CLASSIFICATION

Outline

2

- Introduction
- Naïve Bayes Classifiers
- K-Nearest Neighbor Classifiers
- Decision Trees
- Top-Down Induction of Decision Trees (TDIDT)
- Entropy and Information Gain
- ID3

Classification

Introduction

3

- Classification is dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as classes.
- Examples:
  - customers who are likely to buy or not buy a particular product in a supermarket
  - people who are at high, medium or low risk of acquiring a certain illness
  - people who closely resemble, slightly resemble or do not resemble someone seen committing a crime

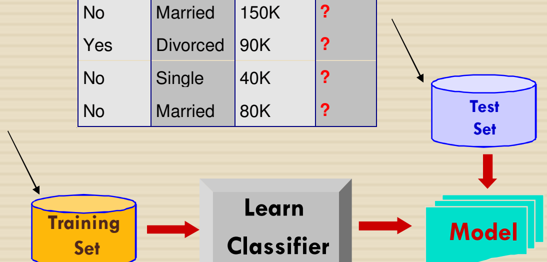
Classification Example

4

categorical categorical continuous class

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

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Classification: Application 1

5

- Direct Marketing
  - Goal: Reduce cost of mailing by targeting a set of consumers likely to buy a new cell-phone product.
  - Approach:
    - Use the data for a similar product introduced before.
    - We know which customers decided to buy and which decided otherwise. This {buy, don't buy} decision forms the class attribute.
    - Collect various demographic, lifestyle, and company-interaction related information about all such customers.
      - Type of business, where they stay, how much they earn, etc.
    - Use this information as input attributes to learn a classifier model.

Classification

Classification: Application 2

6

- Fraud Detection
  - Goal: Predict fraudulent cases in credit card transactions.
  - Approach:
    - Use credit card transactions and the information on its account-holder as attributes.
      - When does a customer buy, what does he buy, how often he pays on time, etc
    - Label past transactions as fraud or fair transactions. This forms the class attribute.
    - Learn a model for the class of the transactions.
    - Use this model to detect fraud by observing credit card transactions on an account.

Classification

7

Naïve Bayes Classifiers

Naïve Bayes Classifiers

8

- This method does not use rules, a decision tree or any other explicit representation of the classifier.
- Rather It uses probability theory to find the most likely of the possible classifications.

Train Arrival Probability Example

9

- Four mutually exclusive and exhaustive events are defined
  - E1 – train cancelled
  - E2 – train ten minutes or more late
  - E3 – train less than ten minutes late
  - E4 – train on time or early.
- With the following Probability:
  - $P(E1) = 0.05$
  - $P(E2) = 0.1$
  - $P(E3) = 0.15$
  - $P(E4) = 0.7$
- $P(E1) + P(E2) + P(E3) + P(E4) = 1$
- The probability can be calculated by counting

Training Set

10

- The training set constitutes the results of a sample of trials that we can use to predict the classification of other (unclassified) instances.

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Probability-based Classification

11

- How should we use probabilities to find the most likely classification for an unseen instance such as this one?

weekday	winter	high	heavy	????

- One straightforward (but flawed) way is just to look at the frequency of each of the classifications in the training set and choose the most common one.
- So it will be on time
- Thus you are right 70% of the time only

Probability-based Classification

12

- Prior Probability:
  - P(class = on time) = 14/20 = 0.7
- Conditional Probability:
  - P(class = on time | season = winter)=2/6=0.33
- What is P(class = on time | day = weekday and season = winter and wind = high and rain = heavy)
- there are only two instances in the training set with that combination of attribute values and basing any estimates of probability on these is unlikely to be helpful.

Example

13

- To obtain a reliable estimate of the four classifications a more indirect approach is needed.
- We could start by using conditional probabilities based on a single attribute.
  - P(class = on time | season = winter) = 2/6 = 0.33
  - P(class = late | season = winter) = 1/6 = 0.17
  - P(class = very late | season = winter) = 3/6 = 0.5
  - P(class = cancelled | season = winter) = 0/6 = 0

Naïve Bayed Algorithm

14

- The Naive Bayes algorithm gives us a way of combining the prior probability and conditional probabilities in a single formula, which we can use to calculate the probability of each of the possible classifications in turn.
- Then choose the classification with the largest value.

The Naïve Bayes Classification Algorithm

- The posterior probability of class ci occurring for the specified instance can be shown to be proportional to:
- P(ci) × P(a1=v1 and a2=v2 . . . and an=vn | ci)
- And is equal to
- P(ci)× P(a1=v1|ci)×P(a2=v2|ci)×. . .×P(an=vn|ci)
- Calculate this product for each value of i from 1 to k and choose the classification that has the largest value.

Conditional and Prior Probabilities

	class = on time	class = late	class = very late	class = cancelled
day = weekday	**9/14 = 0.64**	**1/2 = 0.5**	**3/3 = 1**	**0/1 = 0**
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	**2/14 = 0.14**	**2/2 = 1**	**2/3 = 0.67**	**0/1 = 0**
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	**4/14 = 0.29**	**1/2 = 0.5**	**1/3 = 0.33**	**1/1 = 1**
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 =0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	**1/14 = 0.07**	**1/2 = 0.5**	**2/3 = 0.67**	**1/1 = 1**
Prior Probability	**14/20 = 0.70**	**2/20 = 0.10**	**3/20 = 0.15**	**1/20 = 0.05**

Posterior Probabilities

17

- class = on time
  - $0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$
- class = late
  - $0.10 \times 0.50 \times 1.00 \times 0.50 \times 0.50 = 0.0125$
- class = very late
  - $0.15 \times 1.00 \times 0.67 \times 0.33 \times 0.67 = 0.0222$
- class = cancelled
  - $0.05 \times 0.00 \times 0.00 \times 1.00 \times 1.00 = 0.0000$
- The largest value is for class = very late.

Classification

Naïve Bayes Approach Problems

18

- Assume categorical attributes (not continuous)
  - Work around: Use clustering to convert the continuous attributes to categorical ones
- Estimating probabilities by relative frequencies can give a poor estimate if the number of instances with a given attribute/value combination is small.
  - Use a complicated formula for calculating probability instead of counting.

Classification

19

Nearest Neighbor Classification

Classification

Nearest Neighbor Classification

20

- Nearest Neighbour classification is mainly used when all attribute values are continuous, although it can be modified to deal with categorical attributes.
- The idea is to estimate the classification of an unseen instance using the classification of the instance or instances that are closest to it, in some sense that we need to define.

Classification

K-Nearest Neighbor (K-NN)

21

- Find the k training instances that are closest to the unseen instance.
- Take the most commonly occurring classification for these k instances.
- k is a small integer such as 3 or 5

Classification

Training Data Set

22

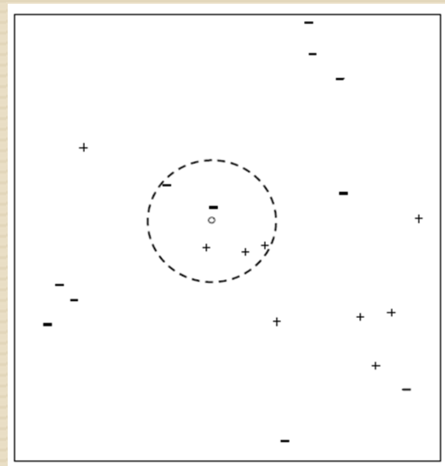
- Two classes
- Two attributes
- How to classify (9.1,11)

Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

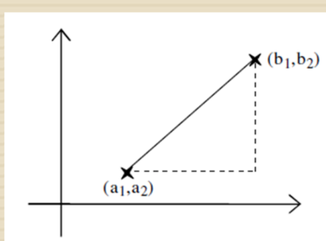
Classification

5-NN Classifier

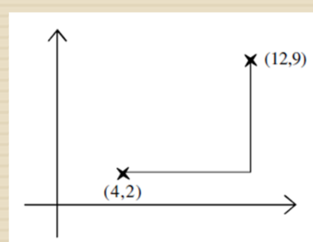
- The five nearest neighbours are labelled with three + signs and two – signs,
- so a basic 5-NN classifier would classify the unseen instance as 'positive' by a form of majority voting.



Distance Measures



Euclidean



City Block

- Maximum dimension distance:
  - The largest absolute difference between any pair of corresponding attribute values

Normalization

25

- When using the Euclidean (and others) distance formula, large values frequently swamp the small ones.
- To overcome this problem we generally normalise the values of continuous attributes so the values of each attribute run from 0 to 1.
- Also we can weight the contributions of the different attributes.

26

Decision Trees

Decision Trees: The Golf Example

27

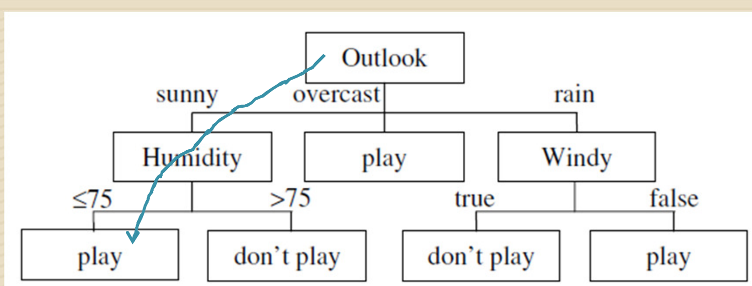
Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

Classes
play, don't play
Outlook
sunny, overcast, rain
Temperature
numerical value
Humidity
numerical value
Windy
true, false

What are the rules that determine the decision whether or not to play each day?

The Golfer Decision Tree

28



If tomorrow the values of *Outlook, Temperature, Humidity and Windy are sunny, 74°F, 71% and false* respectively, what would the decision be?

Decision Trees Functions

29

- Decision trees have two different functions:
  - data compression
    - The two representations are equivalent in the sense that for each of the 14 instances the given values of the four attributes will lead to identical classifications
  - Prediction
    - It can be used to predict the values of other instances not in the training set

TDIDT

30

- Top-Down Induction of Decision Trees
- The method produces decision rules in the implicit form of a decision tree.
- Decision trees are generated by repeatedly splitting on the values of attributes.
- This process is known as recursive partitioning.

TDIDT: BASIC ALGORITHM

31

- IF all the instances in the training set belong to the same class THEN return the value of the class
- ELSE
  - (a) Select an attribute A to split on+
  - (b) Sort the instances in the training set into subsets, one for each value of attribute A
  - (c) Return a tree with one branch for each non-empty subset, each branch having a descendant subtree or a class value produced by applying the algorithm recursively

- \+ Never select an attribute twice in the same branch

32

Entropy and Information Gain

Entropy

33

- Entropy is an information-theoretic measure of the 'uncertainty' contained in a training set, due to the presence of more than one possible classification.

Minimize Entropy → Maximize Information Gain

- If there are K classes, we can denote the proportion of instances with classification i by pi for i = 1 to K.
- The value of pi is the number of occurrences of class i divided by the total number of instances, which is a number between 0 and 1 inclusive.

Classification

Entropy – cont.

34

- The entropy of the training set is denoted by E. It is measured in 'bits' of information and is defined by the formula:

$$E = -\sum_{i=1}^{K} p_i \log_2 p_i$$

- It takes its minimum value (zero) iff all the instances have the same classification (k=1).
- Entropy takes its maximum value when the instances are equally distributed amongst the K possible classes (pi=1/K for any i).

Classification

Lens24 Data

35

Estart = −(4/24) log2(4/24) − (5/24) log2(5/24) − (15/24) log2(15/24)
= 0.4308 + 0.4715 + 0.4238

Value of attribute				Class
age	specRx	astig	tears	
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

classes
1: hard contact lenses
2: soft contact lenses
3: no contact lenses

age
1: young
2: pre-presbyopic
3: presbyopic

specRx
(spectacle prescription)
1: myopia
2: high hypermetropia

astig
(whether astigmatic)
1: no
2: yes

tears
(tear production rate)
1: reduced
2: normal

Using Entropy for Attribute Selection

36

- The process of decision tree generation by repeatedly splitting on attributes is equivalent to partitioning the initial training set into smaller training sets repeatedly, until the entropy of each of these subsets is zero
- At any stage of this process, splitting on any attribute has the property that the average entropy of the resulting subsets will be less than (or occasionally equal to) that of the previous training set.

Splitting on Age

37

Training set 1 (age = 1)
Entropy $E1 = -(2/8)\,log2(2/8) - (2/8)\,log2(2/8) - (4/8)\,log2(4/8)$
$= 0.5 + 0.5 + 0.5 = 1.5$

Training set 2 (age = 2)
Entropy $E2 = -(1/8)\,log2(1/8) - (2/8)\,log2(2/8) - (5/8)\,log2(5/8)$
$= 0.375 + 0.5 + 0.4238 = 1.2988$

Training Set 3 (age = 3)
Entropy $E3 = -(1/8)\,log2(1/8) - (1/8)\,log2(1/8) - (6/8)\,log2(6/8)$
$= 0.375 + 0.375 + 0.3113 = 1.0613$

$Enew = (8/24)E1+(8/24)E2+(8/24)E3 = 1.2867$ bits

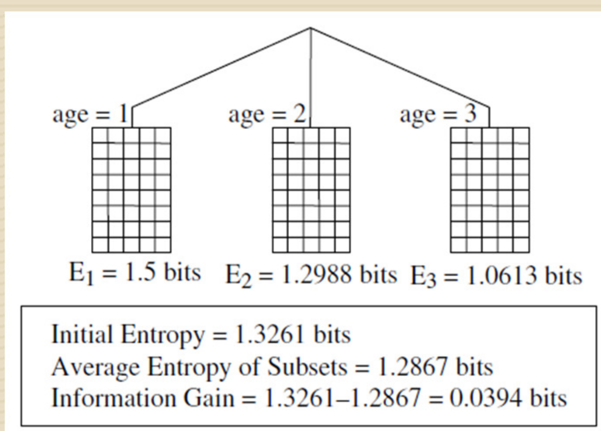
Value of attribute				Class
age	specRx	astig	tears	
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1				
1				
1				
1				
1				

Value of attribute				Class
age	specRx	astig	tears	
2	1	1	1	3
2	1	1	2	2
2				
2				
2				
2				
2				
2				

Value of attribute				Class
age	specRx	astig	tears	
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Splitting on Age – cont.

38



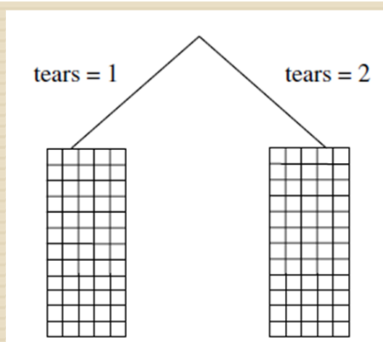
Maximising Information Gain

39

- attribute age
  - Enew = 1.2867
  - Information Gain = 1.3261 – 1.2867 = 0.0394 bits
- attribute specRx
  - Enew = 1.2866
  - Information Gain = 1.3261 – 1.2866 = 0.0395 bits
- attribute astig
  - Enew = 0.9491
  - Information Gain = 1.3261 – 0.9491 = 0.3770 bits
- attribute tears
  - Enew = 0.7773
  - Information Gain = 1.3261 – 0.7773 = 0.5488 bits

Splitting on Attribute tears

40



- The process of splitting on nodes is repeated for each branch of the evolving decision tree, terminating when the subset at every leaf node has entropy zero

41

ID3

ID3: Iterative Dichotomiser

42

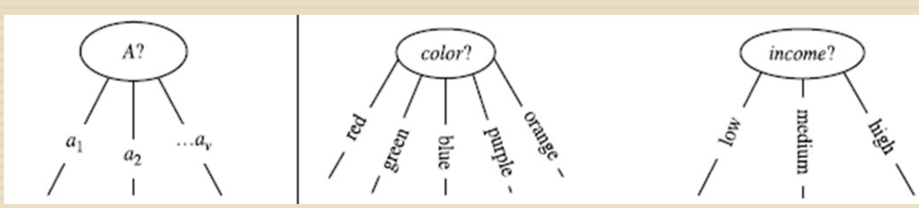
- ID3 (and many others) constructs decision trees in a top-down recursive divide-and-conquer manner.
- ID3 starts with a training set of tuples and their associated class labels.
- The training set is recursively partitioned into smaller subsets as the tree is being built.

ID3 Algorithm

- Step 1: Check the instances in the training set C
  - If all are positive, then create YES node and halt.
  - If all are negative, create a NO node and halt.
  - Otherwise select a feature, F with values v1, ..., vn and create a decision node.
- Step 2: Partition the training instances in C into subsets C1, C2, ..., Cn according to the values of V.
- Step 3: apply the algorithm recursively to each of the sets Ci.
- Features (attributes) are selected based on information gain.

Partition the training instances – Case 1

- If *A is discrete-valued, then one branch is* grown for each known value of *A*



Partition the training instances – Case 2

45

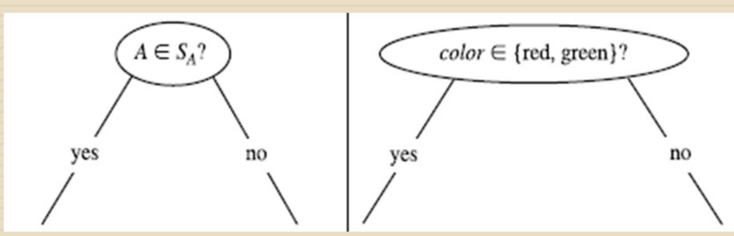
- If *A is continuous-valued, then two branches are* grown, corresponding to

A <= split point and A > split point.

Partition the training instances – Case 3

46

- If *A is discrete-valued* and a binary tree must be produced, then the test is of the form $A \in S_A$ *, where SA is the* splitting subset for *A*



Example of ID3

47

- We want ID3 to decide whether the weather is amenable to playing baseball.
- We have 14 days data.
- The target classification is "should we play baseball?" which can be yes or no.
- The weather attributes are outlook, temperature, humidity, and wind speed. They can have the following values:
  - outlook = { sunny, overcast, rain }
  - temperature = {hot, mild, cool }
  - humidity = { high, normal }
  - wind = {weak, strong }

Classification

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

48 Classification

Example of ID3

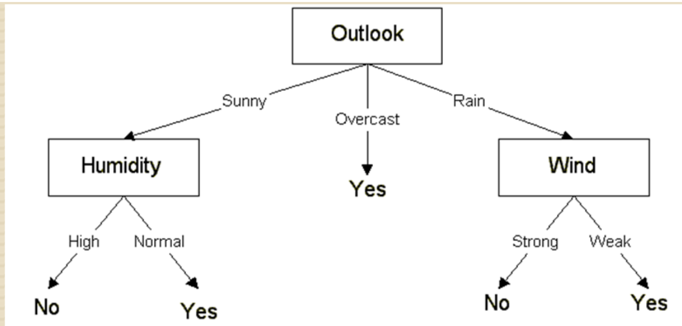
- We need to find which attribute will be the root node in our decision tree.
- The gain is calculated for all four attributes:
  - Gain(S, Outlook) = 0.246
  - Gain(S, Temperature) = 0.029
  - Gain(S, Humidity) = 0.151
  - Gain(S, Wind) = 0.048
- Outlook attribute has the highest gain, therefore it is used as the decision attribute in the root node.

Example of ID3

- Since Outlook has three possible values, the root node has three branches (sunny, overcast, rain).
- The next question is "what attribute should be tested at the Sunny branch node?" Since we=have used Outlook at the root, we only decide on the remaining three attributes: Humidity, Temperature, or Wind.
- S_{sunny} = {D1, D2, D8, D9, D11} = 5 examples from table 1 with outlook = sunny
  - Gain(S_{sunny}, Humidity) = 0.970
  - Gain(S_{sunny}, Temperature) = 0.570
  - Gain(S_{sunny}, Wind) = 0.019
- Humidity has the highest gain; therefore, it is used as the decision node.
- This process goes on until all data is classified perfectly or we run out of attributes.

Final Decision Tree

51



IF outlook = sunny AND humidity = high THEN playball = no
IF outlook = rain AND humidity = high THEN playball = no
IF outlook = rain AND wind = strong THEN playball = yes
IF outlook = overcast THEN playball = yes
IF outlook = rain AND wind = weak THEN playball = yes

ID3 Applications

52

- ID3 has been incorporated in a number of commercial rule-induction packages.
  - medical diagnosis,
  - credit risk assessment of loan applications,
  - equipment malfunctions by their cause,
  - classification of soybean diseases,
  - and web search classification.

Summary

53

- Introduction
- Naïve Bayes Classifiers
- K-Nearest Neighbor Classifiers
- Decision Trees
- Top-Down Induction of Decision Trees (TDIDT)
- Entropy and Information Gain
- ID3