Machine Learning
CSE - 465

Lecture - 06

Lecture 06
Decision Tree

Content credit: Data mining lab, Brigham Young University, Utah, USA

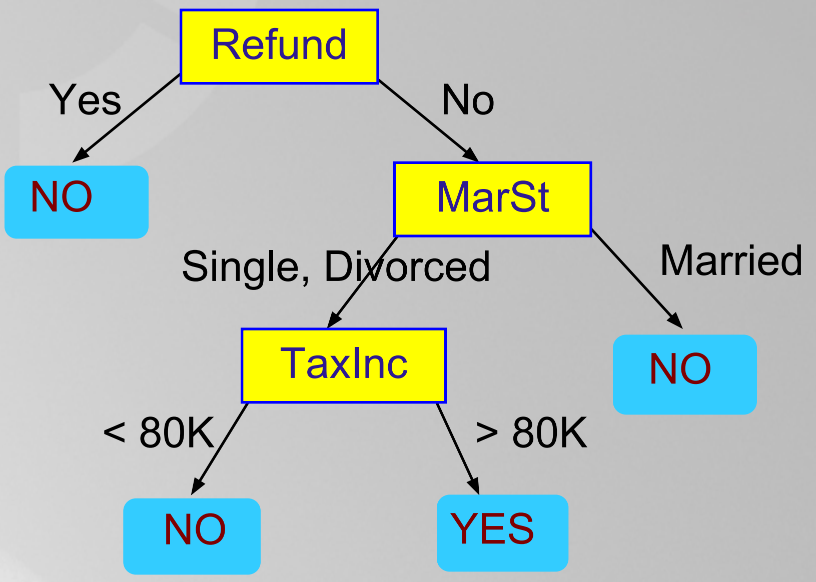
Outline

- Example of a Decision Tree
- ID3 Example
- Entropy
- Information Gain
- Overfitting and Underfitting

Example of a Decision Tree

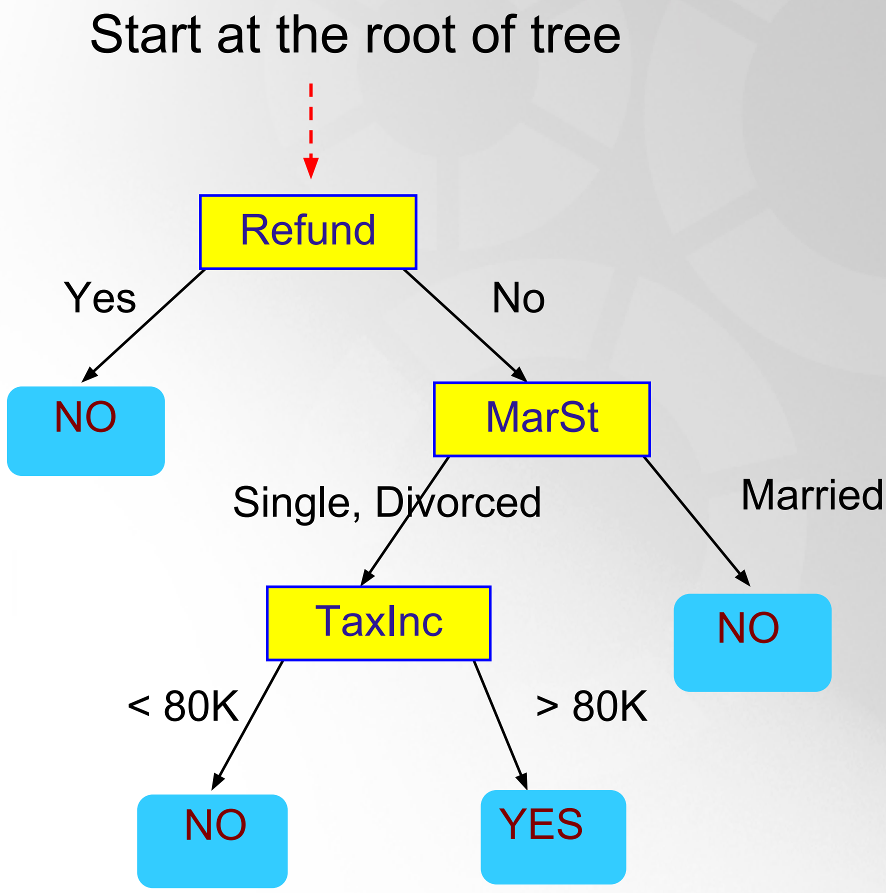
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

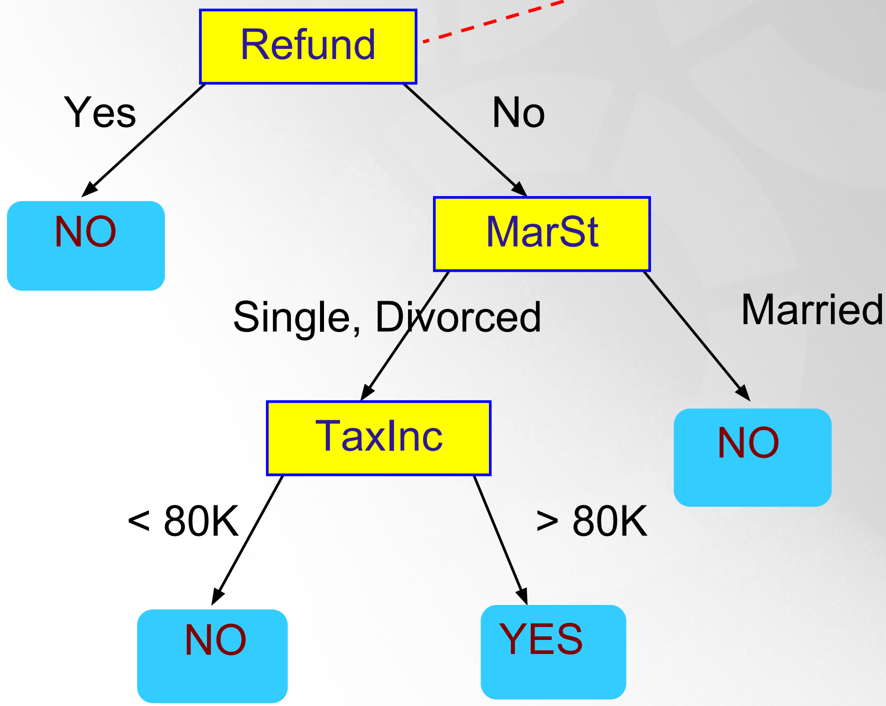
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

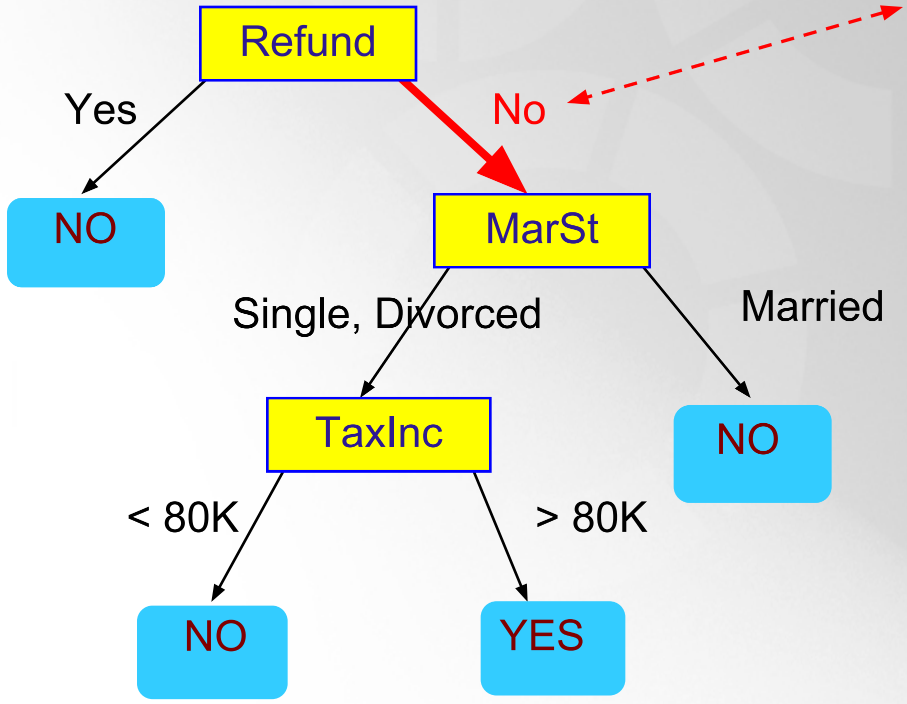
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

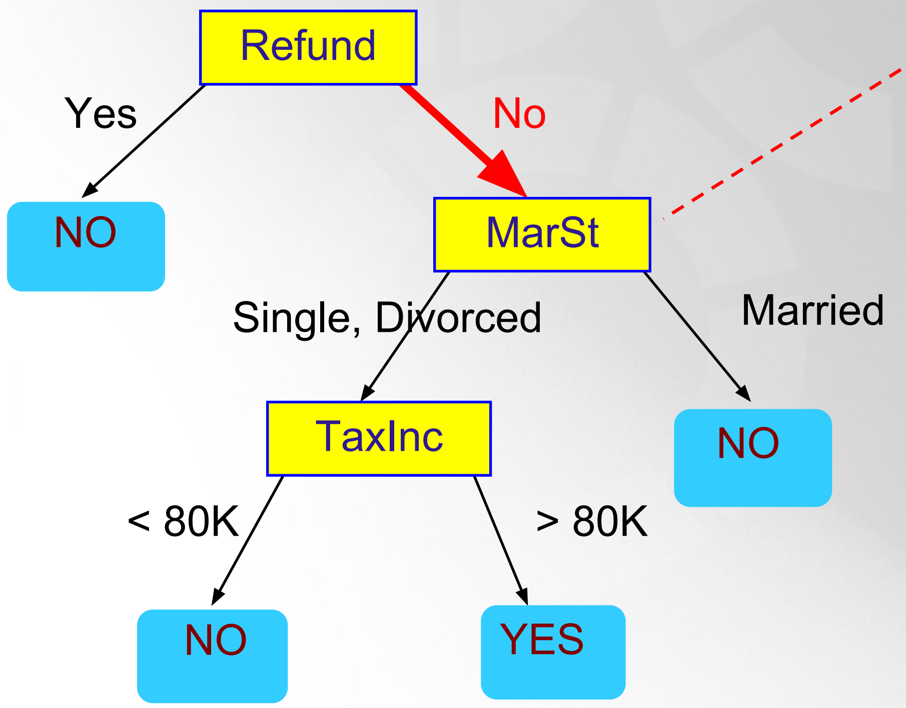
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



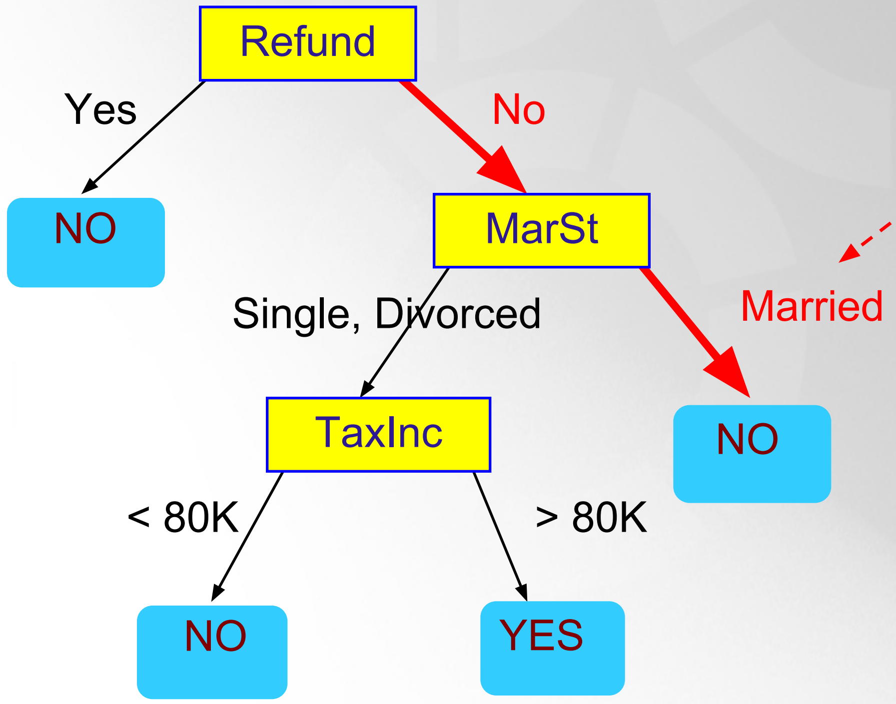
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

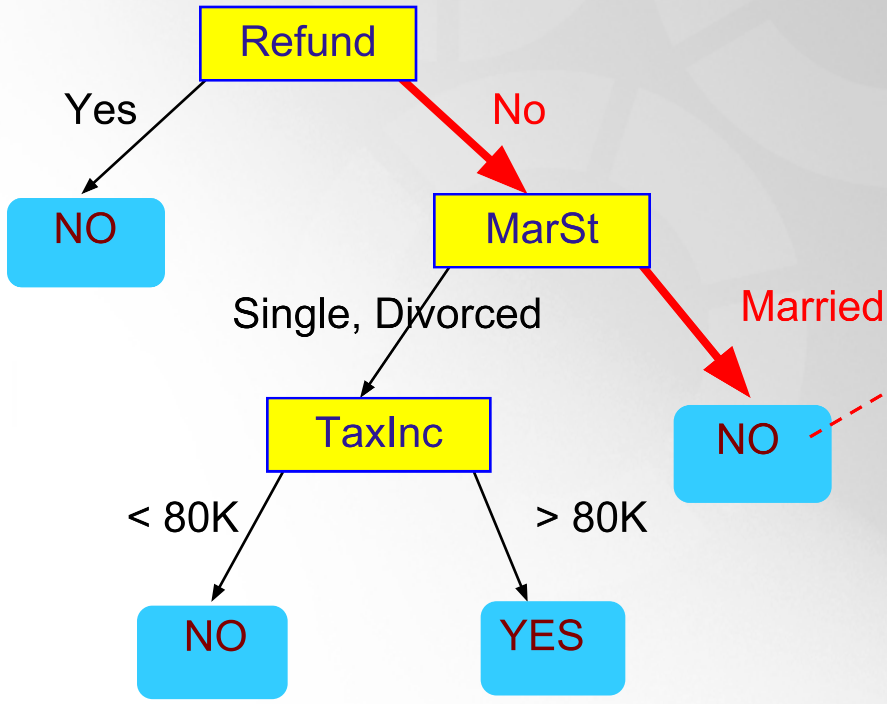
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

Decision Tree Learning: ID3

Function ID3(*Training-set, Attributes*)

- If all elements in *Training-set* are in same class, then return leaf node labeled with that class
- Else if *Attributes* is empty, then return leaf node labeled with majority class in *Training-set*
- Else if *Training-Set* is empty, then return leaf node labeled with default majority class
- Else
  - Select and remove *A* from *Attributes*
  - Make *A* the root of the current tree
  - For each value *V* of *A*
    - Create a branch of the current tree labeled by *V*
    - *Partition_V* ← Elements of *Training-set* with value *V* for *A*
    - Induce-Tree(*Partition_V*, *Attributes*)
    - Attach result to branch *V*

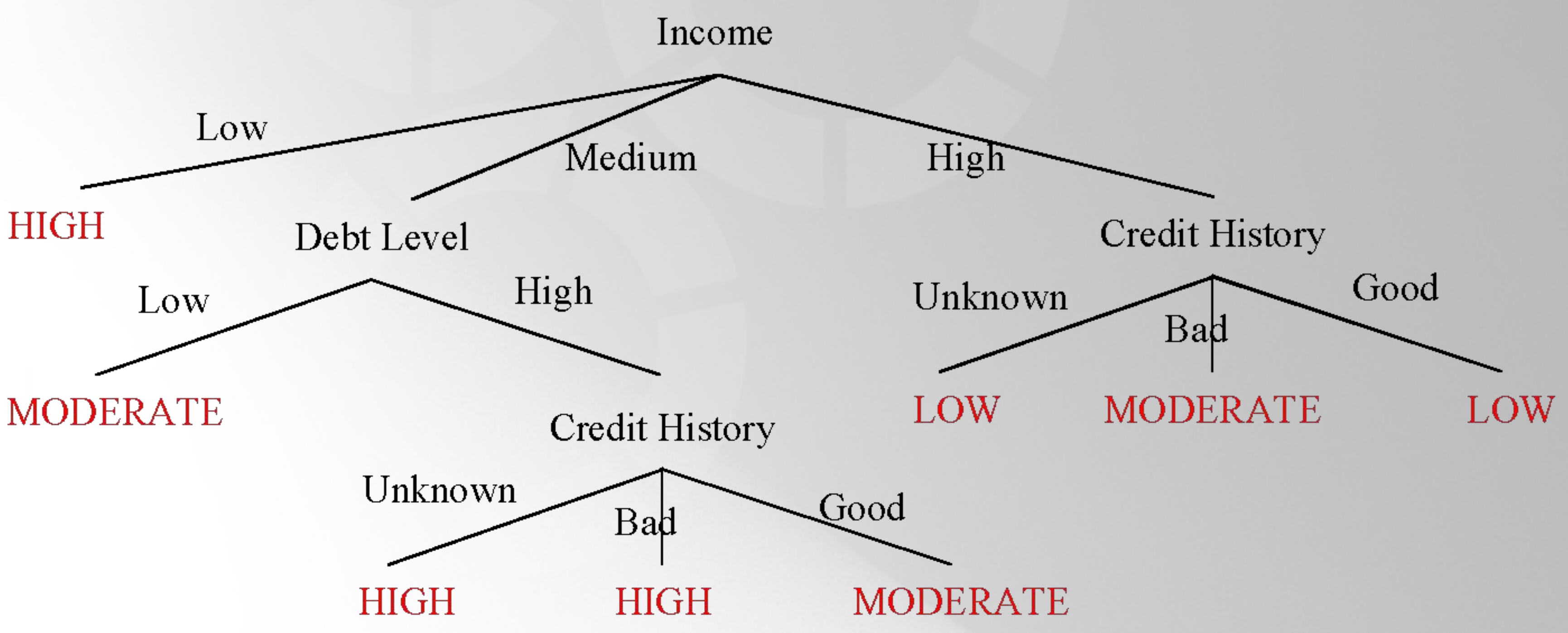
Illustrative Training Set

Risk Assessment for Loan Applications

Client #	Credit History	Debt Level	Collateral	Income Level	RISK LEVEL
1	Bad	High	None	Low	**HIGH**
2	Unknown	High	None	Medium	**HIGH**
3	Unknown	Low	None	Medium	**MODERATE**
4	Unknown	Low	None	Low	**HIGH**
5	Unknown	Low	None	High	**LOW**
6	Unknown	Low	Adequate	High	**LOW**
7	Bad	Low	None	Low	**HIGH**
8	Bad	Low	Adequate	High	**MODERATE**
9	Good	Low	None	High	**LOW**
10	Good	High	Adequate	High	**LOW**
11	Good	High	None	Low	**HIGH**
12	Good	High	None	Medium	**MODERATE**
13	Good	High	None	High	**LOW**
14	Bad	High	None	Medium	**HIGH**

ID3 Example

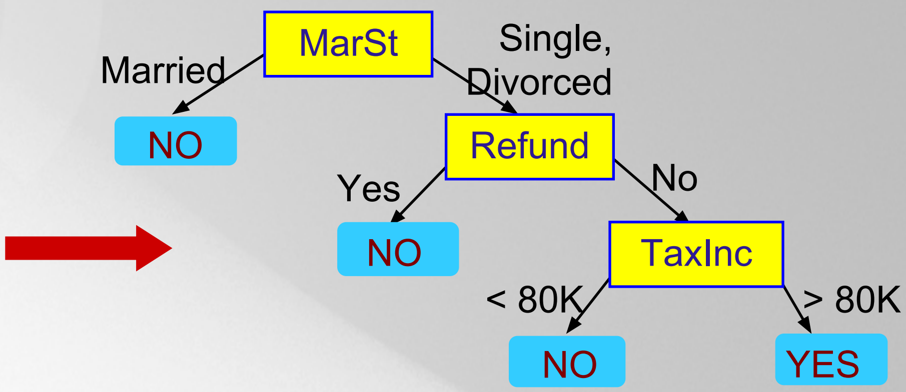
Attach subtrees at appropriate places.



Non-Uniqueness

- Decision trees are not unique:
  - Given a set of training instances *T*, there generally exists a number of decision trees that are consistent with (or fit) *T*

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



ID3's Question

Given a training set, which of all of the decision trees consistent with that training set should we pick?

More precisely:

Given a training set, which of all of the decision trees consistent with that training set has the greatest likelihood of correctly classifying unseen instances of the population?

Entropy (as information)

- Entropy at a given node t:

$$\boxed{Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)}$$

(where $p(j \mid t)$ is the relative frequency of class j at node t)

Based on Shannon's information theory

For simplicity, assume only 2 classes *Yes* and *No*

Assume *t* is a set of messages sent to a receiver that must guess their class

If *p(Yes | t)*=1 (resp., *p(No | t)*=1), then the receiver guesses a new example as *Yes* (resp., *No*). No message need be sent.

If *p(Yes | t)*=*p(No | t)*=0.5, then the receiver cannot guess and must be told the class of a new example. A 1-bit message must be sent.

If 0<*p(Yes | t)*<1, then the receiver needs less than 1 bit on average to know the class of a new example.

Entropy (as homogeneity)

- Think chemistry/physics
  - Entropy is measure of disorder or homogeneity
  - Minimum (0.0) when homogeneous / perfect order
  - Maximum (1.0, in general log C) when most heterogeneous / complete chaos
- In ID3
  - Minimum (0.0) when all records belong to one class, implying most information
  - Maximum (log C) when records are equally distributed among all classes, implying least information
  - Intuitively, the smaller the entropy the purer the partition

Examples of Computing Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Information Gain

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

(where parent Node, p is split into k partitions, and n_i is number of records in partition i)

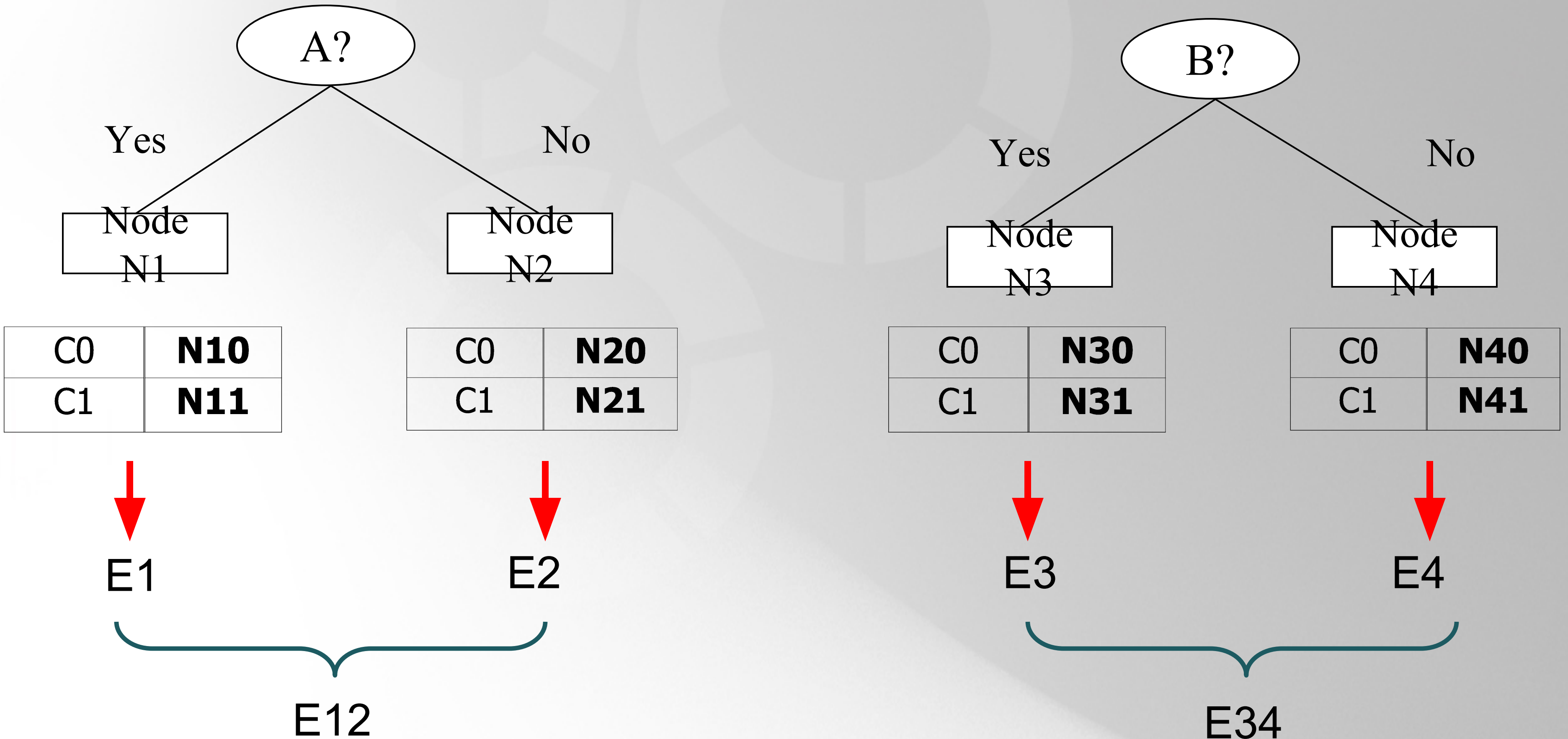
  – Measures reduction in entropy achieved because of the split maximize
  – ID3 chooses to split on the attribute that results in the largest reduction, i.e, (maximizes GAIN)
  – Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Computing Gain

Before Splitting:

C0	**N00**
C1	**N01**

→ E0

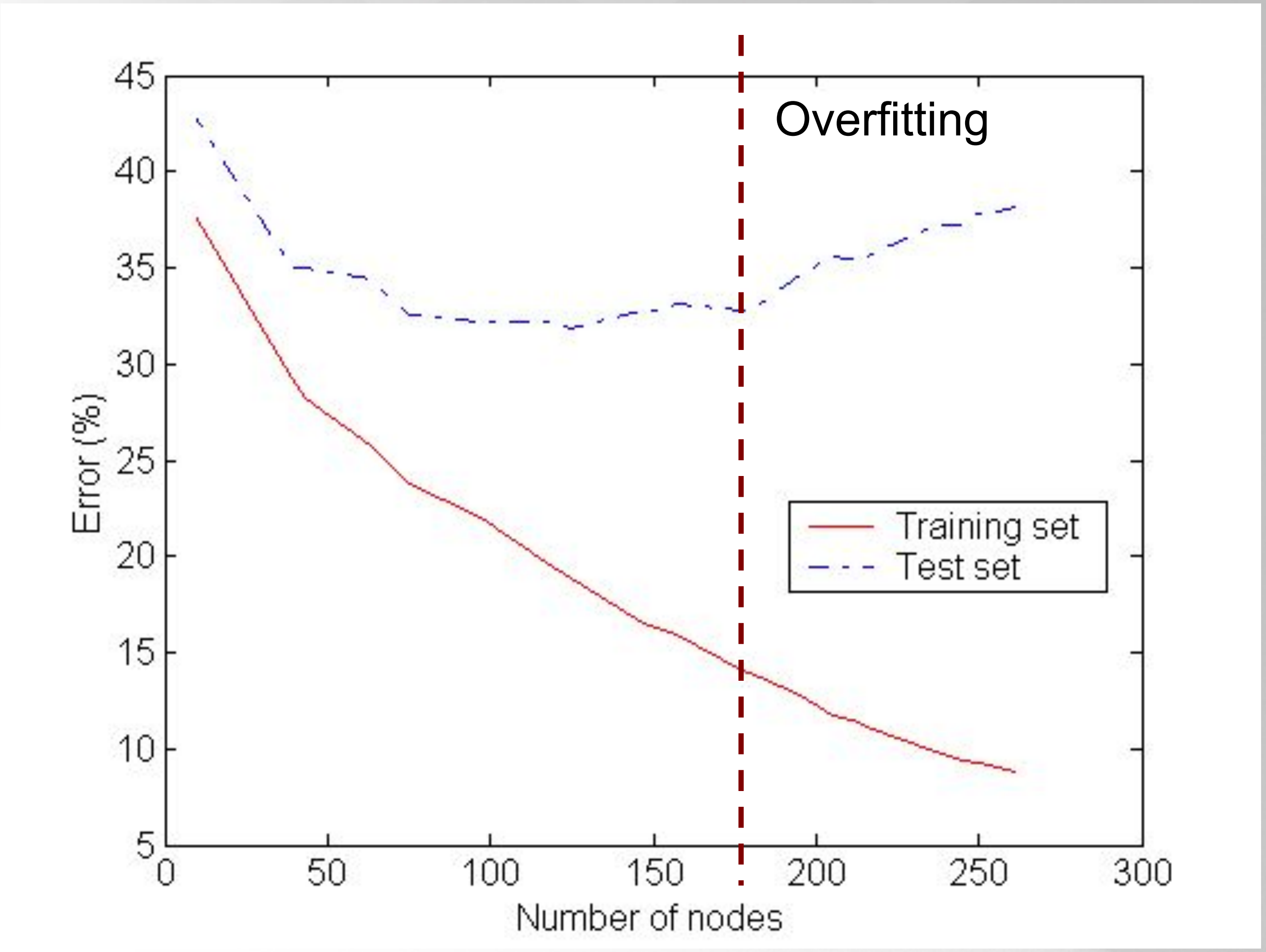


Gain = E0 – E12 vs. E0 – E34

Overfitting and Underfitting

- Overfitting:
  - Given a model space H, a specific model $h \in H$ is said to overfit the training data if there exists some alternative model $h' \in H$, such that h has smaller error than h' over the training examples, but h' has smaller error than h over the entire distribution of instances
- Underfitting:
  - The model is too simple, so that both training and test errors are large

Detecting Overfitting



Overfitting in Decision Tree Learning

- Overfitting results in decision trees that are more complex than necessary
  - Tree growth went too far
  - Number of instances gets smaller as we build the tree (e.g., several leaves match a single example)

- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records

Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Good accuracy
- Disadvantages:
  - Axis-parallel decision boundaries
  - Redundancy
  - Need data to fit in memory
  - Need to retrain with new data

Thank You