Classification

Toon Calders

Technische Universiteit
Eindhoven
University of Technology

Sheets are largely based on the those provided by
Tan, Steinbach, and Kumar. *Introduction to Data Mining*

Where innovation starts

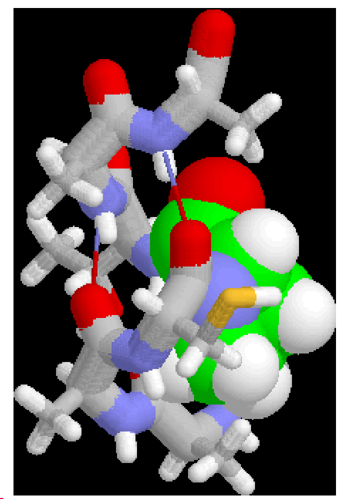
Classification: Definition

- Given a ***training set***
  - Relation over ***attributes***, one of the attributes is the ***class***.
- Find a ***model*** for the class attribute
- That allows for:
  - predicting previously unseen records accurately.

Examples of Classification Task

- Predicting tumor cells as benign or malignant

- Classifying credit card transactions as legitimate or fraudulent



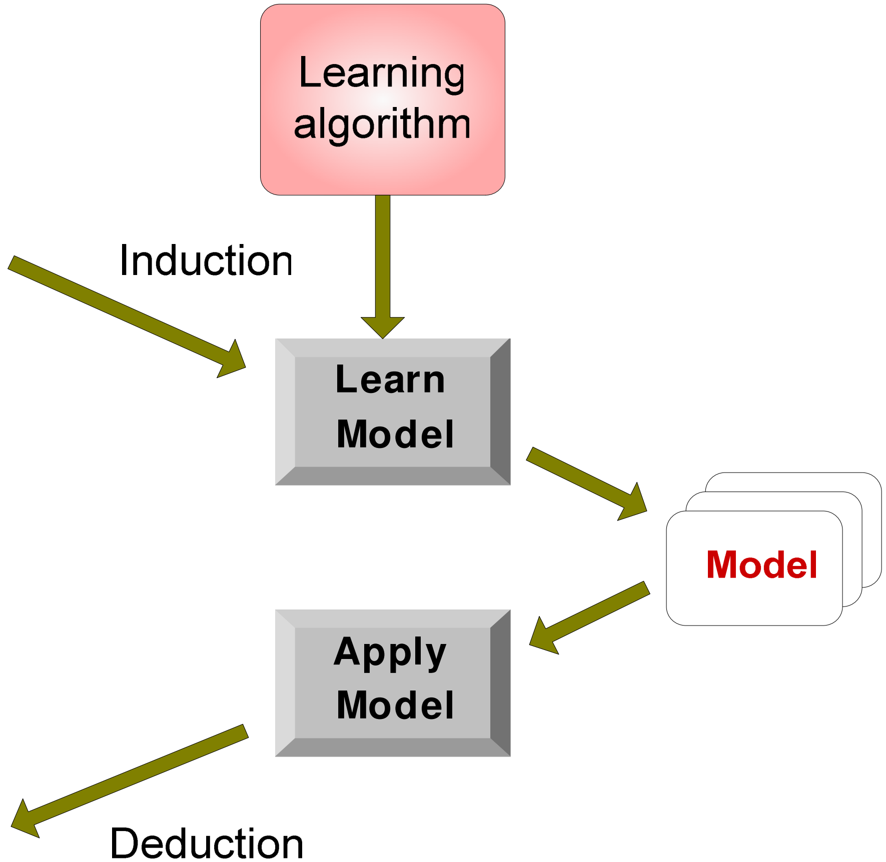
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil

- Categorizing news stories as finance, weather, entertainment, sports, etc

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

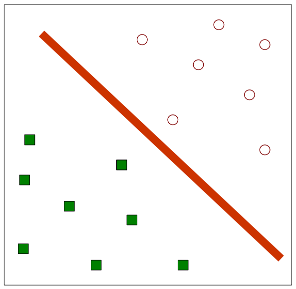
Training Set

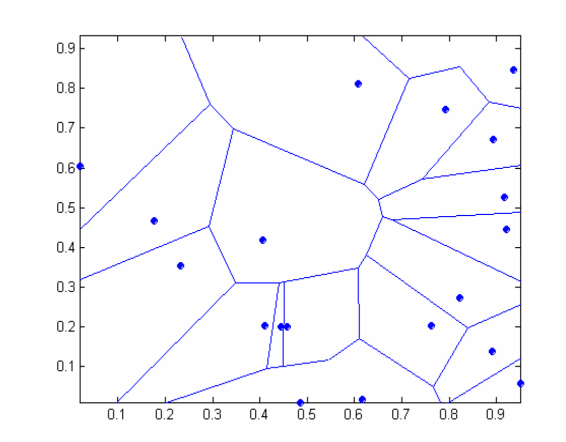


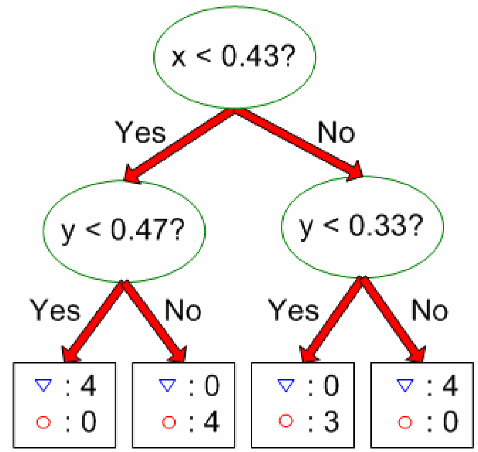
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

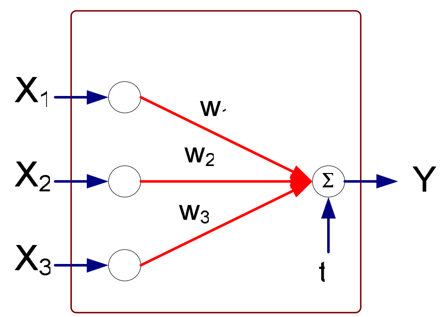
Many different types of models







R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians



Technische Universiteit
Eindhoven
University of Technology

Outline

- K-nearest neighbors
  - Distance measures

- Decision trees
  - *Induction* of a decision tree
  - Hunt's algorithm
  - Issues with decision trees

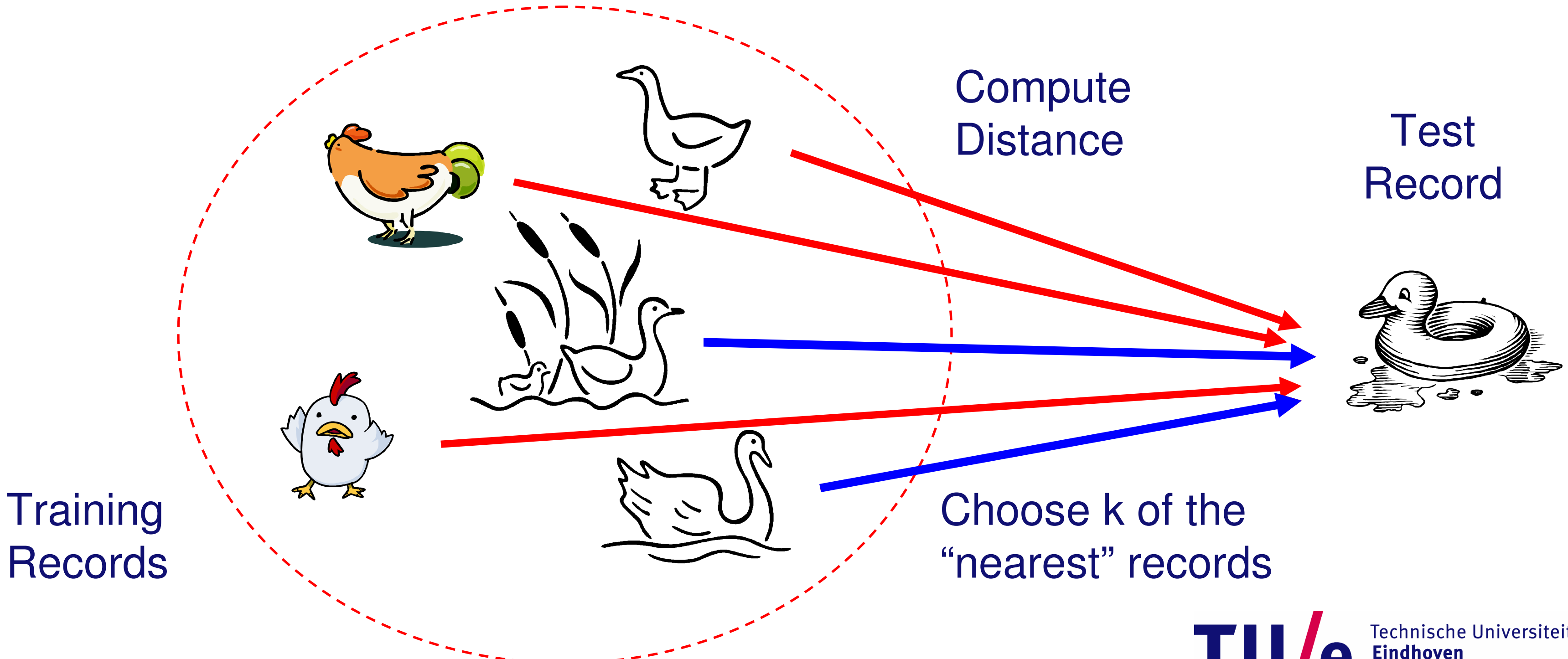
Outline

- K-nearest neighbors
  - Distance measures

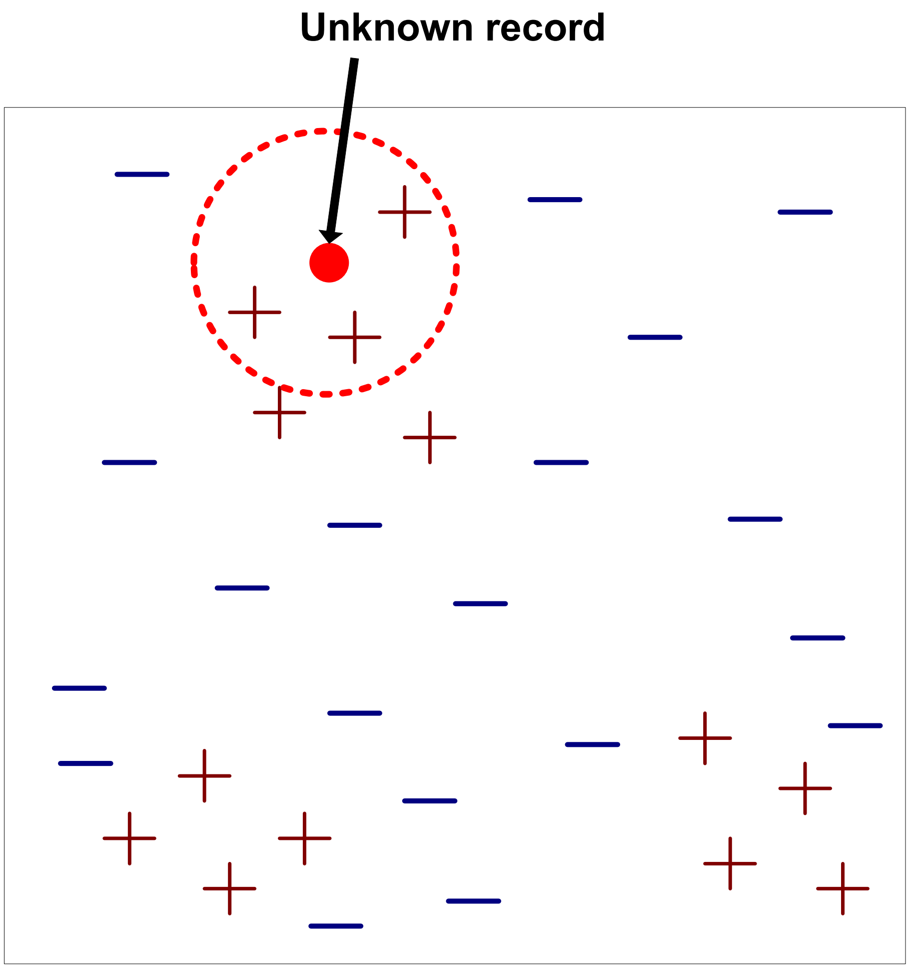
- Decision trees
  - *Induction* of a decision tree
  - Hunt's algorithm
  - Issues with decision trees

Nearest Neighbor Classifiers

- **Basic idea:**
  - **If it walks like a duck, quacks like a duck, then it's probably a duck**



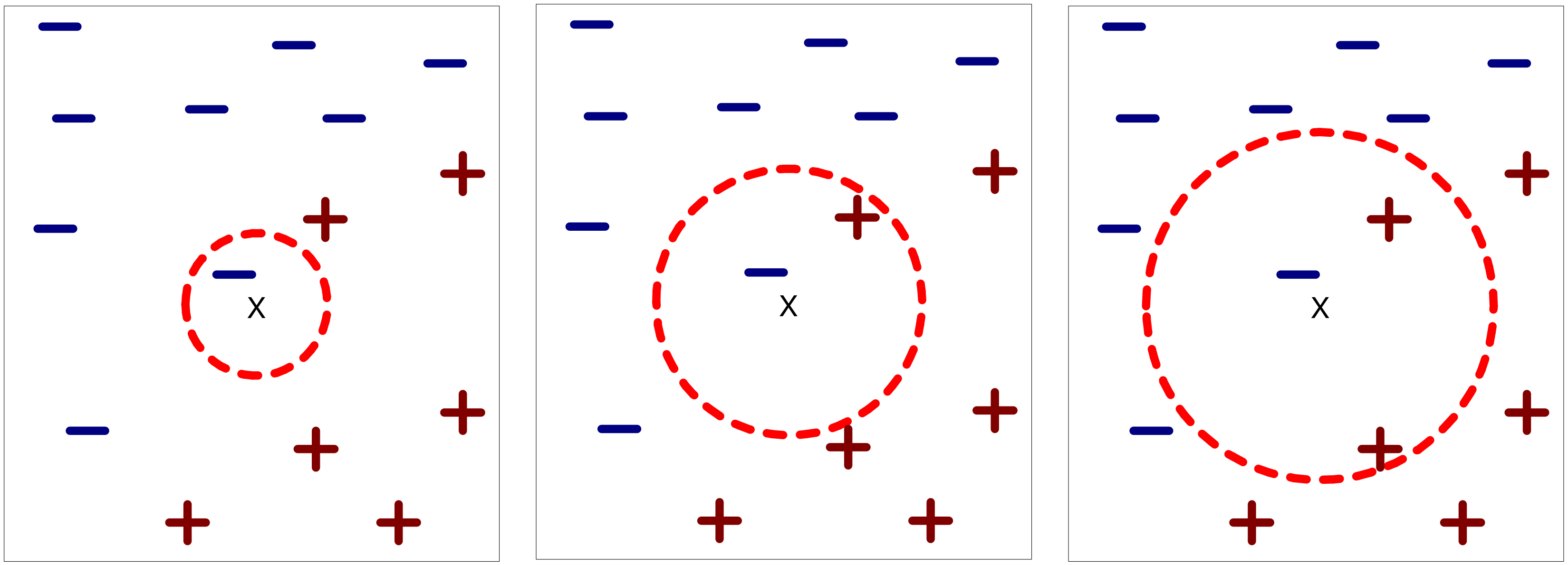
Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record

Definition of Nearest Neighbor

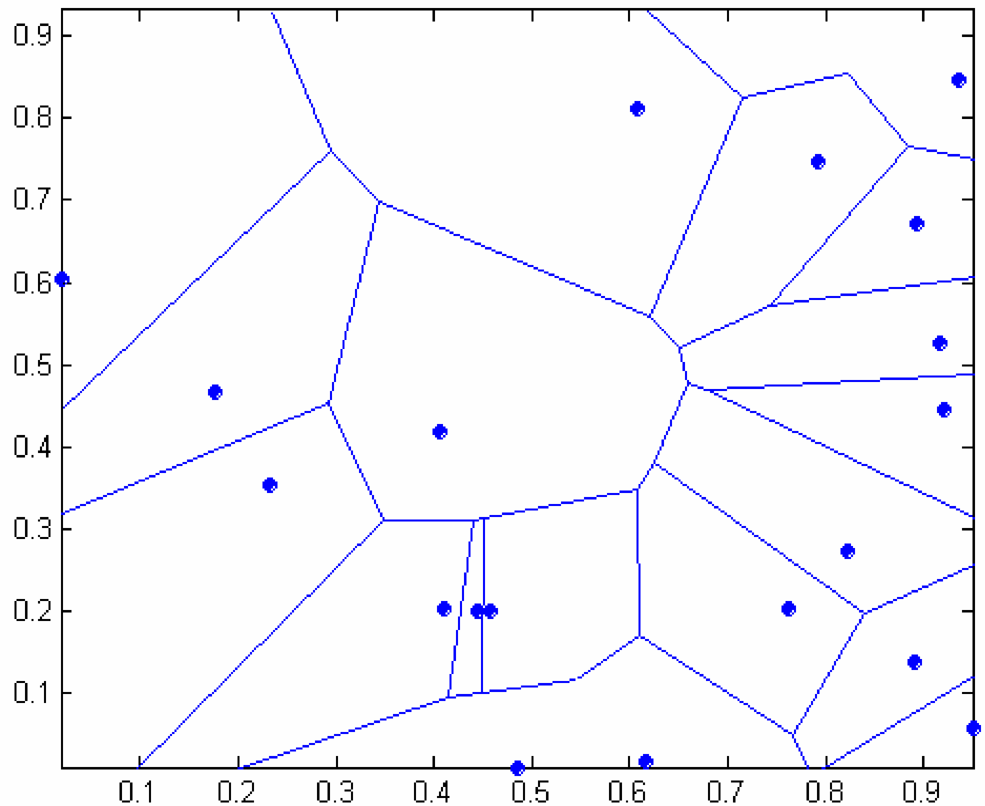


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

1 nearest-neighbor

Voronoi Diagram



Nearest Neighbor Classification

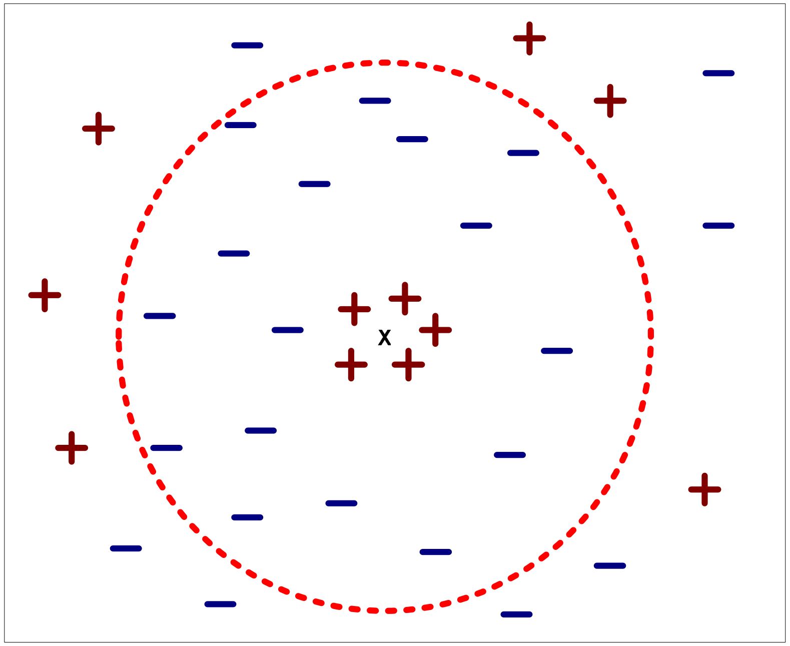
- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification…

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Classification…

- **Problem with Euclidean measure:**
  - **High dimensional data**
    - **curse of dimensionality**
  - **Can produce counter-intuitive results**

1 1 1 1 1 1 1 1 1 1 1 0	vs	1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1		0 0 0 0 0 0 0 0 0 0 0 1
d = 1.4142		d = 1.4142

Outline

- K-nearest neighbors
  - Distance measures
    - Example: for strings and sequences

- Decision trees
  - *Induction* of a decision tree
  - Hunt’s algorithm
  - Issues with Decision trees

Distance Measures …

- Choosing the correct distance function is essential
  - Eucledian, Minkowski
  - Mahalanobis
  - Simple Matching Coefficient
  - Jaccard measure
  - Tanimoto Coefficient
  - Cosine Measure

- Example: distance measure for strings

Edit Distance

- Distance between two strings: minimal number of operations to transform one into another
  - Insert a character
  - Delete a character
  - Replace a character with aother

- Example:
  - paard → paad → parad → parade distance = 3
  - eauivlaent → equivlaent → equivaent → equivalent distance = 3

Edit Distance: Algorithm

	_	P	A	A	R	D
_						
P						
A						
R						
A						
D						
E						

Fill a matrix
entry i,j: edit distance between
t[1..i] en s[1..j]

Edit distance: algoritme

	_	P	A	A	R	D
_	0	1	2	3	4	5
P	1					
A	2					
R	3					
A	4					
D	5					
E	6					

Filling the matrix: recursively

d[i,j] = min { d(i-1, j) + 1 (del)
d(i,j-1) + 1 (ins)
d(i-1,j-1) + cost }
(match of subst.)

Edit distance: algoritme

	_	P	A	A	R	D
_	0	1	2	3	4	5
P	1	0	1	2	3	4
A	2	1	0	1	2	3
R	3	2	1	1	1	2
A	4	3	2	1	2	2
D	5	4	3	2	2	2
E	6	5	4	3	3	3

Distance for DNA Sequences

- Matching in BLAST (Basic Local Alignment and Search Tool) is based on this type of match

- Similarity is defined as the maximal match

```
ATGGCGT
***  !**
ATG-AGT
```

- Not every replacement is equally likely
  - Evolutionary theory

BLOSUM62 Substitution Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	−1	−2	−2	0	−1	−1	0	−2	−1	−1	−1	−1	−2	−1	1	0	−3	−2	0
R	−1	5	0	−2	−3	1	0	−2	0	−3	−2	2	−1	−3	−2	−1	−1	−3	−2	−3
N	−2	0	6	1	−3	0	0	0	1	−3	−3	0	−2	−3	−2	1	0	−4	−2	−3
D	−2	−2	1	6	−3	0	2	−1	−1	−3	−4	−1	−3	−3	−1	0	−1	−4	−3	−3
C	0	−3	−3	−3	9	−3	−4	−3	−3	−1	−1	−3	−1	−2	−3	−1	−1	−2	−2	−1
Q	−1	1	0	0	−3	5	2	−2	0	−3	−2	1	0	−3	−1	0	−1	−2	−1	−2
E	−1	0	0	2	−4	2	5	−2	0	−3	−3	1	−2	−3	−1	0	−1	−3	−2	−2
G	0	−2	0	−1	−3	−2	−2	6	−2	−4	−4	−2	−3	−3	−2	0	−2	−2	−3	−3
H	−2	0	1	−1	−3	0	0	−2	8	−3	−3	−1	−2	−1	−2	−1	−2	−2	2	−3
I	−1	−3	−3	−3	−1	−3	−3	−4	−3	4	2	−3	1	0	−3	−2	−1	−3	−1	3
L	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4	−2	2	0	−3	−2	−1	−2	−1	1
K	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5	−1	−3	−1	0	−1	−3	−2	−2
M	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5	0	−2	−1	−1	−1	−1	1
F	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6	−4	−2	−2	1	3	−1
P	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7	−1	−1	−4	−3	−2
S	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4	1	−3	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5	−2	−2	0
W	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11	2	−3
Y	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	−1
V	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

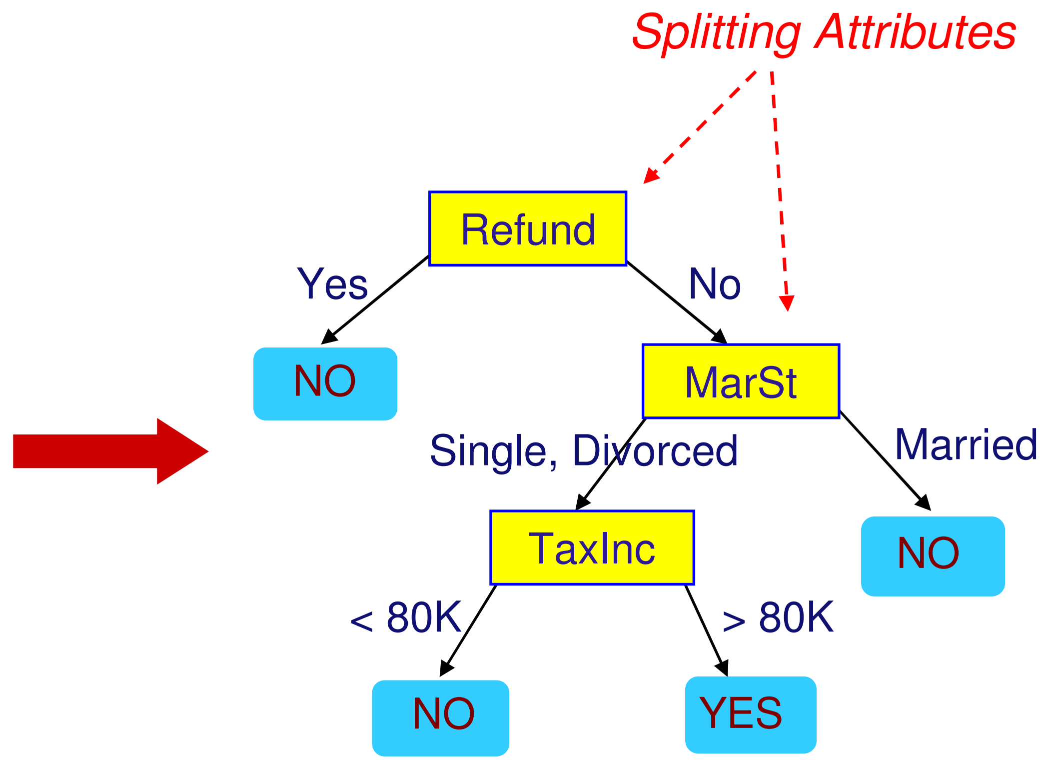
Outline

- K-nearest neighbors
  - Distance measures

- Decision trees
  - *Induction* of a decision tree
  - Hunt's algorithm
    – Local optimal criterion
    – Gini-Index
  - Issues with decision trees

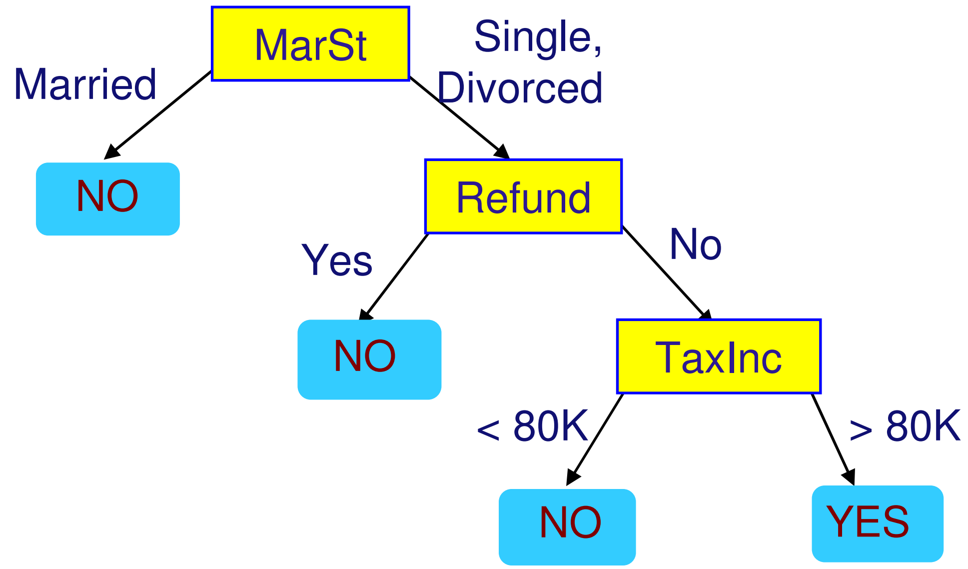
Example of a Decision Tree

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



Another Example of 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**

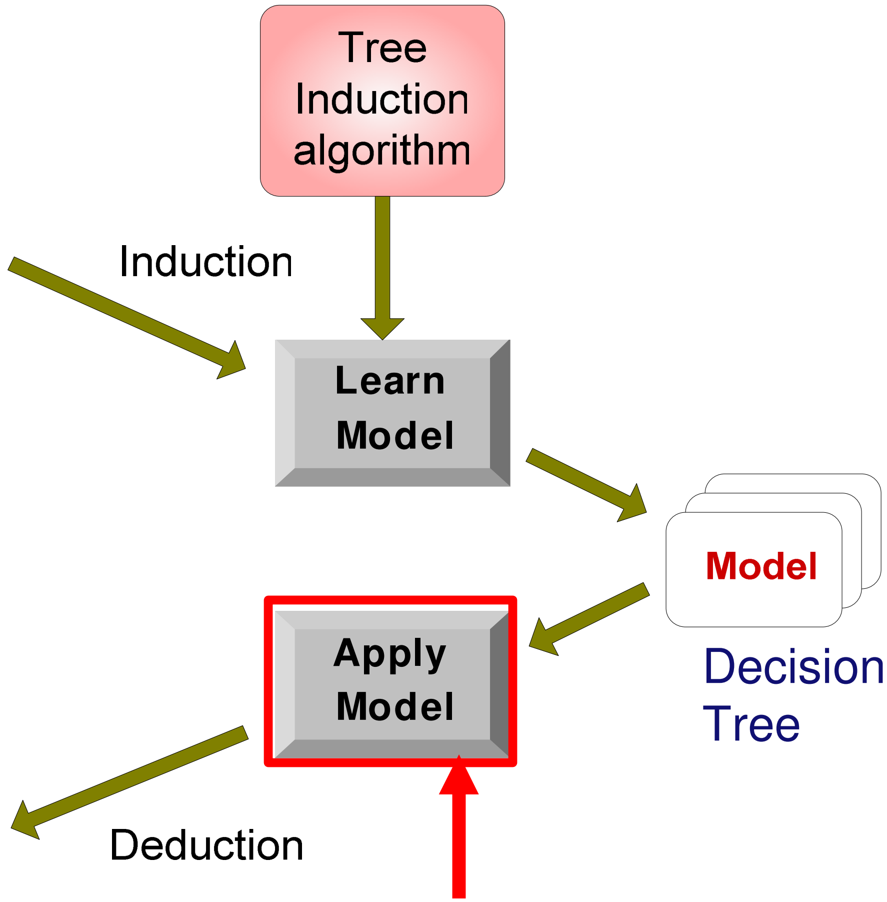


There could be more than one tree that fits the same data!

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

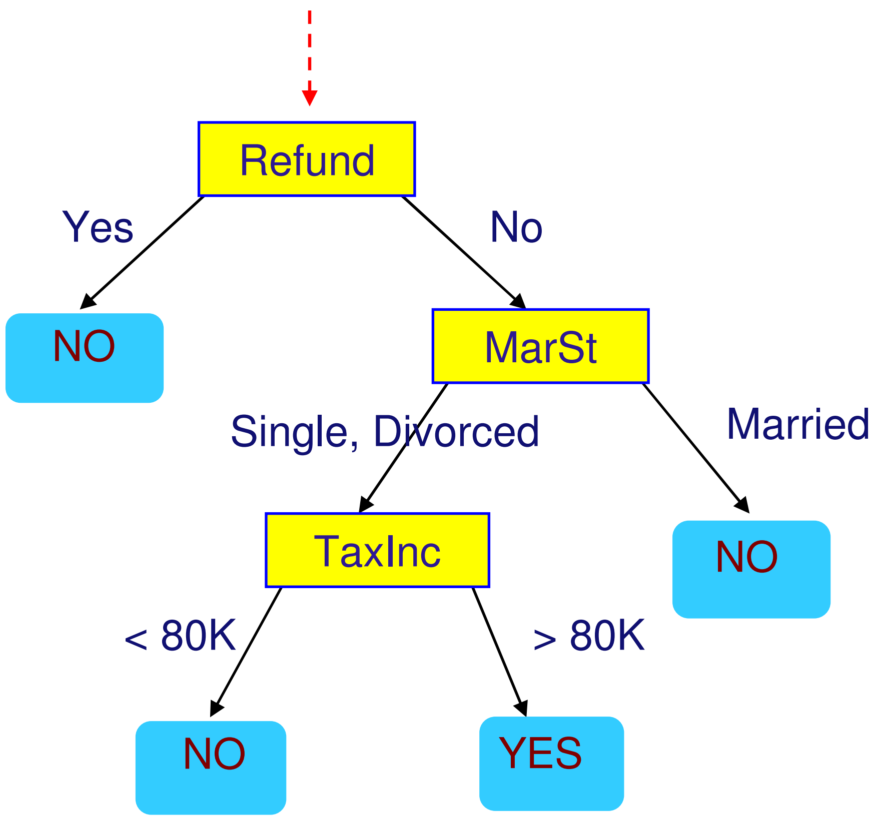


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

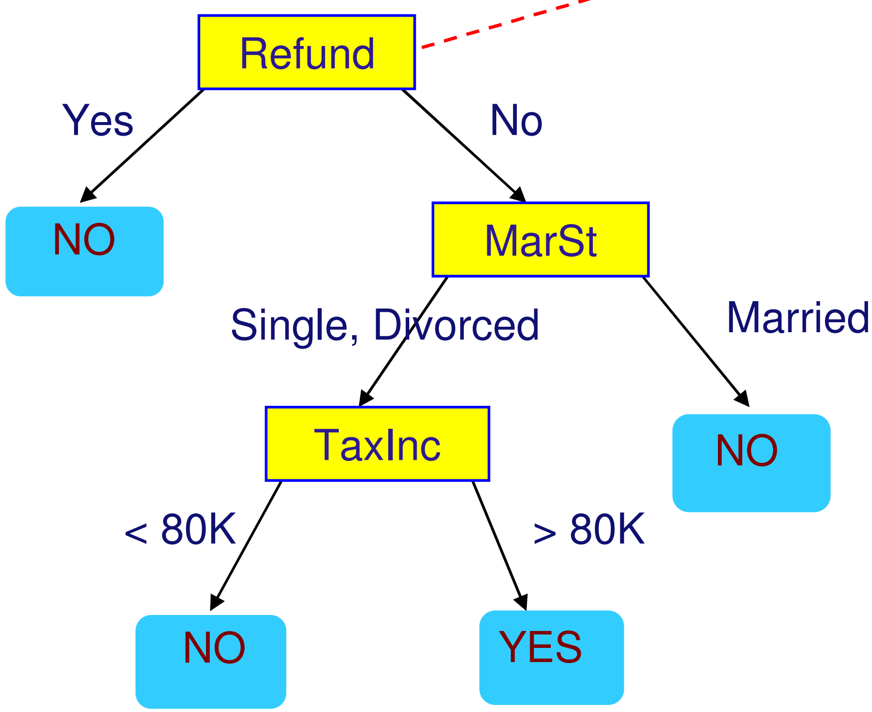
Apply Model to Test Data

Start from the root of tree.



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

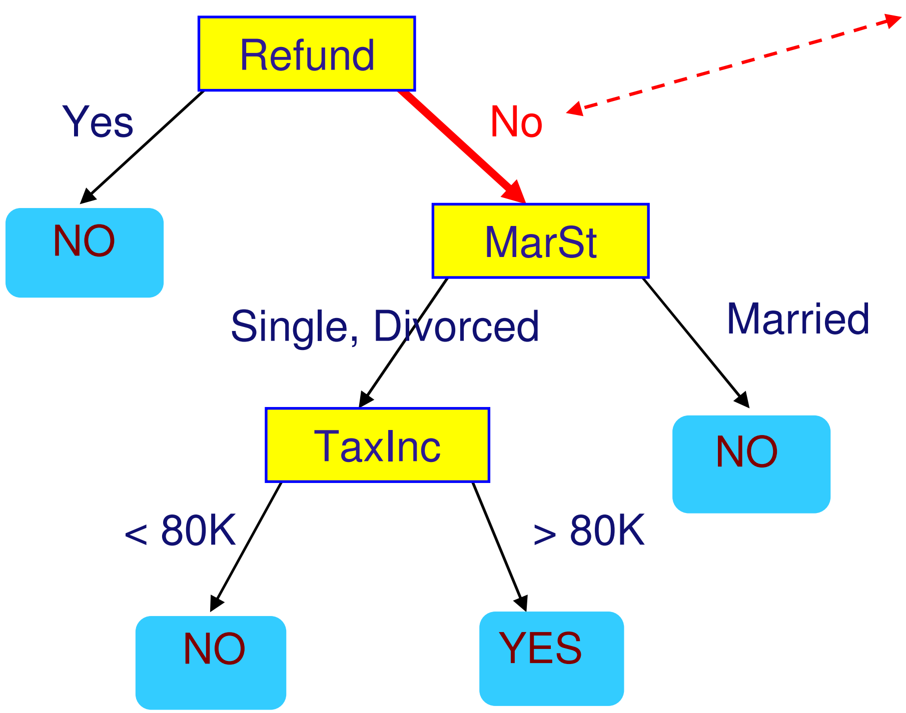


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

TU/e Technische Universiteit Eindhoven University of Technology

Apply Model to Test Data

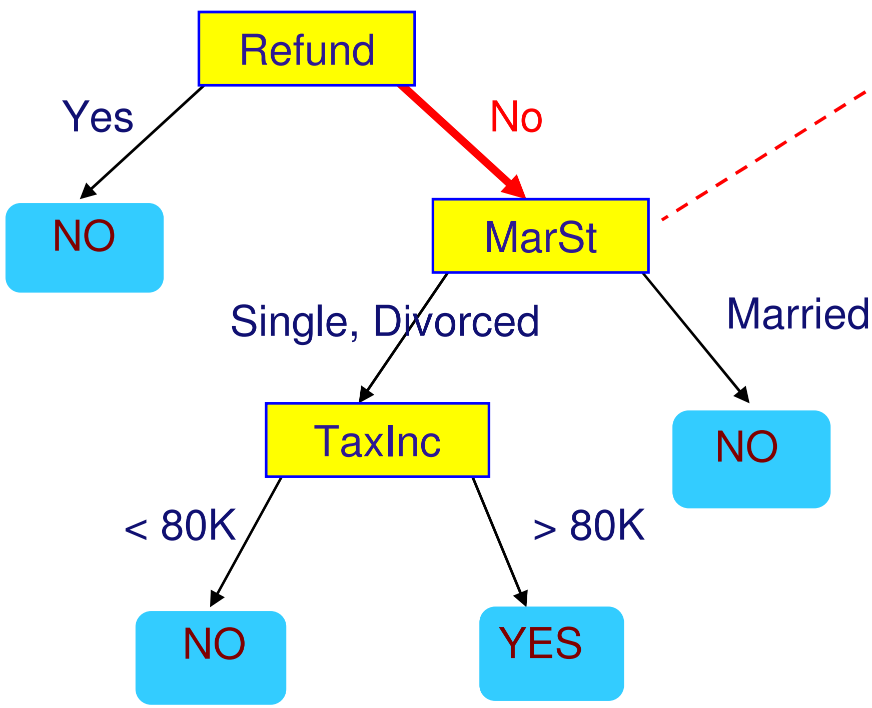
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Technische Universiteit
Eindhoven
University of Technology

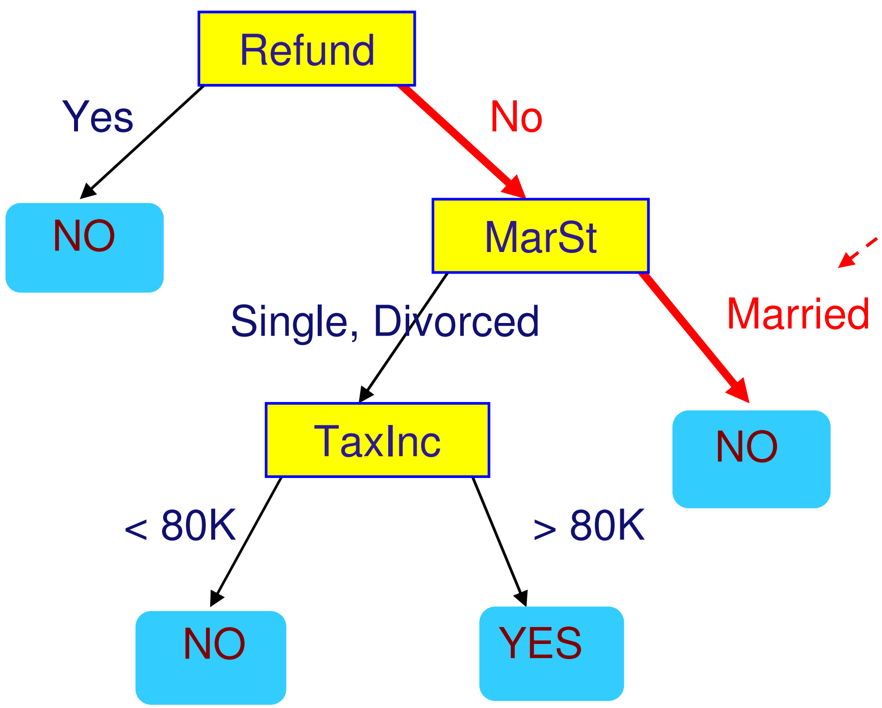
Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



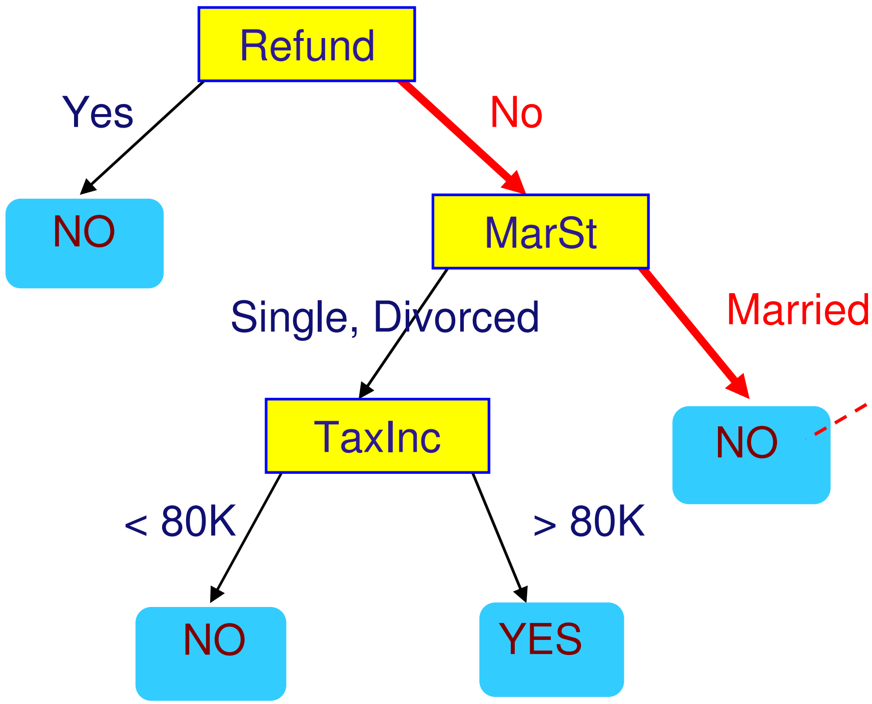
Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

Outline

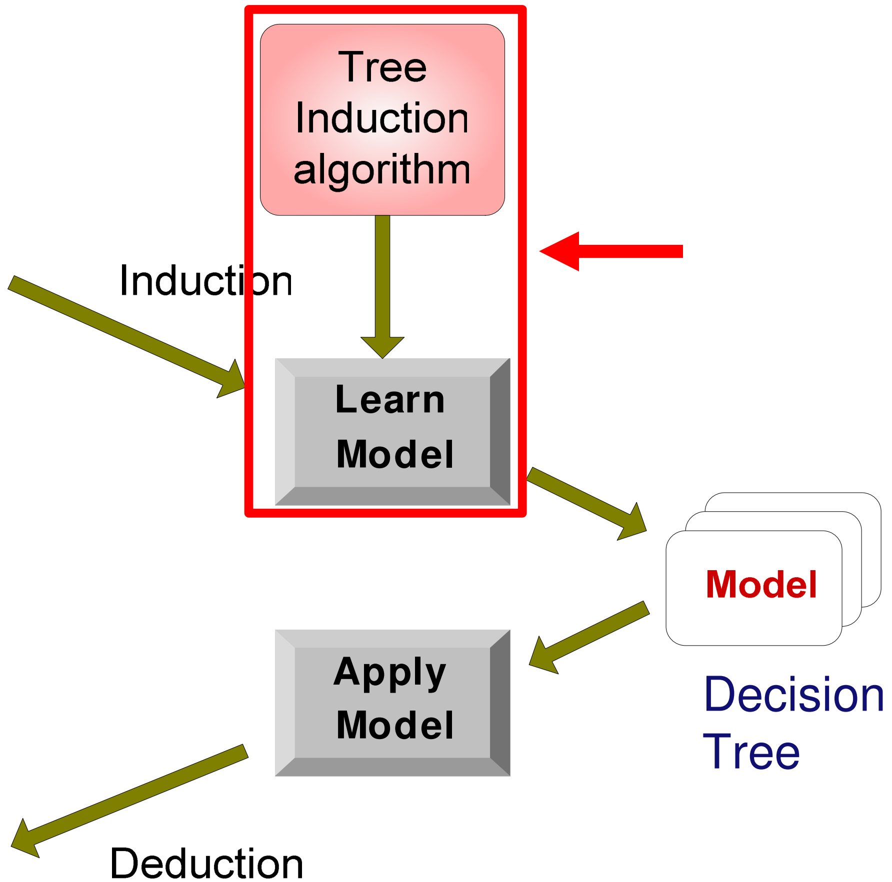
- K-nearest neighbors
  - Distance measures

- Decision trees
  - *Induction* of a decision tree
  - Hunt’s algorithm
    - Local optimal criterion
    - Gini-index
  - Issues with decision trees

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

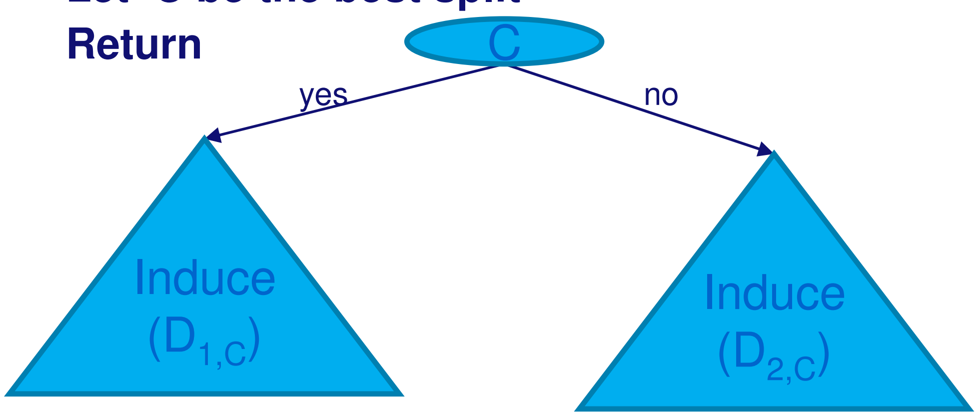
Test Set

General Structure of Hunt's Algorithm

Input: Dataset D
Output: Decision tree t

Induce(D):

- **If all tuples t in D have label + then**
  - **return** +
- **If all tuples t in D have label - then**
  - **return** -
- **For all split criteria C:**
  - $\mathbf{D_{1,C} = \{ t \text{ in } D \mid t \text{ satisfies } C\}}$
  - $\mathbf{D_{2,C} = D - D_1}$
  - **Measure Quality(D_1, D_2)**
- **Let C be the best split**
- **Return**



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

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

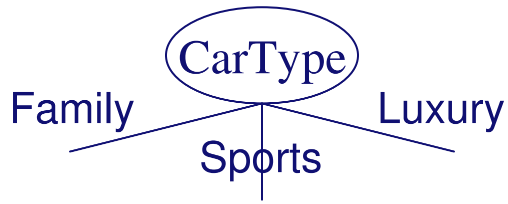
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

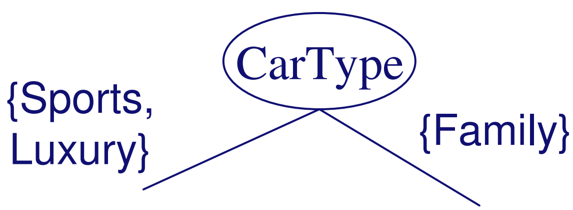
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

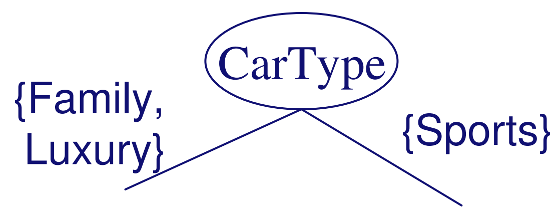
- **Multi-way split:** Use as many partitions as distinct values.



- **Binary split:** Divides values into two subsets. Need to find optimal partitioning.

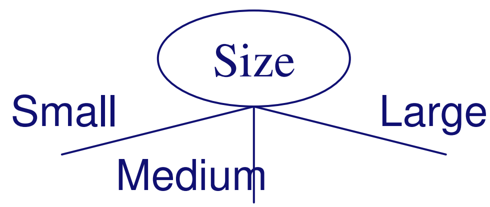


OR

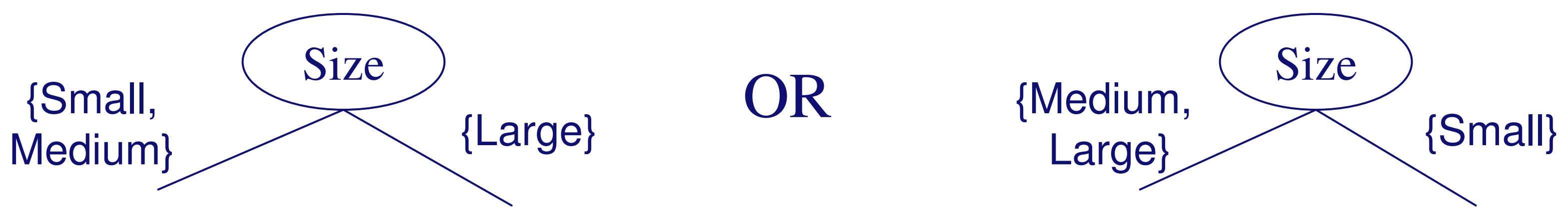


Splitting Based on Ordinal Attributes

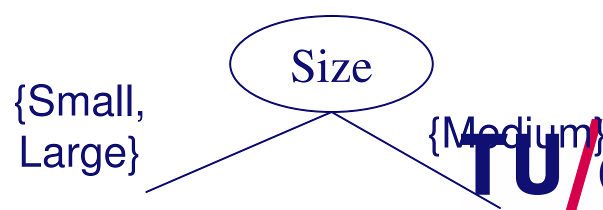
- **Multi-way split:** **Use as many partitions as distinct values.**



- **Binary split:** **Divides values into two subsets. Need to find optimal partitioning.**



- **What about this split?**

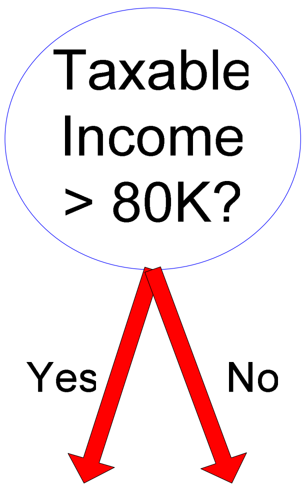


Splitting Based on Continuous Attributes

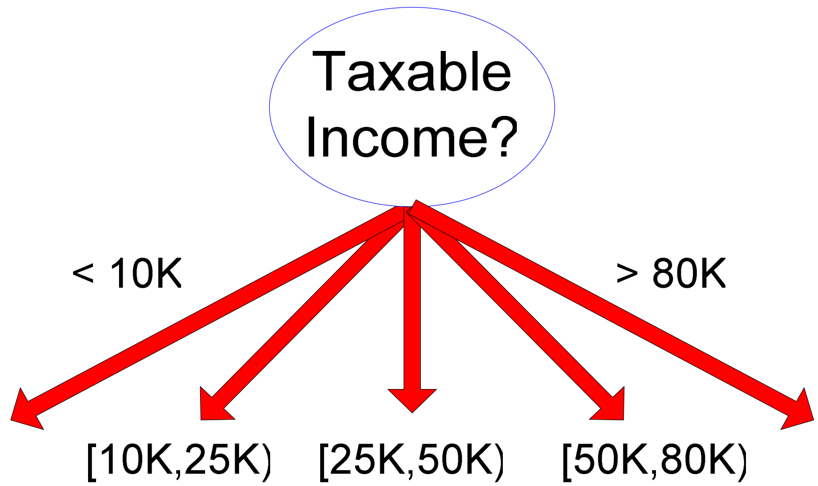
- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

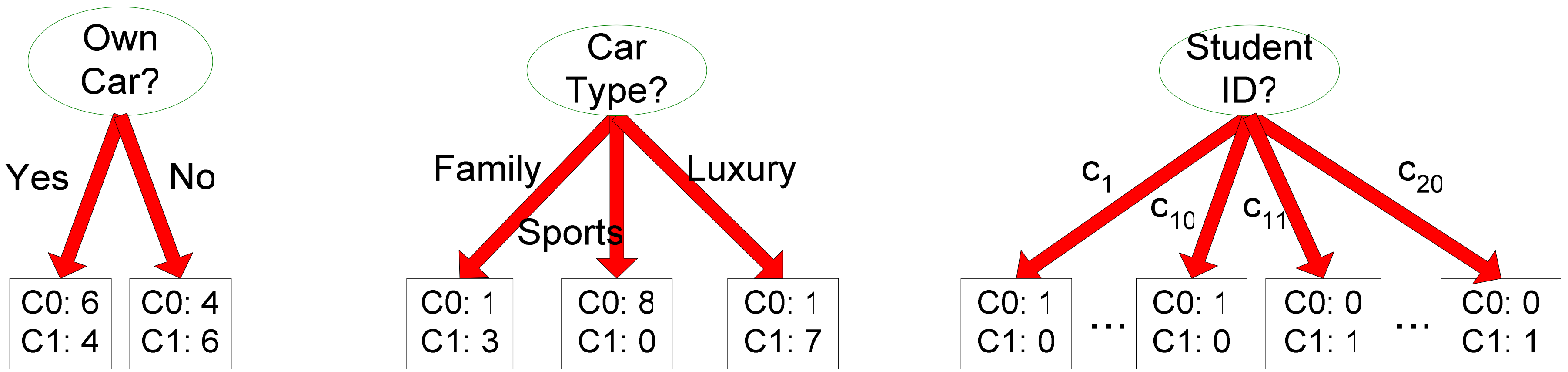
Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

How to determine the Best Split

- **Greedy approach:**
  - **Nodes with homogeneous class distribution are preferred**
- **Need a measure of node impurity:**

C0: 5	C0: 9
C1: 5	C1: 1

Non-homogeneous,

High degree of impurity

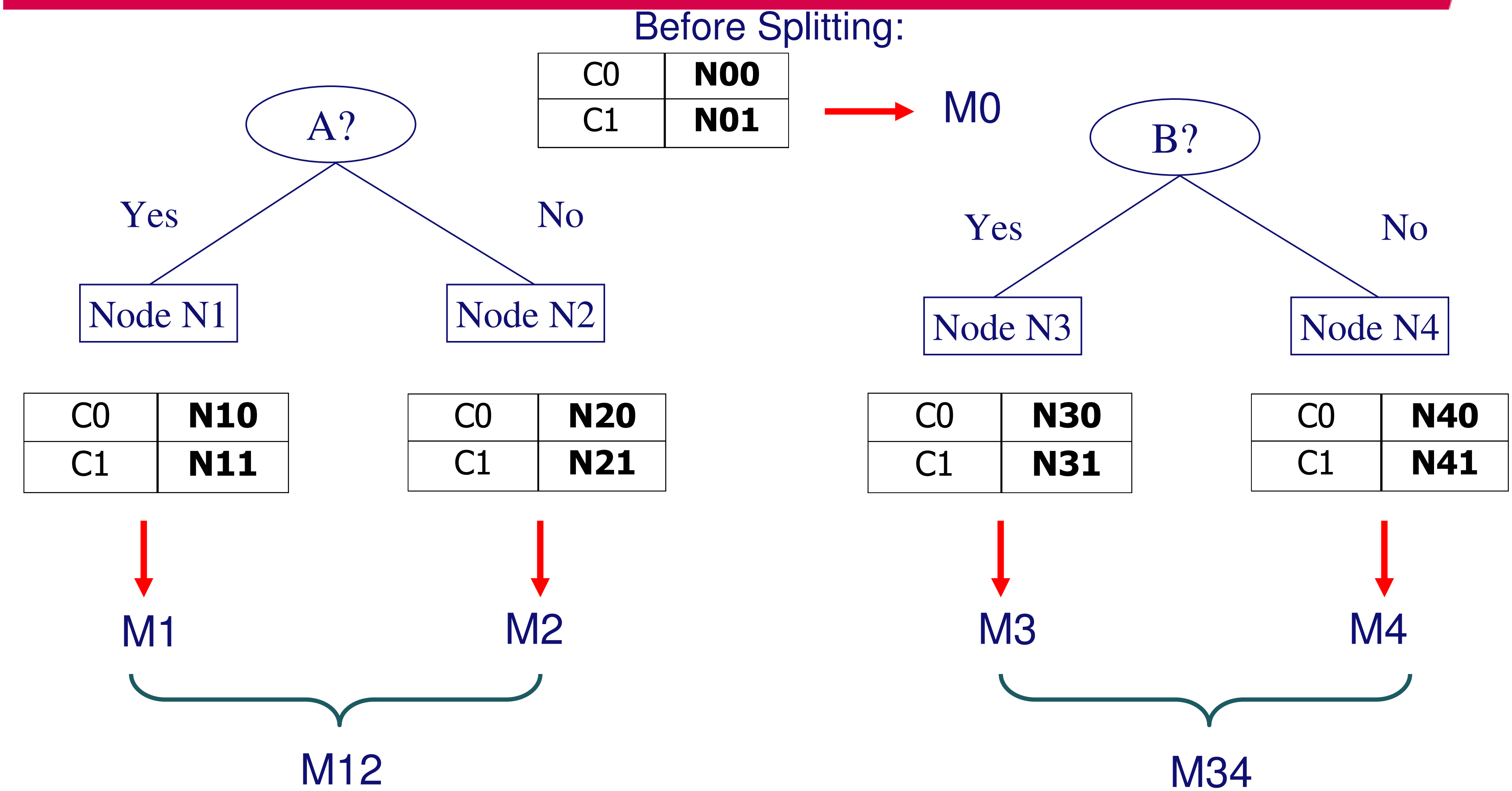
Homogeneous,

Low degree of impurity

Measures of Node Impurity

- **Gini Index**
- **Entropy**
- **Misclassification error**

How to Find the Best Split



Gain = M0 – M12 vs M0 – M34

Measure of Impurity: GINI

- **Gini Index for a given node t :**

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

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

- **Maximum $(1 - 1/n_c)$ (records equally distributed)**

- **Minimum 0 (all records in one class)**

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

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

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

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

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

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

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

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

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

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Splitting Based on GINI

- **Used in CART, SLIQ, SPRINT.**
- **When a node p is split into k partitions (children), the quality of split is computed as,**

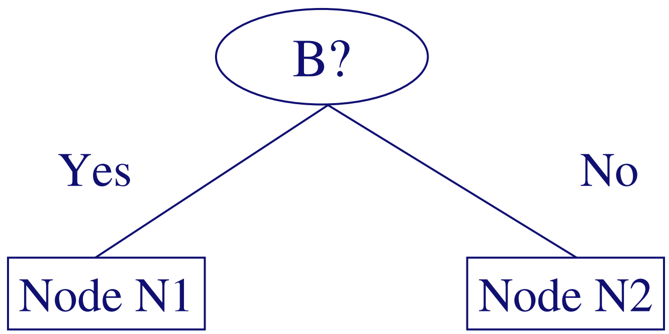
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i **= number of records at child i,**

n **= number of records at node p.**

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



	Parent
C1	**6**
C2	**6**
Gini = 0.500	

Gini(N1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.408$

Gini(N2)
$= 1 - (1/5)^2 - (4/5)^2$
$= 0.32$

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

Gini(Children)
$= 7/12 * 0.408 +$
$5/12 * 0.32$
$= 0.371$

Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Outline

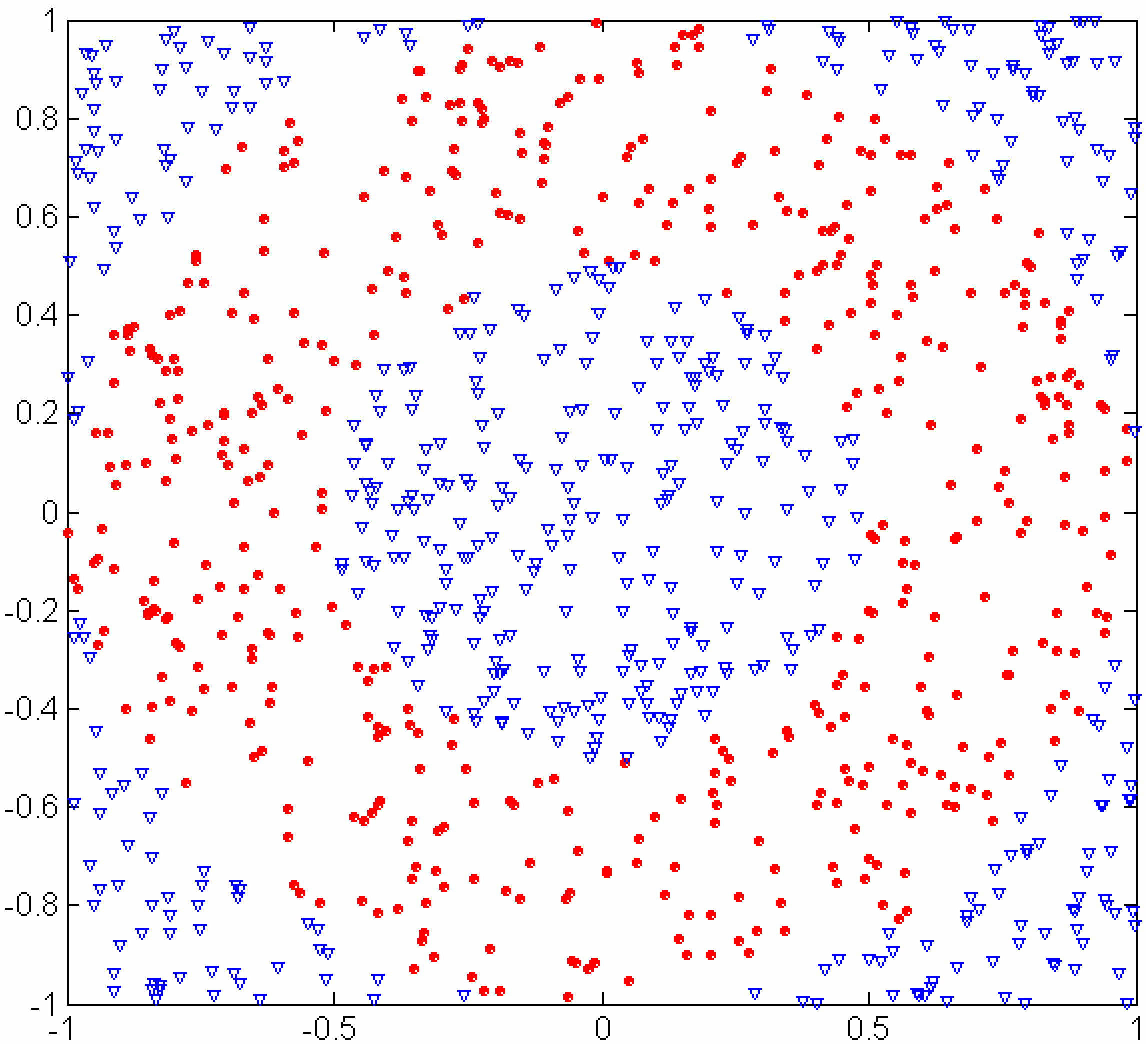
- K-nearest neighbors
  - Distance measures

- Decision trees
  - *Induction* of a decision tree
  - Hunt’s algorithm
    - Local optimal criterion
    - Gini-index
  - Issues with decision trees

Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

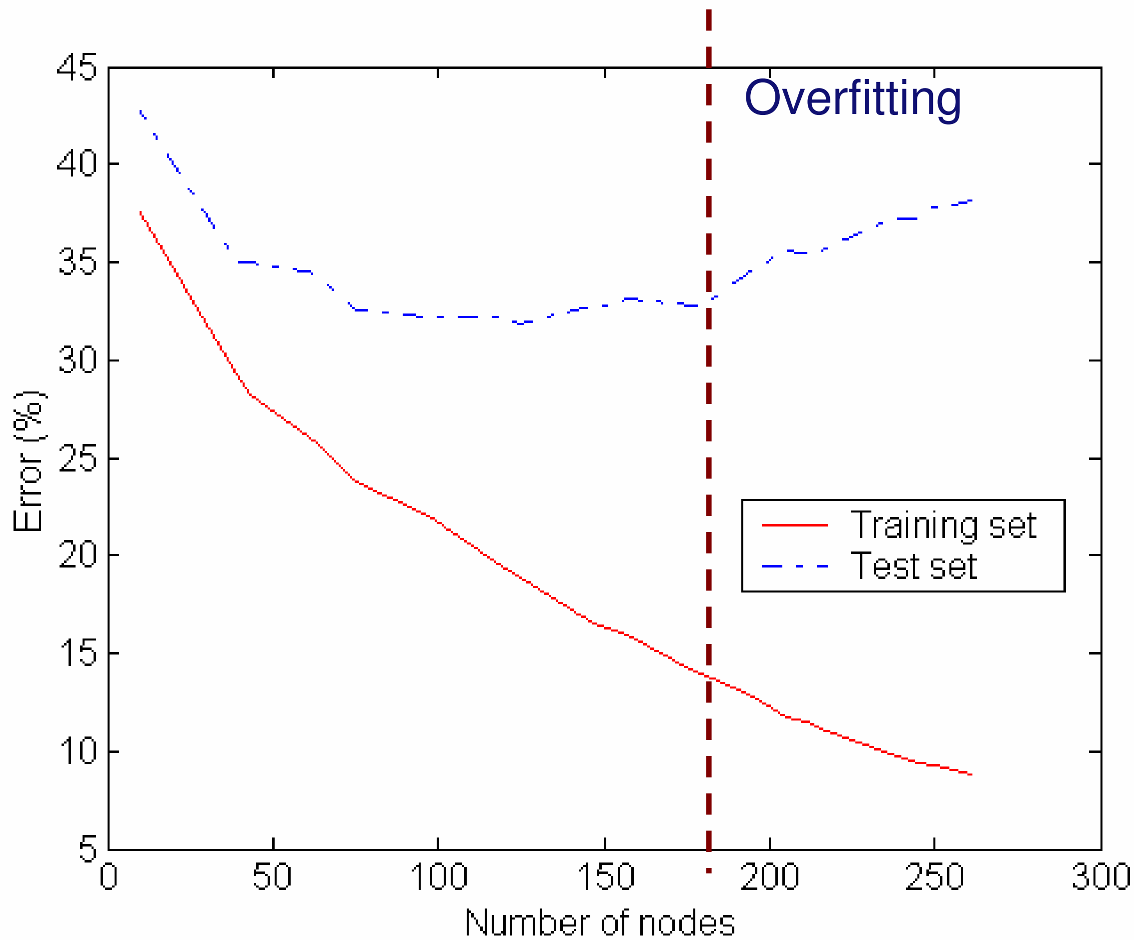
$0.5 \leq \mathrm{sqrt}(x_1^2+x_2^2) \leq 1$

Triangular points:

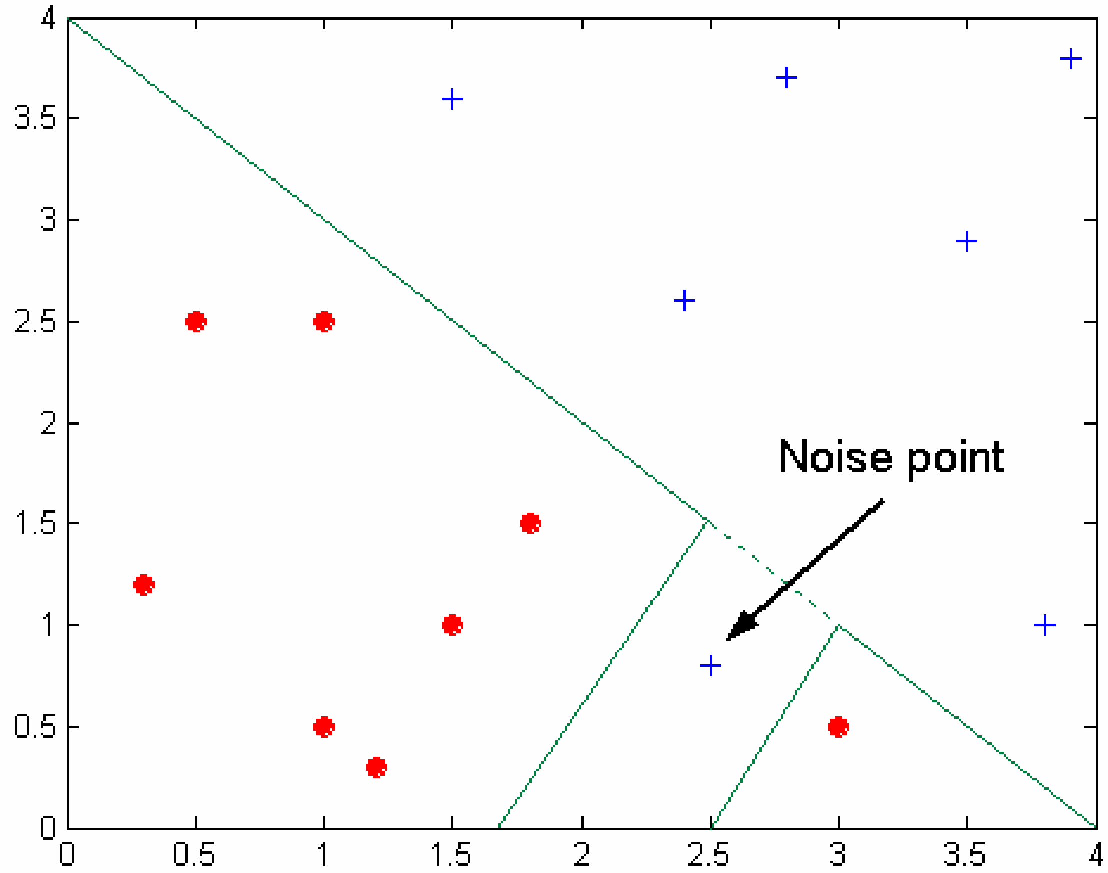
$\mathrm{sqrt}(x_1^2+x_2^2) > 0.5$ or

$\mathrm{sqrt}(x_1^2+x_2^2) < 1$

Underfitting and Overfitting



Overfitting due to Noise



Decision boundary is distorted by noise point

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
- Need new ways for estimating errors

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - all instances belong to the same class
    - all the attribute values are the same
  - More restrictive conditions:
    - if number of instances becomes too small
    - If class distribution becomes independent of attributes
    - If expanding the current node does not improve impurity measures.

How to Address Overfitting…

- **Post-pruning**
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming, replace sub-tree by a leaf node → use a validation set
  - Class label of leaf node is determined from majority class of instances in the sub-tree

Conclusion

- Classification problem
  - Learning a model on labeled data
  - Model used to predict class of new examples
- K-nearest neighbor
  - Distance function essential
- Decision trees
  - Hunt's algorithm
  - Split criteria
  - Stopping condition

TU/e Technische Universiteit Eindhoven University of Technology