CST413 – Machine Learning

Module-2 (Supervised Learning)

Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Naive Bayes, Decision tree algorithm ID3.

The Bayesian classifier

- The Bayesian classifier is an algorithm for classifying multiclass datasets.
- This is based on the Bayes' theorem in probability theory.
- The classifier is also known as "naïve Bayes Algorithm" where the word "naive" is an English word with the following meanings: simple, unsophisticated, or primitive

Conditional probability

- The probability of the occurrence of an event A given that an event B has already occurred is called the conditional probability of A given B and is denoted by P(A|B).

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \; if P(B) \neq 0$$

- Ex:
- B –FIRST EVENT; manufacturing bulbs from factory B
- A---second event –a bulb to be defective

❑ P(A|B) is called the posterior probability of A given B.

❑ P(B|A) is called the likelihood of B given A.

$$P(A \mid B) = P(A \cap B) / P(B)$$
$$P(B \mid A) = P(B \cap A) / P(A)$$
$$P(A \cap B) = P(B \cap A)$$
$$P(A \mid B).P(B) = P(B \mid A).P(A)$$
$$P(A \mid B) = \frac{P(B \mid A).P(A)}{P(B)}$$

- Posterior: Conditional probability distribution representing what parameters are likely after observing data object:
- Likelihood: The probability of falling under specific class

Bayes Theorem

- **Theorem**
- Let A and B any two events in a random experiment.

$$P(A \mid B) = \frac{P(B \mid A).P(A)}{P(B)} \; if P(B) \neq 0$$

- If P(B) ≠ 0, then

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

- A is called the proposition and B is called the evidence.
- P(A) is called the prior probability of proposition and P(B) is called the prior probability of evidence.
- P(A|B) is called the posterior probability of A given B.
- P(B|A) is called the likelihood of B given A.

Generalisation

- Let the sample space be divided into disjoint events B1,B2,...,Bn and A be any event.
- Then we have

$$P(B_K \mid A) = \frac{P(A \mid B_K)P(B_K)}{\sum_{i=1}^{n} P(A \mid B_i)P(B_i)}$$

Bayes Theorem

- **Goal:** To determine the most probable hypothesis(Class), given the data *D* plus any initial knowledge about the prior probabilities of the various hypotheses in *H*.

Independent events

- Two events A and B are said to be independent
  - if $P(A \cap B)=P(A)P(B)$.
- Three events A,B,C are said to be pairwise independent if
  - $P(B \cap C)=P(B)P(C)$
  - $P(C \cap A)=P(C)P(A)$
  - $P(A \cap B)=P(A)P(B)$
- Three events A,B,C are said to be mutually independent if
  - $P(B \cap C)=P(B)P(C)$
  - $P(C \cap A)=P(C)P(A)$
  - $P(A \cap B)=P(A)P(B)$
  - $P(A \cap B \cap C)=P(A)P(B)P(C)$

Bayes Theorem – Problem 1

- **Consider a set of patients coming for treatment in a certain clinic. Let A denote the event that a "Patient has liver disease" and B the event that a "Patient is an alcoholic." It is known from experience that 10% of the patients entering the clinic have liver disease and 5% of the patients are alcoholics. Also, among those patients diagnosed with liver disease, 7% are alcoholics. Given that a patient is alcoholic, what is the probability that he will have liver disease?**

Bayes Theorem - Solution

- P(A) = 10% = 0.10
- P(B) = 5% = 0.05
- P(B/A) = 7% = 0.07
- $P(\frac{A}{B}) = \frac{P(\frac{B}{A})P(A)}{P(B)}$
- $P(\frac{A}{B}) = \frac{0.07x0.10}{0.05} = 0.14$

Bayes Theorem – Problem 2

- **Three factories A, B, C of an electric bulb manufacturing company produce respectively 35%. 35% and 30% of the total output. Approximately 1.5%, 1% and 2% of the bulbs produced by these factories are known to be defective. If a randomly selected bulb manufactured by the company was found to be defective, what is the probability that the bulb was manufactures in factory A?**

Bayes Theorem - Solution

- Let A;B;C denote the events that a randomly selected bulb was manufactured in factory A, B, C respectively. Let D denote the event that a bulb is defective. We have the following data:

$$P(A) = 0.35, \quad P(B) = 0.35, \quad P(C) = 0.30$$
$$P(D|A) = 0.015, \quad P(D|B) = 0.010, \quad P(D|C) = 0.020$$

- We are required to find P(A/D). By the generalisation of the Bayes' theorem we have:

$$P(A|D) = \frac{P(D|A)P(A)}{P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C)}$$

$$= \frac{0.015 \times 0.35}{0.015 \times 0.35 + 0.010 \times 0.35 + 0.020 \times 0.30}$$
$$= 0.356.$$

Naïve Bayes Classifier

- Assumption
  - The naive Bayes algorithm is based on the following assumptions:
  - All the features are independent and are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.
  - The data has class-conditional independence, which means that events are independent so long as they are conditioned on the same class value.
  - These assumptions are, in general, true in many real world problems. It is because of these assumptions, the algorithm is called a naive algorithm

Sl. No.	Swim	Fly	Crawl	Class
1	Fast	No	No	Fish
2	Fast	No	Yes	Animal
3	Slow	No	No	Animal
4	Fast	No	No	Animal
5	No	Short	No	Bird
6	No	Short	No	Bird
7	No	Rarely	No	Animal
8	Slow	No	Yes	Animal
9	Slow	No	No	Fish
10	Slow	No	Yes	Fish
11	No	Long	No	Bird
12	Fast	No	No	Bird

Basic idea

- Suppose we have a training data set consisting of N examples having n features.
- Let the features be named as(F1,...,Fn).
- A feature vector is of the form(f1,f2,...,fn). Associated with each example, there is a certain class label.
- Let the set of class labels be{c1,c2,...,cp}. Suppose we are given a test instance having the feature vector X=(x1,x2,...,xn)

- We are required to determine the most appropriate class label that should be assigned to the test instance.
- For this purpose we compute the following conditional probabilities
  - –P(c1|X),P(c2|X),…,P(cp|X).
  - –and choose the maximum among them.
  - –Let the maximum probability be P(ci|X).
  - –Then,we choose ci as the most appropriate class label for the test instance having X as the feature vector.

- The various probabilities in the above expression are computed as follows:

$$p(C_k) = \frac{No.ofexampleswithclasslabel\ C_k}{TotalNo.ofexamples}$$

P(xj|ck)=No. of examples with jth feature equal to x j and class label ck/ No. of examples with class label ck

Algorithm: Naive Bayes

Let there be a training data set having n features $F_1, \ldots, F_n$. Let f_1 denote an arbitrary value of F_1, f_2 of F_2, and so on. Let the set of class labels be $\{c_1, c_2, \ldots, c_p\}$. Let there be given a test instance having the feature vector

$$X = (x_1, x_2, \ldots, x_n).$$

We are required to determine the most appropriate class label that should be assigned to the test instance.

Step 1. Compute the probabilities $P(c_k)$ for $k = 1, \ldots, p$.

Step 2. Form a table showing the conditional probabilities

$$P(f_1|c_k), \quad P(f_2|c_k), \quad \ldots \quad , P(f_n|c_k)$$

for all values of $f_1, f_2, \ldots, f_n$ and for $k = 1, \ldots, p$.

Step 3. Compute the products

$$q_k = P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k)P(c_k)$$

for $k = 1, \ldots, p$.

Step 4. Find j such $q_j = \max\{q_1, q_2, \ldots, q_p\}$.

Step 5. Assign the class label c_j to the test instance X.

Predicting a class label using naïve Bayesian classification.

- Problem 1

Sl. No.	Swim	Fly	Crawl	Class
1	Fast	No	No	Fish
2	Fast	No	Yes	Animal
3	Slow	No	No	Animal
4	Fast	No	No	Animal
5	No	Short	No	Bird
6	No	Short	No	Bird
7	No	Rarely	No	Animal
8	Slow	No	Yes	Animal
9	Slow	No	No	Fish
10	Slow	No	Yes	Fish
11	No	Long	No	Bird
12	Fast	No	No	Bird

- ***X*** = (*swim = slow, Fly = rarely, crawl = No*)

Step 1. We compute following probabilities.

$$P(c_1) = \frac{\text{No. of records with class label “Animal”}}{\text{Total number of examples}}$$

$$= 5/12$$

$$P(c_2) = \frac{\text{No. of records with class label “Bird”}}{\text{Total number of examples}}$$

$$= 4/12$$

$$P(c_3) = \frac{\text{No of records with class label “Fish”}}{\text{Total number of examples}}$$

$$= 3/12$$

Step 2. We construct the following table of conditional probabilities:

Class	Features								
	Swim (F_1) f_1			Fly (F_2) f_2				Crawl (F_3) f_3	
	Fast	Slow	No	Long	Short	Rarely	No	Yes	No
Animal (c_1)	2/5	2/5	1/5	0/5	0/5	1/5	4/5	2/5	3/5
Bird (c_2)	1/4	0/4	3/4	1/4	2/4	0/4	1/4	0/4	4/4
Fish (c_3)	1 3	2/3	0/3	0/3	0/3	0/3	3/3	1/3	2/3

Table 6.3: Table of the conditional probabilities $P(f_i|c_k)$

Step 3. We now calculate the following numbers:

$$
\begin{aligned}
q_1 &= P(x_1|c_1)P(x_2|c_1)P(x_3|c_1)P(c_1) \\
&= (2/5) \times (1/5) \times (3/5) \times (5/12) \\
&= 0.02 \\
q_2 &= P(x_1|c_2)P(x_2|c_2)P(x_3|c_2)P(c_2) \\
&= (0/4) \times (0/4) \times (3/4) \times (4/12) \\
&= 0 \\
q_3 &= P(x_1|c_3)P(x_2|c_3)P(x_3|c_3)P(c_3) \\
&= (2/3) \times (0/3) \times (3/3) \times (3/12) \\
&= 0
\end{aligned}
$$

Step 4. Now

$$\max\{q_1, q_2, q_3\} = 0.0^{2}$$

Step 5. The maximum is q_1 an it corresponds to the class label

c_1 = " Animal".

So we assign the class label "Animal" to the test instance "(Slow, Rarely, No)".

Example 2:

- Given the following data on a certain set of patients seen by a doctor, can the doctor conclude that a person having chills, fever, mild headache and without running nose has the flu?

chills	running nose	headache	fever	has flu
Y	N	mild	Y	N
Y	Y	no	N	Y
Y	N	strong	Y	Y
N	Y	mild	Y	Y
N	N	no	N	N
N	Y	strong	Y	Y
N	Y	strong	N	N
Y	Y	mild	Y	Y

Predicting a class label using naïve Bayesian classification.

Table 6.1 Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Predicting a class label using naïve Bayesian classification.

- The data tuples are described by the attributes age, income, student, and credit rating.
- The class label attribute, buys computer, has two distinct values (namely, {yes, no}).
- Let C1 correspond to the class buys computer = yes and C2 correspond to buys computer = no.
- The tuple we wish to classify is
- ***X*** = (*age = youth, income = medium, student = yes, credit rating = fair*)

Predicting a class label using naïve Bayesian classification.

- We need to maximize P(X/Ci)P(Ci), for i = 1, 2.
- P(Ci), the prior probability of each class, can be computed based on the training tuples:
  - *P*(*buys computer = yes*) = 9/14 = 0.643
  - *P*(*buys computer = no*) = 5/14 = 0.357

Predicting a class label using naïve Bayesian classification.

- To compute P(X/Ci), for i = 1, 2, we compute the following conditional probabilities:
- P(age = youth | buys computer = yes) = 2/9 = 0.222
- P(age = youth | buys computer = no) = 3/5 = 0.600
  - P(income = medium | buys computer = yes) = 4/9 = 0.444
  - P(income = medium | buys computer = no) = 2/5 = 0.400
- P(student = yes | buys computer = yes) = 6/9 = 0.667
- P(student = yes | buys computer = no) = 1/5 = 0.200
  - P(credit rating = fair | buys computer = yes) = 6/9 = 0.667
  - P(credit rating = fair | buys computer = no) = 2/5 = 0.400

Predicting a class label using naïve Bayesian classification.

- Using the above probabilities, we obtain
  - P(X | buys_computer = yes)

    =P(age = youth | buys computer = yes) x P(income = medium | buys computer = yes) x P(student = yes | buys computer = yes) x P(credit rating = fair | buys computer = yes)

  = 0.222x0.444x0.667x0.667 = 0.044

Predicting a class label using naïve Bayesian classification.

- Similarly, P(X | buys computer = no) = 0.600x0.400x0.200x0.400 = 0.019.

- CLASS LABEL=YES

Using numeric features with Naïve Bayes Algorithm

- The naive Bayes algorithm can be applied to a data set only if the features are categorical.
- This is so because, the various probabilities are computed using the various frequencies and the frequencies can be counted only if each feature has a limited set of values.
- If a feature is numeric, it has to be discretized before applying the algorithm.
- The discretization is effected by putting the numeric values into categories known as bins.
- Because of this discretization is also known as binning.

- This is ideal when there are large amounts of data. There are several different ways to discretize a numeric feature.

1.If there are natural categories or cutpoints in the distribution of values,use these cutpoints to create the bins. For example, let the data consists of records of times when certain activities were carried out.

- The categories, or bins, may be created as follows

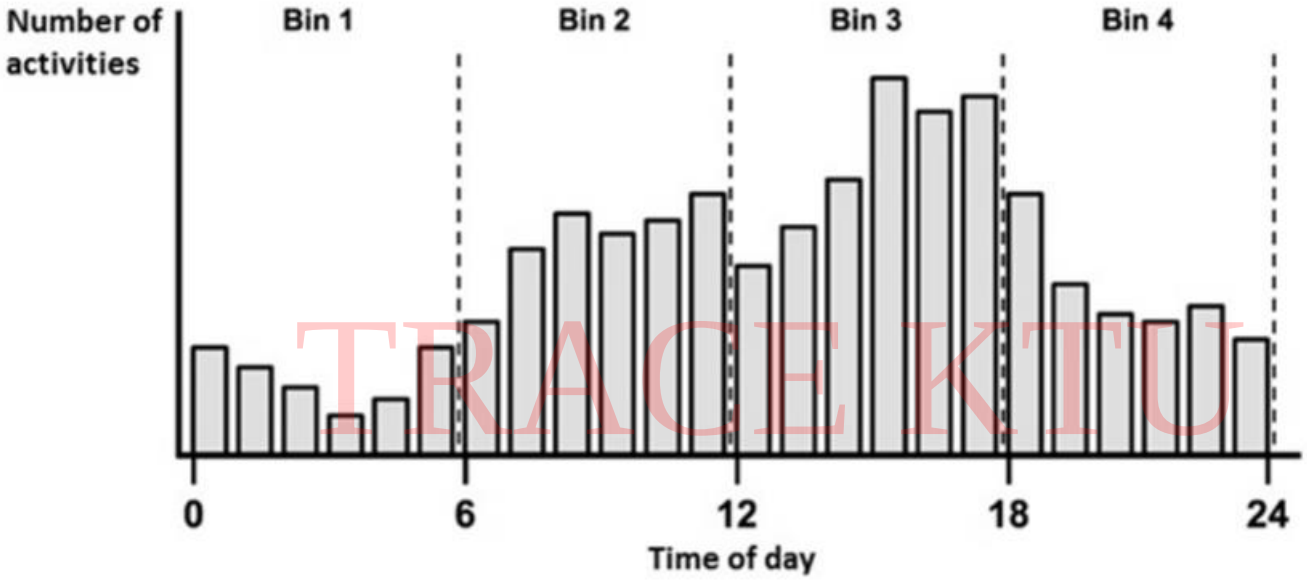


Figure 6.3: Discretization of numeric data: Example

- If there are no obvious cut points, we may discretize the feature using quantiles.
- We may divide the data into 2 bins-median,three bins with tertiles,four bins with quartiles,or five bins with quintiles, etc.
-

Maximum Likelihood Estimation (ML estimation)

- To develop a Bayesian classifier, we need the probabilities P(x|ck)for the class labels c1,...,ck.
- These probabilities are estimated from the given data.
- There is need to know whether the sample is truly random so that the computed probabilities are good approximations to true probabilities.
- If they are good approximations of true probabilities, then there would be an underlying probability distribution.
- Suppose we have reasons to believe that the underlying distribution has a particular form, say binomial, Poisson or normal.
-

- **These forms are defined by probability functions or probability density functions.**
- There are parameters which define these functions, and these parameters are to be estimated to test whether a given data follow some particular distribution.
- ***Maximum likelihood estimation is particular method to estimate the parameters of a probability distribution.***

Definition

- ***Maximum likelihood estimation(MLE) is a method of estimating the parameters of a statistical model, given observations.***
- ***MLE attempts to find the parameter values that maximize the likelihood function, given the observations.***
- ***The resulting estimate is called a maximum likelihood estimate, which is also abbreviated as MLE.***

General MLE Method

- Suppose we have a random sample
  - $X=\{x1,...,xn\}$
- taken from a probability distribution having the probability mass function or probability density function $p(x|\theta)$
  - where x denotes a value of the random variable and
- θ denotes the set of parameters that appear in the function.
- The likelihood of sample X is a function of the parameter θ and is defined as

$$l(\theta) = p(x_1|\theta)p(x_2|\theta)\ldots p(x_n|\theta).$$

- In maximum likelihood estimation,
- we find the value of θ that makes the value of the likelihood function maximum.
- we define the **log likelihood function** as the logarithm of the likelihood function:

$$L(\theta) = \log l(\theta)$$
$$= \log p(x_1|\theta) + \log p(x_2|\theta) + \cdots + \log p(x_n|\theta).$$

Special cases:

- Parameter estimation:[derivation of all]Refer note
  - Bernoulli density
  - Binomial distribution
  - Gaussian (Normal) density
  - Poission distribution
  - Geometric distribution
  - Exponential distribution=$f_{X_i}(x_i) = \lambda e^{-\lambda x_i}$.

Bernoulli Density

The probability function of X is given by

$$f(x|p) = p^x(1-p)^{1-x}, \quad x = 0, 1.$$

In this function, the probability p is the only parameter.

Estimation of p

Consider a random sample $X = \{x_1, \ldots, x_n\}$ taken from a Bernoulli distribution with the probability function $f(x|p)$. The log likelihood function is

$$\begin{aligned} L(p) &= \log f(x_1|p) + \cdots + \log f(x_n|p) \\ &= \log p^{x_1}(1-p)^{1-x_1} + \cdots + \log p^{x_n}(1-p)^{1-x_n} \\ &= [x_1 \log p + (1-x_1)\log(1-p)] + \cdots + [x_n \log p + (1-x_n)\log(1-p)] \end{aligned}$$

To find the value of p that maximizes $L(p)$ we set up the equation

$$\frac{dL}{dp} = 0,$$

that is,

$$\left[\frac{x_1}{p} - \frac{1-x_1}{1-p}\right] + \cdots + \left[\frac{x_n}{p} - \frac{1-x_n}{1-p}\right] = 0.$$

Solving this equation, we have the maximum likelihood estimate of p as

$$\hat{p} = \frac{1}{n}(x_1 + \cdots + x_n).$$

2. Multinomial density

Suppose that the outcome of a random event is one of K classes, each of which has a probability of occurring p_i with

$$p_1 + \cdots + p_K = 1.$$

We represent each outcome by an ordered K-tuple $\mathbf{x} = (x_1, \ldots, x_K)$ where exactly one of $x_1, \ldots, x_K$ is 1 and all others are 0. $x_i = 1$ if the outcome in the i-th class occurs. The probability function can be expressed as

$$f(\mathbf{x}|p_1, \ldots, p_K) = p_1^{x_1} \ldots p_K^{x_K}.$$

Here, $p_1, \ldots, p_K$ are the parameters.

We choose n random samples. The i-the sample may be represented by

$$\mathbf{x}_i = (x_{1i}, \ldots, x_{Ki}).$$

The values of the parameters that maximizes the likelihood function can be shown to be

$$\hat{p}_k = \frac{1}{n}(x_{k1} + x_{k2} + \cdots + x_{kn}).$$

Problem

- A coin is tossed 100 times and lands heads 62 times. What is the maximum likelihood estimate for θ, the probability of heads.

- Suppose the data x1, x2, . . . , xn is drawn from a $N(\mu, \sigma 2)$ distribution, where μ and σ are unknown. Find the maximum likelihood estimate for the pair $(\mu, \sigma 2)$

- Suppose data x1, ..., xn are independent and identically distributed drawn from an exponential distribution exp(λ). Find the maximum likelihood for λ.

Example 3. Light bulbs

Suppose that the lifetime of *Badger* brand light bulbs is modeled by an exponential distribution with (unknown) parameter λ. We test 5 bulbs and find they have lifetimes of 2, 3, 1, 3, and 4 years, respectively. What is the MLE for λ?

answer: We need to be careful with our notation. With five different values it is best to use subscripts. Let X_j be the lifetime of the i^{th} bulb and let x_i be the value X_i takes. Then each X_i has pdf $f_{X_i}(x_i) = \lambda e^{-\lambda x_i}$. We assume the lifetimes of the bulbs are independent, so the joint pdf is the product of the individual densities:

$$f(x_1, x_2, x_3, x_4, x_5 \,|\, \lambda) = (\lambda e^{-\lambda x_1})(\lambda e^{-\lambda x_2})(\lambda e^{-\lambda x_3})(\lambda e^{-\lambda x_4})(\lambda e^{-\lambda x_5}) = \lambda^5 e^{-\lambda(x_1+x_2+x_3+x_4+x_5)}.$$

Note that we write this as a conditional density, since it depends on λ. Viewing the data as fixed and λ as variable, this density is the likelihood function. Our data had values

$$x_1 = 2,\ x_2 = 3,\ x_3 = 1,\ x_4 = 3,\ x_5 = 4.$$

So the likelihood and log likelihood functions with this data are

$$f(2, 3, 1, 3, 4 \,|\, \lambda) = \lambda^5 e^{-13\lambda}, \quad \ln(f(2, 3, 1, 3, 4 \,|\, \lambda) = 5 \ln(\lambda) - 13\lambda$$

Finally we use calculus to find the MLE:

$$\frac{d}{d\lambda}(\text{log likelihood}) = \frac{5}{\lambda} - 13 = 0 \ \Rightarrow\ \boxed{\hat{\lambda} = \frac{5}{13}}.$$

MAP

Maximum A Posterior Estimation

MLE is great, but it is not the only way to estimate parameters! This section introduces an alternate algorithm, Maximum A Posteriori (MAP).The paradigm of MAP is that we should chose the value for our parameters that is the most likely given the data. At first blush this might seem the same as MLE, however notice that MLE chooses the value of parameters that makes the *data* most likely. Formally, for IID random variables $X_1, \ldots, X_n$:

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n)$$

In the equation above we trying to calculate the conditional probability of unobserved random variables given observed random variables. When that is the case, think Bayes Theorem! Expand the function f using the continuous version of Bayes Theorem.

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n) && \text{Now apply Bayes Theorem} \\
&= \underset{\theta}{\text{argmax}}\ \frac{f(X_1, X_2, \ldots, X_n | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Ahh much better}
\end{aligned}$$

Note that f, g and h are all probability densities. I used different symbols to make it explicit that they may have different functions. Now we are going to leverage two observations. First, the data is assumed to be IID so we can decompose the density of the data given θ. Second, the denominator is a constant with respect to θ. As such its value does not affect the argmax and we can drop that term. Mathematically:

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ \frac{\prod_{i=1}^{n} f(X_i | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Since the samples are IID} \\
&= \underset{\theta}{\text{argmax}}\ \prod_{i=1}^{n} f(X_i | \theta) g(\theta) && \text{Since } h \text{ is constant with respect to } \theta
\end{aligned}$$

As before, it will be more convenient to find the argmax of the log of the MAP function, which gives us the final form for MAP estimation of parameters.

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}} \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i | \theta)) \right)$$

Using Bayesian terminology, the MAP estimate is the mode of the "posterior" distribution for θ. If you look at this equation side by side with the MLE equation you will notice that MAP is the argmax of the exact same function *plus* a term for the log of the prior.

Decision Trees

Decision Trees- Entropy, Information Gain, Tree construction, ID3,

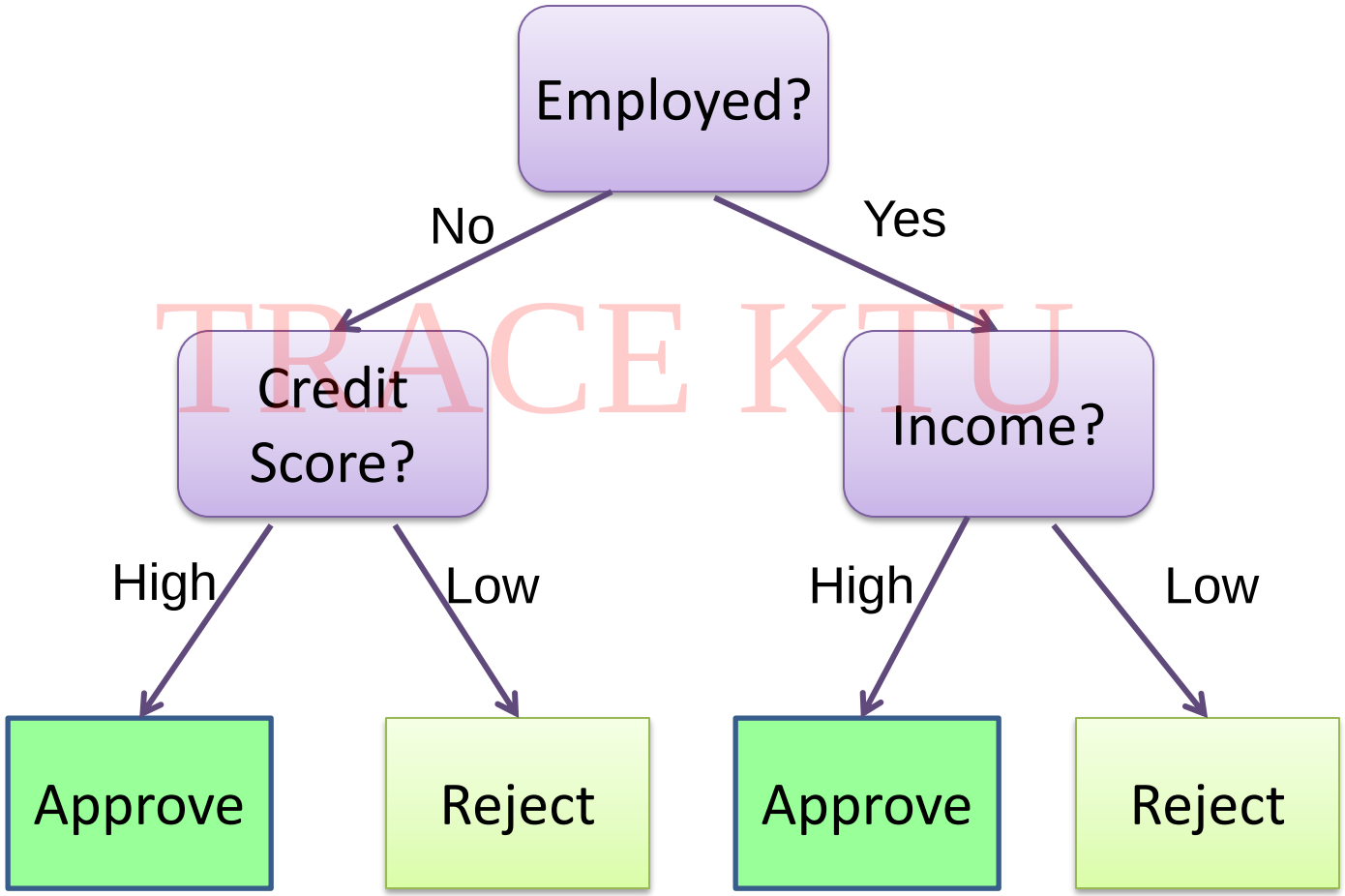
Issues in Decision Tree learning- Avoiding Over-fitting, Reduced Error Pruning,

Decision Trees

- A decision tree is a classifier in the form of a tree structure with two types of nodes:
  - Decision node(Root,Internal): Specifies a choice or test of some attribute, with one branch for each outcome
  - Leaf node: Indicates classification of an example
- A Decision tree is a flowchart like tree structure, where
  - each internal node denotes a test on an attribute,
  - each branch represents an outcome of the test,
  - each leaf node (terminal node) holds a class label.

Decision Tree Example 1

Whether to approve a loan



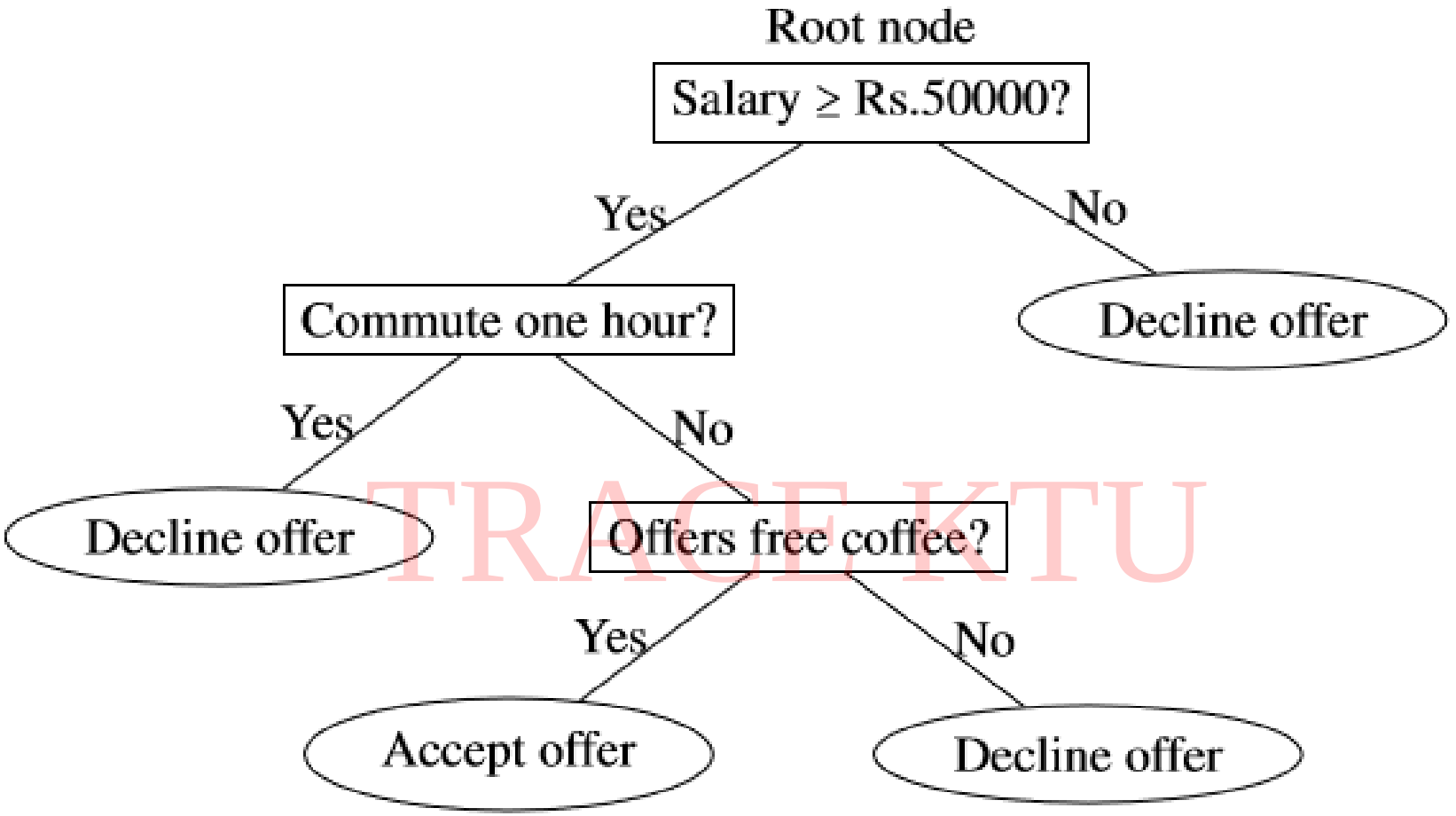


Figure 8.1: Example for a decision tree

Two types of decision trees:

- Two types of decision trees.

1.Classification trees Tree models where the target variable can take a discrete set of values are called classification trees.

  - In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

2. Regression trees: Decision trees where the target variable can take continuous values (real numbers) like the price of a house, or a patient's length of stay in a hospital, are called regression trees.

Construction of tree

- A decision tree is a hierarchical model for supervised learning whereby the local region is identified in a sequence of recursive splits in a smaller number of steps.
- A decision tree is composed of internal decision nodes and terminal leaves.
- Each *decision node m* implements a test function $f_m(\boldsymbol{x})$ with discrete outcomes labeling the branches.

- Given an input, at each node, a test is applied and one of the branches is taken depending on the outcome.
- This process starts at the root and is repeated recursively until a *leaf node* is hit, at which point the value written in the leaf constitutes the output.

Issues

- Given some training examples, what decision tree should be generated?
- One proposal: prefer the smallest tree that is consistent with the data (Bias)
  - the tree with the least depth?
  - the tree with the fewest nodes?
- Possible method:
  - search the space of decision trees for the smallest decision tree that fits the data

Nam	Features				Class label
	gives birth	aquatic animal	aerial animal	has legs	
human	yes	no	no	yes	mammal
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
bat	yes	no	yes	yes	bird
pigeon	no	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.1: The vertebrate data set

This stage of the classification can be represented as in Figure 8.3.

Name	Gives birth	Aquatic animal	Aerial animal	Has legs	Class label
human	yes	no	no	yes	mammal
bat	yes	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish

Table 8.2: The subset of Table 8.1 with “gives birth” = ”yes"

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
pigeon	no	no	yes	yes	bird
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.3: The subset of Table 8.1 with “gives birth” = ”no"

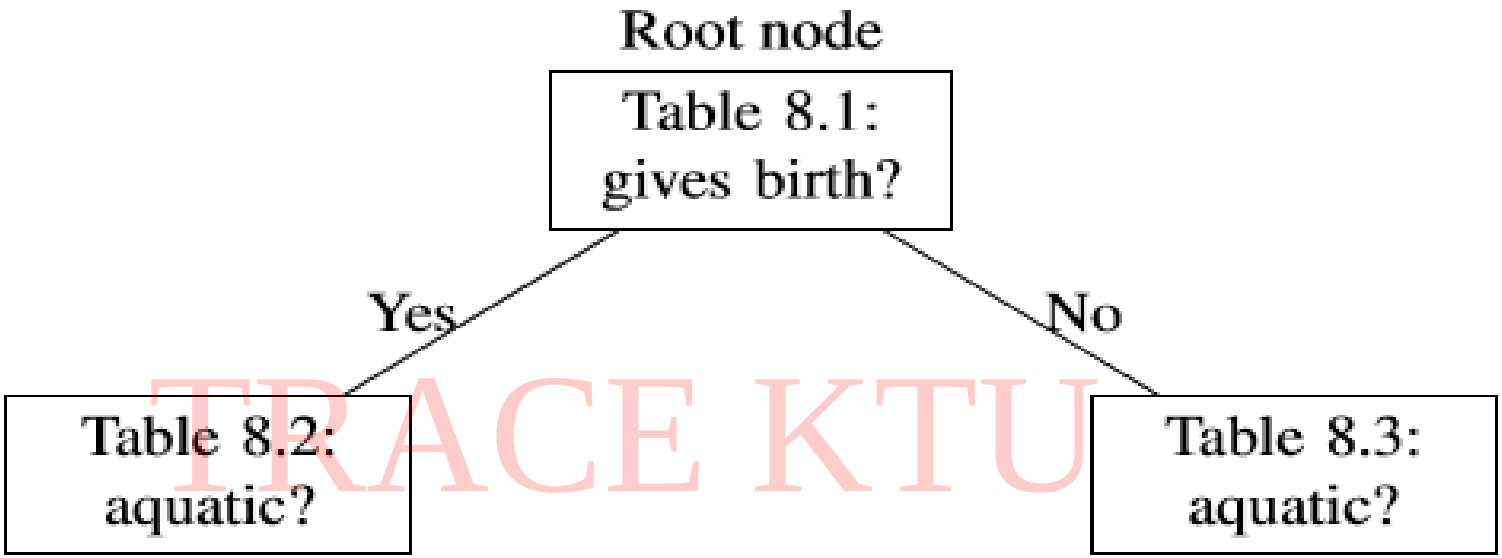


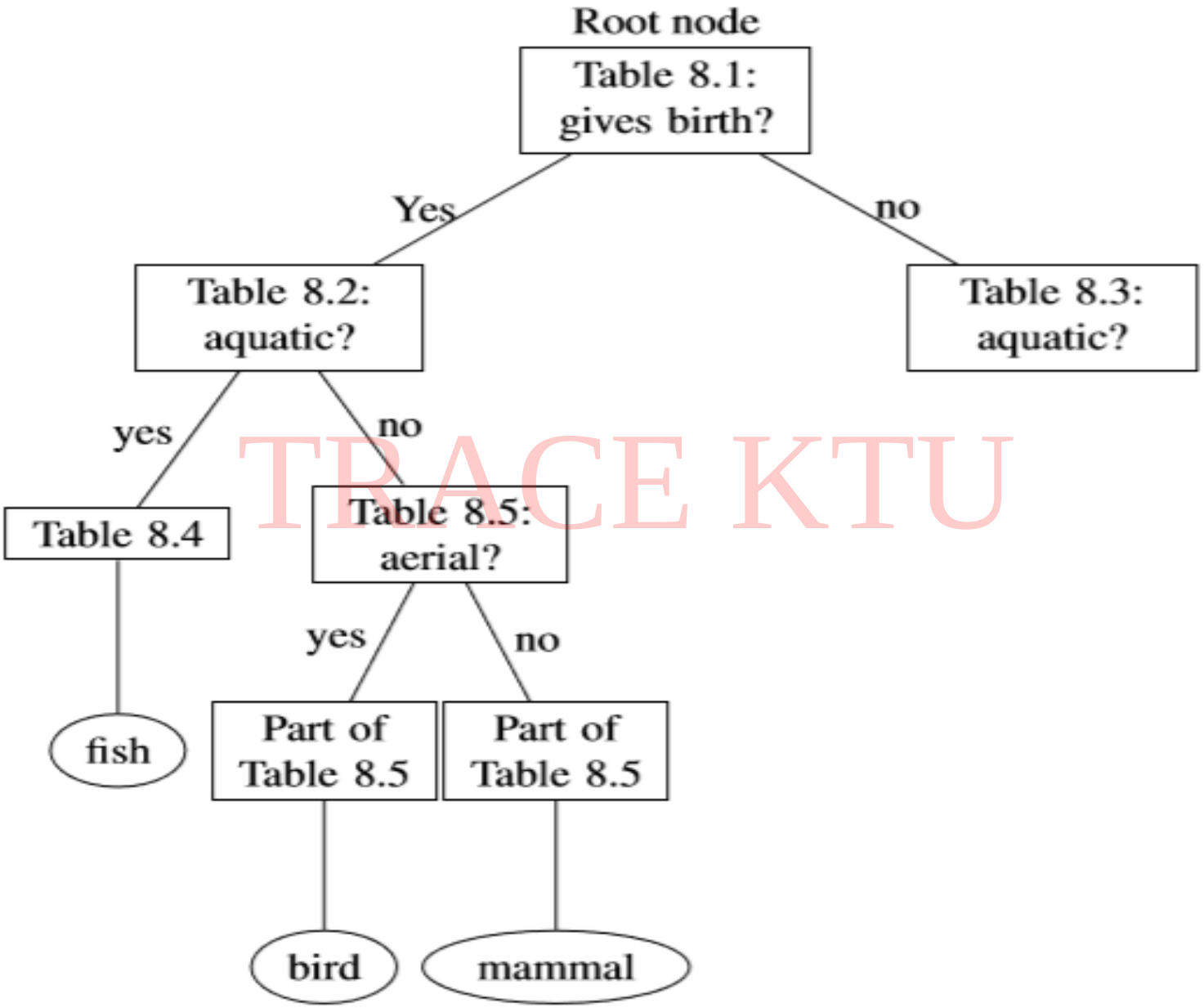
Figure 8.3: Classification tree

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
human	yes	no	no	yes	mammal
bat	yes	no	yes	yes	bird
cat	yes	no	no	yes	mammal

Table 8.5: The vertebrate data set

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
shark	yes	yes	no	no	fish

Table 8.4: The vertebrate data set


Root node
Table 8.1:
gives birth?
Yes
no
Table 8.2:
aquatic?
Table 8.3:
aquatic?
yes
no
Table 8.4
Table 8.5:
aerial?
yes
no
fish
Part of
Table 8.5
Part of
Table 8.5
bird
mammal

STEP 3:

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
pigeon	no	no	yes	yes	bird
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.3: The subset of Table 8.1 with “gives birth” = ”no"

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
salmon	no	yes	no	no	fish

Table 8.6: The vertebrate data set

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
frog	no	semi	no	yes	amphibian
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.7: The vertebrate data set

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
python	no	no	no	no	reptile
pigeon	no	no	yes	yes	bird

Table 8.8: The vertebrate data set

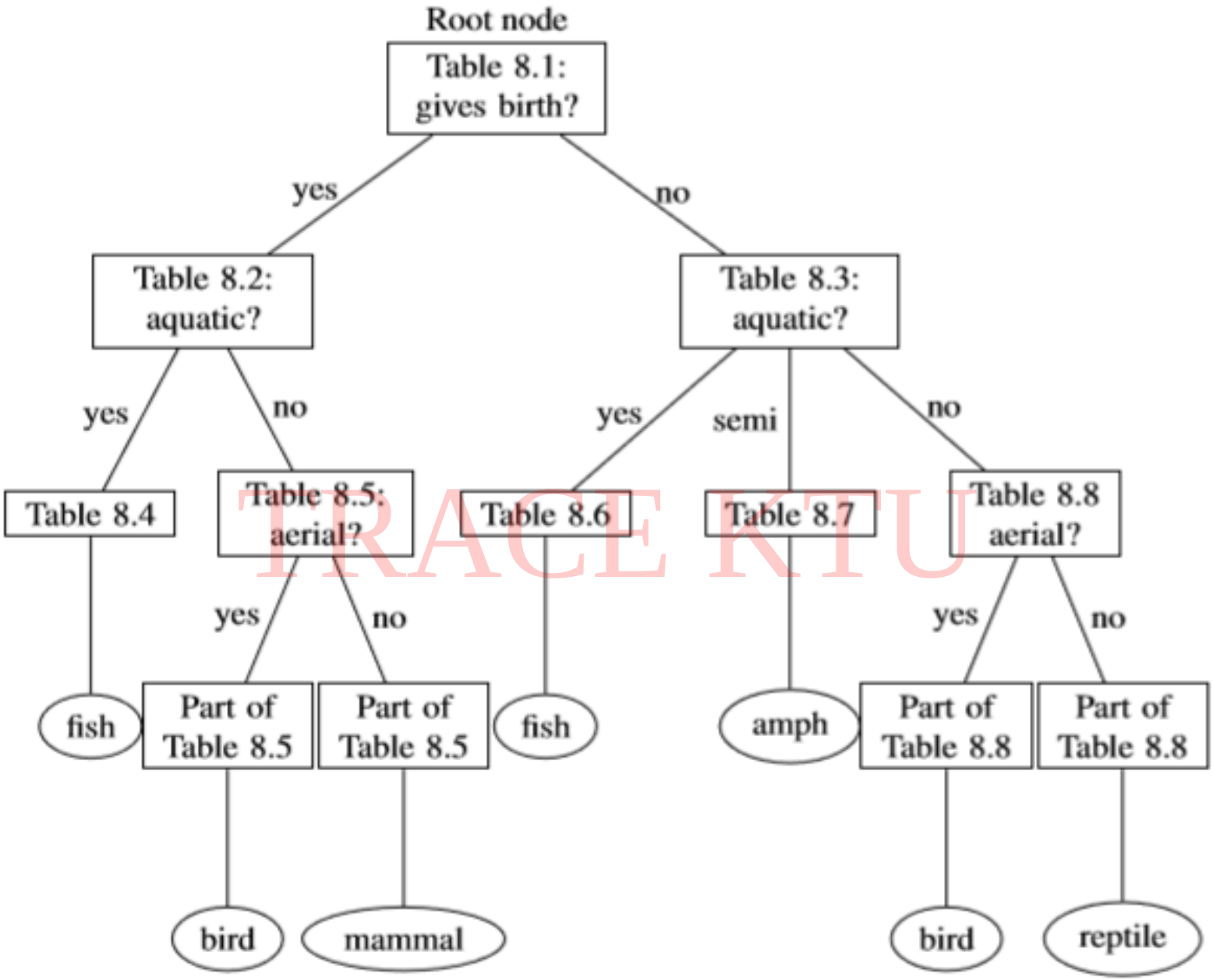


Figure 8.5: Classification tree

8.3.2 Classification tree in rule format

The classification tree shown in Figure 8.5 can be presented as a set of algorithm.

Algorithm for classification of vertebrates

```
1.  if give birth = "yes" then
2.      if aquatic = "yes" then
3.          return class = "fish"
4.      else
5.          if aerial = "yes" then
6.              return class = "bird"
7.          else
8.              return class = "mammal"
9.          end if
10.     end if
11. else
12.     if aquatic = "yes" then
13.         return class = "fish"
```

```
14.        end if
15.        if aquatic = "semi" then
16.              return  class = "amphibian"
17.        else
18.              if aerial = "yes" then
19.                    return  class = "amphibian"
20.              else
21.                    return  class = "reptile"
22.              end if
23.        end if
24.  end if
```

Feature selection measures

- The most important problem in implementing the decision tree algorithm is deciding which features are to be considered as the root node and at each level.
- Several methods have been developed to assign numerical values to the various features such that the values reflect the relative importance of the various features.
- These are called the feature selection measures.
- Two of the popular feature selection measures are **information gain and Gini index.**

What Is Entropy?

- Any subset composed of only a single class is called a pure class.
- The degree to which a subset of examples contains only a single class is known as purity.
- ***Entropy is a measure of "impurity" in a dataset.***
- Entropy, as it relates to machine learning, is a measure of the randomness in the information being processed.
- The higher the entropy, the harder it is to draw any conclusions from that information.

What Is Entropy?

Definition

- Consider a segment S of a dataset having c number of class labels.
- Let pi be the proportion of examples in S having the i^{th} class label.
- The entropy of S is defined as

$$\text{Entropy}\,(S) = \sum_{i=1}^{c} -p_i \log_2(p_i).$$

- Let the data segment S has only two class labels, say, “yes” and “no”.
- If p is the proportion of examples having the label “yes”
- the proportion of examples having label “no” will be 1−p.
- In this case, the entropy of S is given by

$$\text{Entropy}\,(S) = -p\log_2(p) - (1-p)\log_2(1-p)$$

Nam	Features				Class label
	gives birth	aquatic animal	aerial animal	has legs	
human	yes	no	no	yes	mammal
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
bat	yes	no	yes	yes	bird
pigeon	no	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.1: The vertebrate data set

Entropy of data inTable8.1

- Number of examples with class label “amphi” = 3
- Number of examples with class label “bird” = 2
- Number of examples with class label “fish” = 2
- Number of examples with class label “mammal” = 2
- Number of examples with class label “reptile” = 1
- Total number of examples = 10

Let“xxx”be some class label.

$$\text{Entropy } (S) = \sum_{\text{for all classes “xxx”}} -p_{\text{xxx}} \log_2 (p_{\text{xxx}})$$

$$
\begin{aligned}
&= -p_{\text{amphi}} \log_2(p_{\text{amphi}}) - p_{\text{bird}} \log_2(p_{\text{bird}}) \\
&\quad - p_{\text{fish}} \log_2(p_{\text{fish}}) - p_{\text{mammal}} \log_2(p_{\text{mammal}}) \\
&\quad - p_{\text{reptile}} \log_2(p_{\text{reptile}}) \\
&= -(3/10)\log_2(3/10) - (2/10)\log_2(2/10) \\
&\quad - (2/10)\log_2(2/10) - (2/10)\log_2(2/10) \\
&\quad - (1/10)\log_2(1/10) \\
&= 2.2464
\end{aligned}
$$

Entropy of data inTable8.2

Name	Gives birth	Aquatic animal	Aerial animal	Has legs	Class label
human	yes	no	no	yes	mammal
bat	yes	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish

class labels appear in this segment, namely, “bird”, “fish” and “mammal”.

Number of examples with class label “bird”	1
Number of examples with class label “fish”	1
Number of examples with class label “mammal”	2
Total number of examples	4

$$\text{Entropy}(S) = \sum_{\text{for all classes "xxx"}} -p_{\text{xxx}} \log_2(p_{\text{xxx}})$$

$$= -p_{\text{bird}} \log_2(p_{\text{bird}}) - p_{\text{fish}} \log_2(p_{\text{fish}}) - p_{\text{mammal}} \log_2(p_{\text{mammal}})$$

$$= -(1/4)\log_2(1/4) - (1/4)\log_2(1/4) - (2/4)\log_2(2/4)$$

$$= -(1/4) \times (-2) - (1/4) \times (-2) - (2/4) \times (-1)$$

$$= 1.5$$

EntropyofdatainTable8.3

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
pigeon	no	no	yes	yes	bird
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Number of examples with class label "amphi"	3
Number of examples with class label "bird"	1
Number of examples with class label "fish"	1
Number of examples with class label "reptile"	1
Total number of examples	6

Therefore, we have:

$$\text{Entropy}\,(S) = \sum_{\text{for all classes "xxx"}} -p_{\text{xxx}} \log_2(p_{\text{xxx}})$$

$$= -p_{\text{amphi}} \log_2(p_{\text{amphi}}) - p_{\text{bird}} \log_2(p_{\text{bird}}) - p_{\text{fish}} \log_2(p_{\text{fish}}) - p_{\text{reptile}} \log_2(p_{\text{reptile}})$$

$$= -(3/6)\log_2(3/6) - (1/6)\log_2(1/6) - (1/6)\log_2(1/6) - (1/6)\log_2(1/6)$$

$$= 1.7925$$

Day	outlook	temperature	humidity	wind	playtennis
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

Training examples for the target concept “PlayTennis”

What Is Entropy?

$$\begin{aligned}\text{Entropy}\,(S) &= -p_{\text{yes}}\log_2(p_{\text{yes}}) - p_{\text{no}}\log_2(p_{\text{no}}) \\ &= -(9/14)\times\log_2\left(9/14\right) - (5/14)\times\log_2\left(5/14\right) \\ &= 0.9405\end{aligned}$$

What is information Gain?

- Information gain tells us how important a given attribute of the feature vectors is.
- How much “information a feature gives us about the class”.
- We will use it to decide the ordering of attributes in the nodes of a decision tree.

What is information Gain?

- Let *S* be a set of examples,
- *A* be a feature (or, an attribute),
- S_v be the subset of *S* with *A* = *v*,
- and Values (*A*) be the set of all possible values of *A*.
- Then the *information gain of an attribute A relative to the set S*, denoted by Gain (*S,A*), is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v).$$

Nam	Features				Class label
	gives birth	aquatic animal	aerial animal	has legs	
human	yes	no	no	yes	mammal
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
bat	yes	no	yes	yes	bird
pigeon	no	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.1: The vertebrate data set

$$|S| = 10$$
$$\text{Entropy}\,(S) = 2.2464.$$

/e denote the information gain corresponding to the feature "xxx" by Gain (S, xxx).

1. **Computation of Gain $(S,$ gives birth$)$**

$$A_1 = \text{gives birth}$$
$$\text{Values of } A_1 = \{\text{“yes”}, \text{“no”}\}$$
$$S_{A_1=\text{yes}} = \text{Data in Table 8.2}$$
$$|S_{A_1=\text{yes}}| = 4$$
$$\text{Entropy}\,(S_{A_1=\text{yes}}) = 1.5 \quad \text{(See Eq.(8.1))}$$
$$S_{A_1=\text{no}} = \text{Data in Table 8.3}$$
$$|S_{A_1=\text{no}}| = 6$$
$$\text{Entropy}\,(S_{A_1=\text{no}}) = 1.7925 \quad \text{(See Eq.(8.2))}$$

$$\text{Gain}(S, A_1) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A_1)} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - \frac{|S_{A_1=\text{yes}}|}{|S|} \times \text{Entropy}(S_{A_1=\text{yes}}) - \frac{|S_{A_1=\text{no}}|}{|S|} \times \text{Entropy}(S_{A_1=\text{no}})$$

$$= 2.2464 - (4/10) \times 1.5 - (6/10) \times 1.7925$$

$$= 0.5709$$

Computation of Gain (S, aquatic)

$$
\begin{aligned}
A_2 &= \text{aquatic} \\
\text{Values of } A_2 &= \{\text{“yes”}, \text{“no”}, \text{“semi”}\} \\
S_{A_2=\text{yes}} &= \text{See Table 8.1} \\
|S_{A_2=\text{yes}}| &= 2 \\
\text{Entropy}\,(S_{A_2=\text{yes}}) &= -p_{\text{fish}} \log_2 (p_{\text{fish}}) \\
&= -(2/2) \log_2 (2/2) \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
S_{A_2=\text{no}} &= \text{See Table 8.1} \\
|S_{A_2=\text{no}}| &= 5 \\
\text{Entropy}\,(S_{A_2=\text{no}}) &= -p_{\text{mammal}} \log_2(p_{\text{mammal}}) - p_{\text{reptile}} \log_2(p_{\text{reptile}}) \\
&\quad - p_{\text{bird}} \log_2(p_{\text{bird}}) \\
&= -(2/5) \times \log_2(2/5) - (1/5) \times \log_2(1/5) \\
&\quad - (2/5) \times \log_2(2/5) \\
&= 1.5219
\end{aligned}
$$

$$
\begin{aligned}
S_{A_2=\text{semi}} &= \text{See Table 8.1} \\
|S_{A_2=\text{semi}}| &= 3 \\
\text{Entropy}\,(S_{A_2=\text{semi}}) &= -p_{\text{amphi}} \log_2(p_{\text{amphi}}) \\
&= -(3/3) \times \log_2(3/3) \\
&= 0
\end{aligned}
$$

$$\text{Gain}(S, A_2) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A_2)} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - \frac{|S_{A_1=\text{yes}}|}{|S|} \times \text{Entropy}(S_{A_1=\text{yes}})$$

$$- \frac{|S_{A_1=\text{no}}|}{|S|} \times \text{Entropy}(S_{A_1=\text{no}})$$

$$- \frac{|S_{A_1=\text{semi}}|}{|S|} \times \text{Entropy}(S_{A_1=\text{semi}})$$

$$= 2.2464 - (2/10) \times 0 - (5/10) \times 1.5219 - (3/3) \times 0$$

$$= 1.48545$$

- Find information gain
  - Gain(S,aerial) gain(s,has legs) ??

  - **Attribute with highest information gain is selected as ROOT.**
  - **Reduction Entropy is more.**

Gini indices

- The Gini split index of a data set is another feature selection measure in the construction of classification trees.
- Gini index
  - Consider a data set S having r class labels c1,...,cr.
  - Let pi be the proportion of examples having the class label ci.
  - The Gini index of the data set S, denoted by Gini(S), is defined by

$$\mathrm{Gini}(S) = 1 - \sum_{i=1}^{r} p_i^2 .$$

Example

Let S be the data in Table 8.1. There are four class labels "amphi", "bird", "fish", "mammal" and "reptile". The numbers of examples having these class labels are as follows:

Number of examples with class label "amphi"	= 3
Number of examples with class label "bird"	= 2
Number of examples with class label "fish"	= 2
Number of examples with class label "mammal"	= 2
Number of examples with class label "reptile"	= 1
Total number of examples	= 10

The Gini index of S is given by

$$\begin{aligned}\text{Gini}(S) &= 1 - \sum_{i=1}^{r} p_i^2 \\ &= 1 - (3/10)^2 - (2/10)^2 - (2/10)^2 - (2/10)^2 - (1/10)^2 \\ &= 0.78\end{aligned}$$

Gini split index

- Let S be a set of examples, A be a feature (or, an attribute),
- Sv be the subset of S with A=v, and Values(A)be the set of all possible values of A.
- Then the Gini split index of A relative to S, denoted by $\text{Gini}_{\text{split}(S,A)}$), :

$$\text{Gini}_{\text{split}}(S, A) = \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \times \text{Gini}(S_v).$$

$Gini_{split}$ (S,Givesbirth)=??

Data

Nam	Features				Class label
	gives birth	aquatic animal	aerial animal	has legs	
human	yes	no	no	yes	mammal
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
bat	yes	no	yes	yes	bird
pigeon	no	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Table 8.1: The vertebrate data set

8.8 Gain ratio

The *gain ratio* is a third feature selection measure in the construction of classification trees.

Let S be a set of examples, A a feature having c different values and let the set of values of A be denoted by Values(A). We defined the information gain of A relative to S, denoted by Gain(S, A), by

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v).$$

We now define the*split information* of A relative to S, dented by SplitInformation(S, A), by

$$\text{SplitInformation}(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where $S_1, \ldots S_c$ are the c subsets of examples resulting from partitioning S into the c values of the attribute A. The *gain ratio* of A relative to S, denoted by GainRatio(S, A), by

$$\text{GainRatio}(S, A) = \frac{\text{Gain(S,A)}}{\text{SplitInformation}(S, A)}.$$

Consider the data S given in Table 8.1. Let A denote the attribute "gives birth". We have have already seen that

$$|S| = 10$$
$$\text{Entropy}\ (S) = 2.2464$$
$$\text{Gain}(S, A) = 0.5709$$

Now we have

$$\begin{aligned}
\text{SplitInformation}(S, A) &= -\frac{|S_{\text{yes}}|}{|S|}\log_2\frac{|S_{\text{yes}}|}{|S|} - \frac{|S_{\text{no}}|}{|S|}\log_2\frac{|S_{\text{no}}|}{|S|} \\
&= -\frac{4}{10}\times\log_2\frac{4}{10} - \frac{6}{10}\times\log_2\frac{6}{10} \\
&= 0.9710 \\
\text{GainRatio} &= \frac{0.5709}{0.9710} \\
&= 0.5880
\end{aligned}$$

What is information Gain?

- Calculation of Gain (S, outlook)
- The values of the attribute “outlook” are “sunny”, “ overcast” and “rain”.
- We have to calculate Entropy (S_v) for v = sunny, v = overcast and v = rain.

Day	outlook	temperature	humidity	wind	playtennis
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

Training examples for the target concept “PlayTennis”

What is information Gain?

$$\text{Entropy}\,(S_{\text{sunny}}) = -(3/5) \times \log_2 (3/5) - (2/5) \times \log_2 (2/5)$$
$$= 0.9710$$

$$\text{Entropy}\,(S_{\text{overcast}}) = -(4/4) \times \log_2 (4/4)$$
$$= 0$$

$$\text{Entropy}\,(S_{\text{rain}}) = -(3/5) \times \log_2 (3/5) - (2/5) \times \log_2 (2/5)$$
$$= 0.9710$$

$$\text{Gain}\,(S, \text{outlook}) = \text{Entropy}\,(S) - \frac{|S_{\text{sunny}}|}{|S|} \times \text{Entropy}\,(S_{\text{sunny}}) - \frac{|S_{\text{overcast}}|}{|S|} \times \text{Entropy}\,(S_{\text{overcast}}) - \frac{|S_{\text{rain}}|}{|S|} \times \text{Entropy}\,(S_{\text{rain}})$$

$$= 0.9405 - (5/14) \times 0.9710 - (4/14) \times 0 - (5/14) \times 0.9710$$

$$= 0.2469$$

ID3 Algorithm

- ID3 stands for Iterative Dichotomiser 3
- Algorithm used to generate a decision tree.
- The ID3 algorithm was invented by Ross Quinlan.
- The ID3 follows the Occam's razor principle.
  - A simple model would generalize better than a complex model. This principle is known as Occam's razor, which states that simpler explanations are more plausible and any unnecessary complexity should be shaved off
- Attempts to create the smallest possible decision tree.

▷ Assumptions

- The algorithm uses information gain to select the most useful attribute for classification.
- We assume that there are only two class labels,namely,“+”and“–”.

The examples with class labels “+” are called positive examples and others negative examples.

ID3 Algorithm

The following notations are used in the algorithm:

S	The set of examples
C	The set of class labels
F	The set of features
A	An arbitrary feature (attribute)
Values(A)	The set of values of the feature A
v	An arbitrary value of A
S_v	The set of examples with $A = v$
Root	The root node of a tree

Algorithm ID3(S, F, C)

1. Create a root node for the tree.
2. **if** (all examples in S are positive) **then**
3. **return** single node tree Root with label "+"
4. **end if**
5. **if** (all examples are negative) **then**
6. **return** single node tree Root with label "–"
7. **end if**
8. **if** (number of features is 0) **then**
9. **return** single node tree Root with label equal to the most common class label.
10. **else**
11. Let A be the feature in F with the highest information gain.
12. Assign A to the Root node in decision tree.
13. **for all** (values v of A) **do**
14. Add a new tree branch below Root corresponding to v.
15. **if** (S_v is empty) **then**
16. Below this branch add a leaf node with label equal to the most common class label in the set S.
17. **else**
18. Below this branch add the subtree formed by applying the same algorithm ID3 with the values ID3($S_v, C, F - \{A\}$).
19. **end if**
20. **end for**
21. **end if**

ID3 Algorithm Problem 1

Use ID3 algorithm to construct a decision tree for the data in Table 8.9.

Day	outlook	temperature	humidity	wind	playtennis
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

Table 8.9: Training examples for the target concept "PlayTennis"

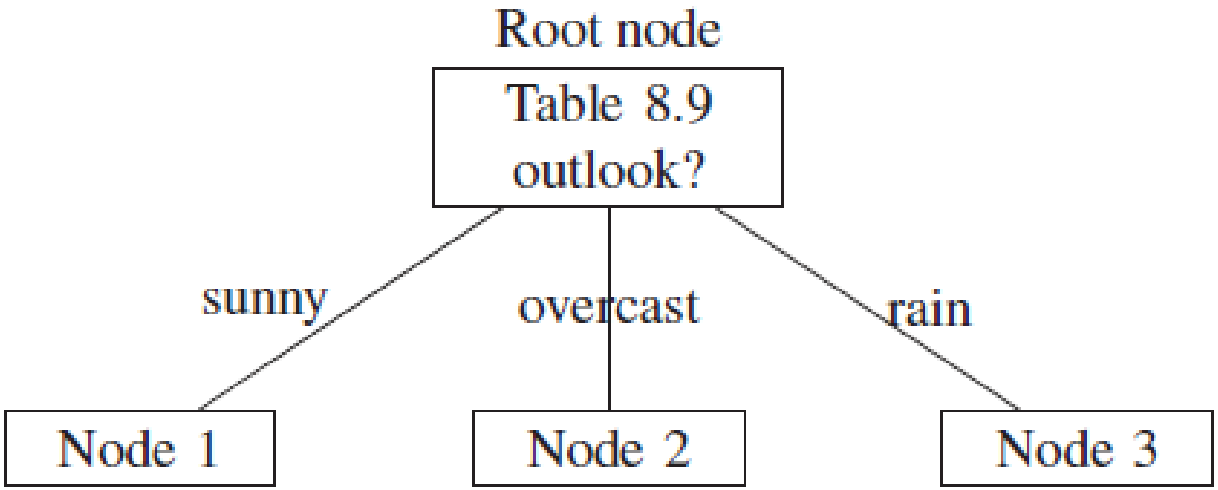
ID3 Algorithm Solution 1

- **Step 1:** We first create a root node for the tree
- **Step 2:** Note that not all examples are positive (class label “yes”) and not all examples are negative (class label “no”). Also the number of features is not zero.
- **Step 3:** We have to decide which feature is to be placed at the root node.
  - For this, we have to calculate the information gains corresponding to each of the four features.

ID3 Algorithm Solution 1

- Step 4: We find the highest information gain whic is the maximum among Gain(S, outlook), Gain(S, temperature), Gain(S, humidity) and Gain(S, wind).
- highest information gain = max{0.2469, 0.0293, 0.151, 0.048}
  - = 0.2469
- This corresponds to the feature “outlook”. Therefore, we place “outlook” at the root node.

ID3 Algorithm Solution 1



ID3 Algorithm Solution 1

▷ Step 5 :

$= S_{\text{outlook=sunny}}$. We have $\left|S^{(1)}\right| = 5$. The examples in $S^{(1)}$ are shown in

Day	outlook	temperature	humidity	wind	playtenn
D1	sunny	hot	high	weak	no
D2	sunny	hot	high	strong	no
D8	sunny	mild	high	weak	no
D9	sunny	cool	normal	weak	yes
D11	sunny	mild	normal	strong	yes

Table 8.10: Training examples with outlook = "sunny"

ID3 Algorithm Solution 1

$$\text{Gain}(S^{(1)}, \text{temp}) = \text{Entropy}(S^{(1)}) - \frac{\left|S^{(1)}_{\text{temp}=\text{hot}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}(S^{(1)}_{\text{temp}=\text{hot}})$$
$$- \frac{\left|S^{(1)}_{\text{temp}=\text{mild}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}(S^{(1)}_{\text{temp}=\text{mild}})$$
$$- \frac{\left|S^{(1)}_{\text{temp}=\text{cool}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}(S^{(1)}_{\text{temp}=\text{cool}})$$
$$= \left[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)\right]$$
$$- (2/5) \times \left[-(2/2)\log(2/2))\right]$$
$$- (2/5) \times \left[-(1/2)\log(1/2) - (1/2)\log_2(1/2)\right]$$
$$- (1/5) \times \left[-(1/1)\log(1/1)\right]$$
$$= 0.5709$$

ID3 Algorithm Solution 1

$$\text{Gain}(S^{(1)}, \text{hum}) = \text{Entropy}(S^{(1)}) - \frac{\left|S^{(1)}_{\text{hum}=\text{high}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}\left(S^{(1)}_{\text{hum}=\text{high}}\right) - \frac{\left|S^{(1)}_{\text{hum}=\text{normal}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}\left(S^{(1)}_{\text{hum}=\text{normal}}\right)$$

$$= \left[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)\right] - (3/5) \times \left[-(3/3)\log(3/3))\right] - (2/5) \times \left[-(2/2)\log(2/2)\right]$$

$$= 0.9709$$

ID3 Algorithm Solution 1

$$\text{Gain}(S^{(1)}, \text{wind}) = \text{Entropy}(S^{(1)}) - \frac{\left|S^{(1)}_{\text{wind}=\text{weak}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}(S^{(1)}_{\text{wind}=\text{weak}})$$

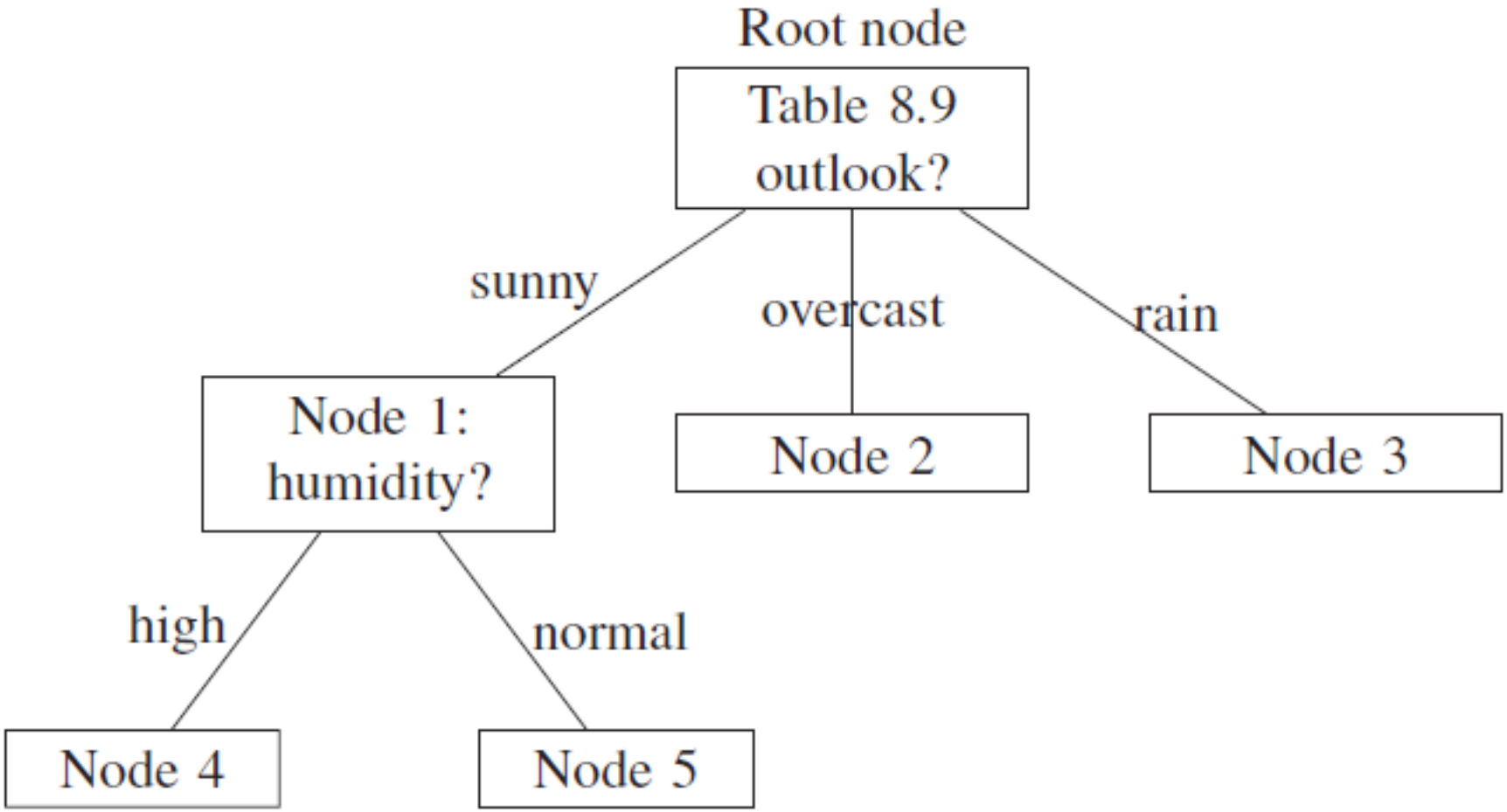
$$- \frac{\left|S^{(1)}_{\text{wind}=\text{strong}}\right|}{\left|S^{(1)}\right|} \times \text{Entropy}(S^{(1)}_{\text{wind}=\text{strong}})$$

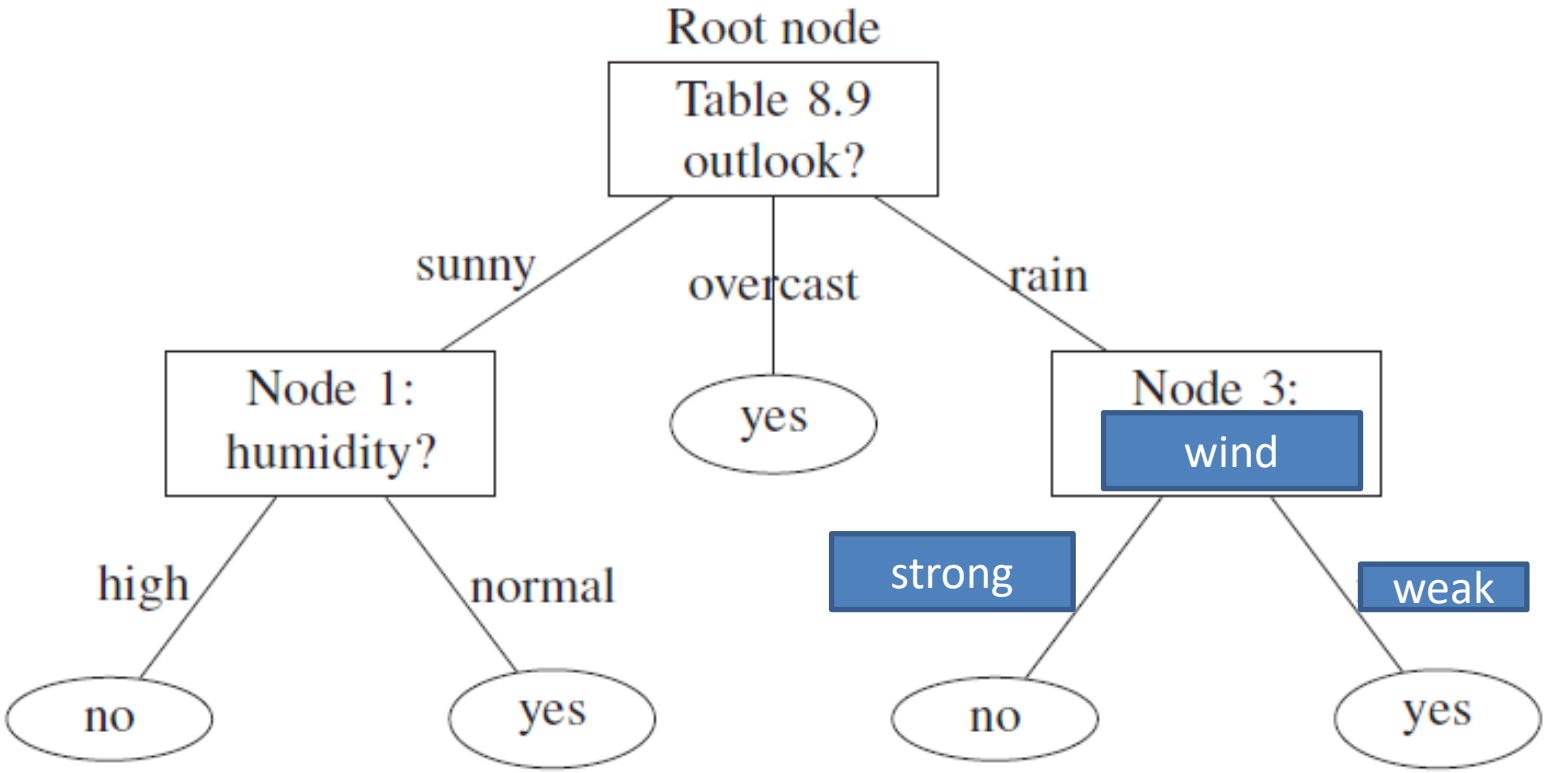
$$= \left[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)\right]$$

$$- (3/5) \times \left[-(2/3)\log(2/3) - (1/3)\log_2(1/3))\right]$$

$$- (2/5) \times \left[-(1/2)\log(1/2) - (1/2)\log(1/2)\right]$$

$$= 0.0110$$


Root node
Table 8.9
outlook?
sunny
overcast
rain
Node 1:
humidity?
Node 2
Node 3
high
normal
Node 4
Node 5


Root node
Table 8.9
outlook?
sunny
overcast
rain
Node 1:
humidity?
yes
Node 3:
wind
high
normal
strong
weak
no
yes
no
yes

Issues in decision Tree learning

- 1. Avoiding overfitting of data
  - When we construct a decision tree, the various branches are grown (that is, sub-branches are constructed) just deeply enough to perfectly classify the training examples.
  - This leads to difficulties when there is noise in the data or when the number of training examples are too small.
  - In these cases the algorithm can produce trees that overfit the training examples.

Approaches to avoiding overfitting:

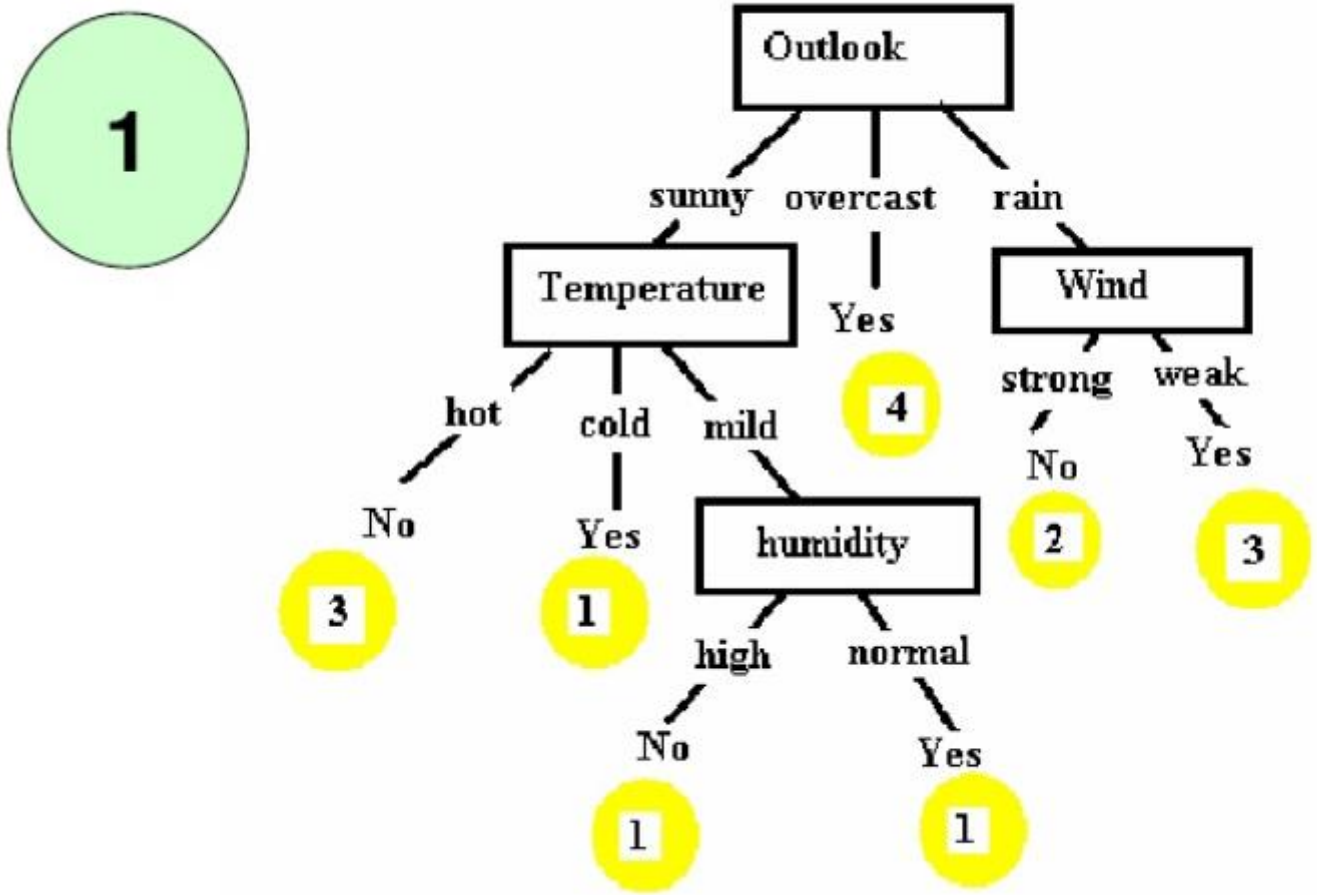
- The main approach to avoid overfitting is pruning.
  - Pruning is a technique that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.
  - Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.

Reduced error pruning

- In reduced error pruning
  - We consider each of the decision tress to be a candidate for pruning.
    - Pruning a decision node consists of removing the sub tree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated to that node.
    - Nodes are removed only if the resulting pruned tree performs no worse than the original over validation set.
    - Nodes are pruned iteratively, always choosing the node whose removal most increases the accuracy over the validation set.
    - Pruning of nodes is continued until further pruning decreases the accuracy over the validation set.

Rule post-pruning

1. Create the decision tree from the training set
2. Convert the tree into an equivalent set of rules
   - Each path corresponds to a rule
   - Each node along a path corresponds to a pre-condition
   - Each leaf classification to the post-condition
3. Prune (generalize) each rule by removing those preconditions whose removal improves accuracy …
   - … over validation set
   - … over training with a pessimistic, statistically inspired, measure
4. Sort the rules in estimated order of accuracy, and consider them in sequence when classifying new instances


1
Outlook
sunny
overcast
rain
Temperature
Yes
4
Wind
hot
cold
mild
strong
weak
No
Yes
No
2
3
3
Yes
1
humidity
high
normal
No
Yes
1
1

1: IF (Outlook = sunny and Temperature = Hot) **THEN** PlayTennis = No

2: IF (Outlook = sunny and Temperature = Cold) **THEN** PlayTennis = Yes

3: IF (Outlook = sunny and Temperature = Mild and Humidity=High) **THEN** PlayTennis = No

4: IF (Outlook = sunny and Temperature = Mild and Humidity=Normal) **THEN** PlayTennis = Yes

5: IF (Outlook = overcast) **THEN** PlayTennis = Yes

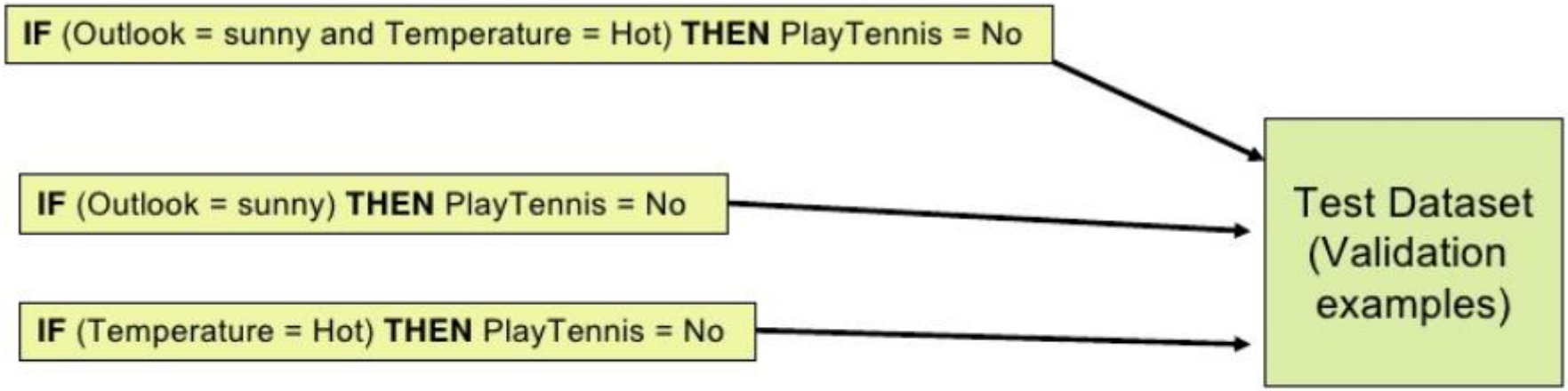
6: IF (Outlook = rain and Wind = Strong) **THEN** PlayTennis = No

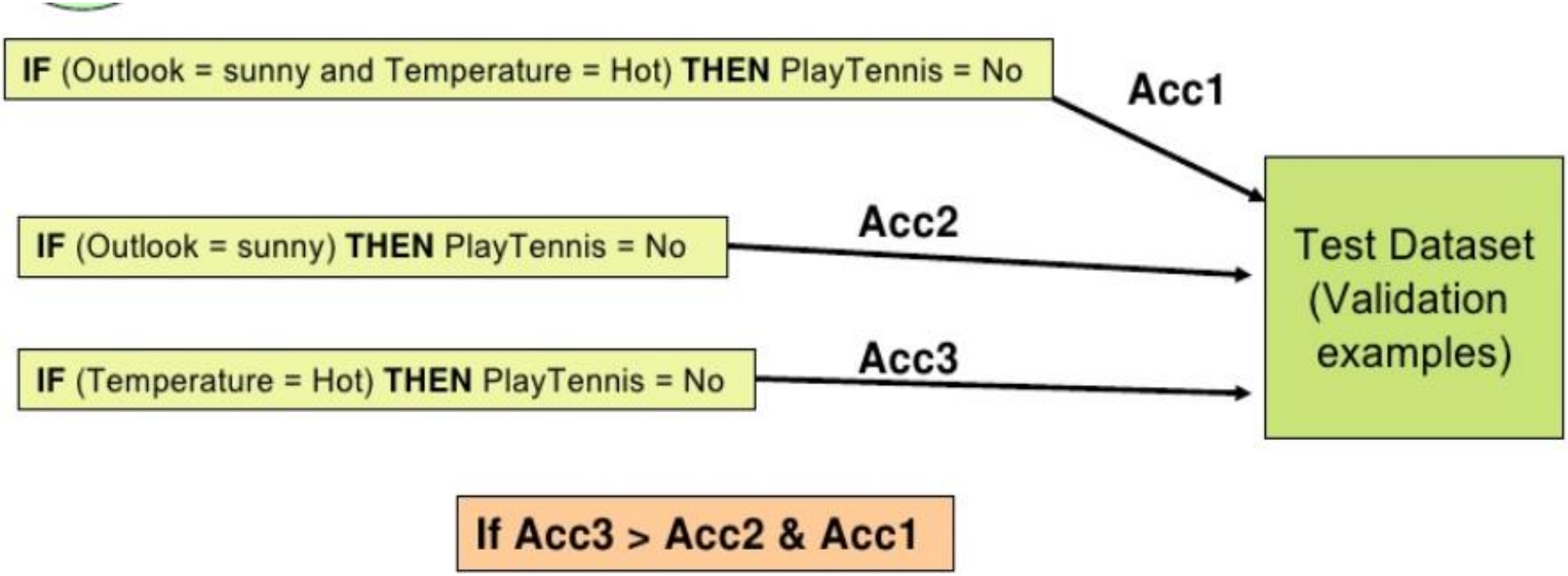
7: IF (Outlook = rain and Wind = Weak) **THEN** PlayTennis = Yes

3

3. Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.

1: IF (Outlook = sunny and Temperature = Hot) **THEN** PlayTennis = No




IF (Outlook = sunny and Temperature = Hot) THEN PlayTennis = No
Acc1
IF (Outlook = sunny) THEN PlayTennis = No
Acc2
IF (Temperature = Hot) THEN PlayTennis = No
Acc3
Test Dataset (Validation examples)
If Acc3 > Acc2 & Acc1

4

4. Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

Problem of missing attributes values

Methods to handle the problem of missing attributes values

- Deleting cases with missing attribute values.
- Replacing a missing attribute value by the most common value of that attribute.
- Assigning all possible values to the missing attribute value .
- Replacing a missing attribute value by the mean for numerical attributes .
- Assigning to a missing attribute value the corresponding value taken from the closest *t* cases, or replacing a missing attribute value by a new value.

Case	Temperature	Headache	Nausea	Decision (Flue)
1	high	?	no	yes
2	very high	yes	no	yes
3	?	no	no	no
4	high	yes	yes	yes
5	high	?	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	?	yes	?	yes

Table 8.15: A dataset with missing attribute values

- Extensions :
  - Continuous valued attributes
  - Alternative measures for selecting attributes
  - Handling training examples with missing attribute values
  - Handling attributes with different costs
  - Improving computational efficiency

Regression Trees

- A regression problem is the problem of determining a relation between one or more independent variables and an output variable which is a real continuous variable
- Then using the relation to predict the values of the dependent variables.
- Regression problems are in general related to prediction of numerical values of variables.
- Trees can also be used to make such predictions.
- A tree used for making predictions of numerical variables is called a prediction tree or a regression tree

Using the data in Table 8.11, construct a tree to predict the values of y.

x_1	1	3	4	6	10	15	2	7	16	0
x_2	12	23	21	10	27	23	35	12	27	17
y	10.1	15.3	11.5	13.9	17.8	23.1	12.7	43.0	17.6	14.9

construct a raw decision tree (a tree constructed without using any standard algorithm) to predict the value of y corresponding to any untabulated values of x1 and x2.

- Step 1. We arbitrarily split the values of x1 into two sets:
  - One set defined by x1<6 and the other set defined by x1≥6. This splits the data into two parts:

x_1	1	3	4	2	0
x_2	12	23	21	35	17
y	10.1	15.3	11.5	12.7	14.9

Table 8.12: Data for regression tree

x_1	6	10	15	7	16
x_2	10	27	23	12	27
y	13.9	17.8	23.1	43.0	17.6

Table 8.13: Data for regression tree

x_1	1	0
x_2	12	17
y	10.1	14.9

(a)

x_1	3	4	2
x_2	23	21	35
y	15.3	11.5	12.7

(b)

x_1	6	7
x_2	10	12
y	13.9	43.0

(c)

x_1	10	15	16
x_2	27	23	27
y	17.8	23.1	17.6

(d)

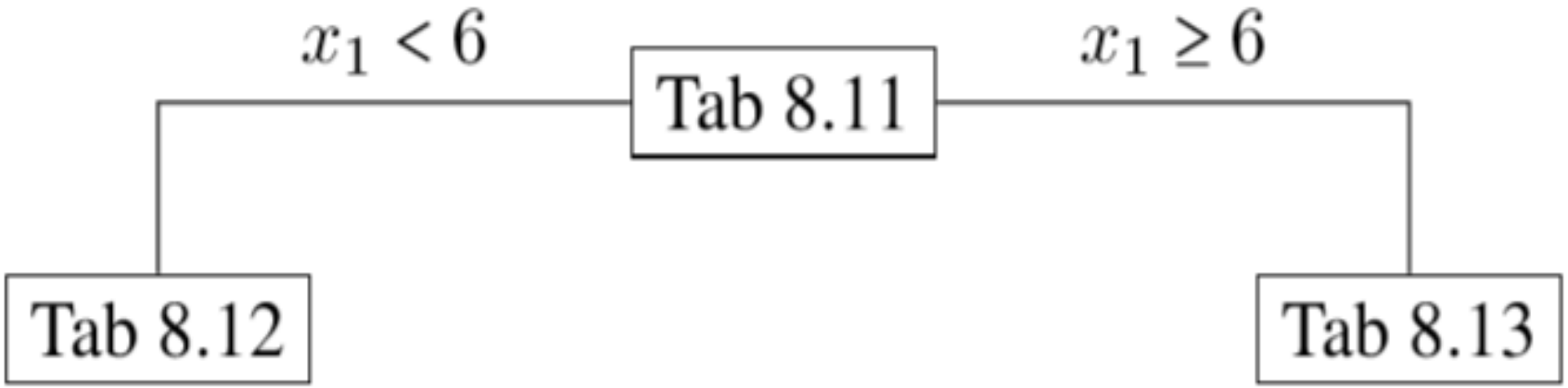


Figure 8.11: Part of a regression tree for Table 8.11

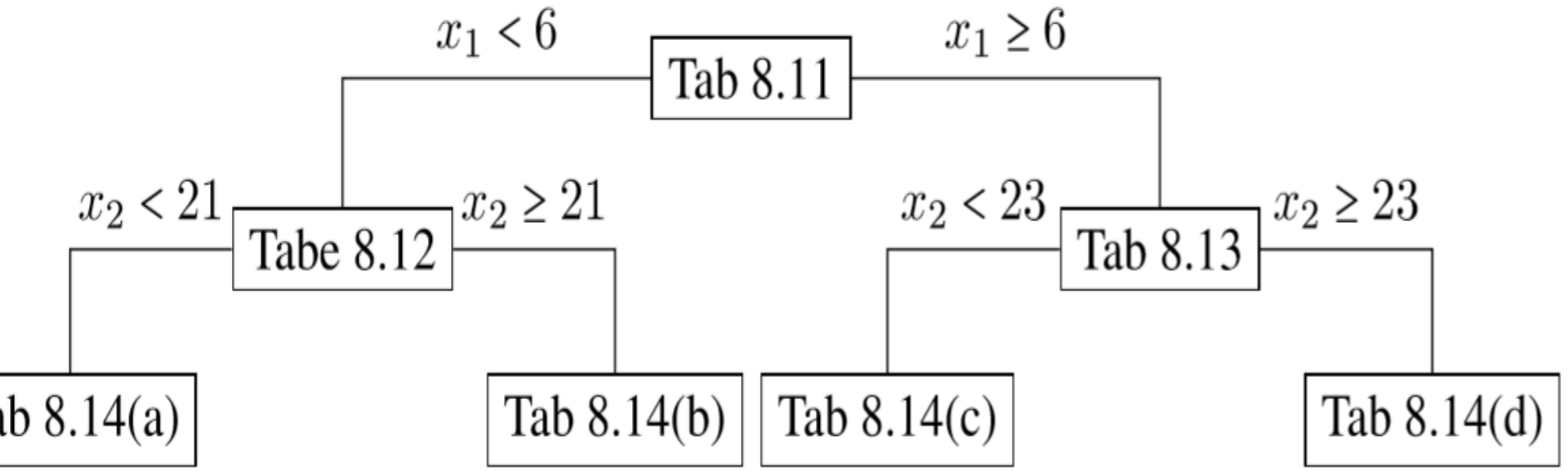


Figure 8.12: Part of regression tree for Table 8.11

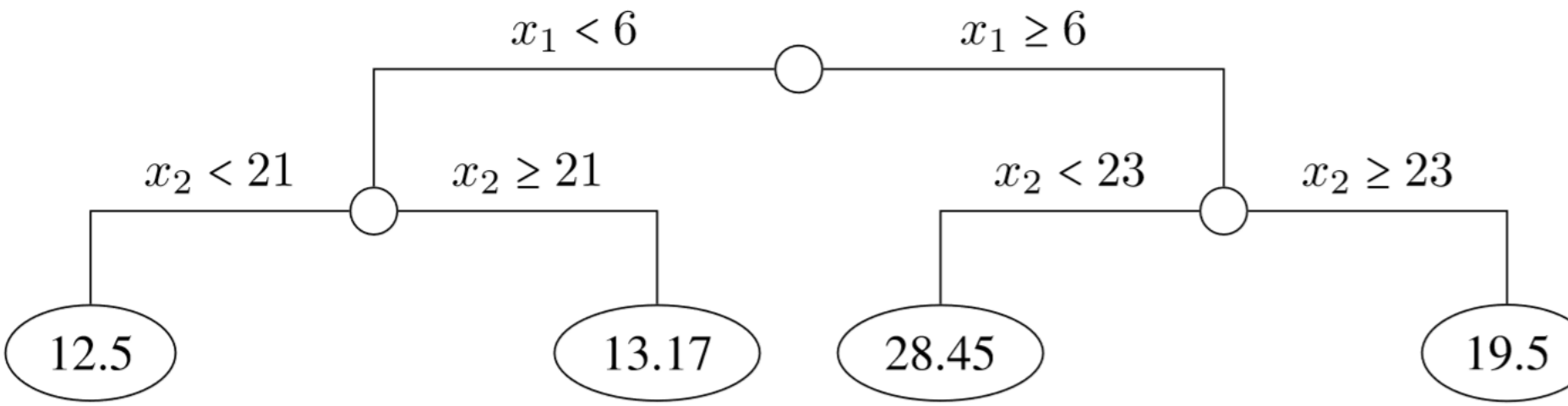


Figure 8.13: A regression tree for Table 8.11

An algorithm for constructing regression trees

▷ Starting with a learning sample, three elements are necessary to determine a regression tree:

1. A way to select a split at every intermediate node
2. A rule for determining when a node is terminal.
3. A rule for assigning a value for the output variable to every terminal node.

tations

$x_1, x_2, \ldots, x_n$	:	The input variables
N	:	Number of samples in the data set
$y_1, y_2, \ldots, y_N$	:	The values of the output variables
T	:	A tree
c	:	A leaf of T
n_c	:	Number of data elements in the leaf c
C	:	The set of indices of data elements which are in the leaf c
m_c	:	The mean of the values of y which are in the leaf c
S_T	:	Sum of squares of errors in T

$$m_c = \frac{1}{n_c} \sum_{i \in C} y_i$$

$$S_T = \sum_{c \in \text{leaves}(T)} \sum_{i \in C} (y_i - m_c)^2$$

Algorithm

1. Start with a single node containing all data points. Calculate mc and S_T.
2. If all the points in the node have the same value for all the independent variables, stop.
3. Otherwise,search over all binary splits of all variables for the one which will reduce S_T as much as possible.

(a) If the largest decrease in S_T would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop and if c is a node where we have stopped, then assign the value mc to the node.

(b) Otherwise, take that split, creating two new nodes.

Step 4. In each new node, go back to Step 1.

Consider the data given in Table 8.11.

1. Computation of S_T for the entire data set. Initially, there is only one node. So, we have:

$$
\begin{aligned}
m_c &= \frac{1}{n_c} \sum_{c \in C} y_i \\
&= \frac{1}{10}(10.1 + 15.3 + \cdots + 14.9) \\
&= 17.99 \\
S_T &= \sum_{c \in \text{leaves}(T)} \sum_{i \in C} (y_i - m_c)^2 \\
&= (10.1 - 17.99)^2 + (15.3 - 17.99)^2 + \cdots + (14.9 - 17.99)^2 \\
&= 817.669
\end{aligned}
$$

2. We have to search every distinct value of x1 and x2 to find the predictor variable and split value which will reduce S_T as much as possible.

3. Let us consider the value 6 of x_1. This splits the data set into two parts c_1 and c_2. Let c_1 be the part defined by $x_1 < 6$ and c_2 the part defined by $x_1 \geq 6$. S_1 is given in Table 8.12 and S_2 by Table 8.13.Now

$$\text{leaves}(T) = \{c_1, c_2\}.$$

Let T_1 be the tree corresponding to this partition. Then

$$\begin{aligned} S_{T_1} &= \sum_{c \in \text{leaves}(T_1)} \sum_{i \in C} (y_i - m_c)^2 \\ &= \sum_{i \in C_1} (y_i - m_{c_1})^2 + \sum_{i \in C_2} (y_i - m_{c_2})^2 \\ m_{c_1} &= \frac{1}{n_{c_1}} \sum_{i \in C_1} y_i \\ &= \frac{1}{5}(10.1 + 15.3 + 11.5 + 12.7 + 14.9) \\ &= 12.9 \end{aligned}$$

$$
\begin{aligned}
m_{c_2} &= \frac{1}{n_{c_2}} \sum_{i \in C_2} y_i \\
&= \frac{1}{5}(13.9 + 17.8 + 23.1 + 43.0 + 17.6) \\
&= 23.08 \\
S_{T_1} &= [(10.1 - 12.9)^2 + \cdots + (14.9 - 12.9)^2] + \\
&\quad [(13.9 - 23.08)^2 + \cdots + (17.6 - 23.08)^2] \\
&= 558.588
\end{aligned}
$$

The reduction in sum of squares of errors is

$$S_T - S_{T_1} = 817.669 - 558.588 = 259.081.$$

4. In this way, we have compute the reduction in the sum of squares of errors corresponding to all other values of x1 and each of the values of x2 .
   - choose the one for which the reduction is maximum.
5. The process has be continued

CARTalgorithm

- The CART,or Classification And Regression Trees methodology,was introduced in 1984 by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone as an umbrella term to refer to the following types of decision trees:

- Classification trees where the target variable is categorical and the tree is used to identify the “class” within which a target variable would likely fall into.
- Regression trees where the target variable is continuous and tree is used to predict it’s value.
- The main elements of CART are:
  - Rules for splitting data at a node based on the value of one variable.(Gini Indicies)
  - Stopping rules for deciding when a branch is terminal and can be split no more.
  - A prediction for the target variable in each terminal node.

Day	outlook	temperature	humidity	wind	playtennis
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

Regression

- Regression is a supervised learning problem where there is an input x an output y and the task is to learn the mapping from the input to the output.
- we assume a model, that is, a relation between x and y containing a set of parameters, say, θ in the following form: $y=g(x,\theta)$
- g $(x,\theta)$is the regression function.

- Definition:
  - A regression problem is the problem of determining a relation between one or more independent variables and an output variable which is a real continuous variable, given a set of observed values of the set of independent variables and the corresponding values of the output variable.

What is Regression Analysis?

- Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor).
- This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.
- For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.

1. Predict a system that can predict the price of a used car.
   - Inputs are the car attributes : brand, year, engine capacity, mileage, and other information:
   - The output is the price of the car.
2. Consider the navigation of a mobile robot, say an autonomous car.
   - The output is the angle by which the steering wheel should be turned at each time, to advance without hitting obstacles and deviating from the route.
   - Inputs are provided by sensors on the car like a video camera, GPS, and so forth.

Different Regression models

1. Simple linear regression :
   - Assume that there is only one independent variable x. If the relation between x and y is modeled by the relation
     - **y=a+bx (best fit straight line** (also known as regression line).
   - then we have a simple linear regression.
2. Multiple regression :
   - Let there be more than one independent variable, say x1, x2, ..., xn, and let the relation between y and the independent variables be modeled as
     - **$y=\alpha 0+\alpha 1x1+\cdots+\alpha n\ xn$**
   - then it is case of multiple linear regression or multiple regression.

- Polynomial regression :
  - Let there be only one variable x and let the relation between x y be modeled as
  -

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

  - for some positive integer n>1, then we have polynomial regression

Simple linear regression

- Let x be the independent predictor variable and y the dependent variable. Assume that we have a set of observed values of x and y:
- A simple linear regression model defines the relationship between x and y using a line defined by an equation in the following form:
  - $y=\alpha+\beta x$
- In order to determine the optimal estimates of α and β, an estimation method known as Ordinary Least Squares (OLS) is used.

OLS method

- In the OLS method, the values of y-intercept(α) and slope (β) are chosen such that they minimize the sum of the squared errors;
- that is, the sum of the squares of the vertical distance between the predicted y-value and the actual y-value
- Let ˆ yi be the predicted value of yi.
- Then the sum of squares of errors is given by:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} [y_i - (\alpha + \beta x_i)]^2$$

x	x_1	x_2	$\cdots$	x_n
y	y_1	y_2	$\cdots$	y_n

Table 7.1: Data set for simple linear regression

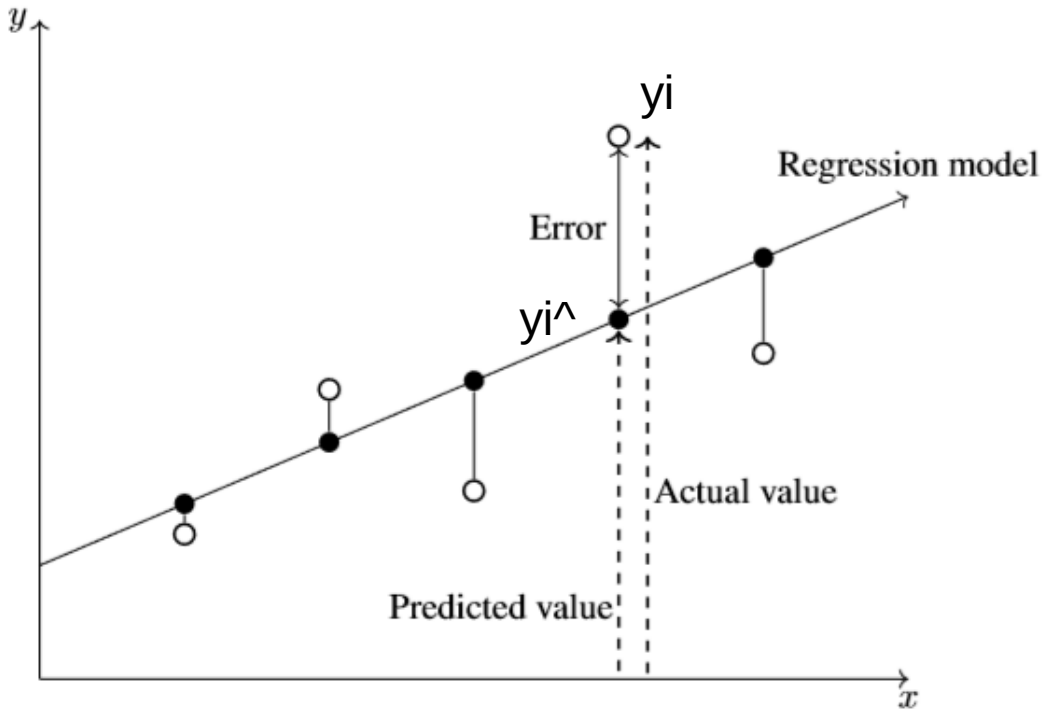


Figure 7.1: Errors in observed values

- So we are required to
  - find the values of α and β such that E is minimum. Using methods of calculus,
  - we can show that the values of a and b, which are respectively the values of α and β for which E is minimum,can be solved using..

$$\sum_{i=1}^{n} y_i = na + b \sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2$$

Formulas to find a and b

Recall that the means of x and y are given by

$$\bar{x} = \frac{1}{n}\sum x_i$$

$$\bar{y} = \frac{1}{n}\sum y_i$$

and also that the variance of x is given by

$$\text{Var}\,(x) = \frac{1}{n-1}\sum (x_i - \bar{x_i})^2.$$

The *covariance of* x *and* y, denoted by $\text{Cov}(x, y)$ is defined as

$$\text{Cov}(x, y) = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y})$$

It can be shown that the values of a and b can be computed using the following formulas:

$$b = \frac{\text{Cov}(x, y)}{\text{Var}(x)}$$

$$a = \bar{y} - b\bar{x}$$

Remarks

Example

Obtain a linear regression for the data in Table 7.2 assuming that y is the independent variable

x	1.0	2.0	3.0	4.0	5.0
y	1.00	2.00	1.30	3.75	2.25

Linear Regression Problem1

Question: Find linear regression equation for the following two sets of data:

x	2	4	6	8
y	3	7	5	10

Linear Regression Solution

x	y	x^2	
2	3	4	
4	7	16	
6	5	36	
8	10	64	
$\sum x$ = 20	$\sum y$ = 25	$\sum x^2$ = 120	

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$b = \frac{4\times144 - 20\times25}{4\times120 - 400}$$

b = 0.95

$$a = \frac{\sum y - b(\sum x)}{n}$$

$$a = \frac{26 - 0.95\times20}{4}$$

a = 1.5

Linear regression is given by:

y = a + bx

y = 1.5 + 0.95 x

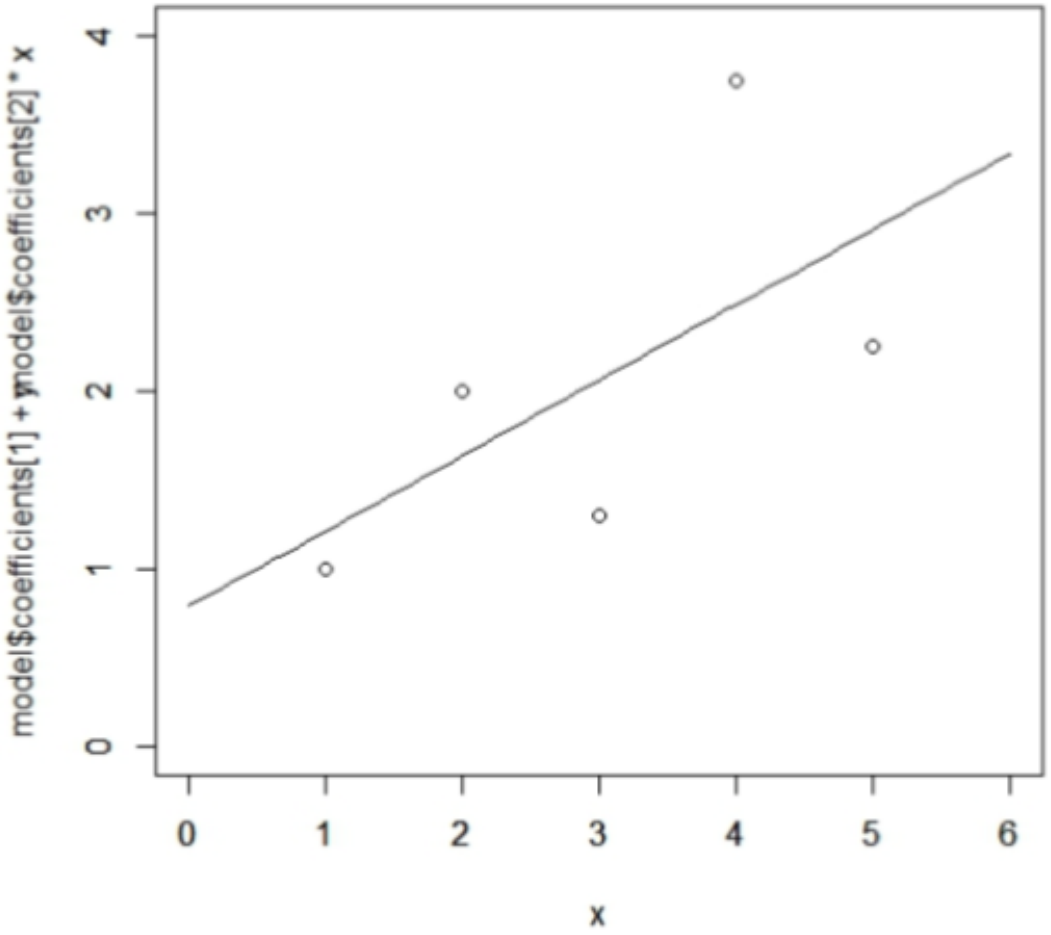


Figure 7.2: Regression model for Table 7

Polynomial regression

- Let x be the independent predictor variable and y the dependent variable.
- A polynomial regression model defines the relationship between x and y by an equation in the following form:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_k x^k.$$

- To determine the optimal values of the parameters α0, α1, . . ., αk the method of ordinary least squares is used. The values of the parameters are those values which minimizes the sum of squares:

$$E = \sum_{i=1}^{n} [y_i - (\alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \cdots + \alpha_k x_i^k)]^2.$$

$$
\begin{aligned}
\sum y_i &= \alpha_0 n + \alpha_1 \left(\sum x_i\right) + \cdots + \alpha_k \left(\sum x_i^k\right) \\
\sum y_i x_i &= \alpha_0 \left(\sum x_i\right) + \alpha_1 \left(\sum x_i^2\right) + \cdots + \alpha_k \left(\sum x_i^{k+1}\right) \\
\sum y_i x_i^2 &= \alpha_0 \left(\sum x_i^2\right) + \alpha_1 \left(\sum x_i^3\right) + \cdots + \alpha_k \left(\sum x_i^{k+2}\right) \\
&\vdots \\
\sum y_i x_i^k &= \alpha_0 \left(\sum x_i^k\right) + \alpha_1 \left(\sum x_i^{k+1}\right) + \cdots + \alpha_k \left(\sum x_i^{2k}\right)
\end{aligned}
$$

- solving this system of linear equations, we get the optimal values for the parameters.

Polynomial Regression Problem1

Question: Find a quadratic regression model for the following data:

x	3	4	5	6	7
y	2.5	3.2	3.8	6.5	11.5

Polynomial Regression Solution

- Let the quadratic regression model be

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2.$$

- The values of α_0, α_1 and α_2 which minimises the sum of squares of errors are a0, a1 and a2 which satisfy the following system of equations:

$$\sum y_i = n a_0 + a_1 \left(\sum x_i\right) + a_2 \left(\sum x_i^2\right)$$
$$\sum y_i x_i = a_0 \left(\sum x_i\right) + a_1 \left(\sum x_i^2\right) + a_2 \left(\sum x_i^3\right)$$
$$\sum y_i x_i^2 = a_0 \left(\sum x_i^2\right) + a_1 \left(\sum x_i^3\right) + a_2 \left(\sum x_i^4\right)$$

x	Y	∑x^2	∑y^2
3	2.5		
4	3.2		
5	3.8		
6	6.5		
7	11.5		
25	27.5		

Polynomial Regression Solution

- Using the given data we have

$$27.5 = 5a_0 + 25a_1 + 135a_2$$
$$158.8 = 25a_0 + 135a_1 + 775a_2$$
$$966.2 = 135a_0 + 775a_1 + 4659a_2$$

- Solving this system of equations we get

$$a_0 = 12.4285714$$
$$a_1 = -5.5128571$$
$$a_2 = 0.7642857$$

- The required quadratic polynomial model is

$$y = 12.4285714 - 5.5128571x + 0.7642857x^2.$$

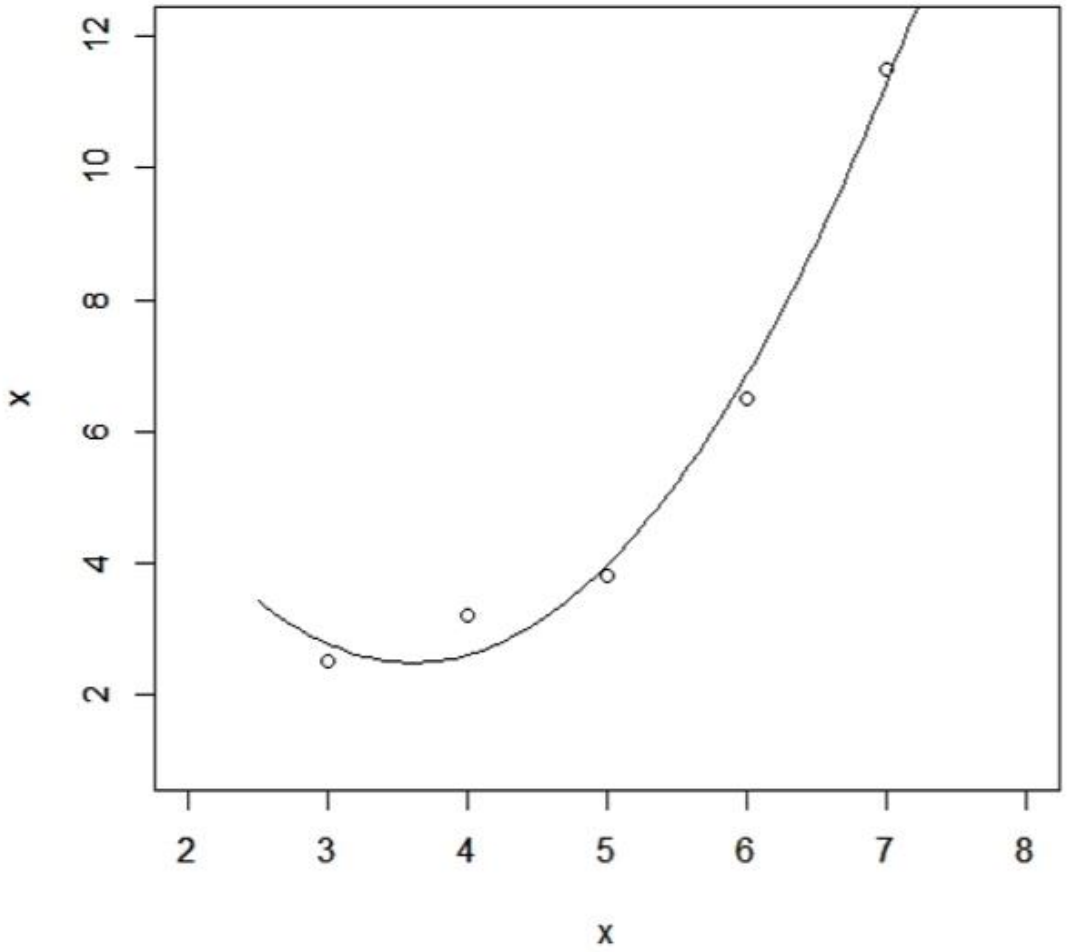


Figure 7.3: Plot of quadratic polynomial model

Multiple linear regression

- The multiple linear regression model defines the relationship between the N independent variables and the dependent variable by an equation of the following form:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N$$

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{N1} \\ 1 & x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & & & & \\ 1 & x_{1n} & x_{2n} & \cdots & x_{Nn} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$$

- Then it can be shown that the regression coefficients are given by

$$B = (X^T X)^{-1} X^T Y$$

Variables (features)	Values (examples)			
	Example 1	Example 2	$\cdots$	Example n
x_1	x_{11}	x_{12}	$\cdots$	x_{1n}
x_2	x_{21}	x_{22}	$\cdots$	x_{2n}
$\cdots$				
x_N	x_{N1}	x_{N2}	$\cdots$	x_{Nn}
y (outcomes)	y_1	y_2	$\cdots$	y_n

Table 7.3: Data for multiple linear regression

Multiple Linear Regression Problem1

Question: Fit a multiple linear regression model to the following data:

x_1	1	1	2	0
x_2	1	2	2	1
y	3.25	6.5	3.5	5.0

Multiple Linear Regression Solution 1

- The multiple linear regression model for this problem has the form n=2 N=4

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

- The computations are

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 0 & 1 \end{bmatrix}, \quad Y = \begin{bmatrix} 3.25 \\ 6.5 \\ 3.5 \\ 5.0 \end{bmatrix}, \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 4 & 4 & 6 \\ 4 & 6 & 7 \\ 6 & 7 & 10 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} \frac{11}{4} & \frac{1}{2} & -2 \\ \frac{1}{2} & 1 & -1 \\ -2 & -1 & 2 \end{bmatrix}$$

$$B = (X^T X)^{-1} X^T Y$$

$$= \begin{bmatrix} 2.0625 \\ -2.3750 \\ 3.2500 \end{bmatrix}$$

- The required model is

$$y = 2.0625 - 2.3750x_1 + 3.2500x_2.$$