Internal Assessment Test 2 Scheme and Solutions– November 2023

Sub:	Artificial Intelligence and Machine Learning – Set 2			Sub Code:	18CS71		Branch:	ISE
Date:	25-11-2023	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	**7 C**	**OBE**

<u>Answer any FIVE FULL Questions</u>		Marks Distribution	Max Marks
1	• Finding the overall probabilities • Applying Naïve Bayes to calculate the conditional probabilities • Predicting the result	2 M 6M 2M	10 M
2	• Finding the overall probabilities • Applying Naïve Bayes to calculate the conditional probabilities • Predicting the result	2 M 6M 2M	10 M
3	• Explaining KNN algorithm for discrete values • Pseudo Code	6 M 4 M	10 M
4.a 4.b	• Explaining locally weighted regression with ex. • Explanation of Q-learning	5 M 5 M	10 M
5.a 5.b	• Bayesian belief networks explanation with example • EM algorithm explanation	5 M 5 M	10 M
6.a 6.b	• Explaining CADET system with example • Explanation of Radial Basis function	6 M 4 M	10 M
7.a 7.b	• Explanation of maximum likelihood hypothesis • Bayes theorem formula with all the terms explanation	8M 2M	10 M

SN	Question	Marks	CO	BT
1	Classify the test data {**Red, SUV, Domestic**} using NAÏVE Bayes classifier for the dataset shown below.	10	CO2	L3

Color	**Type**	**Origin**	**Stolen**
Red	Sports	Domestic	Yes
Red	Sports	Domestic	No
Red	Sports	Domestic	Yes
Yellow	Sports	Domestic	No
Yellow	Sports	Imported	Yes
Yellow	SUV	Imported	No
Yellow	SUV	Imported	Yes
Yellow	SUV	Domestic	No
Red	SUV	Imported	No
Red	Sports	Imported	Yes

Solution:

Our task is to predict the target value ***(yes or no)*** of the target concept ***Stolen*** for this new instance

The probabilities of the different target values can easily be estimated based on their frequencies over the 10 training examples.

- P(Yes) = 5/10 = 0.5
- P(No) = 5/10 = 0.5

For new data {Red, SUV, Domestic} we need to classify the result

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

= argmax P (Vj) * P (Red|Vj) * P (SUV|Vj) * P (Domestic|Vj), Vj ={Yes, No}

Now we need to find the conditional probabilities for the test data w.r.t '**Yes**' as mentioned below.

- P(Red|Vj=Yes) = 3/5 = 0.6
- P(SUV|Vj=Yes) = 1/5 = 0.2
- P(Domestic|Vj=Yes) = 2/5 = 0.4

Now we need to find the conditional probabilities for the test data w.r.t '**No**'as mentioned below.

- P(Red|Vj=No) = 2/5 = 0.4
- P(SUV|Vj=No) = 3/5 = 0.6
- P(Domestic|Vj=No) = 3/5 = 0.6

Finally for the test data we have the formula as below.

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

VNB {Yes} = P (Yes)*P (Red|Yes)*P (SUV|Yes)*P (Domestic|Yes) = 0.5*0.6*0.2*0.4 = 0.024

VNB {No} = P (No)* P (Red|No)*P (SUV|No)*P (Domestic|No) = 0.5*0.4*0.6*0.6 = 0.072

So for new data {Red, SUV, Domestic} the result is No

2	Estimate the conditional probability of each attributes {Color, Legs, Height, Smell} for the species class {M, H} using the data given in the table. Using these probabilities estimate the probabilities values for the new instance **{Color=Green, Legs=2,Height=Tall and Smelly=No}**	10	CO2	L3

SN	Color	Legs	Height	Smell	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

Solution:

Our task is to predict the target value ***(M or H)*** of the target concept ***Species*** for this new instance

The probabilities of the different target values can easily be estimated based on their frequencies over the 10 training examples.

- P(M) = 4/8 = 0.4
- P(H) = 4/8 = 0.4

For new data {Green, 2, Tall, No} we need to classify the result

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

= argmax P (Vj) * P (Green|Vj) * P (2|Vj) * P (Tall|Vj)*P(No|Vj), Vj ={M, H}

Now we need to find the conditional probabilities for the test data w.r.t '**M**' as mentioned below.

- P(Green|Vj=M) = 2/4
- P(2|Vj=M) = 1/4
- P(Tall|Vj=M) = 1/4
- P(No|Vj=M) = 1/4

Now we need to find the conditional probabilities for the test data w.r.t '**H**'as mentioned below.

- P(Green|Vj=H) = 1/4
- P(2|Vj=H) = 4/4
- P(Tall|Vj=H) = 2/4
- P(No|Vj=H) = 3/4

Finally for the test data we have the formula as below.

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

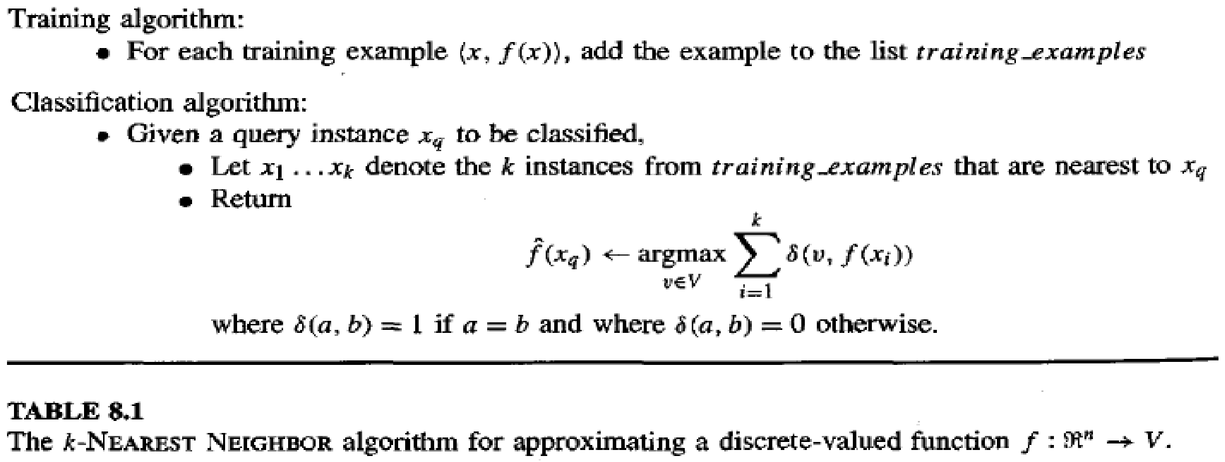
VNB {M} = P (M)*P (Green|M)*P (2|M)*P (Tall|M) *P(No|M) = 2/4*1/4*1/4*1/4

VNB {H} = P (H)*P (Green|H)*P (2|H)*P (Tall|H) *P(No|H) = 1/4*4/4*2/4*3/4

So for new data {Green, 2,Tall,No} the result is H

3	Explain the K – nearest neighbor algorithm for approximating a discrete –valued function f-> Rn-V with pseudo code.	10	CO3	L2

Solution:



Training algorithm:

- For each training example $\langle x, f(x)\rangle$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

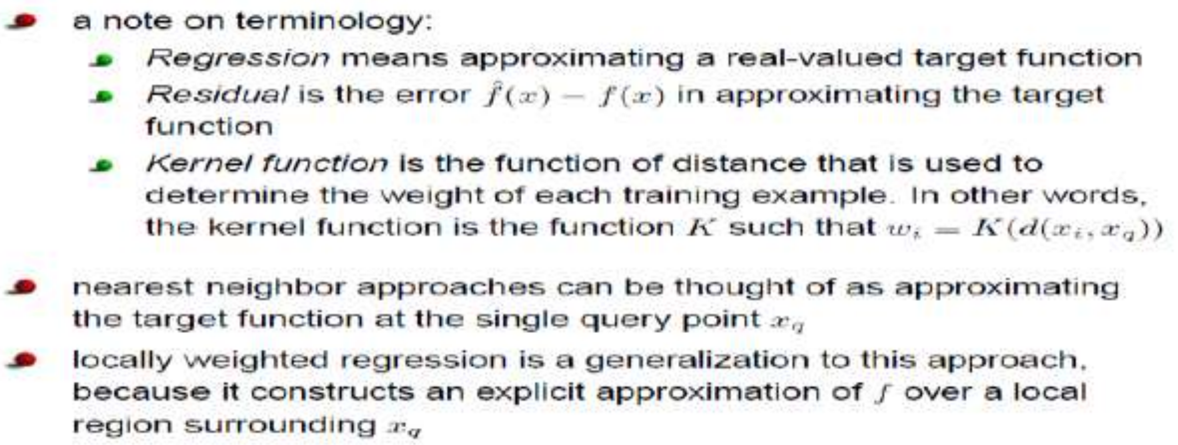
TABLE 8.1
The k-Nearest Neighbor algorithm for approximating a discrete-valued function $f : \Re^n \rightarrow V$.

Pseudo Code:

1. Load the data
2. Initialize the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
   1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are cosine, etc.
   2. Sort the calculated distances in ascending order based on distance values
   3. Get top k rows from the sorted array
   4. Get the most frequent class of these rows
   5. Return the predicted class

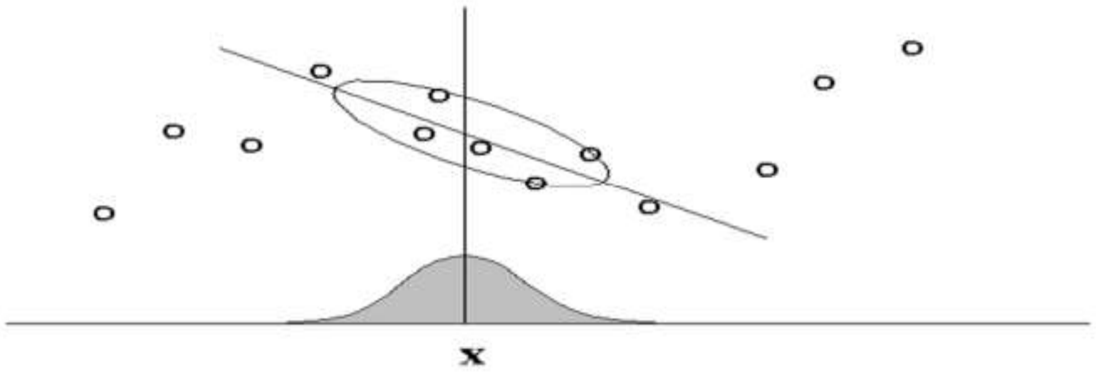
4.a	Explain locally weighted linear regression with an example.	5	CO3	L2

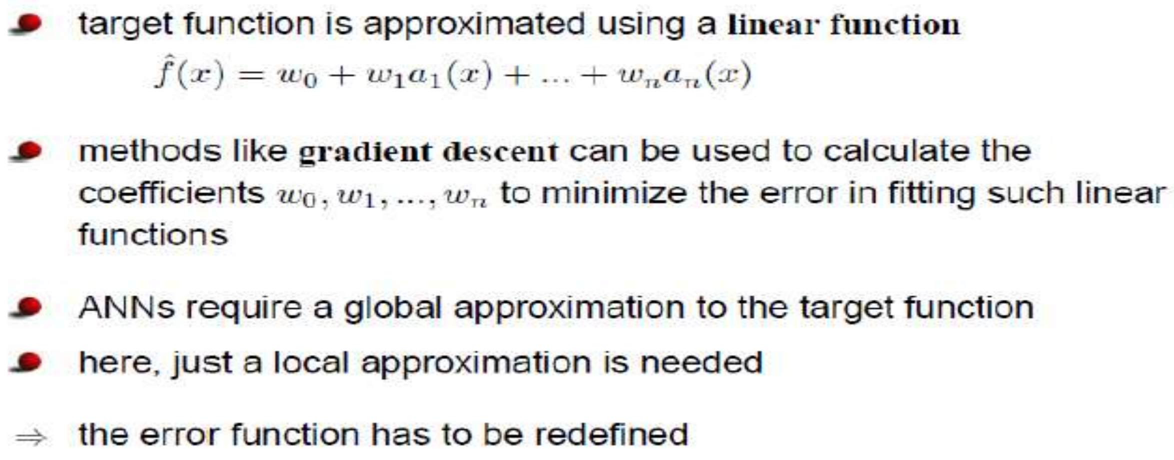
Solution:



- a note on terminology:
  - *Regression* means approximating a real-valued target function
  - *Residual* is the error $\hat{f}(x) - f(x)$ in approximating the target function
  - *Kernel function* is the function of distance that is used to determine the weight of each training example. In other words, the kernel function is the function K such that $w_i = K(d(x_i, x_q))$
- nearest neighbor approaches can be thought of as approximating the target function at the single query point x_q
- locally weighted regression is a generalization to this approach, because it constructs an explicit approximation of f over a local region surrounding x_q

-





- target function is approximated using a **linear function**

$$\hat{f}(x) = w_0 + w_1 a_1(x) + \ldots + w_n a_n(x)$$

- methods like **gradient descent** can be used to calculate the coefficients $w_0, w_1, \ldots, w_n$ to minimize the error in fitting such linear functions

- ANNs require a global approximation to the target function
- here, just a local approximation is needed

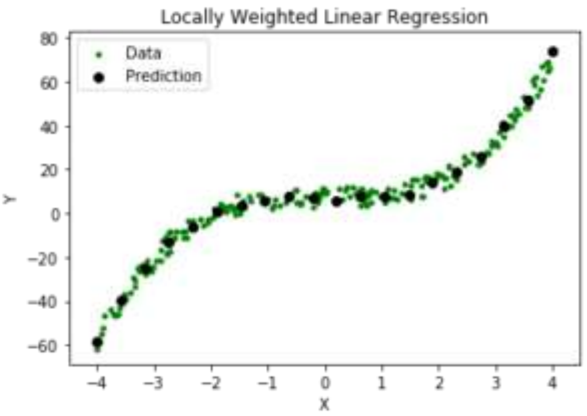
⇒ the error function has to be redefined

Example

- Consider a query point x = 5.0 and let x^{(1)} and x^{(2) be two points in the training set such that x^{(1)} = 4.9 and x^{(2)} = 3.0.
Using the formula w^{(i)} = exp(frac{-(x^{(i)} - x)^2}{2tau^2}) with tau = 0.5:
w^{(1)} = exp(frac{-(4.9 - 5.0)^2}{2(0.5)^2}) = 0.9802
w^{(2)} = exp(frac{-(3.0 - 5.0)^2}{2(0.5)^2}) = 0.000335

- So, J(theta) = 0.9802*(theta^Tx^{(1)} - y^{(1)}) + 0.000335*(theta^Tx^{(2)} - y^{(2)})
Thus, the weights fall exponentially as the distance between x and x^{(i)} increases and so does the contribution of error in prediction for x^{(i)} to the cost.

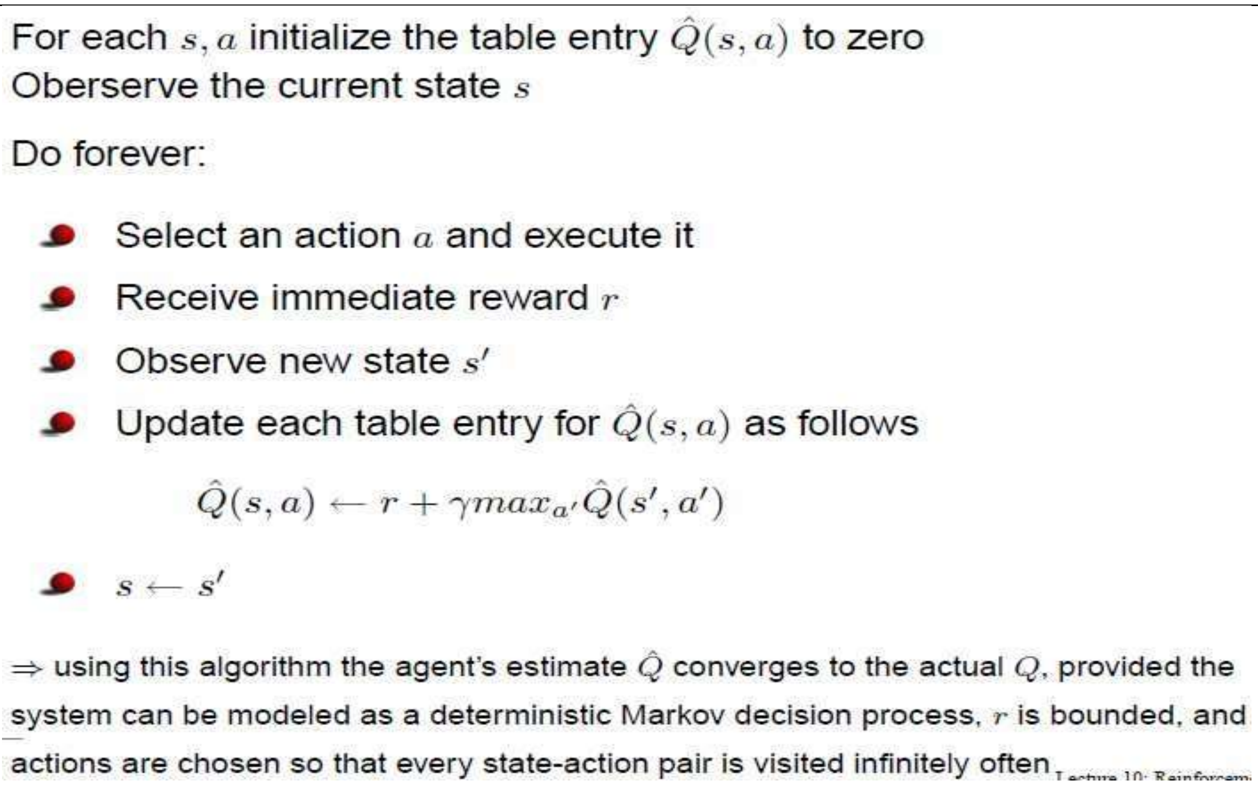
Consequently, while computing theta, we focus more on reducing (theta^Tx^{(i)} - y^{(i)})^2 for the points lying closer to the query point (having larger value of w^{(i)}).



Steps involved in locally weighted linear regression are:

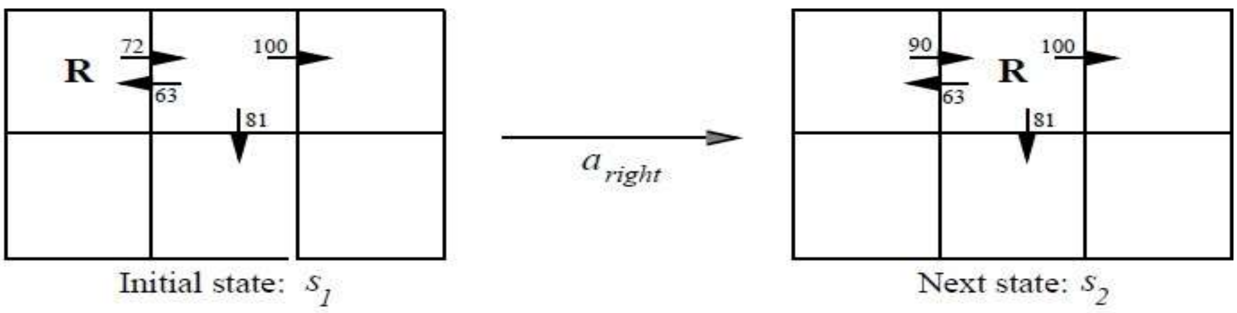
- Compute theta to minimize the cost. J(theta) = $sum_{i=1}^{m} w^{(i)}(theta^Tx^{(i)} - y^{(i)})^2.
- Predict Output: for given query point x,
- return: theta^Tx

4.b	Write a note on Q-learning.	5M	CO3	L2
Solution:				



```
For each $s, a$ initialize the table entry $\hat{Q}(s,a)$ to zero
Oberserve the current state $s$
Do forever:
    Select an action $a$ and execute it
    Receive immediate reward $r$
    Observe new state $s'$
    Update each table entry for $\hat{Q}(s,a)$ as follows
        $\hat{Q}(s,a) \leftarrow r + \gamma max_{a'} \hat{Q}(s',a')$
    $s \leftarrow s'$
```

$\Rightarrow$ using this algorithm the agent's estimate $\hat{Q}$ converges to the actual Q, provided the system can be modeled as a deterministic Markov decision process, r is bounded, and actions are chosen so that every state-action pair is visited infinitely often Lecture 10: Reinforcem



$$\begin{aligned}\hat{Q}(s_1, a_{right}) &\leftarrow r + \gamma \cdot \max_{a'} \hat{Q}(s_2, a') \\ &\leftarrow 0 + 0.9 \cdot \max\{66, 81, 100\} \\ &\leftarrow 90\end{aligned}$$

- each time the agent moves, Q Learning propagates $\hat{Q}$ estimates *backwards* from the new state to the old

5.a	Explain Bayesian Belief Networks and conditional independence with example.	6M	CO3	L2

Solution:

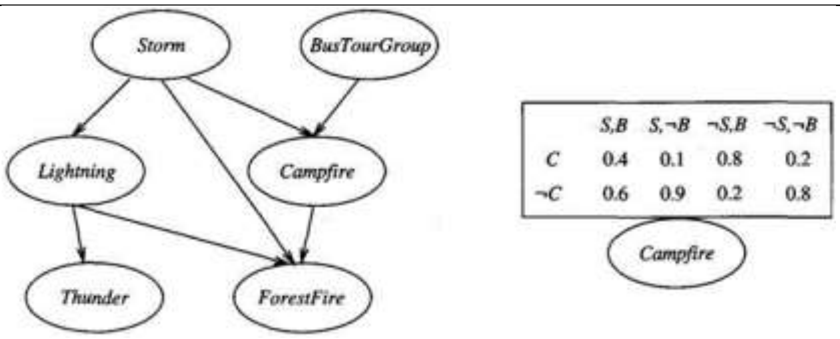
A Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities.

Bayesian belief networks allow stating conditional independence assumptions that apply to subsets of the variables

Representation

A Bayesian belief network represents the joint probability distribution for a set of variables.

Bayesian networks (BN) are represented by directed acyclic graphs.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

The Bayesian network in above figure represents the joint probability distribution over the boolean variables *Storm, Lightning, Thunder, ForestFire, Campfire,* and *BusTourGroup*

A Bayesian network (BN) represents the joint probability distribution by specifying a set of *conditional independence assumptions.*

- BN represented by a directed acyclic graph, together with sets of local conditional probabilities.
- Each variable in the joint space is represented by a node in the Bayesian network.
- The network arcs represent the assertion that the variable is conditionally independent of its non-descendants in the network given its immediate predecessors in the network.
- A ***conditional probability table*** **(CPT)** is given for each variable, describing the probability distribution for that variable given the values of its immediate predecessors.

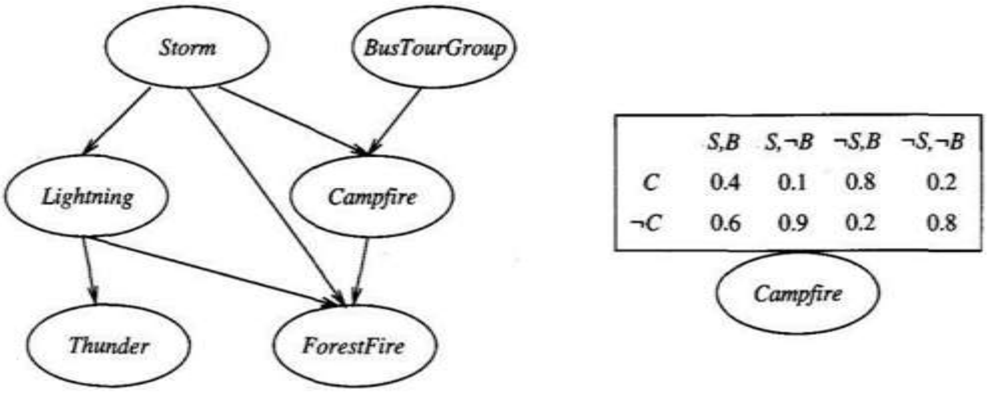
The joint probability for any desired assignment of values (y1, . . . , yn) to the tuple of network variables (Y1 . . . Ym) can be computed by the formula

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

Where, Parents(Yi) denotes the set of immediate predecessors of Yi in the network.

Example:

Consider the node ***Campfire***. The network nodes and arcs represent the assertion that ***Campfire*** is conditionally independent of its non-descendants ***Lightning*** and ***Thunder***, given its immediate parents Storm and ***BusTourGroup***.



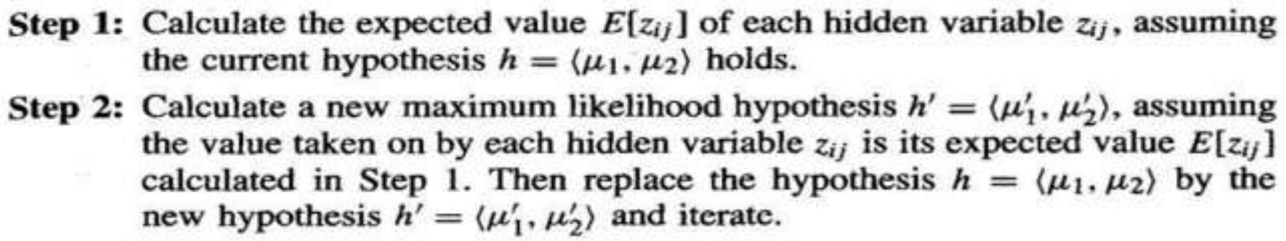
	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

This means that once we know the value of the variables ***Storm*** and ***BusTourGroup***, the variables ***Lightning*** and ***Thunder*** provide no additional information about ***Campfire*** The conditional probability table associated with the variable ***Campfire.*** The assertion is

P(Campfire = True | Storm = True, BusTourGroup = True) = 0.4

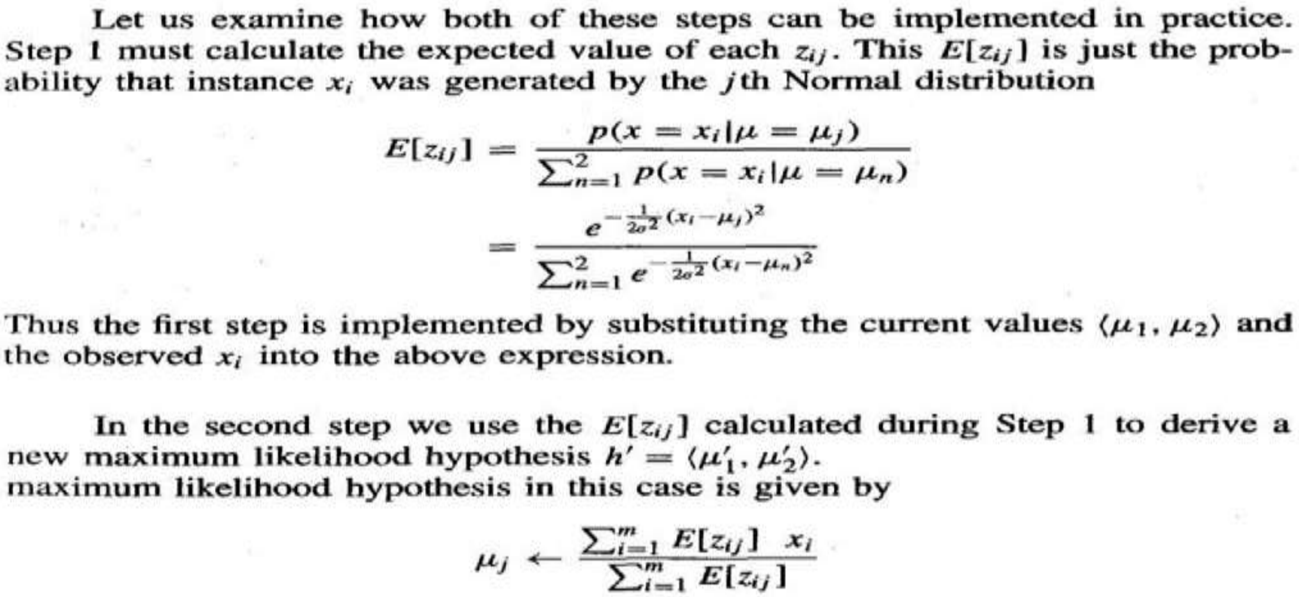
5.b	Explain the EM Algorithm in detail.	4M	CO3	L2

Solution:



Step 1: Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming the current hypothesis $h = \langle \mu_1, \mu_2 \rangle$ holds.

Step 2: Calculate a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated in Step 1. Then replace the hypothesis $h = \langle \mu_1, \mu_2 \rangle$ by the new hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$ and iterate.



Let us examine how both of these steps can be implemented in practice. Step 1 must calculate the expected value of each z_{ij}. This $E[z_{ij}]$ is just the probability that instance x_i was generated by the jth Normal distribution

$$E[z_{ij}] = \frac{p(x = x_i|\mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i|\mu = \mu_n)}$$

$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i-\mu_n)^2}}$$

Thus the first step is implemented by substituting the current values $\langle \mu_1, \mu_2 \rangle$ and the observed x_i into the above expression.

In the second step we use the $E[z_{ij}]$ calculated during Step 1 to derive a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$.

maximum likelihood hypothesis in this case is given by

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}]\ x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

6.a	Explain the CADET System with Case based reasoning with example.	6	CO2	L1

Solution:

Case-based reasoning (CBR) is a learning paradigm based on lazy learning methods and they classify new query instances by analysing similar instances while ignoring instances that are very different from the query.

In CBR represent instances are not represented as real-valued points, but instead, they use a ***rich symbolic*** representation.

CBR has been applied to problems such as conceptual design of mechanical devices based on a stored library of previous designs, reasoning about new legal cases based on previous rulings, and solving planning and scheduling problems by reusing and combining portions of previous solutions to similar problems

A prototypical example of a case-based reasoning

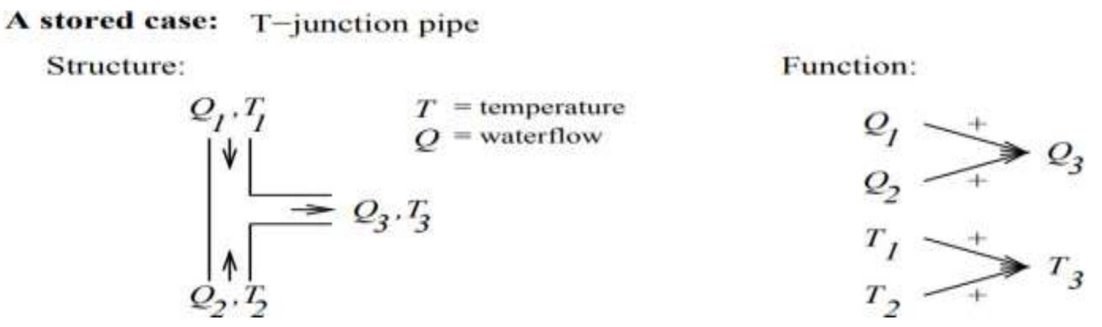
The CADET system employs case-based reasoning to assist in the conceptual design of simple mechanical devices such as water faucets.

It uses a library containing approximately 75 previous designs and design fragments to suggest conceptual designs to meet the specifications of new design problems.

Each instance stored in memory (e.g., a water pipe) is represented by describing both its structure and its qualitative function.

New design problems are then presented by specifying the desired function and requesting the corresponding structure.

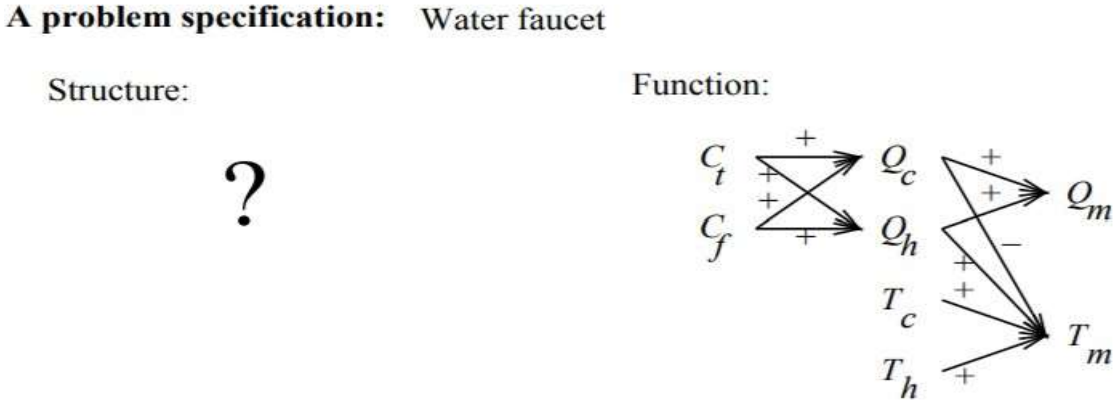
The problem setting is illustrated in below figure



The function is represented in terms of the qualitative relationships among the water- flow levels and temperatures at its inputs and outputs.

In the functional description, an arrow with a "+" label indicates that the variable at the arrowhead increases with the variable at its tail. A "-" label indicates that the variable at the head decreases with the variable at the tail.

- Here Q_c refers to the flow of cold water into the faucet, Q_h to the input flow of hot water, and Q_m to the single mixed flow out of the faucet.
- T_c, T_h, and T_m refer to the temperatures of the cold water, hot water, and mixed water respectively.
- The variable C_t denotes the control signal for temperature that is input to the faucet, and C_f denotes the control signal for waterflow.
- The controls C_t and C_f are to influence the water flows Q_c and Q_h, thereby indirectly influencing the faucet output flow Q_m and temperature T_m.



CADET searches its library for stored cases whose functional descriptions match the design problem. If an exact match is found, indicating that some stored case implements exactly the desired function, then this case can be returned as a suggested solution to the design problem. If no exact match occurs, CADET may find cases that match various subgraphs of the desired functional specification.

6.b	Write a note on Radial basis function.	4	CO2	L1

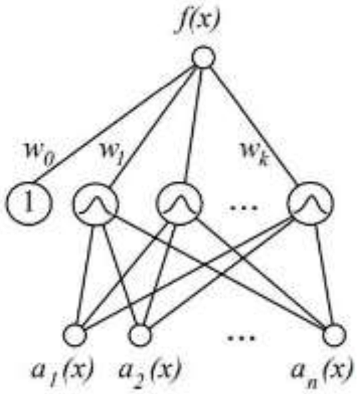
Solution:

Given a set of training examples of the target function, RBF networks are typically trained in a two-stage process.

1. First, the number k of hidden units is determined and each hidden unit u is defined by choosing the values of x_u and σ_u^2 that define its kernel function $K_u(d(x_u, x))$
2. Second, the weights w, are trained to maximize the fit of the network to the training data, using the global error criterion given by

$$E \equiv \frac{1}{2}\sum_{x\in D}(f(x) - \hat{f}(x))^2$$

Because the kernel functions are held fixed during this second stage, the linear weight values w, can be trained very efficiently



Several alternative methods have been proposed for choosing an appropriate number of hidden units or, equivalently, kernel functions.

- One approach is to allocate a Gaussian kernel function for each training example (xi,f (xi)), centring this Gaussian at the point xi. Each of these kernels may be assigned the same width σ^2.
- A second approach is to choose a set of kernel functions that is smaller than the number of training examples. This approach can be much more efficient than the first approach, especially when the number of training examples is large.

7.a	Explain maximum likelihood hypothesis for predicting probabilities.	8	CO3	L1

Solution:

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)**

· 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

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

The probability P(di|h, xi)

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

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

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

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

7.b	Write Bayes Theorem and explain the notations used.	2	CO3	L1

Solution:

Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

Notations

· 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

Bayes theorem is the cornerstone of Bayesian learning methods because it provides a way to calculate the posterior probability P(h|D), from the prior probability P(h), together with P(D) and P(D|h).

Bayes Theorem:

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