

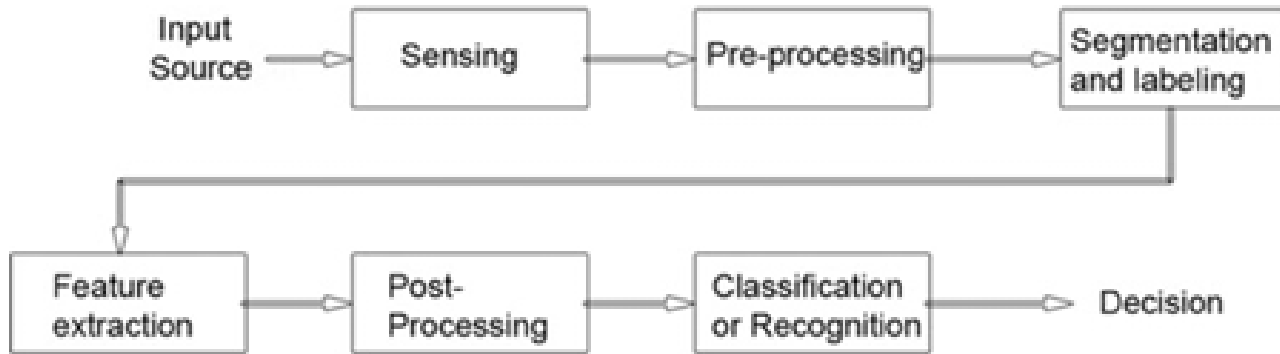
Authentication

Prof. Dr. Helene Dörksen

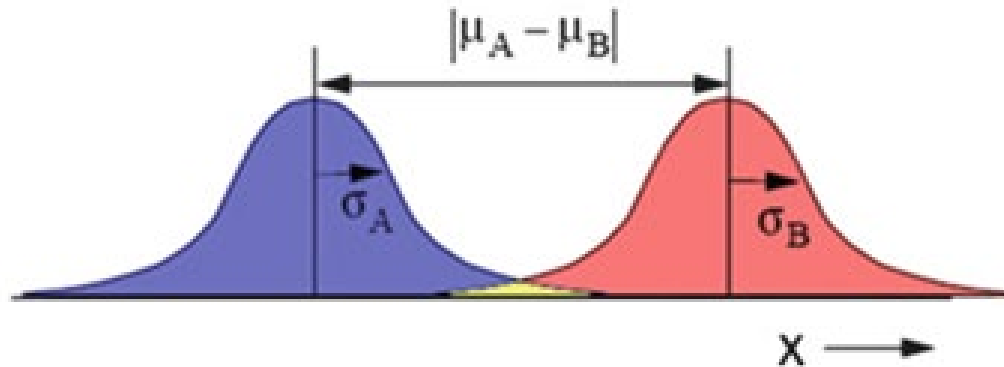
helene.doerksen@th-owl.de

Learned before

Classification principles: start from the end

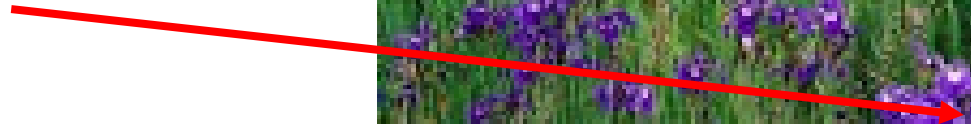
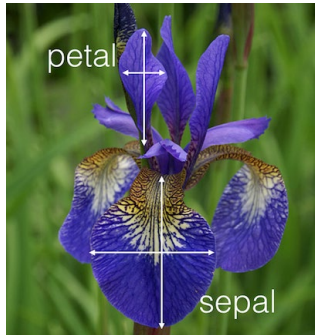


classification
system



good features are
difficult to find !

Example: is your feature good?



Lecture 2:

Nonmetric Methods

Nonmetric Methods

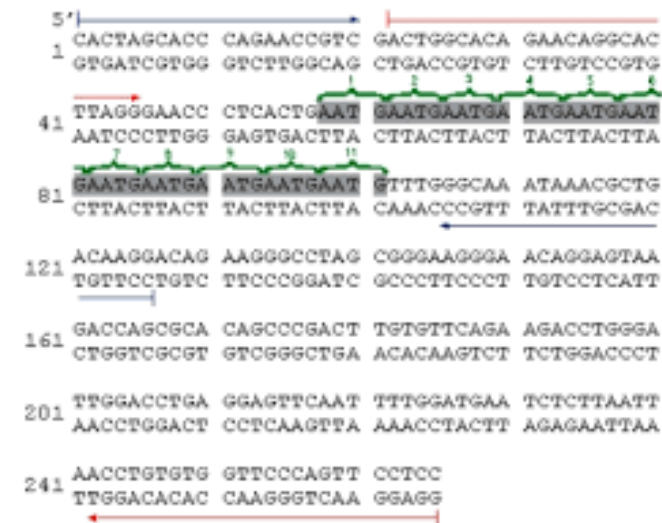
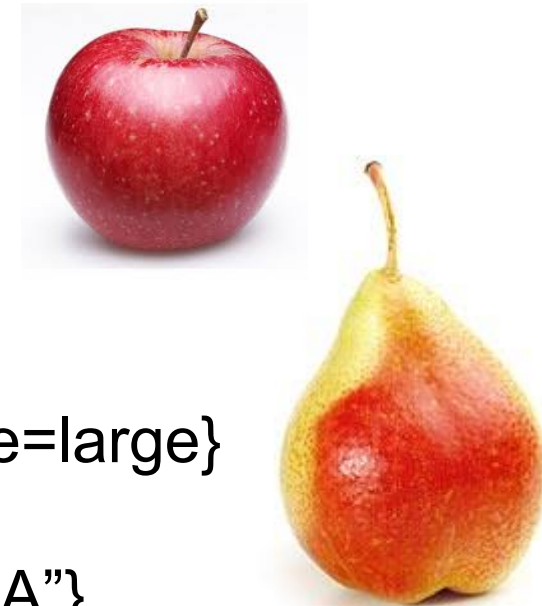
nonmetric = categorical data, i.e.
lists of attributes as features (not numbers)

fruit {color=red, texture=shiny, taste=sweet, size=large}

segment of DNA {sequence “GACTTAGATTCCA”}

solved by:

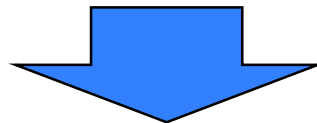
- decision trees
- rule-based classifiers, and
- syntactic (grammar-based) methods



Decision Trees

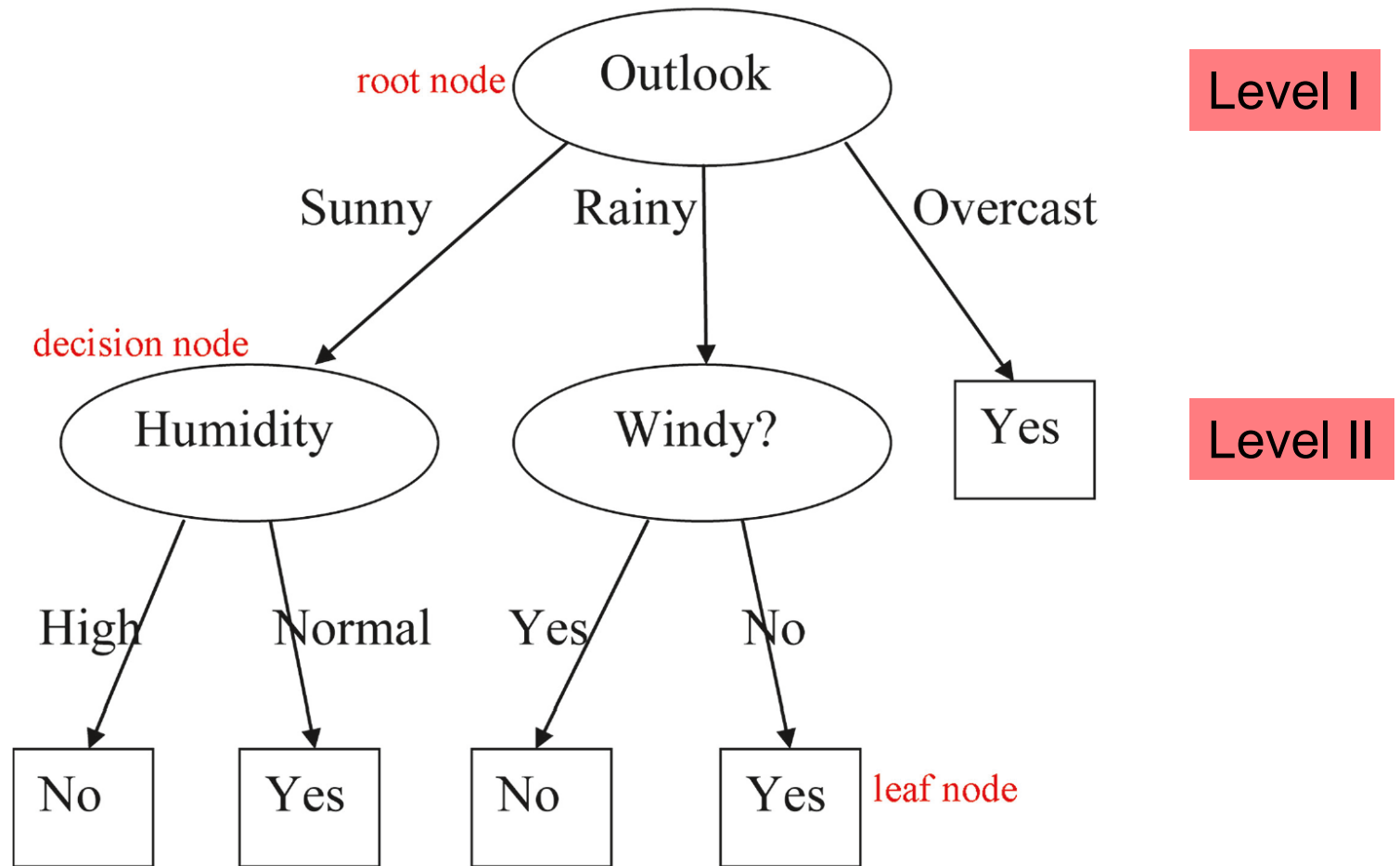
- simple classifier in the form of a hierarchical tree structure
- supervised classification using a *divide-and-conquer* strategy
- directed branching structure with a series of questions
- questions are placed at *decision nodes*
- each node tests particular attribute (feature) and provides a binary or multi-way split
- starting node is known as the *root node* (parent of every other node)

Successive decision nodes are visited until terminal (*leaf node*) is reached, where the class is read



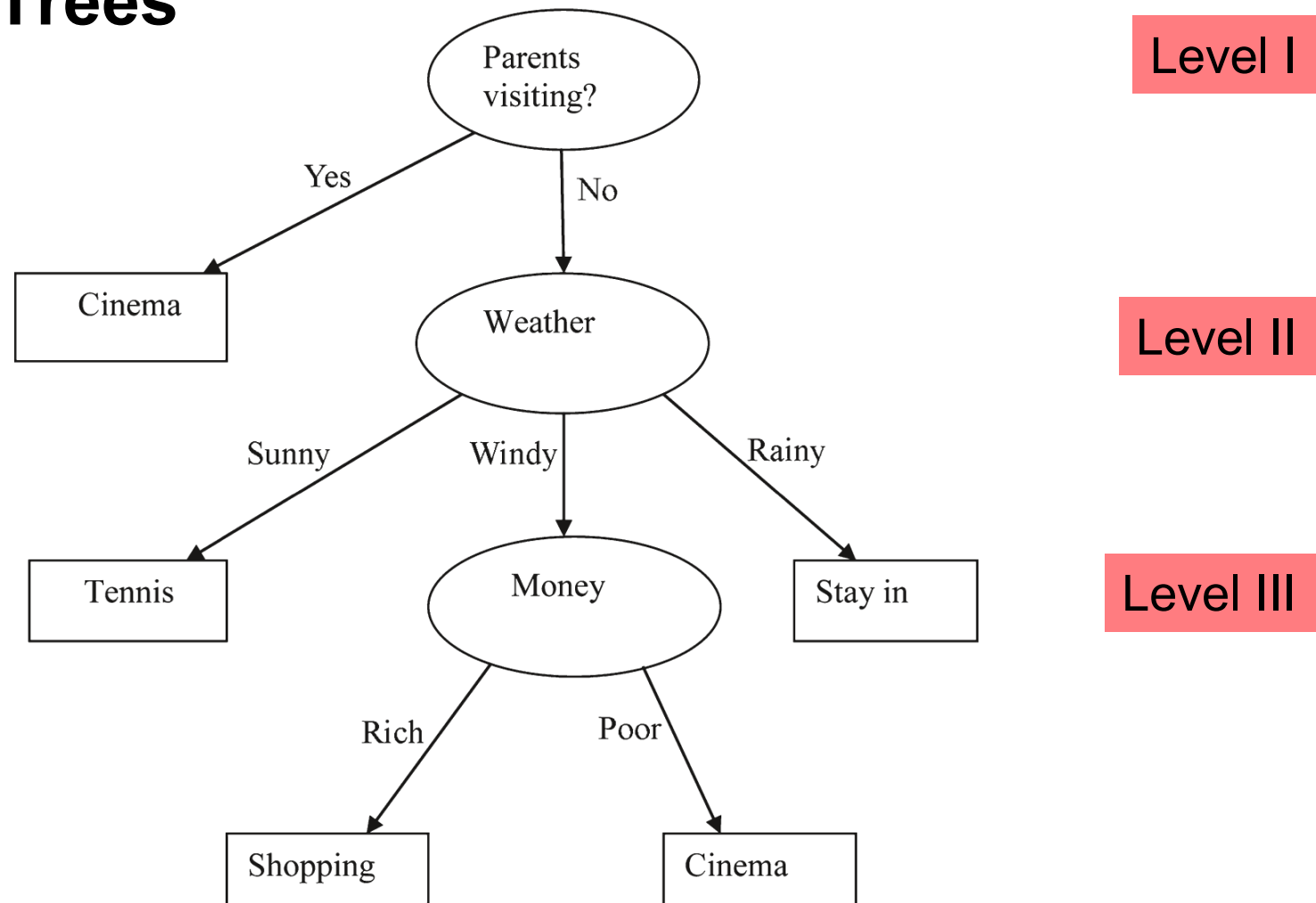
decision tree is upside-down tree, with the root at the top
and the leaves at the bottom!

Decision Trees




Two-level decision tree for determining whether to play tennis

Decision Trees



Three-level decision tree for determining what to do on a Saturday morning

Decision Trees

- decision trees, once constructed, are very fast since they require very little computation
 - **but** the more interesting question is **how to construct** the tree from training data
-
- 
- there are exponentially many decision trees that can be constructed from a given set of features
 - some of the trees will be more accurate than others, finding the optimal tree is not computationally feasible

Decision Trees



- number of efficient algorithms have been developed to create or “grow” a reasonably accurate decision tree in a reasonable amount of time
- algorithms usually employ a greedy strategy that grows the tree using the most informative attribute (feature) at each step and does not allow backtracking

Decision Trees: Information

information = reduction of uncertainty

→ **informative attributes** will result in reduction of uncertainty



color or shape attribute?

$$P(\text{color} = \text{red}) = 1$$

$$P(\text{shape} = \text{round}) = 0.5$$



- attributes with a high probability of occurring carry little information
- attributes that are least expected carry most information

Decision Trees: Information

$E = \text{event}$ (color, shape, etc.)

$P(E) = \text{probability of event}$



$$I(E) = \log \frac{1}{P(E)} = \underbrace{-\log P(E)}$$

units (bits) of information

Decision Trees: Entropy

Entropy is a measure of the disorder or unpredictability in a system

Example: binary (two-class) classification

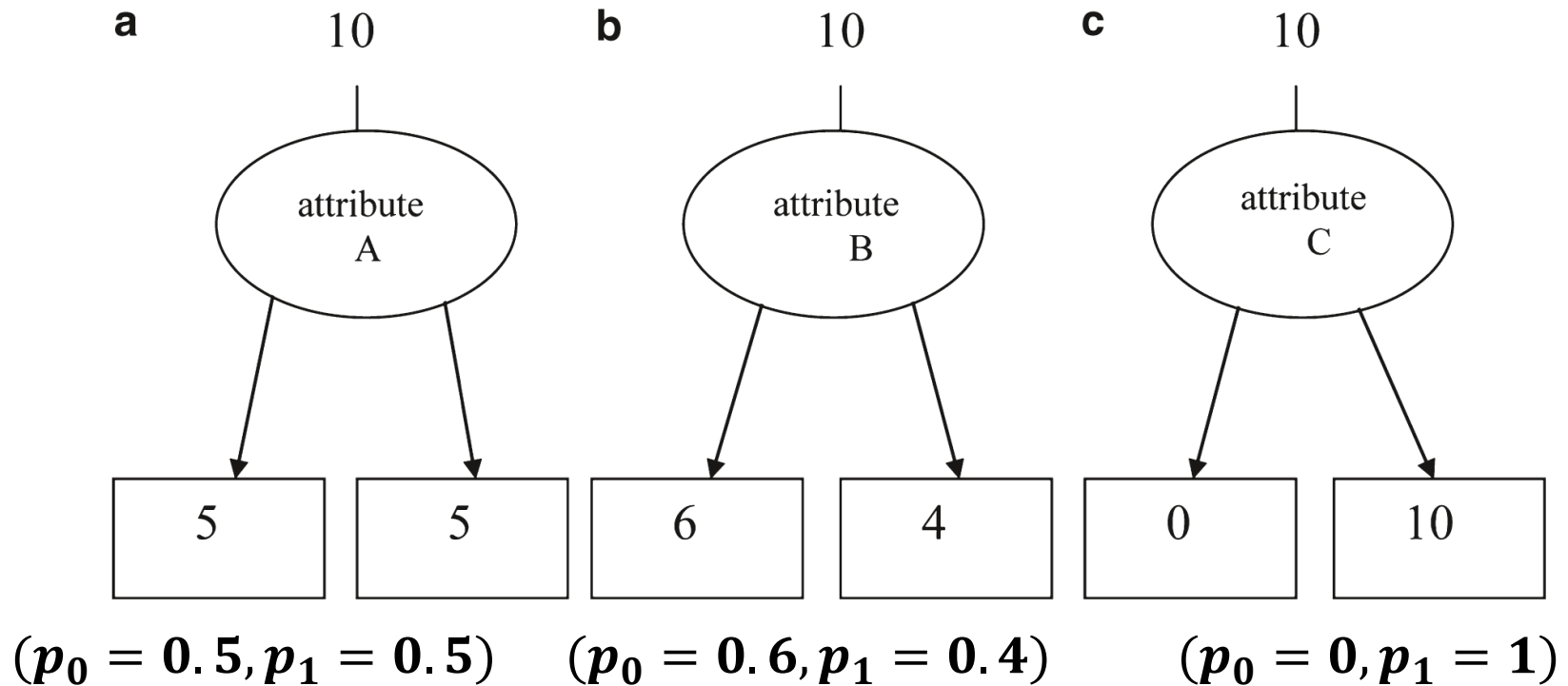
attribute A splits the objects S into (p_0, p_1) ,
with standardisation: $p_1 = 1 - p_0$

$$S = \{1, 2, \dots, 20\}$$

attribute A $\left\{ \begin{array}{l} \text{„object has two positions“} \\ \text{or} \\ \text{„object has } 1 \text{ or } 0\text{“} \end{array} \right.$

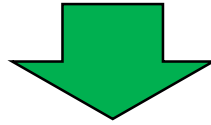
Decision Trees: Entropy

How to decide?



Decision Trees: Entropy

an attribute splits the objects S into (p_0, p_1) ,
with standardisation: $p_1 = 1 - p_0$

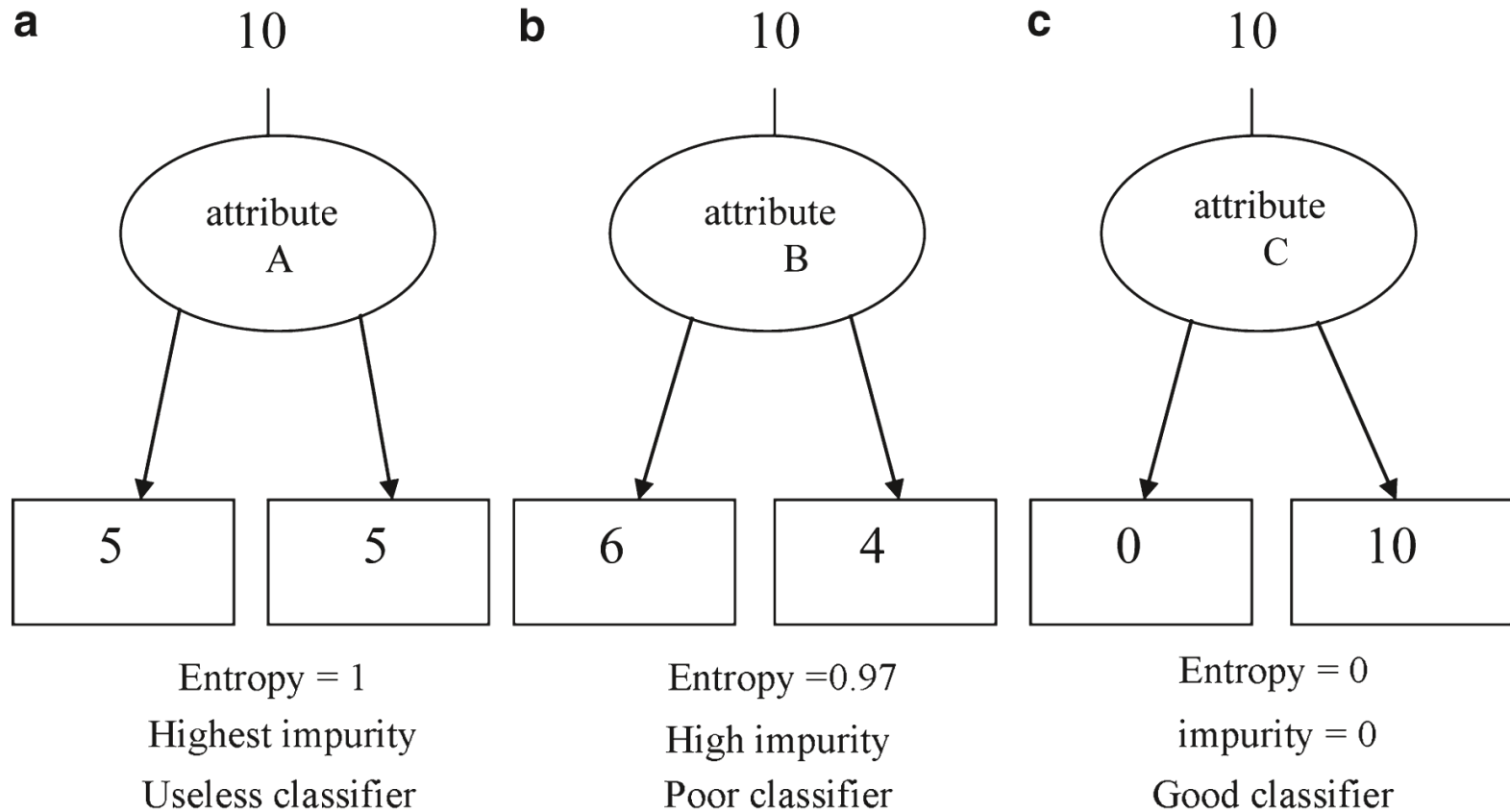


Entropy H of S w.r.t attribute:

$$H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$$

Decision Trees: Entropy

Entropy can be thought of as describing the amount of **impurity** in a set of features at a node

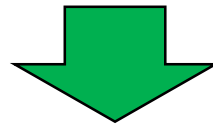


Decision Trees: Entropy

$$H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$$

an attribute splits the objects S into (p_0, p_1, \dots, p_c) ,

with standardisation: $\sum_{i=1}^c p_i = 1$

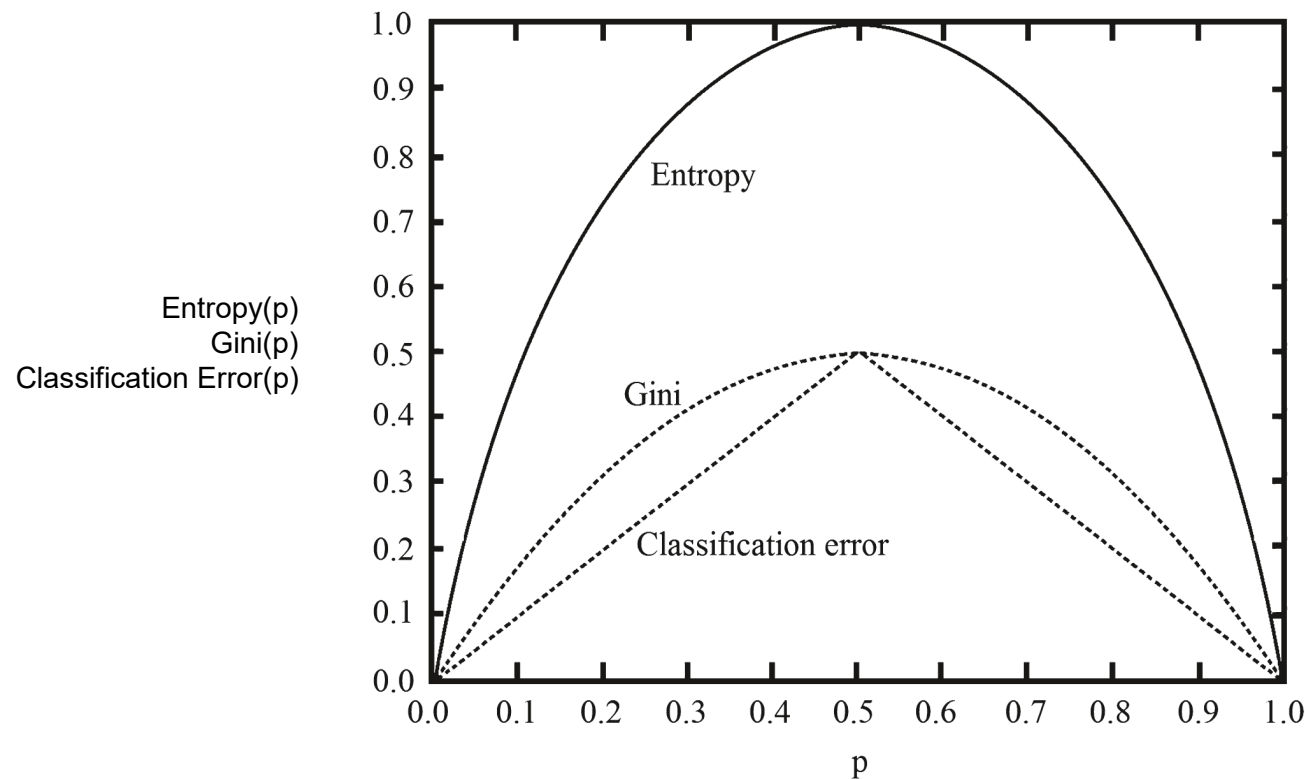


Entropy H of S w.r.t attribute:

$$H(S) = - \sum_{i=1}^c p_i \log_2 p_i$$

Decision Trees: Self Study

- **Gini-impurity** $\text{Gini}(S) = 1 - \sum_i p_i^2$
- **classification error** $\text{classification error}(S) = 1 - \max(p_i)$



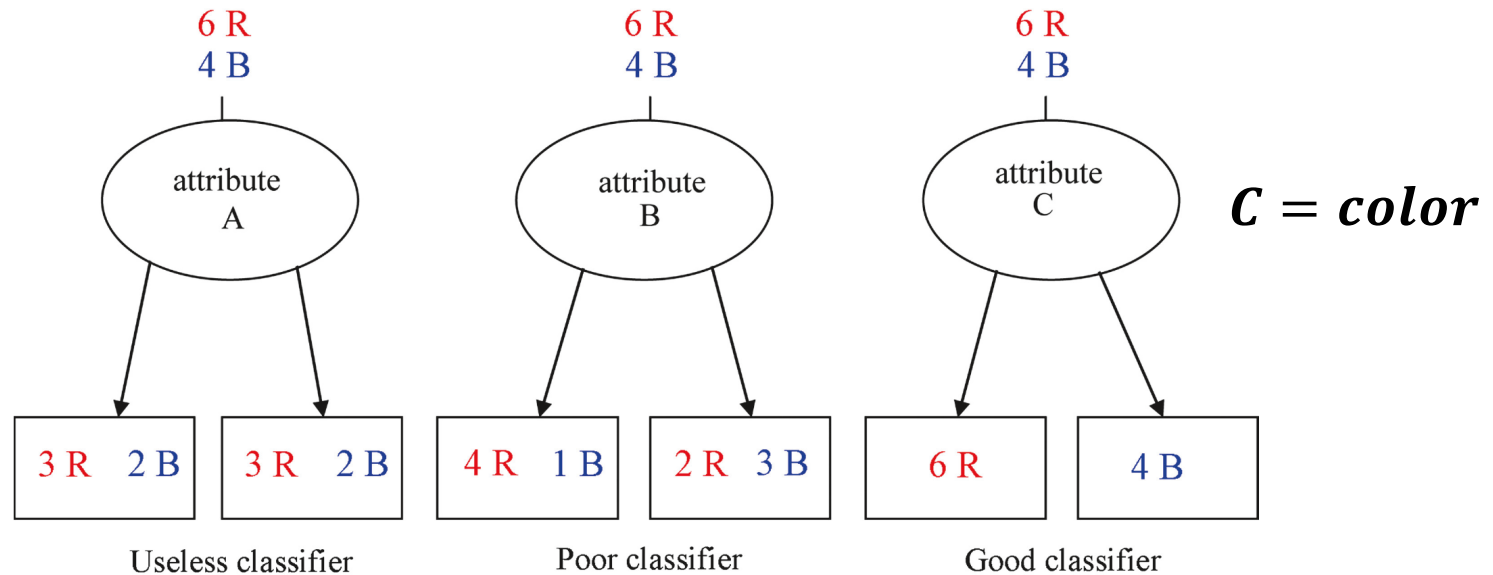
impurity measures for binary classification

Decision Trees: Information Gain

Problem: determine the best attribute to choose for each decision node of the tree

→ best attribute will be the one that best separates instances into **homogeneous subsets**

$$S = \{R, B, R, R, B, R, B, B, R, R\}$$



splitting a mixture of instances

Decision Trees: Information Gain

information gain = value for **reduction of impurity** caused by partitioning the examples according to the attribute

$$\text{Gain}(S, A) = \text{Impurity}(S) + \text{Term}$$

will be explained in later slides



→ bigger gain means bigger reduction of impurity

Decision Trees: Information Gain

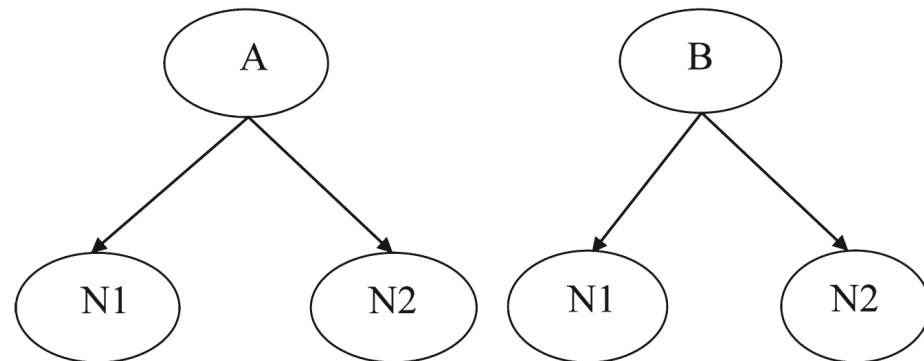
Example: use the Gini index (as measure for impurity) and find the **information gain** of two attributes A and B

$$\text{Gini}(S) = 1 - \sum_i p_i^2$$

- there are two attributes, A and B, which split the data (comprising **12 instances**) into smaller subsets
- before splitting (i.e., the parent node), the Gini index is **0.5** since there is an equal number of cases from both classes

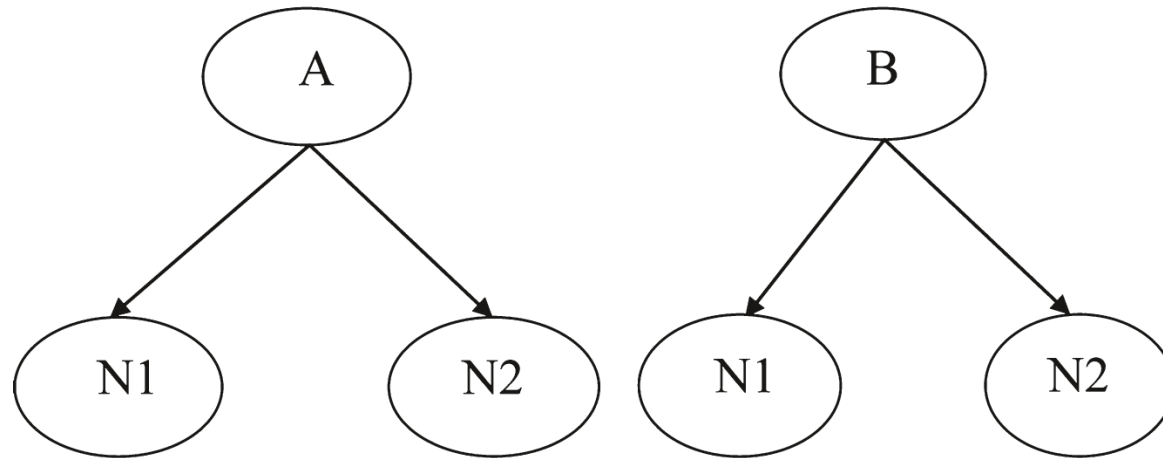
$$S = \{\mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}\}$$

Class	Parent
1	6
2	6
Gini = 0.500	



Decision Trees: Information Gain

$$\text{Gini}(S) = 1 - \sum_i p_i^2$$



Class	N1	N2
1	4	2
2	3	3

Class	N1	N2
1	1	5
2	4	2

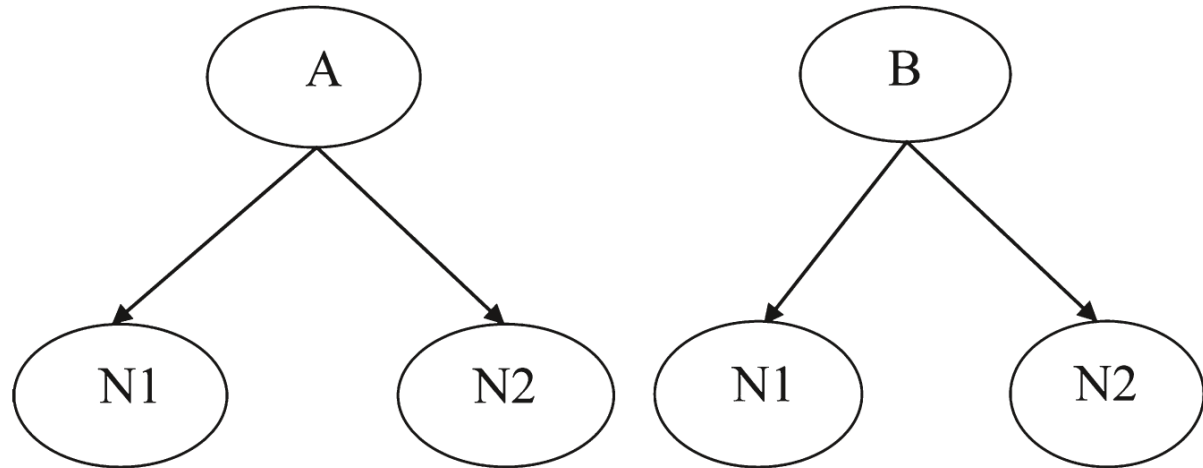
$$1 - [(4/7)^2 + (3/7)^2] = 24/49$$

$$1 - [(2/5)^2 + (3/5)^2] = 12/25$$

$$(7/12) \times 0.4898 + (5/12) \times 0.480 = 0.486 \quad \text{weighted average}$$

Decision Trees: Information Gain

Class	Parent
1	6
2	6
Gini = 0.500	



Class	N1	N2
1	4	2
2	3	3
Gini = 0.486		

Class	N1	N2
1	1	5
2	4	2
Gini = 0.375		



the subsets for attribute B have a smaller Gini index (i.e., smaller impurity), it is preferred to attribute A, or

gain in using attribute B is larger ($\text{Gain}(S, B) = 0.5 - 0.375 = 0.125$) than the gain in using attribute A ($\text{Gain}(S, A) = 0.5 - 0.486 = 0.014$)

Decision Trees: Information Gain

Using the **ID3** (Quinlan 1986) algorithm to build a decision tree

ID3 = top-down greedy search through the space of possible decision trees; name given because it was the third in a series of “interactive dichotomizer” procedures

Examples	Weather	Parents visiting?	Money	Decision (category)
1	Sunny	Yes	Rich	Cinema
2	Sunny	No	Rich	Tennis
3	Windy	Yes	Rich	Cinema
4	Rainy	Yes	Poor	Cinema
5	Rainy	No	Rich	Stay in
6	Rainy	Yes	Poor	Cinema
7	Windy	No	Poor	Cinema
8	Windy	No	Rich	Shopping
9	Windy	Yes	Rich	Cinema
10	Sunny	No	Rich	Tennis

we want to train a decision tree using the examples

Decision Trees: Information Gain

1.Step: find attribute for root node: Weather or Parents visiting? or Money

Cinema 6 $p_1 = 0.6$

Tennis 2 $p_2 = 0.2$

Stay in 1 $p_3 = 0.1$

Shopping 1 $p_4 = 0.1$



$$c = 4$$

$$H(S) = - \sum_{i=1}^c p_i \log_2 p_i$$

Decision (category)
Cinema
Tennis
Cinema
Cinema
Stay in
Cinema
Cinema
Shopping
Cinema
Tennis

$$= -0.6 \log_2 0.6 - 0.2 \log_2 0.2 - 2 \times (0.1 \log_2 0.1) = 1.571$$

Decision Trees: Information Gain

1.Step: find attribute for root node: Weather or Parents visiting? or Money

determine the values of $\text{Gain}(S, \text{parents})$, $\text{Gain}(S, \text{weather})$, and $\text{Gain}(S, \text{money})$

If “parents coming?” is the node, then

- 5 instances will be “yes” with all class “cinema”, i.e. entropy of zero
- 5 instances will be “no”: 2x“tennis,” 1x“stay-in,” 1x “cinema,” 1x“shopping”
 $\text{entropy} = -0.4 \log_2 0.4 - 3 \times (0.2 \log_2 0.2)$

Parents visiting?	Decision (category)
Yes	Cinema
No	Tennis
Yes	Cinema
Yes	Cinema
No	Stay in
Yes	Cinema
No	Cinema
No	Shopping
Yes	Cinema
No	Tennis

$$\begin{aligned} \text{Gain}(S, \text{parents}) &= 1.571 - \left(\frac{5}{10} \times \text{Entropy}(S_{\text{yes}}) - \left(\frac{5}{10} \right) \times \text{Entropy}(S_{\text{no}}) \right) \\ &= 1.571 - (0.5 \times 0 - (0.5) \times (1.922)) = 0.61 \end{aligned}$$

Decision Trees: Information Gain

1.Step: find attribute for root node: Weather or Parents visiting? or Money

Weather	Decision (category)
Sunny	Cinema
Sunny	Tennis
Windy	Cinema
Rainy	Cinema
Rainy	Stay in
Rainy	Cinema
Windy	Cinema
Windy	Shopping
Windy	Cinema
Sunny	Tennis

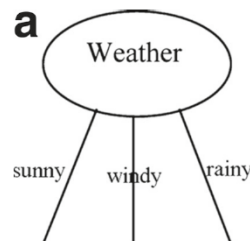
Money	Decision (category)
Rich	Cinema
Rich	Tennis
Rich	Cinema
Poor	Cinema
Rich	Stay in
Poor	Cinema
Poor	Cinema
Rich	Shopping
Rich	Cinema
Rich	Tennis

$$\text{Gain}(S, \text{weather}) = 0.70$$

$$\text{Gain}(S, \text{money}) = 0.2816$$

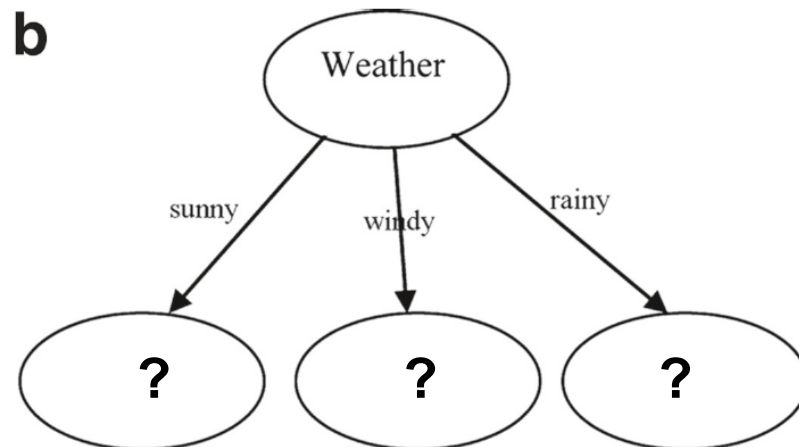
Decision Trees: Information Gain

1.Step: find attribute for root node:



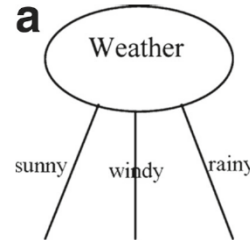
2.Step: find attribute for next nodes:

Examples	Weather	Parents visiting?	Money	Decision (category)
1	Sunny	Yes	Rich	Cinema
2	Sunny	No	Rich	Tennis
3	Windy	Yes	Rich	Cinema
4	Rainy	Yes	Poor	Cinema
5	Rainy	No	Rich	Stay in
6	Rainy	Yes	Poor	Cinema
7	Windy	No	Poor	Cinema
8	Windy	No	Rich	Shopping
9	Windy	Yes	Rich	Cinema
10	Sunny	No	Rich	Tennis



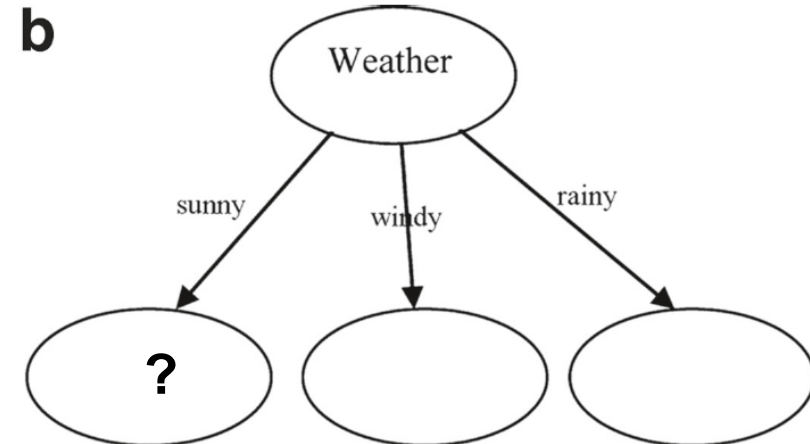
Decision Trees: Information Gain

1.Step: find attribute for root node:



2.Step: find attribute for next nodes:

Examples	Weather	Parents visiting?	Money	Decision (category)
1	Sunny	Yes	Rich	Cinema
2	Sunny	No	Rich	Tennis
3	Windy	Yes	Rich	Cinema
4	Rainy	Yes	Poor	Cinema
5	Rainy	No	Rich	Stay in
6	Rainy	Yes	Poor	Cinema
7	Windy	No	Poor	Cinema
8	Windy	No	Rich	Shopping
9	Windy	Yes	Rich	Cinema
10	Sunny	No	Rich	Tennis

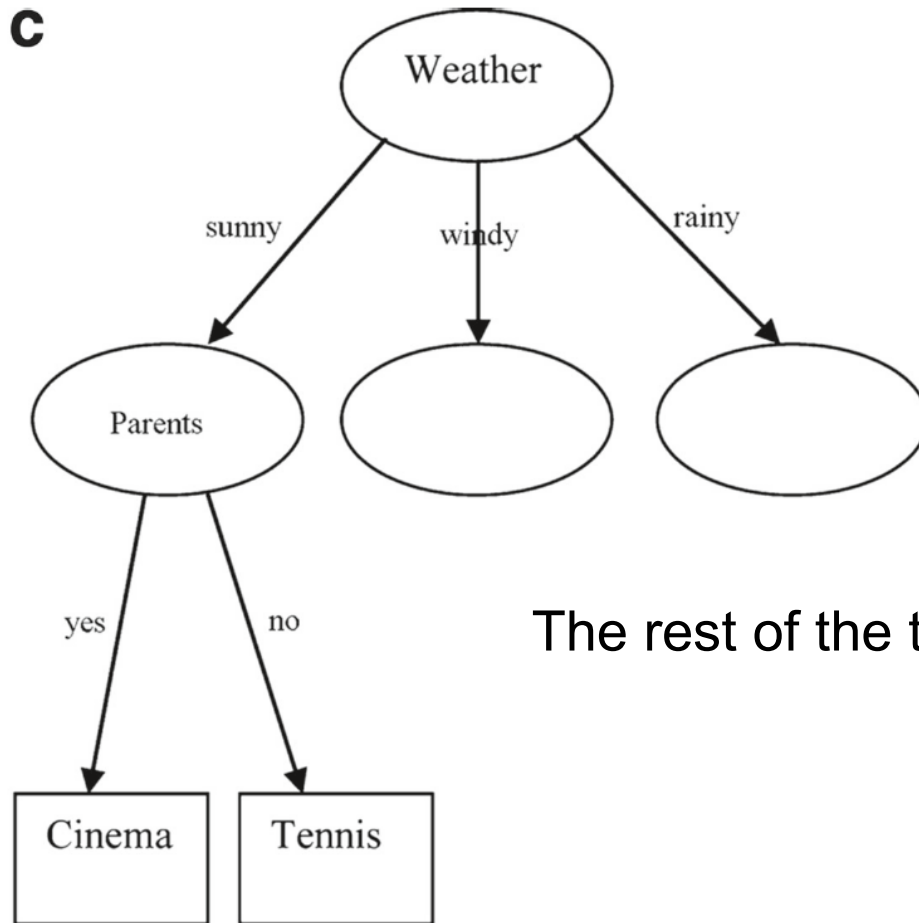


$$\text{Gain}(S_{\text{sunny}}, \text{parents}) = 0.918$$

$$\text{Gain}(S_{\text{sunny}}, \text{money}) = 0$$

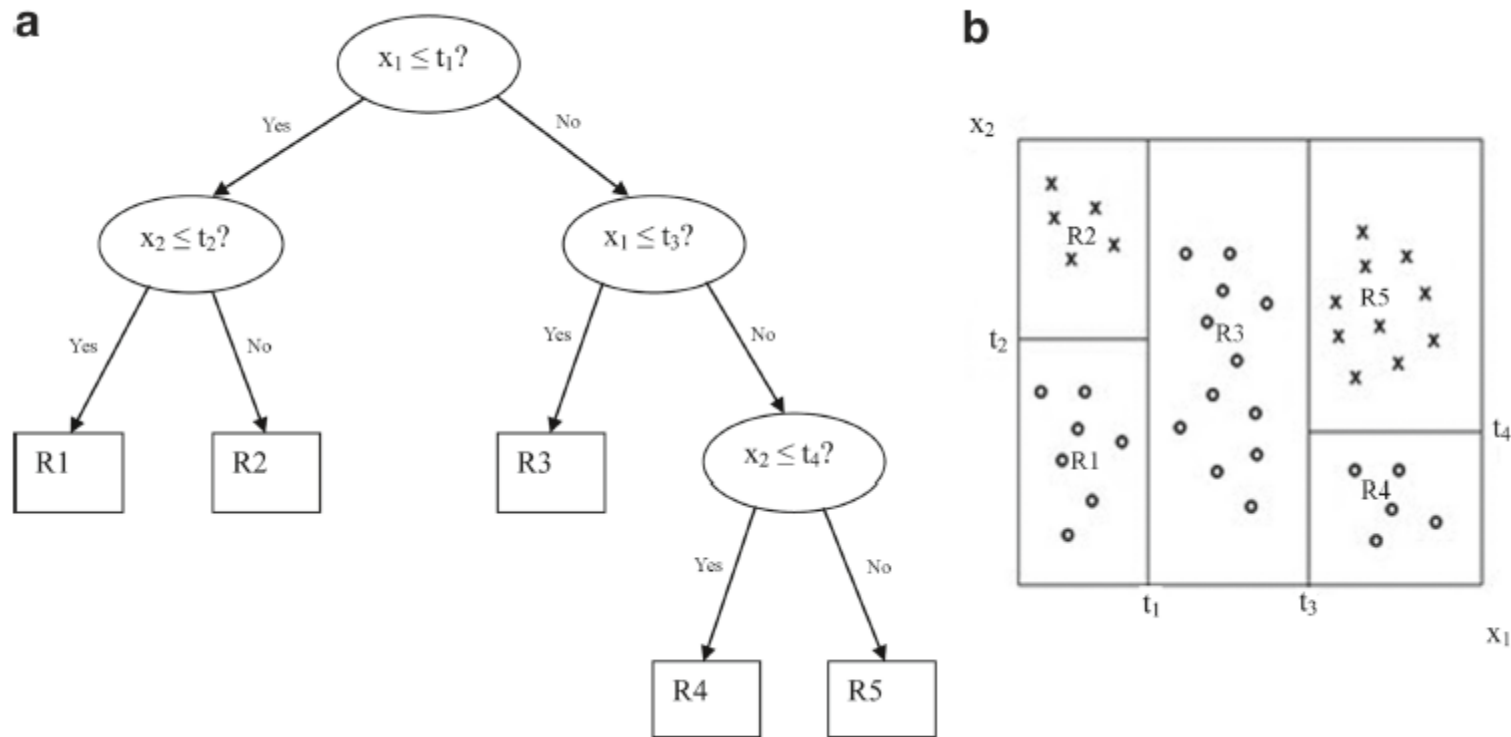
Decision Trees: Information Gain

C



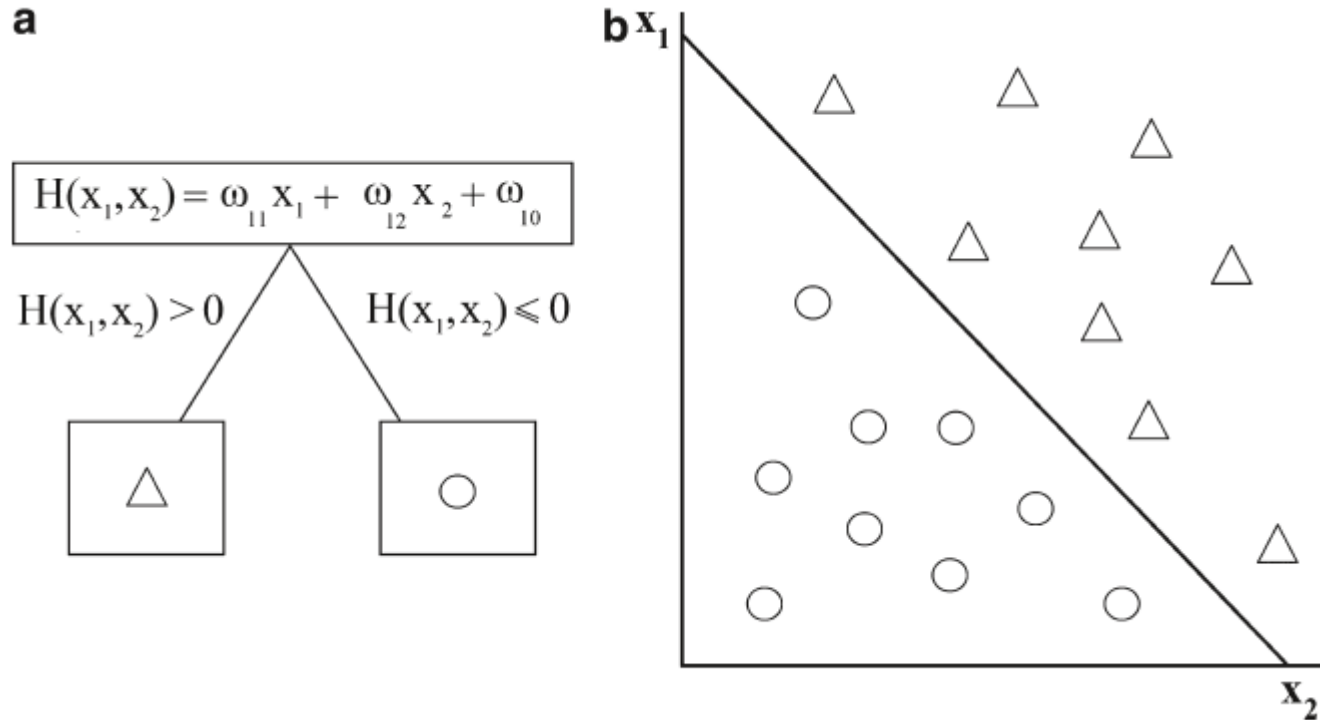
The rest of the tree is left as an exercise!

Decision Trees in Feature Space



(a) decision tree and (b) the resulting decision boundaries in feature space

Decision Trees in Feature Space



(a) decision tree and (b) the resulting decision boundary in feature space

Rule-Based Classifier

Classify records by using a collection of “if...then...” rules

It is possible to extract rules from a decision tree: each path from root to a leaf can be written down as a series of **IF** . . . **THEN** rules

IF (outlook = sunny) **AND** (humidity = high)
THEN do not play tennis

Other Methods

DNA sequence: “**AGCTTGGCATC**” (where A, G, C, and T stand for the nucleic acids adenine, guanine, cytosine, and thymine)

string matching involves finding whether a sub-string appears in a string for a particular shift of characters

edit distance = how many fundamental operations (substitution, insertion, or deletion of a character) are required to transform one string into another

Other Methods

how to transform $x = \text{“excused”}$ into the string $y = \text{“exhausted”}$

	e	x	h	a	u	s	t	e	d
e									
x									
c									
u									
s									
e									
d									

gray arrow indicates no change
black diagonal arrow indicates substitution
black horizontal arrow is an insertion

edit distance = 3

Summary

Nonmetric Methods: features are attributes (not numbers)

- decision trees: information, entropy, information gain
- rule-based classifier
- other methods

Homework: Exercises and Labs

for the next week prepare practical exercises and labs from **Exercises Lec 2** (you will find it in the donwload area)