Machine learning

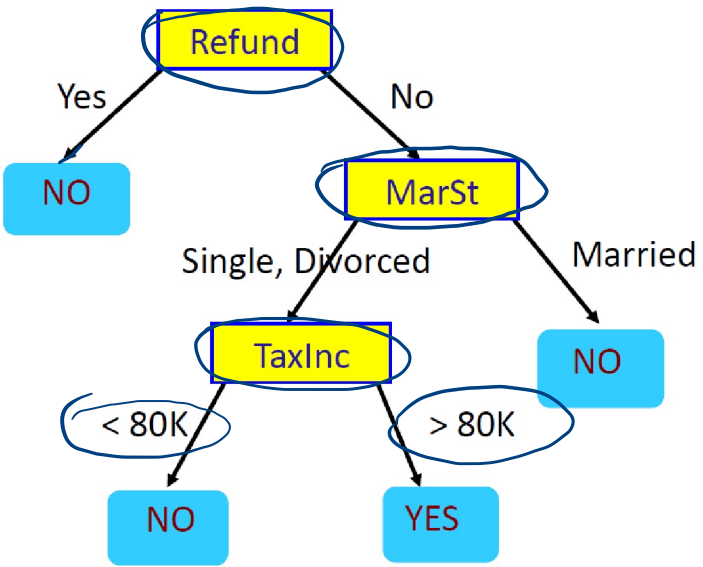
Decision Trees

Mohammad-Reza A. Dehaqani

dehaqani@ut.ac.ir

Slides are mainly adopted form cmu Aarti course

Decision Trees; discrete features, tax fraud detection

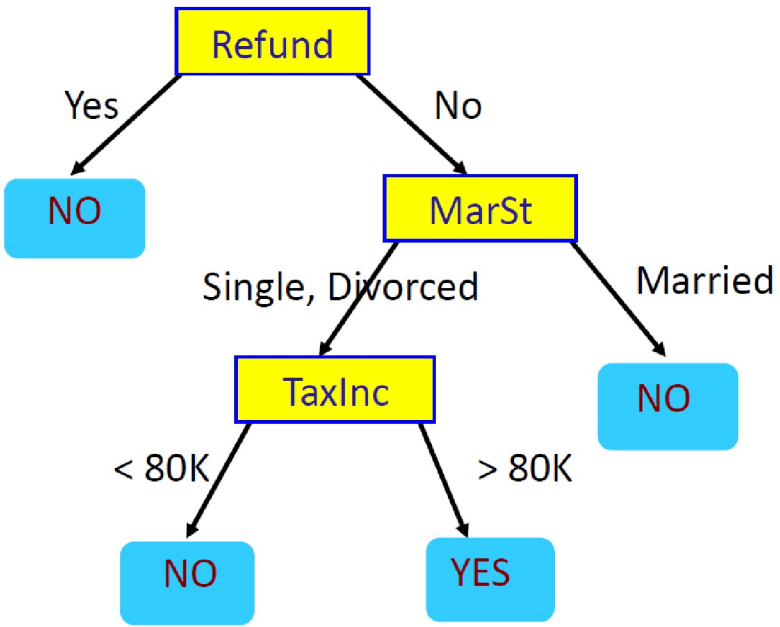


X_1	X_2	X_3	Y
Refund	Marital Status	Taxable Income	Cheat

- Each internal node: test one feature X_i
- Each branch from a node: selects some value for X_i
- Each leaf node: prediction for Y

Prediction: Given a decision tree, how do we assign label to a test point

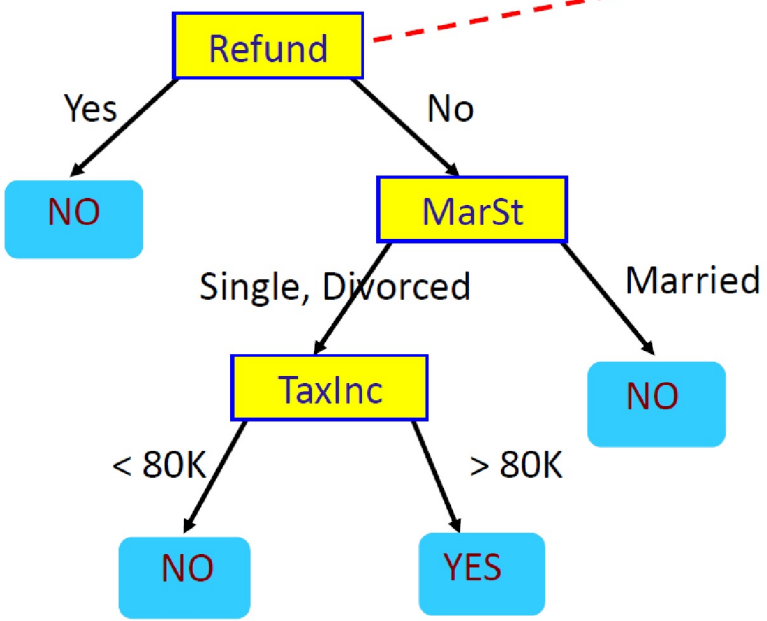
Decision Tree for Tax Fraud Detection



Query Data

X_1	X_2	X_3	Y
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

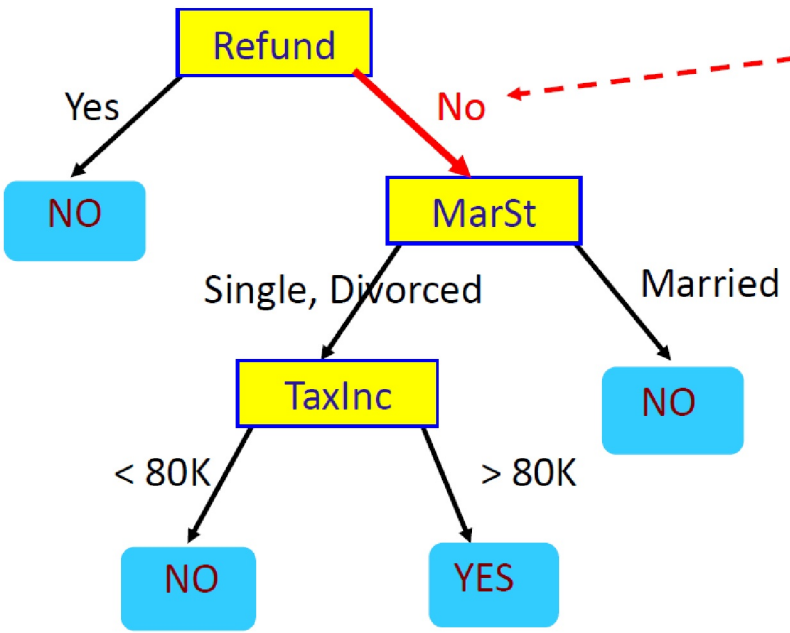
Decision Tree for Tax Fraud Detection



Query Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat
No	Married	80K	?

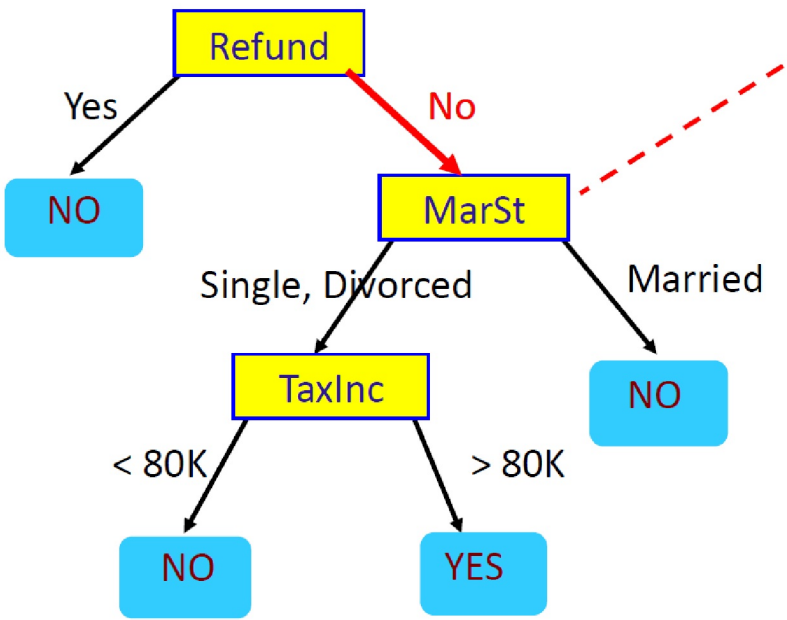
Decision Tree for Tax Fraud Detection



Query Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat
No	Married	80K	?

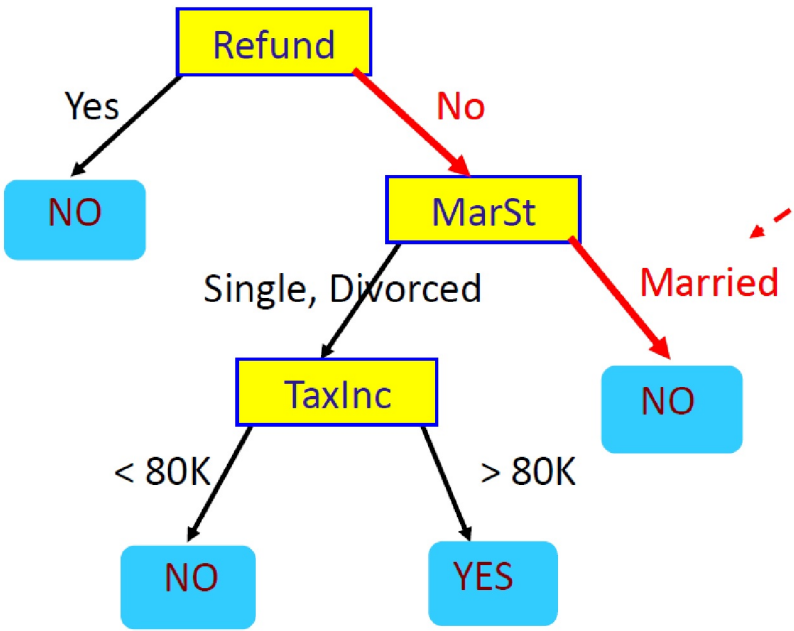
Decision Tree for Tax Fraud Detection



Query Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat
No	Married	80K	?

Decision Tree for Tax Fraud Detection

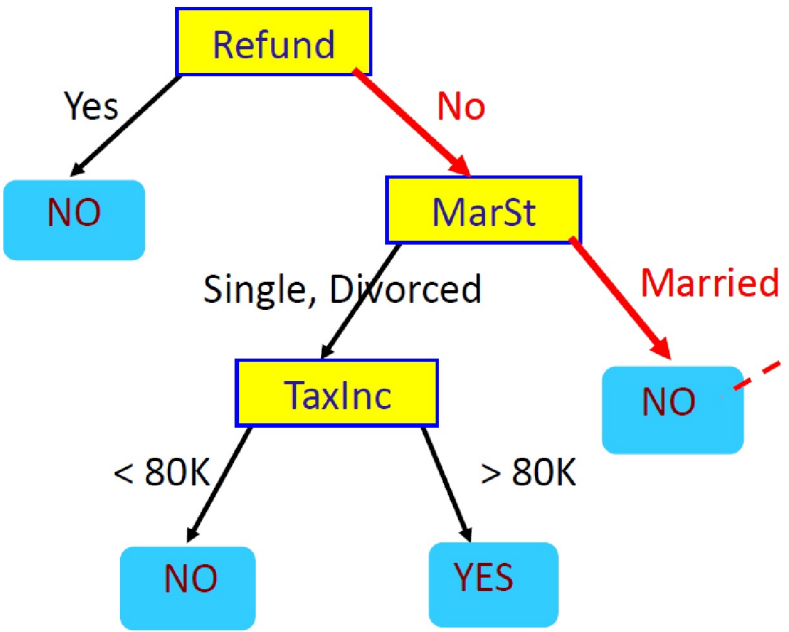


Query Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat
No	Married	80K	?

No

Decision Tree for Tax Fraud Detection



Query Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat
No	Married	80K	?

Assign Cheat to "No"

So far...

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

Discriminative or Generative?

Now ...

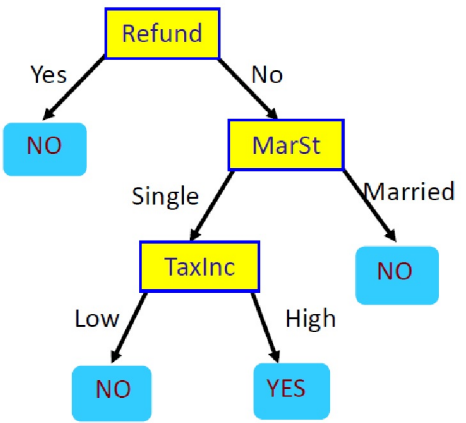
- How do we learn a decision tree from training data

How to learn a decision tree

- Top-down induction [ID3]

Main loop:

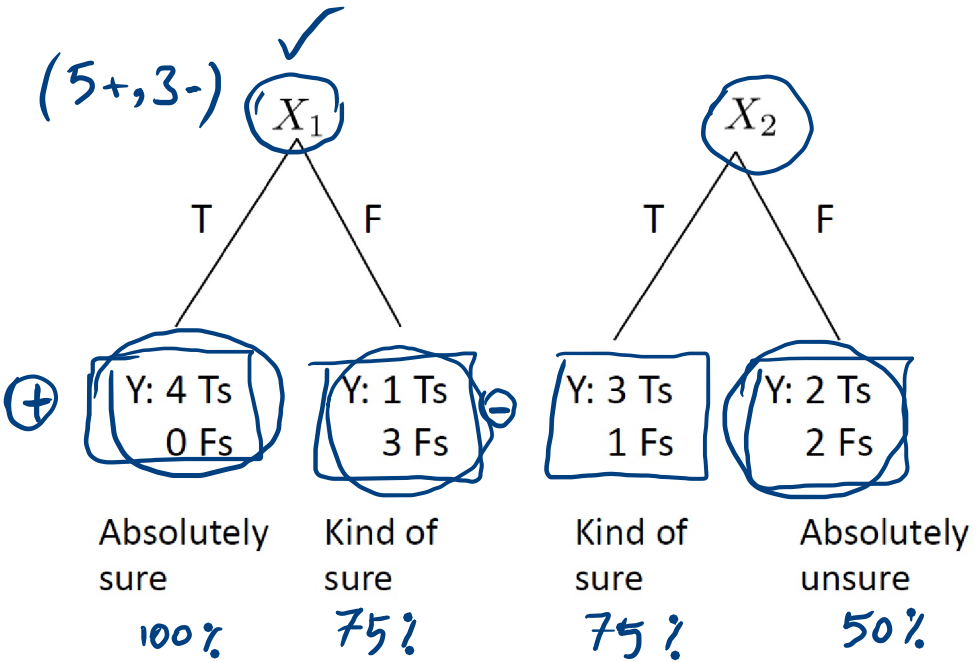
1. $X \leftarrow$ the "best" decision feature for next *node*
2. Assign X as decision feature for *node*
3. For each value of X, create new descendant of *node* (Discrete features)
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf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



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

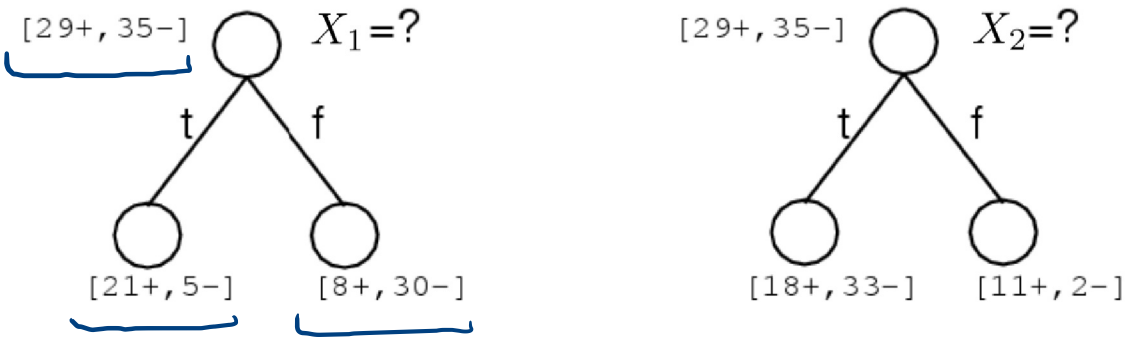
Which feature is best?

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 –
Uniform distribution of labels is bad

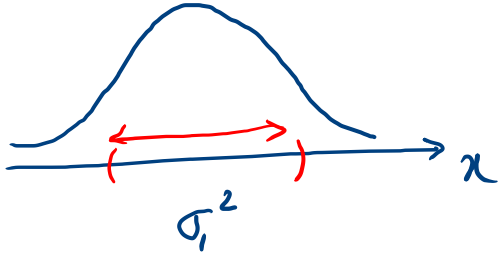
Which feature is best?

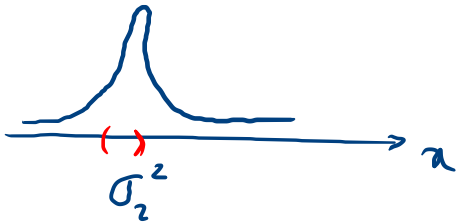


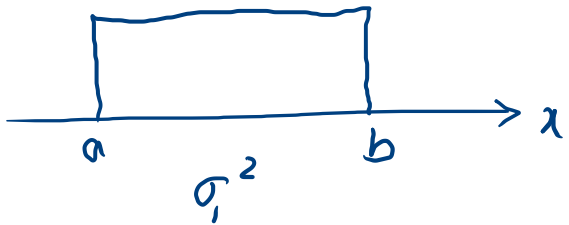
Pick the attribute/feature which yields maximum information gain:

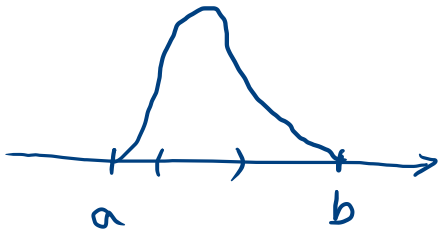
$$\arg\max_i I(Y, X_i) = \arg\max_i [H(Y) - H(Y|X_i)]$$

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









M

Entropy

$$H(X) = -\sum_{i=1}^{m} P(x=i) \log_2 P(x=i)$$

Differential Entropy

$$h(x) = -\int p(x) \log p(x) \, dx$$

Entropy

$H(Y) = -p\log_2 p - (1-p)\log_2(1-p)$

- Entropy of a random variable Y

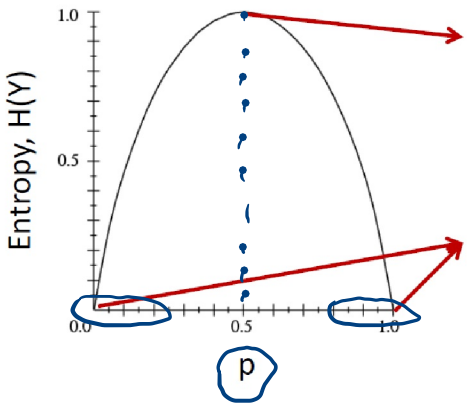
$$H(Y) = -\sum_y P(Y=y)\log_2 P(Y=y)$$

$p^* = 1/2$

? 75 0 1/2
? 25 1 1/2

More uncertainty, more entropy!

Y ~ Bernoulli(p)



0.5
0.5

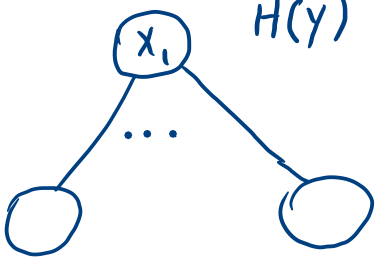
x_1 x_2
0 0
0 1
1 0
1 1

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

$$H(y) = -\sum_{i=1}^{m} \underbrace{P(y=i)}_{\geq 0} \underbrace{\log P(y=i)}_{\leq 0} = -\sum p_i \log p_i$$

$$P(y=i) = \frac{1}{m} \qquad i = 1, \ldots, m$$

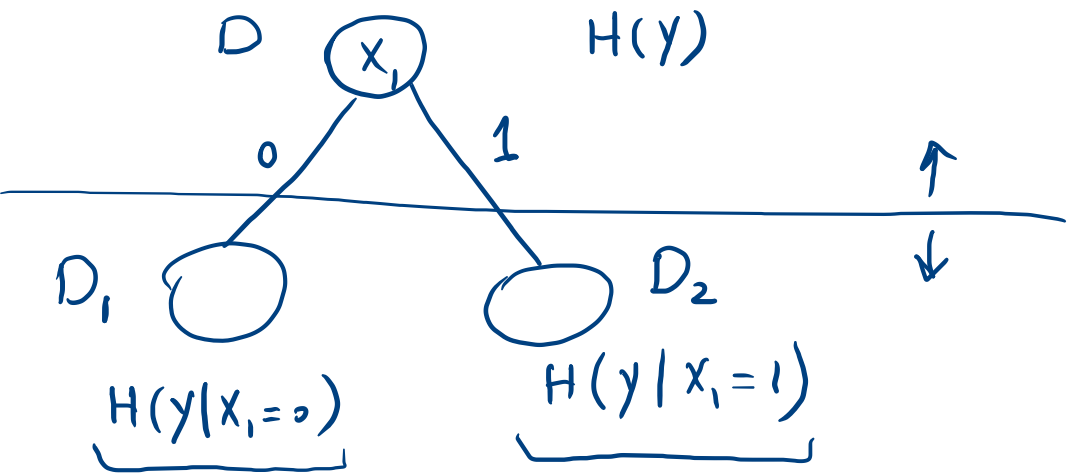
$$\sum_{i=1}^{m} p_i = 1$$



$H(y)$

$H_2(y)$

$$H(y) - H_2(y) \geq 0$$



H(y)

H(y|X)

$D_1 \cup D_2 = D$

$$\frac{|D_1|}{|D|} H(y|X_1=0) + \frac{|D_2|}{|D|} H(y|X_1=1) = H(y|X)$$

(conditional Entropy)

$$H(y|X) \leq H(y) \Rightarrow I(x;y) = H(y) - H(y|X) \geq 0$$

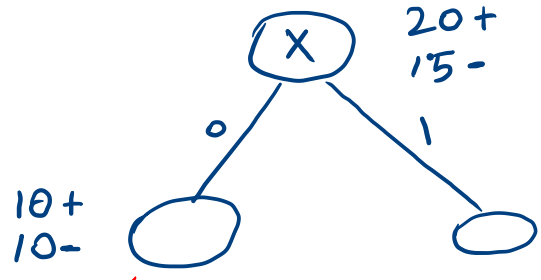
mutual Information

$$I(x;y) = H(y) - H(y|x) = H(x) - H(x|y)$$

$$I(x;y) = 0 \iff x \perp y \equiv p(x,y) = p(x)\,p(y)$$

$H(y)$

$H(y|x)$



$I(X;Y) = ?$

$H(Y) = -\frac{20}{35}\log\frac{20}{35} - \frac{15}{35}\log\frac{15}{35}$

$H(Y|X=0) = -\frac{10}{20}\log\frac{10}{20} - \frac{10}{20}\log\frac{10}{20}$

α

$H(Y|X) = \frac{20}{35}\alpha + \frac{15}{35}\beta \leq H(Y)$

$H(Y|X=1) = -\frac{10}{15}\log\frac{10}{15} - \frac{5}{15}\log\frac{5}{15}$

β

$$H(y|x) = \underbrace{E_x\left[H(y|X=x)\right]} = -\sum_x \sum_y P(X=x, Y=y) \log P(Y=y|X=x)$$

$$I(X;Y) = H(X) - \underbrace{H(X|Y)}_{0} = H(X)$$

$X \sim N(0,1)$

$Y = X^2$ $\qquad$ $E[X^3] = 0$

Information Gain

$I(x_i; y)$

- 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 best to split?

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

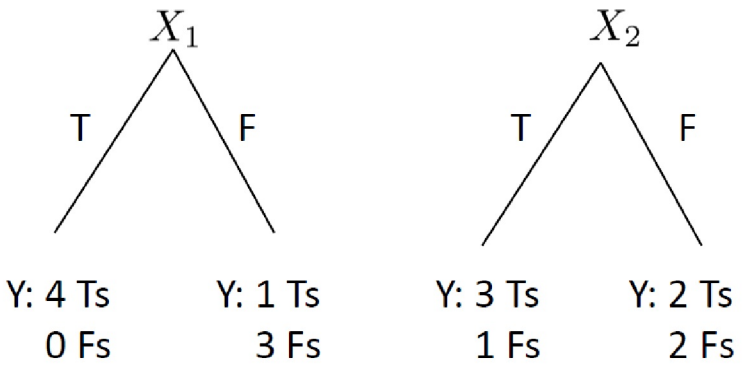
Conditional entropy of Y $\quad H(Y \mid X_i) = \sum_x P(X_i = x) H(Y \mid X_i = x)$

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

Information Gain

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

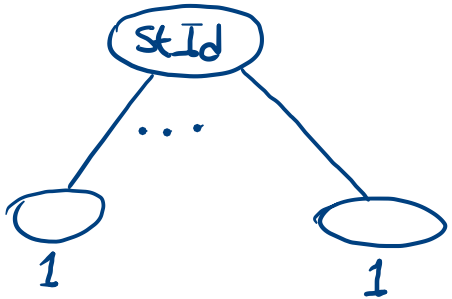
X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F



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

$$\widehat{H}(Y|X_2) = -\frac{1}{2}[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}] - \underbrace{\frac{1}{2}[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}]}_{>0}$$

$$\widehat{H}(Y|X_1) < \widehat{H}(Y|X_2)$$



$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Where,

- |Dj|/|D| acts as the weight of the jth partition.
- v is the number of discrete values in attribute A.

The gain ratio can be defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

Mutual Inf.

C4.5 uses Gain ratio

Gini Index/Impurity

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini}_A(D) = \frac{|D1|}{|D|}\text{Gini}(D_1) + \frac{|D2|}{|D|}\text{Gini}(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

$H(y)$

$H(y|x)$

$I(x;y)$

CART uses Gini index

Handling continuous features

Convert continuous features into discrete by setting a threshold.

What threshold to pick?

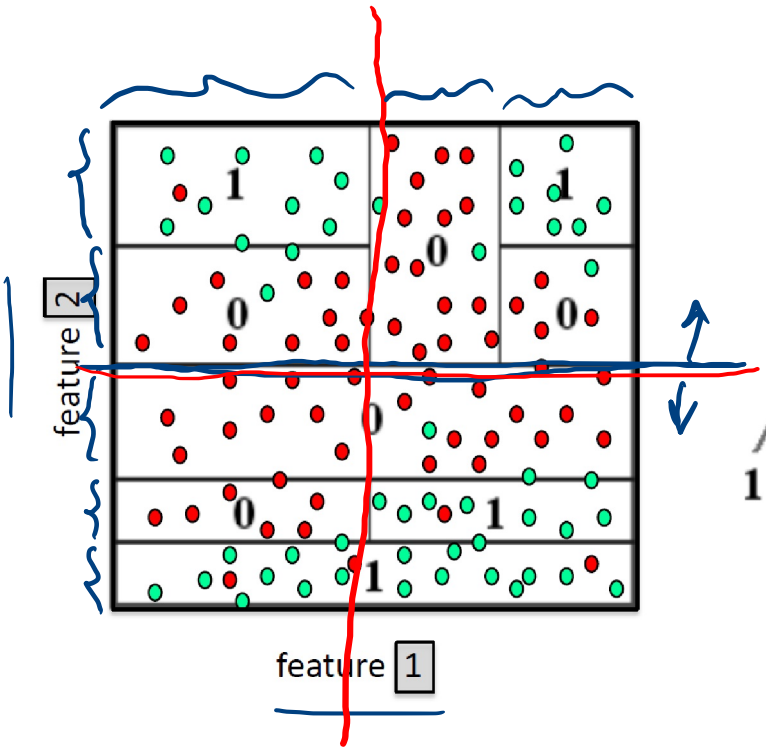
Search for best one as per information gain. Infinitely many??

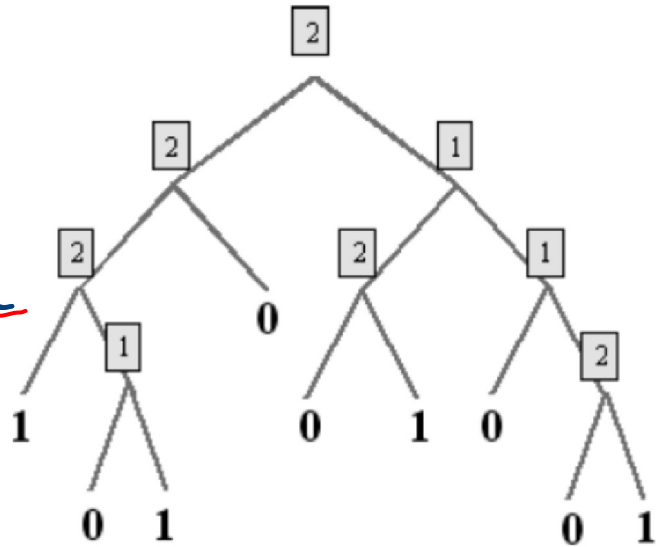
Don't need to search over more than ~ n (number of training data),e.g. say X_1 takes values $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}$ in the training set. Then possible thresholds are

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

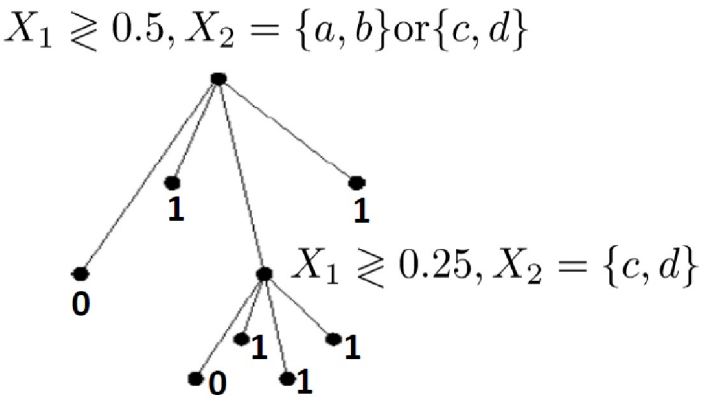
$n-1$

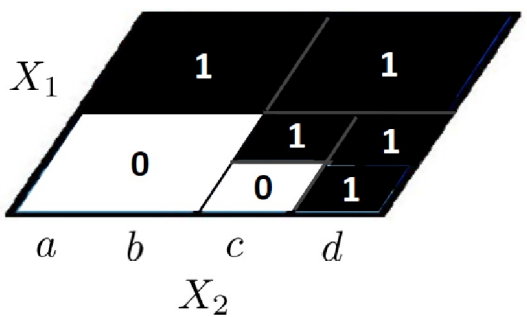
Dyadic decision trees (split on mid-points of features)





Decision Tree more generally



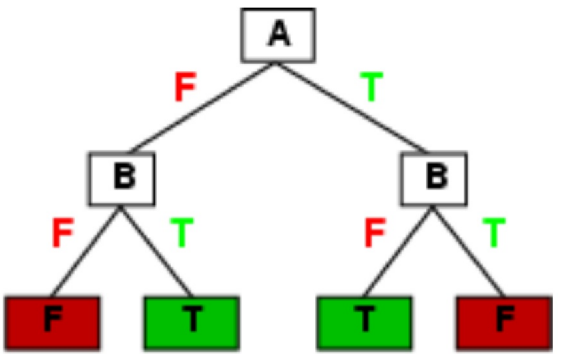


- Features can be discrete, continuous or categorical
- Each internal node: test some set of features $\{X_i\}$
- Each branch from a node: selects a set of value for $\{X_i\}$
- Each leaf node: prediction for Y

Expressiveness of Decision Trees

- Decision trees in general (without pruning) can express any function of the input features.
- E.g., for Boolean functions, truth table row → path to leaf:

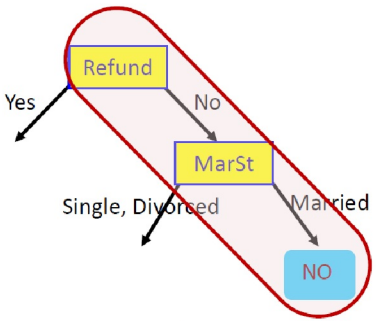
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

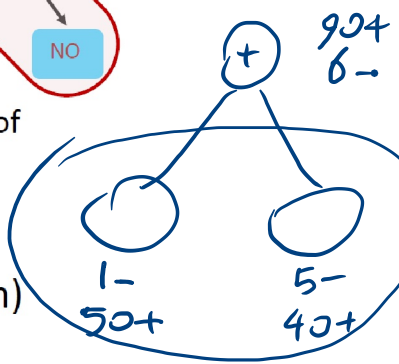


- There is a decision tree which perfectly classifies a training set with one path to leaf for each example - overfitting
- But it won't generalize well to new examples - prefer to find more compact decision trees

When to Stop?

- Many strategies for picking simpler trees:
  - Pre-pruning
    - Fixed depth (e.g. ID3)
    - Fixed number of leaves
  - Post-pruning
    - Chi-square test
      - Convert decision tree to a set of rules
      - Eliminate variable values in rules which are independent of label (using chi-square test for independence)
      - Simplify rule set by eliminating unnecessary rules
  - Information Criteria: MDL(Minimum Description Length)





Information Criteria

- Penalize complex models by introducing cost

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

error ↓ ↑

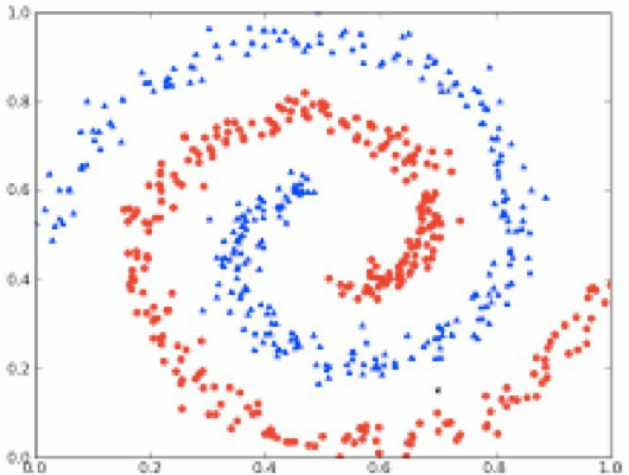
log likelihood — cost

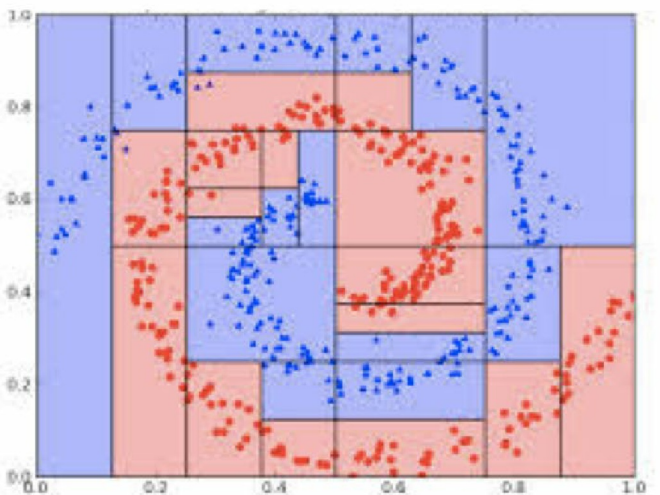
$$\text{loss}(\hat{f}_T(X_i), Y_i) = (\hat{f}_T(X_i) - Y_i)^2 \quad \text{regression}$$
$$= 1_{\hat{f}_T(X_i) \neq Y_i} \quad \text{classification}$$

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

CART – optimization can be solved by dynamic programming

Example of 2-feature decision tree classifier

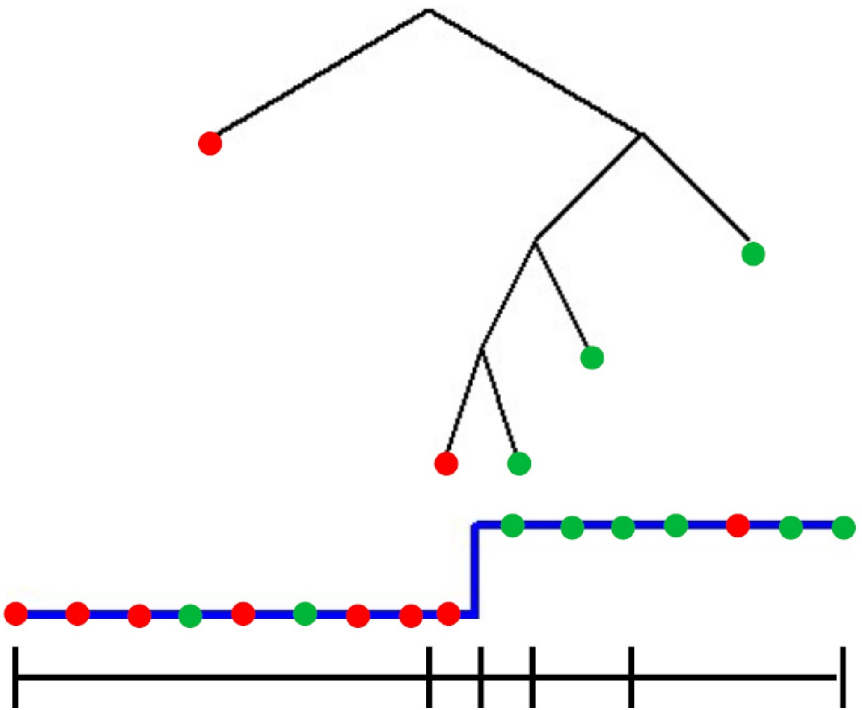




How to assign label to each leaf

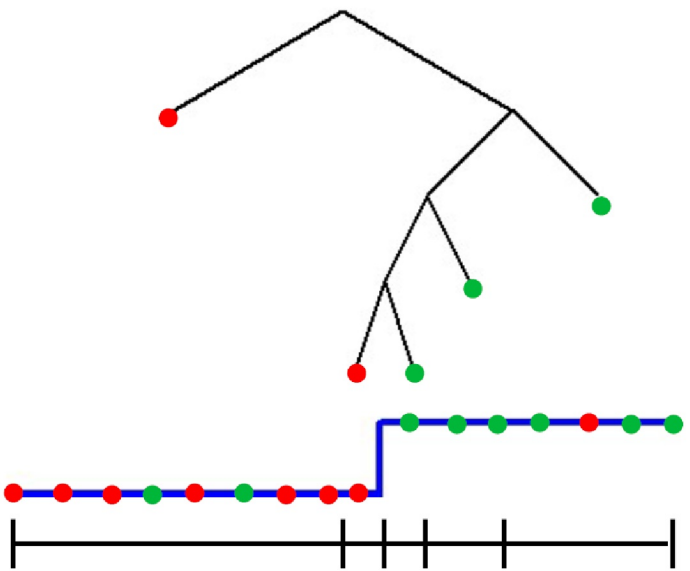
Classification – Majority vote

Regression – ?

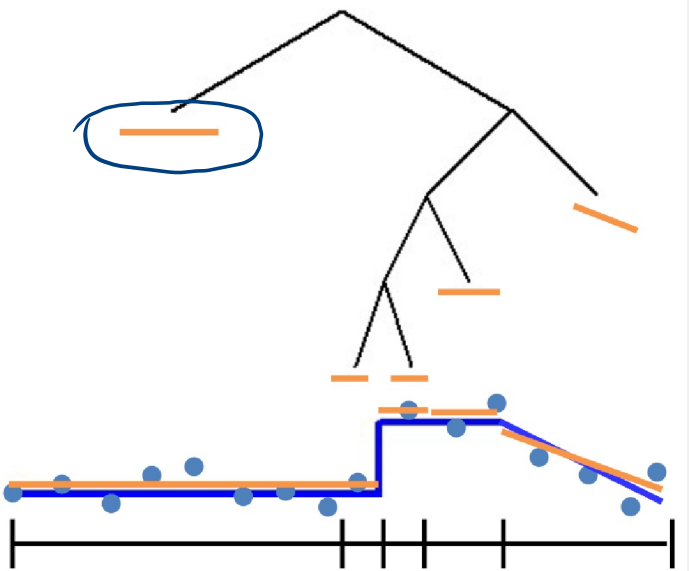


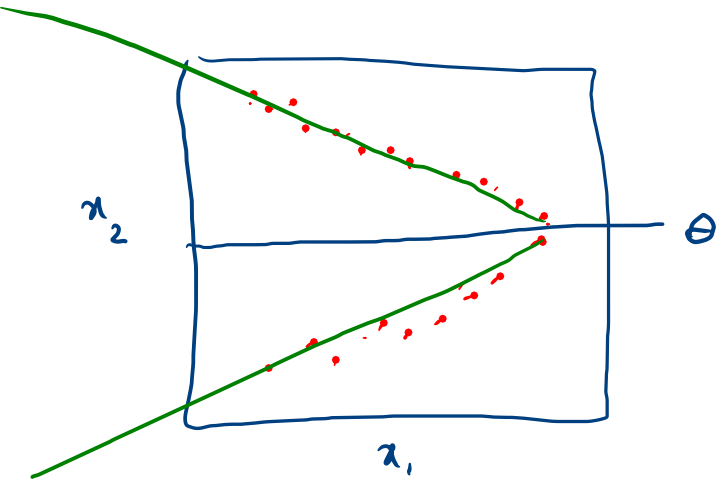
How to assign label to each leaf

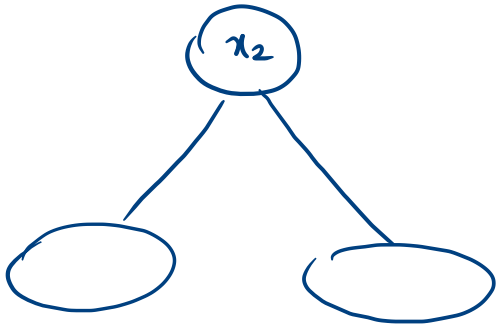
Classification – Majority vote



Regression – Constant/
Linear/Poly fit



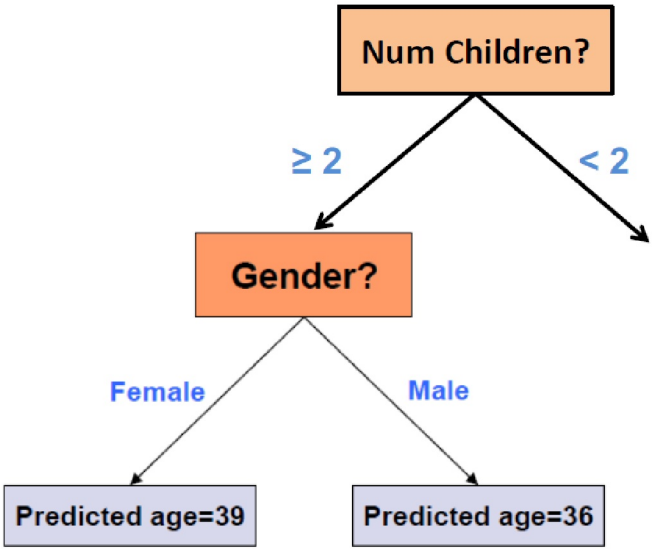




$$H(Y) = -\int p(y) \log p(y) \, dy$$

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)
- Information gain 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: Chi-square test of independence
    - Complexity Penalized/MDL model selection
- Can be used for classification, regression and density estimation too