Decision Trees

Aarti Singh, Eric Xing

Machine Learning 10-701/15-781
Sept 10, 2012

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

How does a decision tree represent a prediction rule?

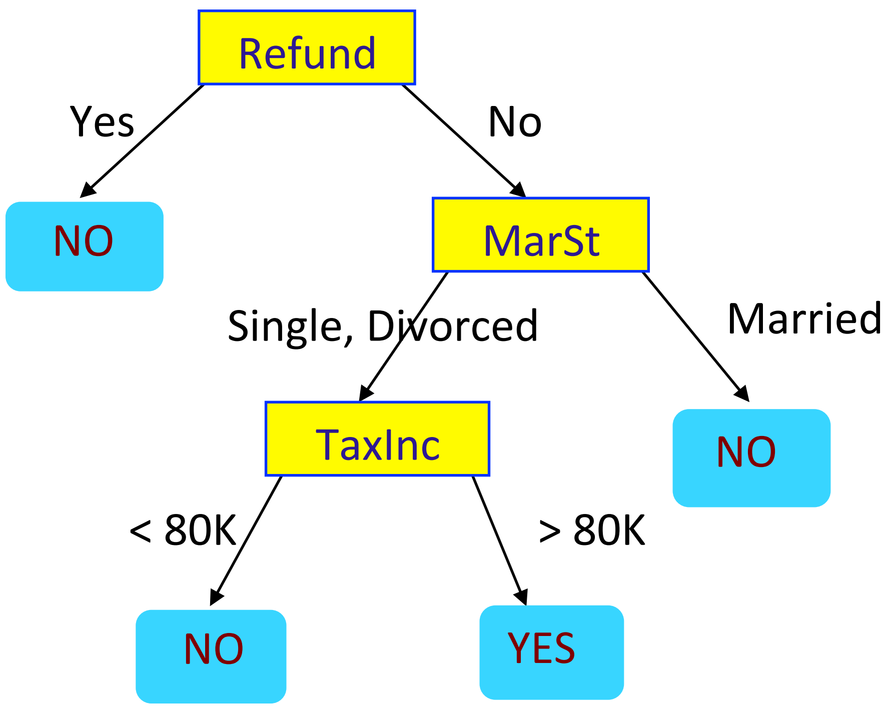
Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$

Data

X_1 Refund	X_2 Marital Status	X_3 Taxable Income	Y Cheat



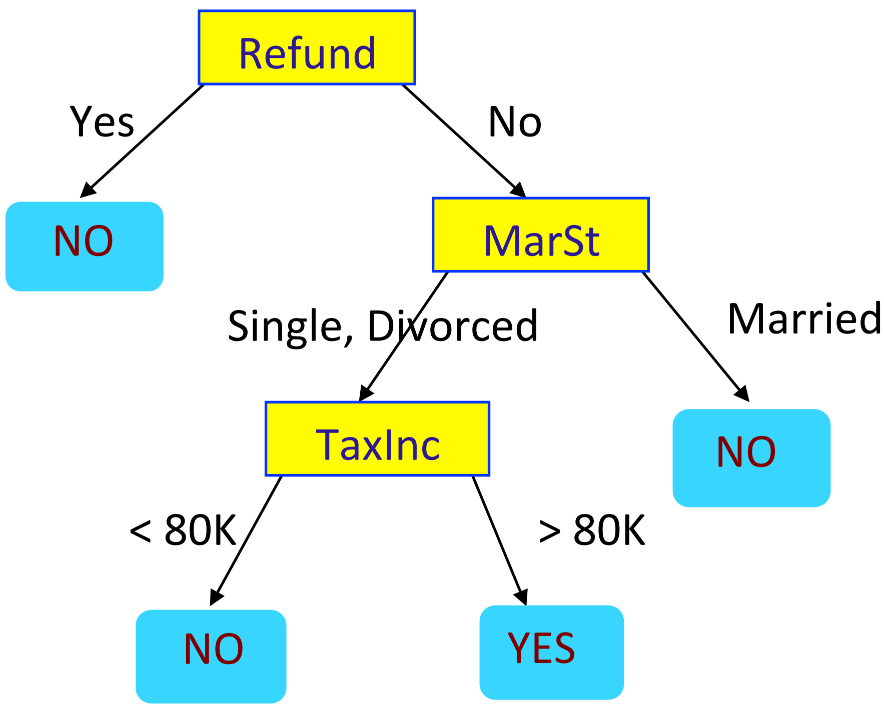
- Each internal node: test one feature X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y

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

Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$



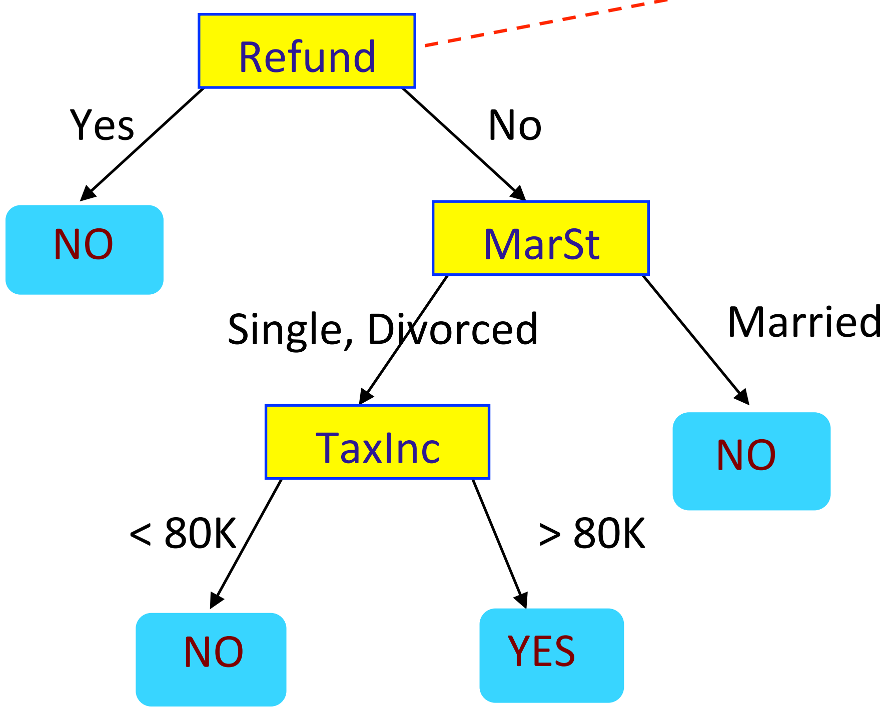
Query Data

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

Decision Tree for Tax Fraud Detection

$$\mathcal{F} - \text{Decision Trees}$$

$$f(X_1, X_2, X_3) \in \mathcal{F}$$



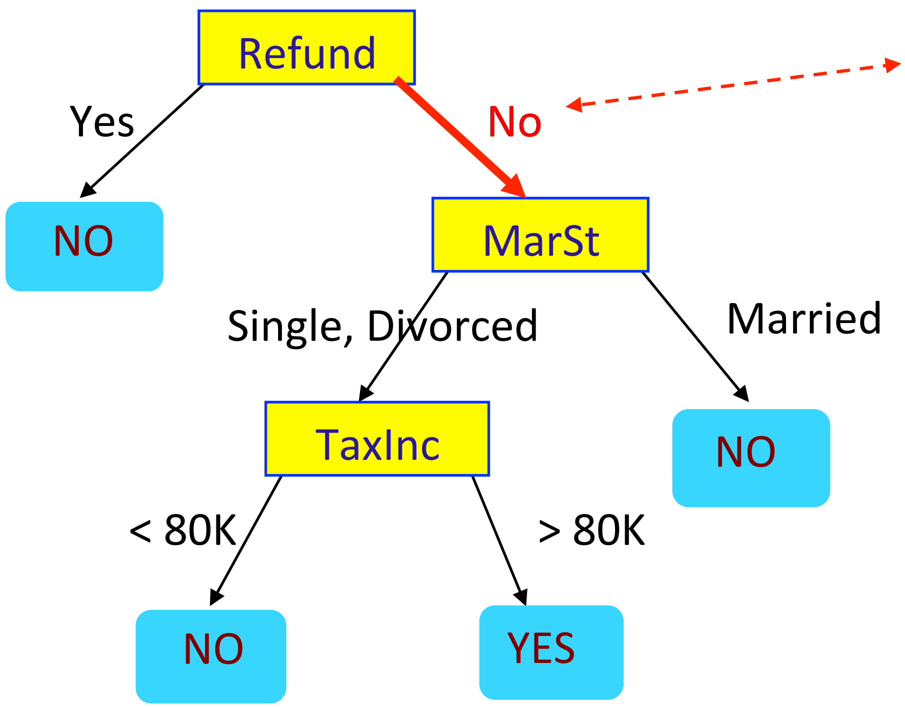
Query Data

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

Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$



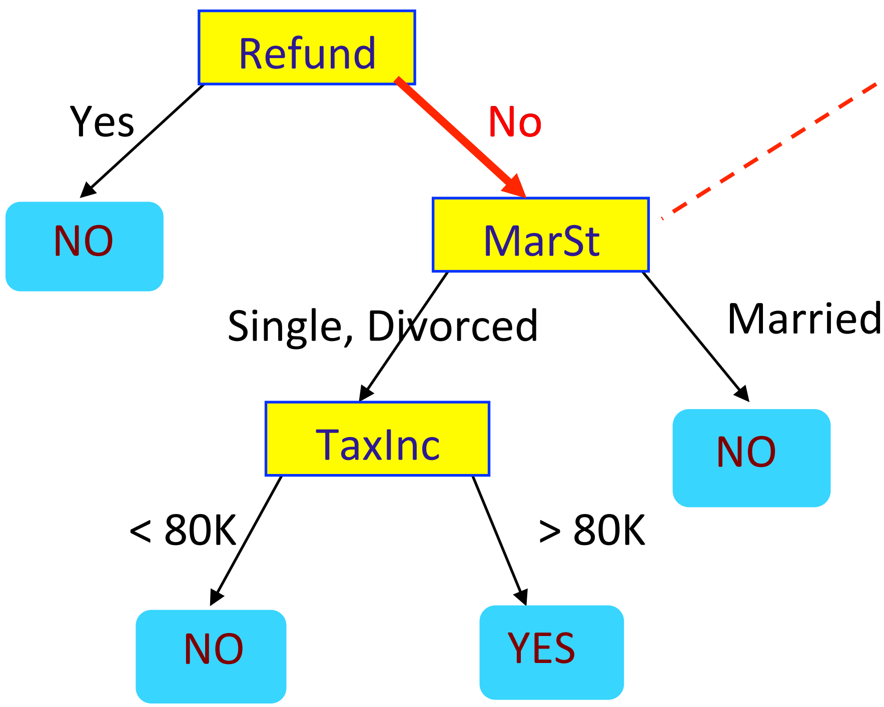
Query Data

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

Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$



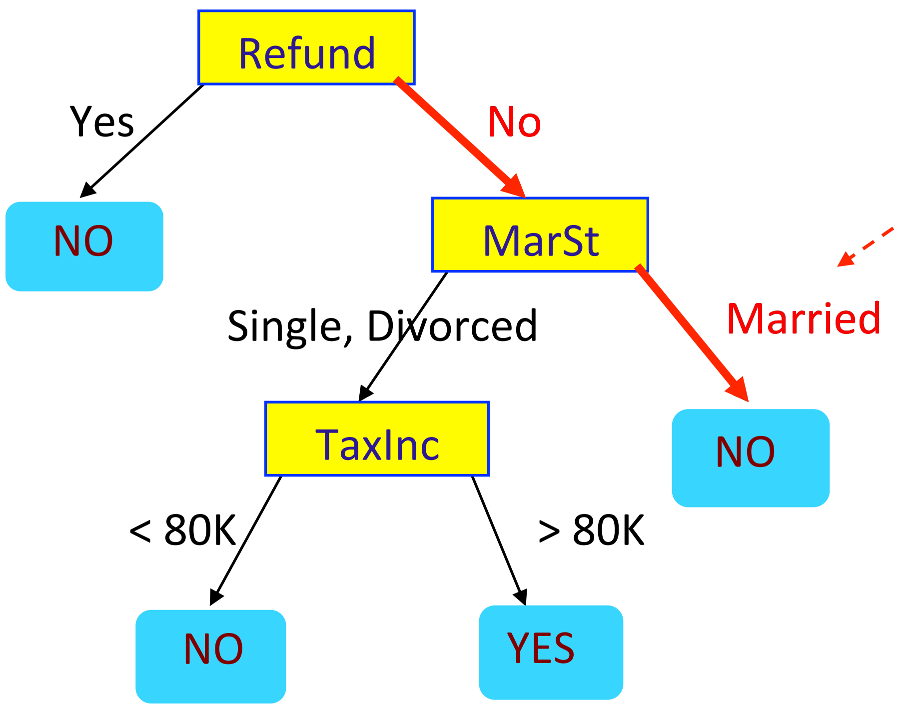
Query Data

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

Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$



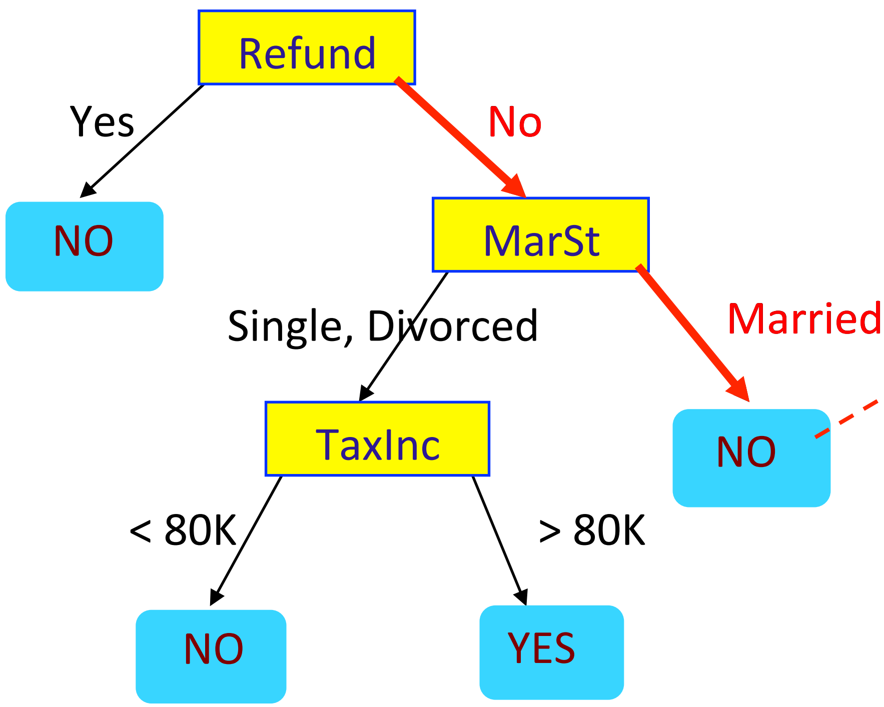
Query Data

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

Decision Tree for Tax Fraud Detection

$\mathcal{F}$ – Decision Trees

$f(X_1, X_2, X_3) \in \mathcal{F}$



Query Data

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

Assign Cheat to “No”

How do we learn a decision tree from training data?

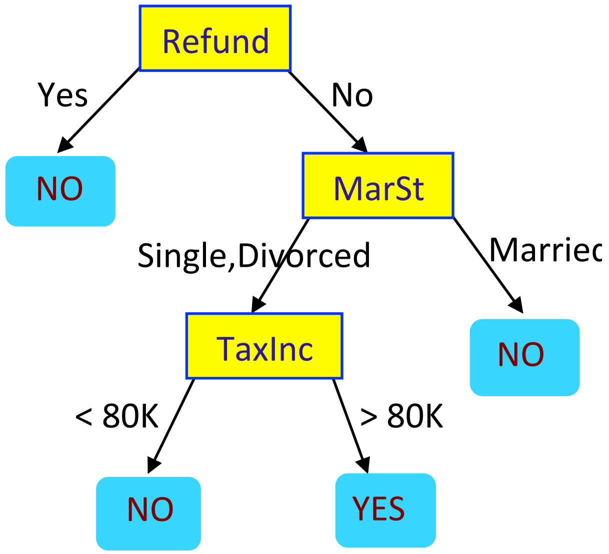
How to learn a decision tree

- Top-down induction [many algorithms ID3, C4.5, CART, …]

We will focus on ID3 algorithm

Repeat:

1. Select "best feature" (X_1, X_2 or X_3) to split
2. For each value that feature takes, sort training examples to leaf nodes
3. Stop if leaf contains all training examples with same label or if all features are used up
4. Assign leaf with majority vote of labels of training examples



Which feature is best to split?

Good split if we are less uncertain about classification after split

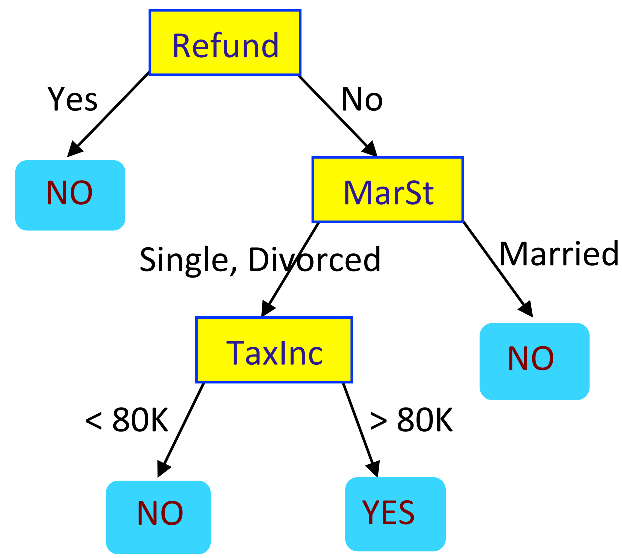
80 training people

Refund X_1

- Yes: 40 Genuine, 0 Cheats — Absolutely sure
- No: 10 Genuine, 30 Cheats — Kind of sure

Marital Status X_2

- Single, Divorced: 30 Genuine, 10 Cheats — Kind of sure
- Married: 20 Genuine, 20 Cheats — Absolutely unsure



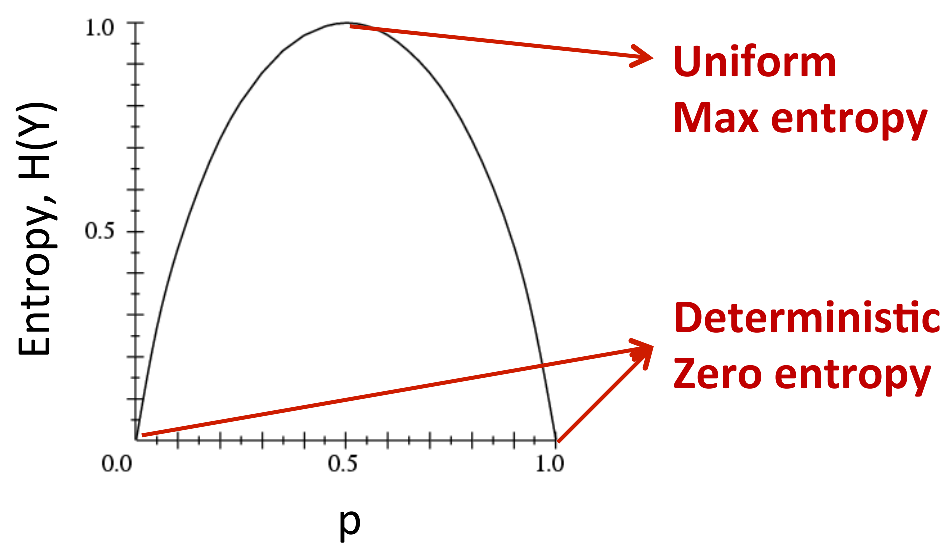
Entropy

- 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)



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

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 best to split?

Pick the attribute/feature which yields maximum information gain:

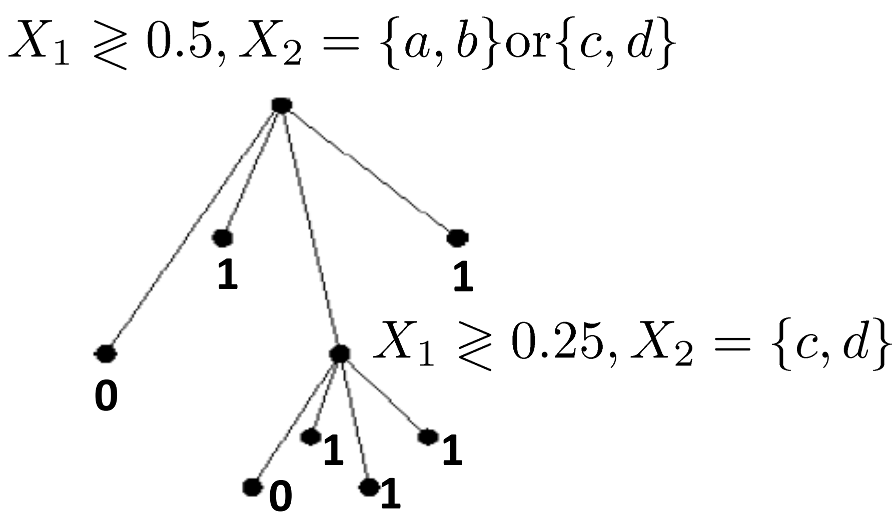
$$\arg\max_i I(Y, X_i) = \arg\max_i [H(Y) - H(Y|X_i)]$$

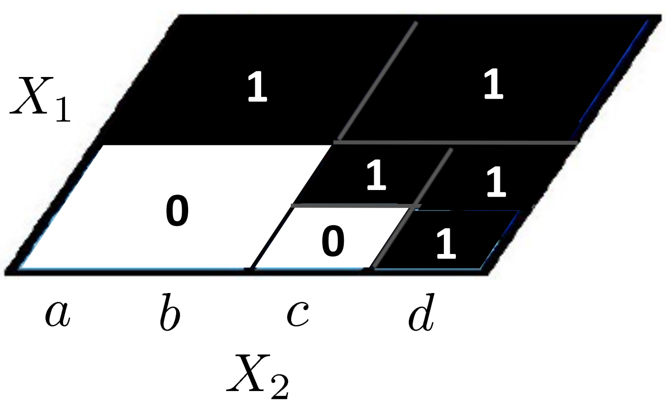
H(Y) – entropy of Y H(Y|X_i) – conditional entropy of Y

Feature which yields maximum reduction in entropy provides maximum information about Y

More generally…

Decision Tree more generally…





- Features can be discrete or continuous
- 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: predict Y

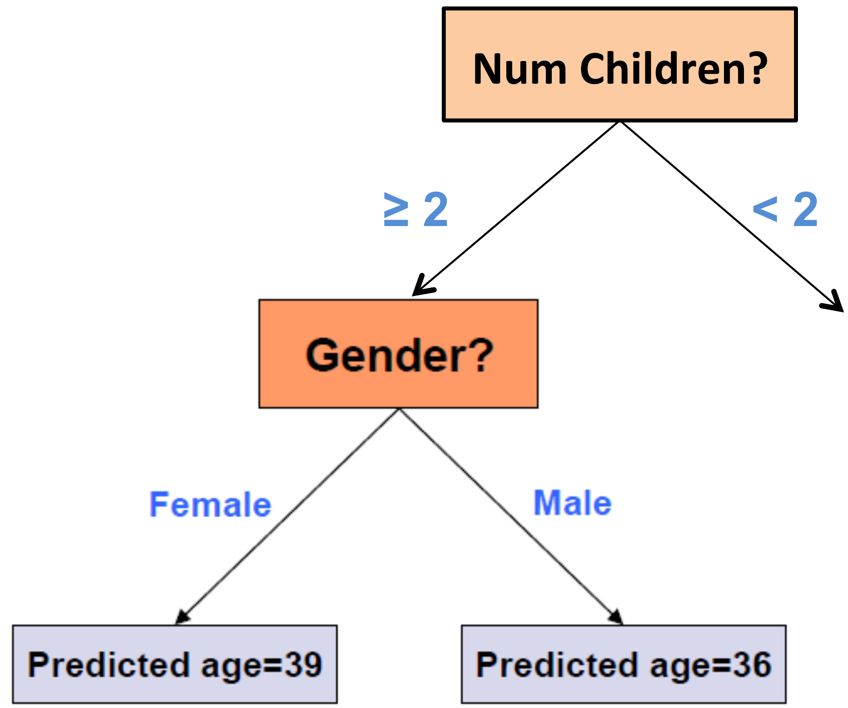
  Majority vote (classification)

  Average or Polynomial fit (regression)

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
:	:	:	:	:



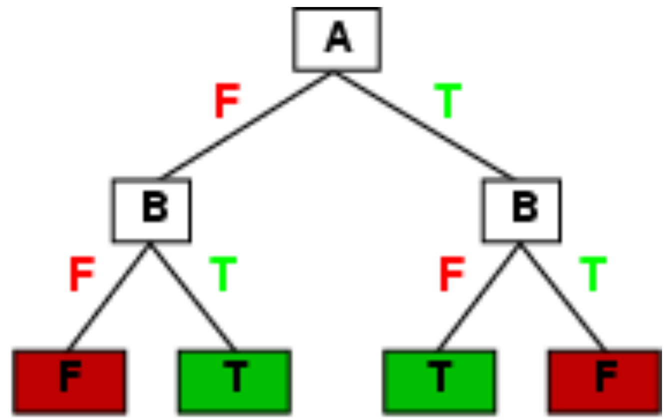
Average (fit a constant) using training data at the leaves

Overfitting

Expressiveness of General Decision Trees

- Decision trees can express any function of the input features.
- E.g., for Boolean features and labels, truth table row → path to leaf:

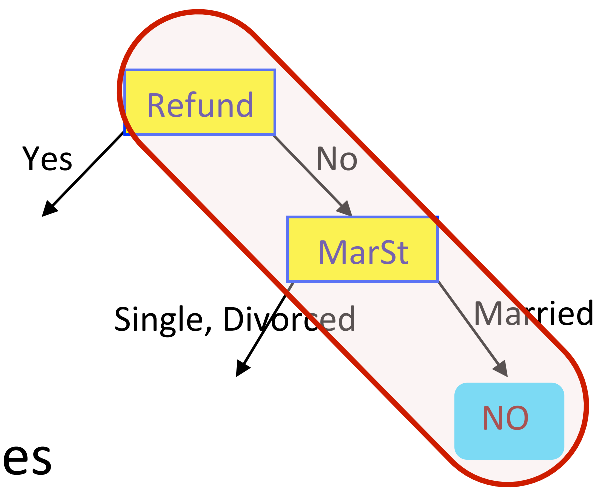
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- There is a decision tree which perfectly classifies a training set with one path to leaf for each example
- 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
    - 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
  - Model Selection by complexity penalization



Model Selection

- Penalize complex models by introducing cost

$$\hat{f} = \arg\min_{T} \Bigg\{ \underbrace{\frac{1}{n}\sum_{j=1}^{n} \text{loss}(\hat{f}_T(X^{(j)}), Y^{(j)})}_{\text{log likelihood}} + \underbrace{\text{pen}(T)}_{\text{cost}} \Bigg\}$$

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

$\text{pen}(T) \propto |T|$ penalize trees with more 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,…)
- 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 model selection