CMR
INSTITUTE OF
TECHNOLOGY

USN

Internal Assesment Test –II, December 2021									
Sub:	**MACHINE LEARNING**							**Code:**	**18MCA53**
Date:	17-12-2021	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	MCA

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module		Marks	OBE	
			CO	RBT
Q1	**Part-I** Discuss the concept of Perceptron with a diagram and also write a note on the problems appropriate for Neural Network.	10	CO3	L3
Q2	**or** Write the Gradient decent Algorithm and visualize the Hypothesis space for gradient decent rule.	10	CO3	L2
Q3	**Part-II** What is linearly in separable problem? Design a two-layer Back propagation network for feed forward network.	10	CO3	L3
Q4	**or** Write the Back Propagation algorithm.	10	CO3	L2
Q5	**Part-III** Explain Bayes Theorem and discuss the need for Maximum A Posteriori hypothesis and Maximum Likelihood Hypothesis.	10	CO3	L2
Q 6	**or** Prove that how maximum likelihood (Bayesian Learning) can be used in any learning algorithm that are used to minimize the squared error between actual output hypothesis and predicted output hypothesis.	10	CO4	L3

CMR
INSTITUTE OF
TECHNOLOGY

USN

Internal Assesment Test –II, December 2021									
Sub:	**MACHINE LEARNING**							**Code:**	**18MCA53**
Date:	17-12-2021	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	MCA

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module		Marks	OBE	
			CO	RBT
Q1	**Part-I** Discuss the concept of Perceptron with a diagram and also write a note on the problems appropriate for Neural Network.	10	CO3	L3
Q2	**or** Write the Gradient decent Algorithm and visualize the Hypothesis space for gradient decent rule.	10	CO3	L2
Q3	**Part-II** What is linearly in separable problem? Design a two-layer Back propagation network for feed forward network.	10	CO3	L3
Q4	**or** Write the Back Propagation algorithm.	10	CO3	L2
Q5	**Part-III** Explain Bayes Theorem and discuss the need for Maximum A Posteriori hypothesis and Maximum Likelihood Hypothesis.	10	CO3	L2
Q 6	**or** Prove that how maximum likelihood (Bayesian Learning) can be used in any learning algorithm that are used to minimize the squared error between actual output hypothesis and predicted output hypothesis.	10	CO4	L3

Q7	**Part-IV** Prove that maximum likelihood hypothesis can be used to predict probabilities.	10	CO4	L3
Q 8	**or** Consider a medical diagnosis problem in which there are two alternative hypotheses: 1: the patient has cancer (+) and 2: the patient does not (-). A patient takes a lab test and the result comes back positive. Given: correct positive result is 98% and a correct negative result is 97% also only 0.008 of the entire population has this disease. Determine if the patient having +ve report has cancer or not using MAP hypothesis.	10	CO4	L3
Q 9	**Part-V** Explain Brute Force Bayes concept learning and derive the posterior probability P(D\h).	10	CO3	L2
Q 10	**or** Apply Naïve Bayes classifier classify the new data (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong).	10	CO3	L3

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Q7	**Part-IV** Prove that maximum likelihood hypothesis can be used to predict probabilities.	10	CO4	L3
Q 8	**or** Consider a medical diagnosis problem in which there are two alternative hypotheses: 1: the patient has cancer (+) and 2: the patient does not (-). A patient takes a lab test and the result comes back positive. Given: correct positive result is 98% and a correct negative result is 97% also only 0.008 of the entire population has this disease. Determine if the patient having +ve report has cancer or not using MAP hypothesis.	10	CO4	L3
Q 9	**Part-V** Explain Brute Force Bayes concept learning and derive the posterior probability P(D\h).	10	CO3	L2
Q 10	**or** Apply Naïve Bayes classifier classify the new data (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong).	10	CO3	L3

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Q1. Discuss the concept of Perceptron with a diagram and also write a note on the problems appropriate for Neural Network.

Representational Power of Perceptrons:

- The perceptron can be viewed as representing a hyperplane decision surface in the n- dimensional space of instances (i.e., points)
- The perceptron outputs a 1 for instances lying on one side of the hyperplane and outputs a -1 for instances lying on the other side, as illustrated in below figure

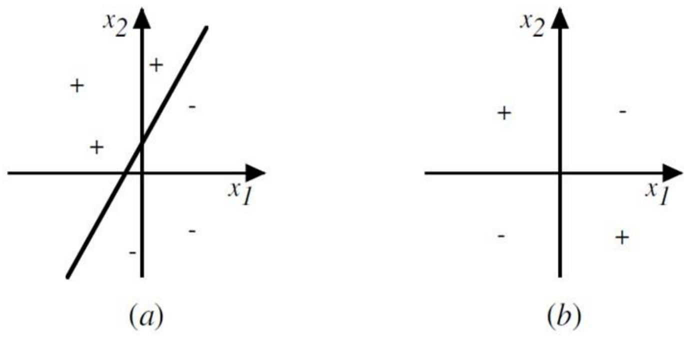
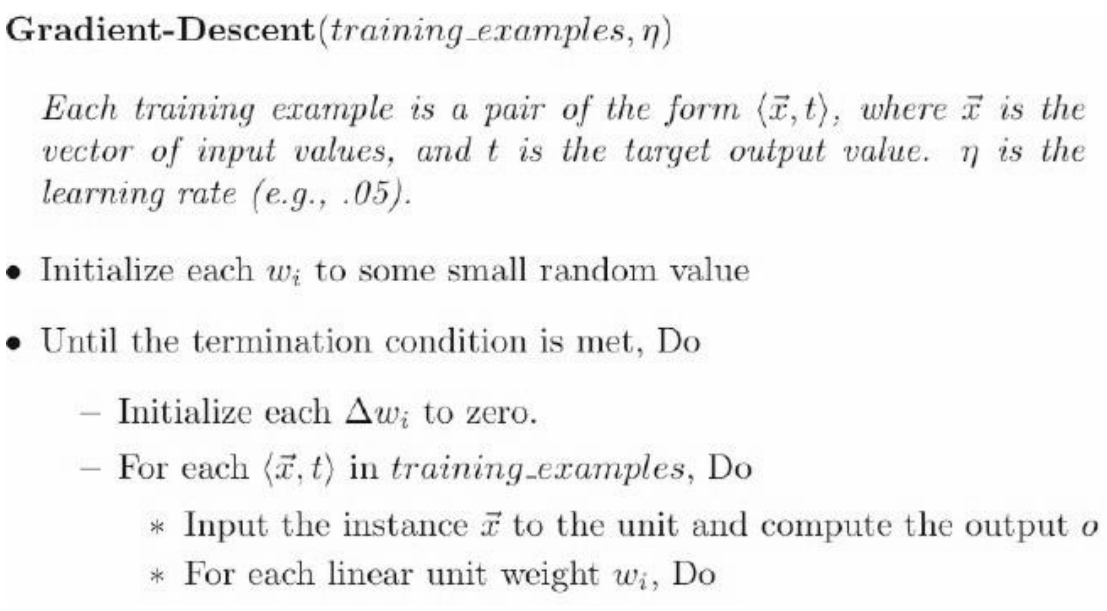


Figure : The decision surface represented by a two-input perceptron. *(a)* A set of training examples and the decision surface of a perceptron that classifies them correctly. *(b)* A set of training examples that is not linearly separable. x_1 and x_2 are the Perceptron inputs. Positive examples are indicated by "+", negative by "-".

Q2

Write the Gradient decent Algorithm and visualize the Hypothesis space for gradient decent rule.

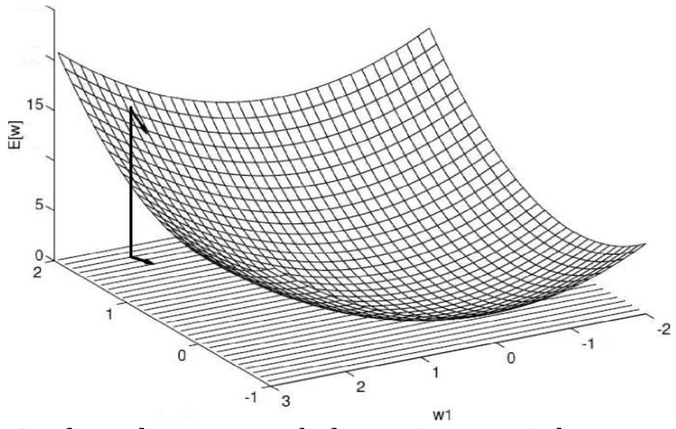


```
Gradient-Descent($training\_examples, \eta$)

    Each training example is a pair of the form $\langle \vec{x}, t \rangle$, where $\vec{x}$ is the
    vector of input values, and $t$ is the target output value. $\eta$ is the
    learning rate (e.g., .05).

• Initialize each $w_i$ to some small random value
• Until the termination condition is met, Do
    – Initialize each $\Delta w_i$ to zero.
    – For each $\langle \vec{x}, t \rangle$ in $training\_examples$, Do
        * Input the instance $\vec{x}$ to the unit and compute the output $o$
        * For each linear unit weight $w_i$, Do
```

$$\Delta w_i := \eta(t - o)x_i$$

$$w_i := w_i + \Delta w_i$$

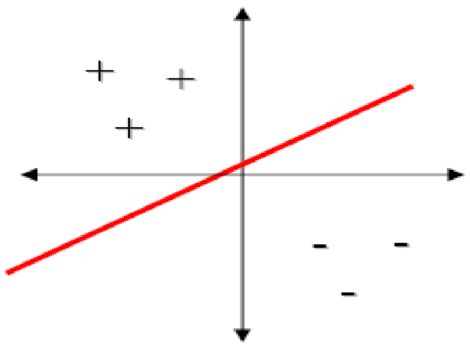
- Perceptron learning converges to a consistent model if D (training set) is linearly separable.
- If the data is not linearly separable than this will not converge.
- If the training examples are not linearly separable, the delta rule converges toward a best-fit approximation to the target concept.
- The key idea behind the *delta rule* is to use *gradient descent* to search the hypothesis space of possible weight vectors to find the weights that best fit the training examples.

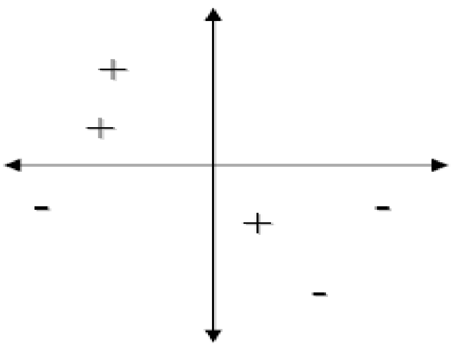


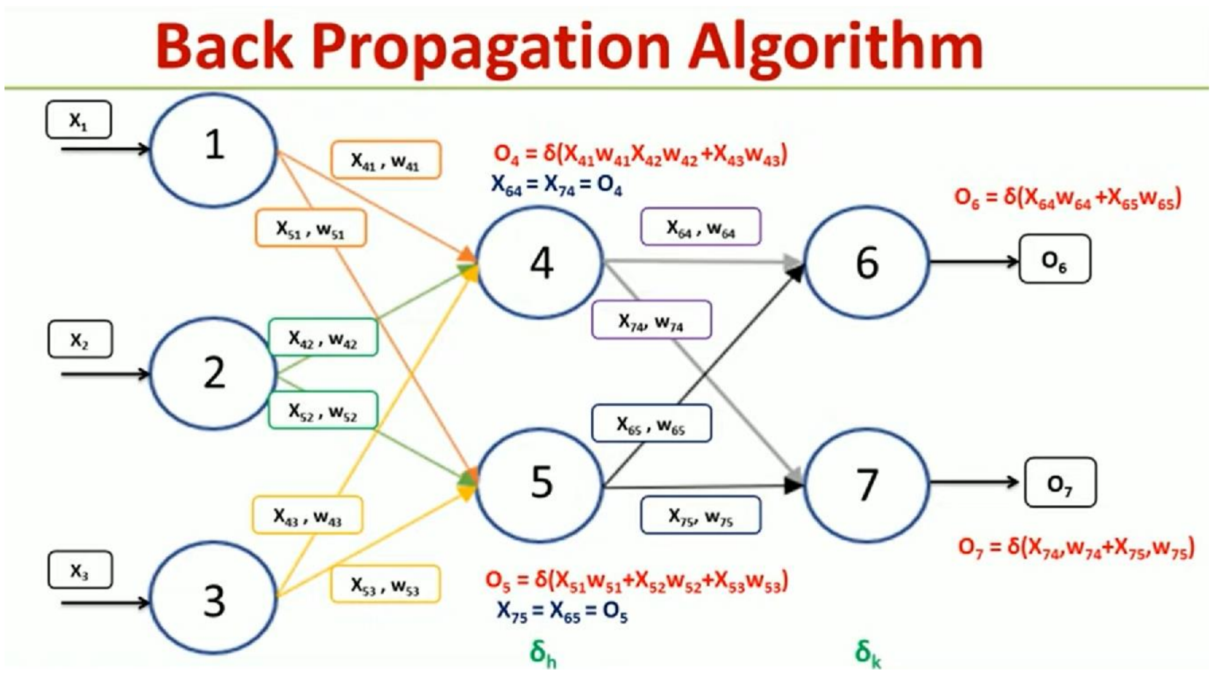
- Gradient descent search determines a weight vector that minimizes E by starting with an arbitrary initial weight vector, then repeatedly modifying it in small steps.
- At each step, the weight vector is altered in the direction that produces the steepest descent along the error surface depicted in above figure. This process continues until the global minimum error is reached.

Q3

What is linearly in separable problem? Design a two-layer Back propagation network for feed forward network.


Linearly separable


Non-linearly separable



Q4

Write the Back Propagation algorithm.

```
Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
Initialize all network weights to small random numbers (e.g., between $-.05$ and $.05$).
Until the termination condition is met, Do
    • For each $\langle \vec{x}, \vec{t} \rangle$ in training_examples, Do
        Propagate the input forward through the network:
        1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit $u$ in
           the network.
        Propagate the errors backward through the network:
        2. For each network output unit $k$, calculate its error term $\delta_k$
               $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$                                  (T4.3)
        3. For each hidden unit $h$, calculate its error term $\delta_h$
               $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$                     (T4.4)
        4. Update each network weight $w_{ji}$
               $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$
           where
               $\Delta w_{ji} = \eta \, \delta_j \, x_{ji}$                                     (T4.5)
```

Q5

Explain Bayes Theorem and discuss the need for Maximum A Posteriori hypothesis and Maximum Likelihood Hypothesis.

Bayes Theorem:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- P(h) prior probability of h, reflects any background knowledge about the chance that h is correct
- P(D) prior probability of D, probability that D will be observed
- P(D|h) probability of observing D given a world in which h holds
- P(h|D) posterior probability of h, reflects confidence that h holds after D has been observed

- We are interested in finding the most probable hypothesis **h ϵ H** given the observed data **D (or at least one of the maximally probable if there** are several).
- Any such maximally probable hypothesis is called a **maximum a posteriori (MAP) hypothesis.**
- **We can determine the MAP hypotheses by using** Bayes theorem to calculate the posterior probability of each candidate hypothesis.
- More precisely, we will say that h_{MAP} **is a MAP hypothesis provided**

$$\begin{aligned} h_{MAP} &\equiv \underset{h \in H}{\text{argmax}}\, P(h|D) \\ &= \underset{h \in H}{\text{argmax}}\, \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h \in H}{\text{argmax}}\, P(D|h)P(h) \end{aligned}$$

- Notice in the final step above we dropped the term **P(D) because it is a constant** independent of **h.**

Maximum Likelihood Hypothesos:

- In some cases, we will assume that every hypothesis in H is equally probable a priori **(P(hi) = P(hj) for all hi and hj in H).**
- $$h_{ML} \equiv \underset{h \in H}{\text{argmax}}\, P(D|h)$$
- Here, P(D\h) is called as the likelihood of the data D given h.
- Any hypothesis that maximizes it id called ML hypothesis.

Q 6

Prove that how maximum likelihood (Bayesian Learning) can be used in any learning algorithm that are used to minimize the squared error between actual output hypothesis and predicted output hypothesis.

A straightforward Bayesian analysis will show that under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a *maximum likelihood (ML) hypothesis*

- Learner L considers an instance space X and a hypothesis space H consisting of some class of real-valued functions defined over X, i.e., (∀ h ∈ H)[h : X → R] and training examples of the form <xi, di>
- The problem faced by L is to learn an unknown target function f : X → R
- A set of m training examples is provided, where the target value of each example is corrupted by random noise drawn according to a Normal probability distribution with zero mean (di = f(xi) + ei)
- Each training example is a pair of the form (xi ,di) where di = f (xi) + ei .
  - – Here f(xi) is the noise-free value of the target function and ei is a random variable representing the noise.
  - – It is assumed that the values of the ei are drawn independently and that they are distributed according to a Normal distribution with zero mean.
- The task of the learner is to *output a maximum likelihood hypothesis* or a *MAP hypothesis assuming all hypotheses are equally probable a priori*.

Using the definition of hML we have

$$h_{ML} = \underset{h \in H}{argmax}\; p(D|h)$$

Assuming training examples are mutually independent given h, we can write P(D|h) as the product of the various (di|h)

$$h_{ML} = \underset{h \in H}{argmax} \prod_{i=1}^{m} p(d_i|h)$$

Given the noise ei obeys a Normal distribution with zero mean and unknown variance σ^2 , each di must also obey a Normal distribution around the true targetvalue f(xi). Because we are writing the expression for P(D|h), we assume h is the correct description of f.

Hence, μ = f(xi) = h(xi)

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - \mu)^2}$$

$$h_{ML} = \underset{h \in H}{argmax} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(x_i))^2}$$

Maximize the less complicated logarithm, which is justified because of the monotonicity of function p

$$h_{ML} = \underset{h \in H}{argmax} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

The first term in this expression is a constant independent of h, and can therefore be discarded, yielding

$$h_{ML} = \underset{h \in H}{argmax} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

Maximizing this negative quantity is equivalent to minimizing the corresponding positive quantity

$$h_{ML} = \underset{h \in H}{argmin} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

Finally, discard constants that are independent of h.

$$h_{ML} = \underset{h \in H}{argmin} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Thus, above equation shows that the maximum likelihood hypothesis hML is the one that minimizes the sum of the squared errors between the observed training values di and the hypothesis predictions h(xi)

Q7

Prove that maximum likelihood hypothesis can be used to predict probabilities.

- Consider the setting in which we wish to learn a nondeterministic (probabilistic) function f : X → {0, 1}, which has two discrete output values.
- We want a function approximator whose output is the probability that f(x) = 1. In other words, learn the target function f ` : X → [0, 1] such that f ` (x) = P(f(x) = 1)

How can we learn f ` using a neural network?

- Use of brute force way would be to first collect the observed frequencies of 1's and 0's for each possible value of x and to then train the neural network to output the target frequency for each x.

What criterion should we optimize in order to find a maximum likelihood hypothesis for f' in this setting?

- First obtain an expression for P(D|h)
- Assume the training data D is of the form D = {(x1, d1) . . . (xm, dm)}, where di is the observed 0 or 1 value for f (xi).
- Both xi and di as random variables, and assuming that each training example is drawn independently, we can write P(D|h) as

$$P(D \mid h) = \prod_{i=1}^{m} P(x_i, d_i \mid h) \qquad \text{equ (1)}$$

Applying the product rule

$$P(D \mid h) = \prod_{i=1}^{m} P(d_i \mid h, x_i) P(x_i) \qquad \text{equ (2)}$$

The probability P(di|h, xi)

$$P(d_i|h, x_i) = \begin{cases} h(x_i) & \text{if } d_i = 1 \\ (1 - h(x_i)) & \text{if } d_i = 0 \end{cases} \qquad \text{equ (3)}$$

Re-express it in a more mathematically manipulable form, as

$$P(d_i|h, x_i) = h(x_i)^{d_i} (1 - h(x_i))^{1-d_i} \qquad \text{equ (4)}$$

Equation (4) to substitute for P(di |h, xi) in Equation (5) to obtain

$$P(D|h) = \prod_{i=1}^{m} h(x_i)^{d_i}(1 - h(x_i))^{1-d_i} P(x_i) \qquad \text{equ (5)}$$

We write an expression for the maximum likelihood hypothesis

$$h_{ML} = \underset{h \in H}{\text{argmax}} \prod_{i=1}^{m} h(x_i)^{d_i}(1 - h(x_i))^{1-d_i} P(x_i)$$

The last term is a constant independent of h, so it can be dropped

$$h_{ML} = \underset{h \in H}{\text{argmax}} \prod_{i=1}^{m} h(x_i)^{d_i}(1 - h(x_i))^{1-d_i} \qquad \text{equ (6)}$$

It easier to work with the log of the likelihood, yielding

$$h_{ML} = \underset{h \in H}{\text{argmax}} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i)) \qquad \text{equ (7)}$$

Equation (7) describes the quantity that must be maximized in order to obtain the maximum likelihood hypothesis in our current problem setting.

Q 8

Consider a medical diagnosis problem in which there are two alternative hypotheses: 1: the patient has cancer (+) and 2: the patient does not (-). A patient takes a lab test and the result comes back positive. Given: correct positive result is 98% and a correct negative result is 97% also only 0.008 of the entire population has this disease. Determine if the patient having +ve report has cancer or not using MAP hypothesis.

P(cancer) = 0.008 P(¬cancer) = 1 – 0.008

= 0.992

P(+|cancer) = 0.98 P(- |cancer) = 1-0.98= 0.02

P(- | ¬cancer) = 0.97 P(+ |¬cancer)=1-0.97= 0.03

Now a new patient, whose test result is positive, Should we diagnose the patient have cancer or not?

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

P(cancer|+) = P(+ |cancer) * P(cancer)

= 0.98 * 0.008

= 0.078

P(¬cancer | +) = P(+ | ¬cancer) * P(¬cancer)

= 0.03 * 0.992

= 0.298

Since,

P(cancer|+) < P(¬cancer | +)

So we can conclude that,

Diagnosis : Not having cancer

Q 9. Explain Brute Force Bayes concept learning and derive the posterior probability P(D\h).

Brute-Force Bayes Concept Learning

Consider the concept learning problem

- Assume the learner considers some finite hypothesis space H defined over the instance space X, in which the task is to learn some target concept c : X → {0,1}.
- Learner is given some sequence of training examples ((x1, d1) . . . (xm, dm)) where xi is some instance from X and where di is the target value of xi (i.e., di = c(xi)).
- The sequence of target values are written as D = (d1 . . . dm).

We can design a straightforward concept learning algorithm to output the maximum a posteriori hypothesis, based on Bayes theorem, as follows:

BRUTE-FORCE MAP LEARNING algorithm:

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{argmax}\ P(h|D)$$

In order specify a learning problem for the BRUTE-FORCE MAP LEARNING algorithm we must specify what values are to be used for P(h) and for P(D|h) ?

Let's choose P(h) and for P(D|h) to be consistent with the following assumptions:

- The training data D is noise free (i.e., di = c(xi))
- The target concept c is contained in the hypothesis space H

Do not have a priori reason to believe that any hypothesis is more probable than any other.

- Assumptions
  - The training data D is noise-free

$$d_i = c(x_i)$$

  - The target concept c is in the hypothesis set H

$$c \in H$$

  - All hypotheses are equally likely

$$P(h) = \frac{1}{|H|}$$

- Choice: Probability of D given h

$$P(D \mid h) = \begin{cases} 1 & \text{if } \forall d \in D, h(d) = c(d) \\ 0 & \text{else} \end{cases}$$

$$P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}$$ Bayes Theorem

$$= \frac{P(D \mid h) \cdot \frac{1}{|H|}}{P(D)}$$ Given our assumptions

If the data is not consistent

$$P(h \mid D) = 0$$

If the data is consistent

$$= \frac{1 \cdot \frac{1}{|H|}}{P(D)} = \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|}$$

$VS_{H,D}$ is the version space

Given: $(\forall i \neq j)(P(h_i \wedge h_j) = 0)$ (the hypotheses are mutually exclusive):

$$P(D) = \sum_{h_i \in H} P(D \mid h_i) P(h_i)$$
$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{1}{|H|}$$
$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|}$$
$$= \frac{|VS_{H,D}|}{|H|}$$

To summarize, Bayes theorem implies that the posterior probability P(h|D) under our assumed P(h) and P(D|h) is

$$P(D|h) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if h is consistent with D} \\ 0 & \text{otherwise} \end{cases}$$

Q 10

Apply Naïve Bayes classifier classify the new data (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong).

Step 1: Convert the data into a frequency table.

Play	Frequency
Yes	9
No	5

Outlook	Yes	No
Sunny	2	3
Overcast	4	0
Rain	3	2

Temperature	Yes	No
Hot	2	2
Mild	4	2
Cool	3	1

Humidity	Yes	No
High	3	4
Normal	6	1

Wind	Yes	No
Strong	3	3
Weak	6	2

Step 2: Create Likelihood table

Play	Frequency	Likelihood
Yes	9	9/14
No	5	5/14

Outlook	Yes	No
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Yes	No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Yes	No

High	3/9	4/5	
Normal	6/9	1/5	

Wind	Yes	No
Strong	3/9	3/5
Weak	6/9	2/5

Step 3:

New instance

X= (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong)

Play Tennis = ?

P(X |Play = Yes) = P(Play = Yes) * P(Outlook = Sunny |Yes) * P(Temperature = Cool|Yes) * P(Humidity = High|Yes) * P(Wind = Strong|Yes)

= 9/14 * 2/9 * 3/9 * 3/9 * 3/9

= 0.0053

P(X|Play = No) = P(Play = No) * P(Outlook = Sunny |No) * P(Temperature = Cool|No) * P(Humidity = High|No) * P(Wind = Strong|No)

= 5/14 * 3/5 * 1/5 * 4/5 * 3/5

= 0.0206

So: 0.0206 > 0.0053

Result : X : PlayTennis = No