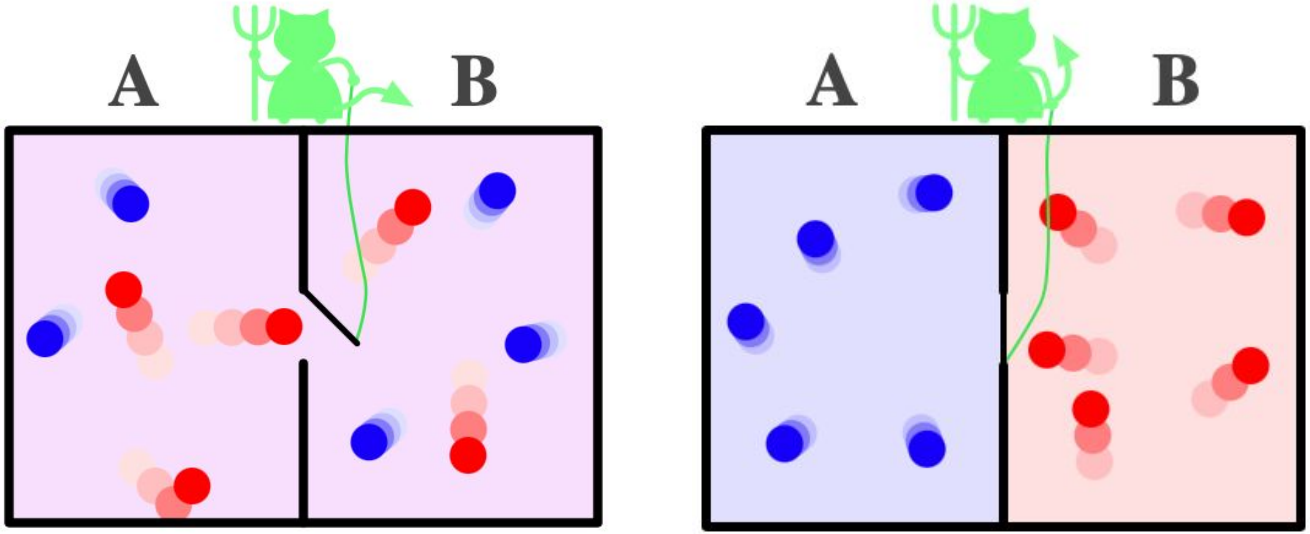
Pick the attribute/feature which yields maximum information gain:

$$\arg\max_i I(Y, X_i) = \arg\max_i [H(Y) - H(Y|X_i)]$$

H(Y) – entropy of Y $H(Y|X_i)$ – conditional entropy of Y

COLUMBIA UNIVERSITY

Entropy: the level of uncertainty



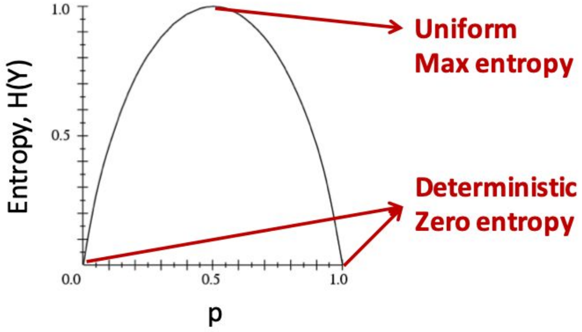
Entropy: the level of uncertainty

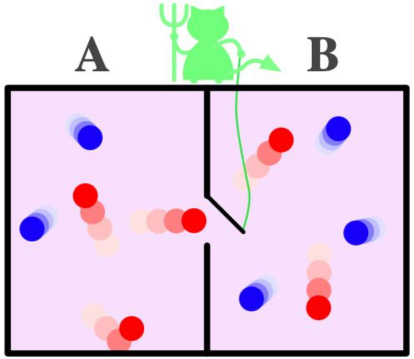
- Entropy of a random variable Y

$$H(Y) = -\sum_{y} P(Y=y)\log_2 P(Y=y)$$

More uncertainty, more entropy!

Y ~ Bernoulli(p)





<u>Information Theory interpretation</u>: $H(Y)$ is the expected number of bits needed to encode a randomly drawn value of Y (under most efficient code)

COLUMBIA UNIVERSITY

Information gain

- Advantage of attribute = decrease in uncertainty
  - Entropy of Y before split

$$H(Y) = -\sum_y P(Y=y)\log_2 P(Y=y)$$

  - Entropy of Y after splitting based on X_i
    - Weight by probability of following each branch

$$\begin{aligned} H(Y \mid X_i) &= \sum_x P(X_i = x) H(Y \mid X_i = x) \\ &= -\sum_x P(X_i = x) \sum_y P(Y = y \mid X_i = x) \log_2 P(Y = y \mid X_i = x) \end{aligned}$$

- Information gain is difference

$$I(Y, X_i) = H(Y) - H(Y \mid X_i)$$

Max Information gain = min conditional entropy

Which feature is considered as “best”

Pick the attribute/feature which yields maximum information gain:

$$\arg\max_i I(Y, X_i) = \arg\max_i [H(Y) - H(Y|X_i)]$$

$$= \arg\min_i H(Y|X_i)$$

Entropy of Y $\quad H(Y) = -\sum_y P(Y=y) \log_2 P(Y=y)$

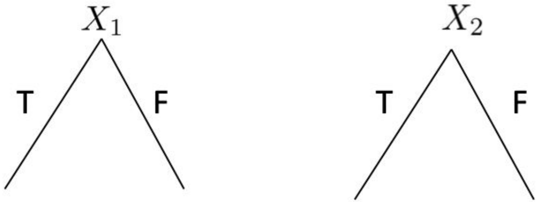
Conditional entropy of Y $\quad H(Y \mid X_i) = \sum_x P(X_i = x) H(Y \mid X_i = x)$

Feature which yields maximum reduction in entropy (uncertainty) provides maximum information about Y

Columbia University

Which feature is considered as “best”

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F



Y: 4 Ts 0 Fs	Y: 1 Ts 3 Fs	Y: 3 Ts 1 Fs	Y: 2 Ts 2 Fs
Absolutely sure	Kind of sure	Kind of sure	Absolutely unsure

$$\widehat{H}(Y|X_1) = -\frac{1}{2}\cancel{[1\log_2 1 + 0\log_2 0]} - \frac{1}{2}[\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}]$$

$$\widehat{H}(Y|X_2) = -\frac{1}{2}[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}] - \underbrace{\frac{1}{2}[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}]}$$

Choose X_1

$$H(Y \mid X_i) = -\sum_x P(X_i = x)\sum_y P(Y = y \mid X_i = x)\log_2 P(Y = y \mid X_i = x)$$

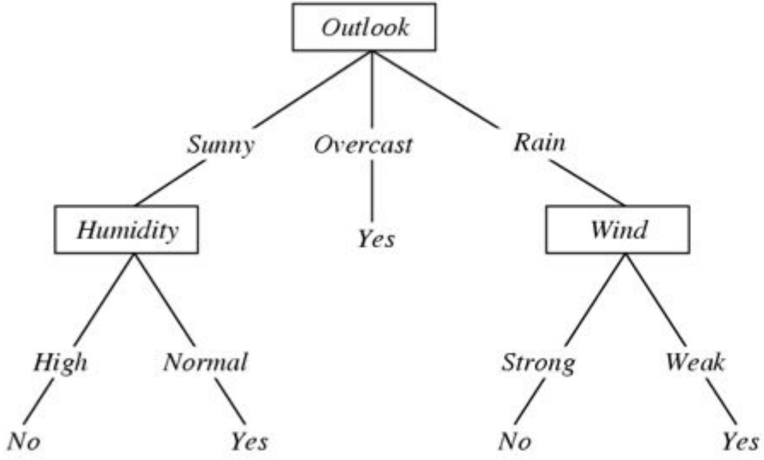
COLUMBIA UNIVERSITY

How to learn and build a decision tree

Top-down induction: (ID3)

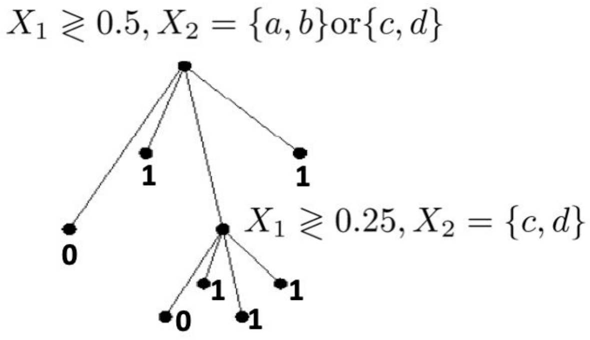
Main loop:

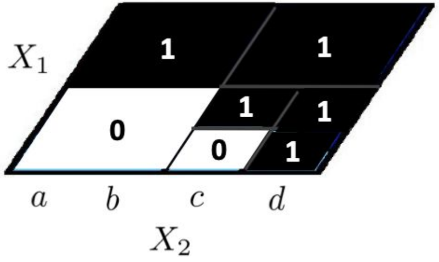
1. $X \leftarrow$ the $\boxed{\text{"best"}}$ decision feature for next *node*
2. Assign X as decision feature for *node*
3. For each value of X, create new descendant of *node* (Discrete features)
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate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



After all features exhausted, assign majority label to each leaf node.

More generally





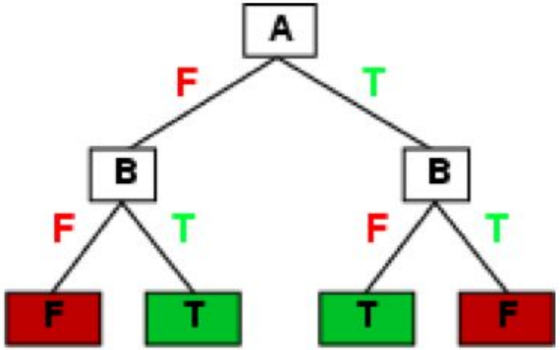
- Features can be discrete, continuous or categorical
- Each internal node: test some set of features $\{X_i\}$
- Each branch from a node: selects a set of value for $\{X_i\}$
- Each leaf node: prediction for Y

COLUMBIA UNIVERSITY

More generally

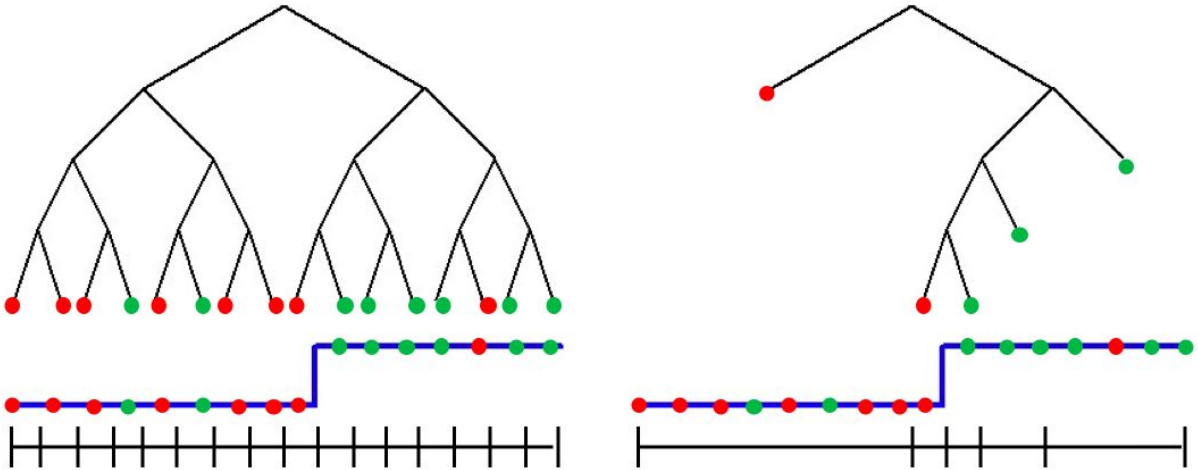
- Decision trees in general (without pruning) can express any function of the input features.
- There is a decision tree which perfectly classifies a training set with one path to leaf for each example - overfitting
- But it won't generalize well to new examples - prefer to find more compact decision trees

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

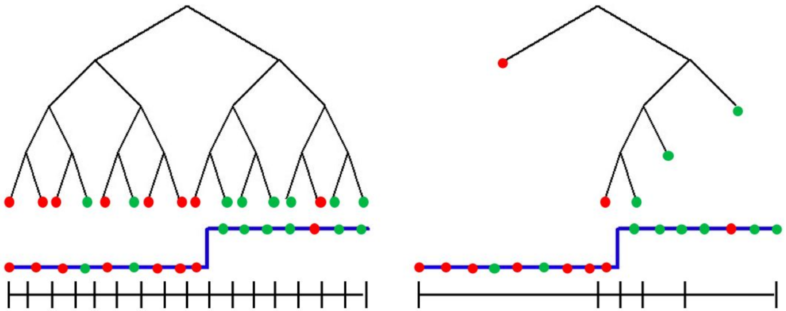


Overfitting

One training example per leaf – overfits, need compact/pruned decision tree



When to stop...



- Occam's razor: *the more assumptions you have to make, the more unlikely an explanation*
- People therefore look for simpler trees:

  - Pre-Pruning:
    - Fixed depth
    - Fixed number of leaves
  - Post-Pruning: Chi-square test of independence
    - Convert decision tree to a set of rules
    - Eliminate variable values in rules which are independent of label (using Chi-square test)
    - Simplify rule set by eliminating unnecessary rules
  - Complexity Penalized/MDL model selection

Information Criteria

- Penalize complex models by introducing cost

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

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

regression

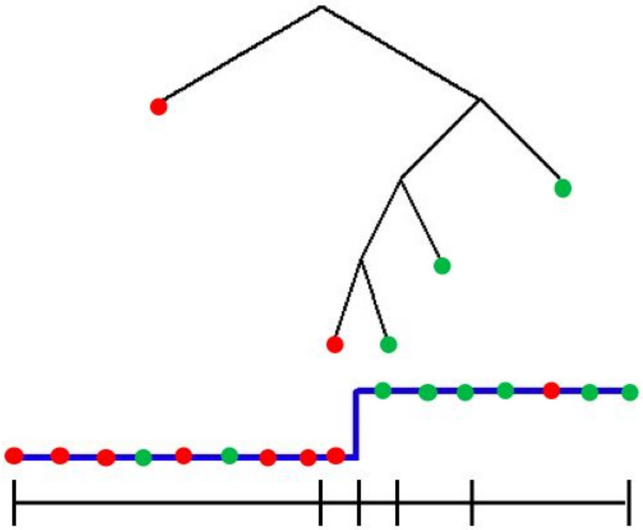
classification

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

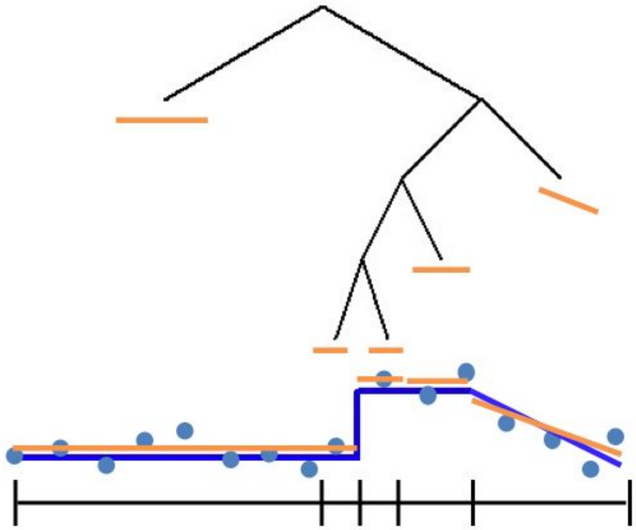
CART – optimization can be solved by dynamic programming

How to assign values to each leaf

Classification – Majority vote



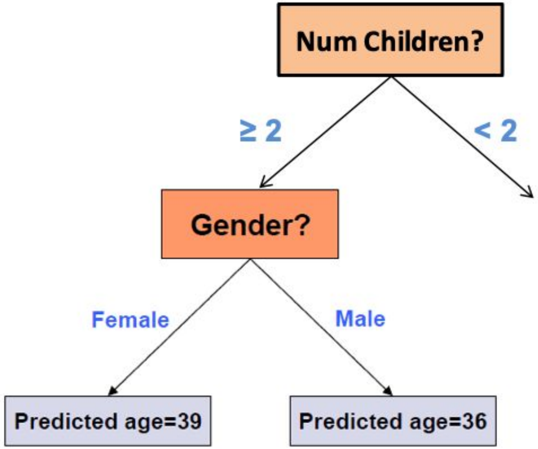
Regression – Constant/ Linear/Poly fit



COLUMBIA UNIVERSITY

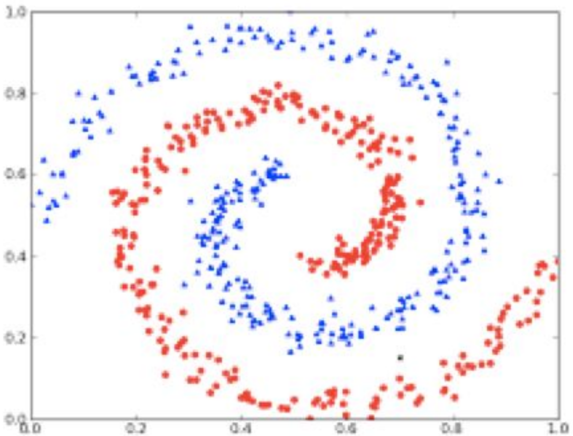
Regression Tree

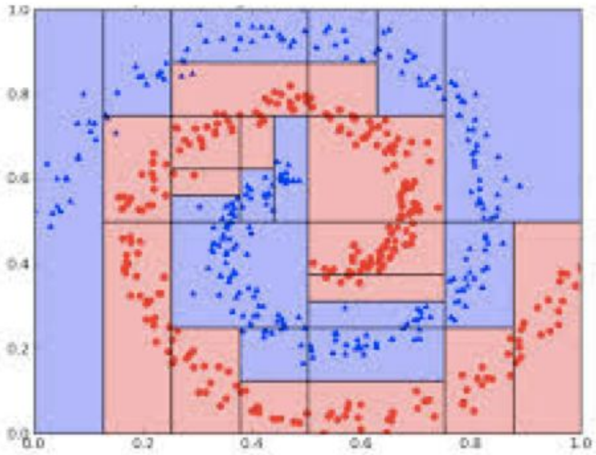
		$X^{(1)}$	$X^{(p)}$	Y
Gender	Rich?	Num. Children	# travel per yr.	Age
F	No	2	5	38
M	No	0	2	25
M	Yes	1	0	72
:	:	:	:	:



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

A 2D example





cs.uchicago.edu

COLUMBIA UNIVERSITY

Takeaways

- Decision trees are one of the most popular data mining tools
  - Interpretability
  - Easy to implement
  - Good performance in practice (for small dimensions!!!)
- Information gain to select attributes (ID3, C4.5, CART...)
- Can be used for classification, regression and density estimation too
- Decision trees will overfit!!! Must use tricks to find “simple trees”, e.g.,
  - Pre-Pruning: Fixed depth/Fixed number of leaves
  - Post-Pruning: Chi-square test of independence
  - Complexity Penalized/MDL model selection

Columbia University

References

- Christopher Bishop: Pattern Recognition and Machine Learning, Chapter 14.4
- Tom Mitchell: Machine Learning, Chapter 3
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Chapter 9
- Ziv Bar-Joseph, Tom Mitchell, Pradeep Ravikumar and Aarti Singh: CMU 10-701

COLUMBIA UNIVERSITY