Machine Learning Module 4 Notes for 3rd IA

Module4: Introduction, **Bayes theorem**, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm

1. Define (i) Prior Probability (ii) Conditional Probability (iii) Posterior Probability.

Ans :

(i) The prior probability is the probability an event will happen before you taken any new evidence into account. The prior probability of an event is the probability of the event computed before the collection of new data. One begins with a prior probability of an event and revises it in the light of new data. For example, if 0.01 of a population has schizophrenia then the probability that a person drawn at random would have schizophrenia is 0.01. This is the prior probability.

Prior is a probability calculated to express one's beliefs about this quantity before some evidence is taken into account. In statistical inferences and Bayesian techniques, priors play an important role in influencing the likelihood for a datum.

(ii) The **conditional probability** of an event *A* given that an event *B* has occurred is written: **P(A|B) and** is calculated using:

$P(A|B) = P(A \cap B)$ **/** $P(B)$ as long as $P(B) > 0$.

Example :

P(A) = 4/52
P(B) = 4/51
P(A and B) = 4/52*4/51= 0.006

$$P\left(\frac{B}{A}\right) = \frac{P(A \text{ and } B)}{P(A)} = \frac{0.006}{0.077} = 0.078$$

iii) A posterior probability, in Bayesian statistics, is the revised or updated probability of an event occurring after taking into consideration new information. Posterior probability is calculated by updating the prior probability using Bayes' theorem. In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred.

Bayes' Theorem Formula

The formula to calculate a posterior probability of A occurring given that B occurred:

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

where:

A, B = events

(B) = greater than zero

P(B|A) = the probability of B occurring given that A is true

P(A) and P(B) =

the probabilities of A occurring and B occurring independently of each other

2. What is conditional Independence?

Ans:

- **Definition:** *X* is *conditionally independent* of *Y* given *Z* if the probability distribution governing *X* is independent of the value of *Y* given the value of *Z*; that is, if

  $(\forall x_i, y_j, z_k)\ P(X= x_i | Y= y_j, Z= z_k) = P(X= x_i | Z= z_k)$

  more compactly, we write

  $P(X | Y, Z) = P(X | Z)$

- **Example:** *Thunder* is conditionally independent of *Rain*, given *Lightning*

  $P(Thunder | Rain, Lightning) = P(Thunder | Lightning)$

- Naive Bayes uses cond. indep. to justify

  $P(X, Y | Z) = P(X | Y, Z)\ P(Y | Z) = P(X | Z)\ P(Y | Z)$

3. What is Bayes Theorem and maximum posterior hypothesis? Derive an equation for MAP Hypothesis and Maximum likelihood (ML) using Bayes theorem.

Ans :

Bayes' theorem is a formula that describes how to update the probabilities of hypotheses when given evidence. It follows simply from the axioms of conditional probability, but can be used to powerfully reason about a wide range of problems involving belief updates.

Given a hypothesis h and evidence D, Bayes' theorem states that the relationship between the probability of the hypothesis before getting the evidence $P(h)$ and the probability of the hypothesis after getting the evidence $P(h|D)$ is

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

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

- $P(h)$ = prior (initial) probability that hypothesis *h* holds , before we observed any training data.
- $P(D)$ = prior probability of training data *D*
- $P(h|D)$ = posterior probability of *h* given *D (it holds after we have seen the training data D)*
- $P(D|h)$ = probability of observing data *D* given some world in which hypothesis *h* holds.

Maximum a posterior (MAP) hypothesis:

- In many learning scenarios , the learner considers some set of candidate hypotheses H and is interested in finding the most probable hypotheses hЄH given the observed data D .
- Any such maximally probable hypothesis is called a maximum posteriori (MAP) hypothesis h_{MAP}:

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} P(h|D) \\
&= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
&= \arg\max_{h \in H} P(D|h)P(h)
\end{aligned}
$$

Maximum Likelihood:

In some cases we will assume that every hypothesis in H is equally probable a priori (***P*(*h*$_i$) = *P*(*h*$_j$) for all *h*$_i$** in H) hen can further simplify and need to consider the term **P(D|h)** is often called the likelihood of the data D given h and hypothesis that maximizes **P(D|h)** is called a ***Maximum likelihood*** (ML) hypothesis h_{ML}

$$
\boxed{h_{ML} = \arg\max_{h_i \in H} P(D|h_i)}
$$

4. Explain how Bayesian theorem can be used for Concept Learning . Describe Brute Force MAP learning Algorithm. Hence Derive the relation for P(h/D) when h is consistent with D and h is not consistent with D.

Ans:

Bayesian Theorem can learn concept by estimating the Maximum A Posterior probability of a given training data examples. Assume that 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}**

Assume fixed set of instances **<*x*$_1$,…, *x*$_m$>**

Assume ***D*** is the set of classifications: ***D* = <*c*(*x*$_1$),…,*c*(*x*$_m$)>**

Assume that the learner has given some sequence of training examples **<<x1,d1><x2,d2>,…….<xm, dm>>** where xi is some instance from X and where **di** is the target value of **xi (i.e di= c(xi)).**

Brute Force MAP Learning Algorithm:

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

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

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

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

Assumptions :

The probability distribution P(h) and P(D|h) is chosen to be consistent with the following assumptions:

1. The training data D is noise free (i.e. di = c(xi))
2. The target concept c is contained in the hypothesis space H
3. We have no *a priori reason* to believe that any hypothesis is more probable than any other.

The Values of P(h) and P(D|h)

- Choose *P*(*h*) to be *uniform* distribution
  - ***P*(*h*) = 1/|*H*| for all *h* in *H***
- Choose ***P*(*D*|*h*):**

$$P(D|h) = \begin{cases} \textbf{1 if di =h(xi) for all di in D(}h\textbf{ consistent with }D\textbf{)} \\ \textbf{0 otherwise} \end{cases}$$

Two cases :

- By Applying Bayes theorem

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

- **Case1 :** When h is inconsistent with training data D:

  $P(h|D) = 0.P(h)/P(D) = 0$

- **Case 2:** When h is consistent with D , we have

  $P(h|D) = (1*1/|H|)/(|V_{SH},D|/|H|)$

  $= 1/|V_{SH},D|$

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

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

5. What is the relevance and features of Bayesian theorem? Explain the practical difficulties of Bayesian theorem. Discuss at least five applications of Bayes learning.

Ans:

Relevance of Bayesian Learning : Bayesian Learning is relevant for two reasons

- **First reason**: **explicit manipulation of probabilities**
  - among the most practical approaches to certain types of learning problems
  - **e.g.** Bayes classifier is **competitive with decision tree** and neural network learning
- **Second reason: useful perspective for understanding learning methods that do not explicitly manipulate probabilities**
  - determine conditions under which algorithms output the most probable hypothesis
  - e.g. justification of the error functions in ANNs
  - e.g.justification of the inductive bias of decision trees

Features of Bayesian Learning Methods :

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
- Bayesian methods can accommodated hypothesis that make probabilities predictions.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities
- Provides a standard of optimal decision making against which other practical methods can be measured.

The practical difficulties of Bayesian theorem:

- **Initial knowledge of many probabilities is required** : It does not tell you how to select a prior. There is no correct way to choose a prior. Bayesian inferences require skills to translate subjective prior beliefs into a mathematically formulated prior. If you do not proceed with caution, you can generate misleading results. It can produce posterior distributions that are heavily influenced by the priors. From a practical point of view, it might sometimes be difficult to convince subject matter experts who do not agree with the validity of the chosen prior.
- **Significant computational costs required :**It often comes with a high computational cost, especially in models with a large number of parameters.

Applications of Bayesian Learning:

- News Categorization
- Medical Diagnosis
- E- mail Spam Detection
- Face Recognition
- Sentiment Analysis
- Digit Recognition
- Weather Prediction

[Note : Students are hereby instructed to provide the detail discussion on their own understanding of the application]

6. Write a Short note on Naïve Bayes classifier. Write and explain the equation for target value output, v_{NB} by the Bayes Classifier.

Ans :

Highly Bayesian learning method is the naïve Bayes learner often called the naïve Bayes Classifier . Bayesian Classifier assumes that all the variables are ***conditionally independent*** given the value of the target variable. The naïve Bayes Classifier applies to learning tasks where each ***instance x is described by a conjunction of attribute values*** and where ***the target function f(x) can take on any value from some finite set V.***A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values **<a_1, a_2 , a_3 ,------- a_n >.** The learner is asked to predict the target value, or classification, for this new instance.

The Bayesian approach to classifying the new instance is to assign the most probable target value, V_{MAP}, given the attribute values <a_1, a_2 , a_3 ,------- a_n >.that describe the instance.

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

We can use Bayes theorem to rewrite this expression as

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

Naive Bayes classifier:

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

7. Illustrate the application of the Bayes Classifier to a concept learning problem of Play Tennis Concept example.

Example: Consider the Play tennis examples given below:

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

Learning Phase:

P(Outlook | Play)

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

P(Temperature | Play)

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

P(Humidity | Play)

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

P(Wind | Play)

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Test Phase:

- Given a new instance,

  $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

- Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

- MAP rule

P(*Yes*|$\mathbf{x}'$): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053
P(*No*|$\mathbf{x}'$): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|$\mathbf{x}'$) < P(*No*|$\mathbf{x}'$), we label $\mathbf{x}'$ to be "*No*".

8. Consider a football game between two rival teams: Team 0 and Team 1. Support Team 0 wins 65% of the time and Team 1 wins the remaining matches. Among the games won by Team 0, only 30% of them come from playing on Team 1's football field. On the other hand, 75% of the victories for Team 1 are obtained while playing at home. If Team 1 is to host the next match between the two teams, which team will most likely emerge as the winner.

Solution:

Let X be the team hosting the match and Y be the winner of the match. Both X and Y can take on values from the set $\{0,1\}$. Then:

Probability Team 0 wins is $P(Y = 0) = 0.65$.

Probability Team 1 wins is $P(Y = 1) = 1 - 0.65 = 0.35$.

Probability Team 1 hosted the match it won is

$$P(X = 1|Y = 1) = 0.75.$$

Probability Team 1 hosted the match won by Team 0 is

$$P(X = 1|Y = 0) = 0.3.$$

$$P(Y = 1|X = 1) = \frac{P(X = 1|Y = 1) \times P(Y = 1)}{P(X = 1)}$$

$$= \frac{P(X = 1|Y = 1) \times P(Y = 1)}{P(X = 1, Y = 1) + P(X = 1, Y = 0)}$$

$$= \frac{P(X = 1|Y = 1) \times P(Y = 1)}{P(X = 1|Y = 1)P(Y = 1) + (X = 1|Y = 0)P(Y = 0)}$$

$$= \frac{0.75 \times 0.35}{0.75 \times 0.35 + 0.3 \times 0.65}$$

$$= 0.5738$$

Hence forth probability of hosting teaming winning chance (0.5738) is more .

9. The following table gives data set about stolen vehicles . Using Naïve Bayes classifier classify the new data (Red, SUV, Domestic) .

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:

The Bayes Naive classifier selects the most likely classification V_{nb} given the attribute values $a_1, a_2, \ldots a_n$. This results in:

$$V_{nb} = \text{argmax}_{v_j \in V} P(v_j) \prod P(a_i|v_j) \quad (1)$$

We generally estimate $P(a_i|v_j)$ using m-estimates:

$$P(a_i|v_j) = \frac{n_c + mp}{n + m} \quad (2)$$

where:

n = the number of training examples for which $v = v_j$
n_c = number of examples for which $v = v_j$ and $a = a_i$
p = a priori estimate for $P(a_i|v_j)$
m = the equivalent sample size

For a given problem We need to estimate:

$$P(a_i|v_j) = \frac{n_c + mp}{n + m}$$

We want to classify a Red Domestic SUV. Note there is no example of a Red Domestic SUV in our data set. Looking back at equation (2) we can see how to compute this. We need to calculate the probabilities

P(Red|Yes), P(SUV|Yes), P(Domestic|Yes) ,

P(Red|No) , P(SUV|No), and P(Domestic|No)

and multiply them by P(Yes) and P(No) respectively . We can estimate these values using equation

```
Yes:                    No:
    Red:                    Red:
        n = 5                   n = 5
        n_c= 3                  n_c = 2
        p = .5                  p = .5
        m = 3                   m = 3
    SUV:                    SUV:
        n = 5                   n = 5
        n_c = 1                 n_c = 3
        p = .5                  p = .5
        m = 3                   m = 3
    Domestic:               Domestic:
        n = 5                   n = 5
        n_c = 2                 n_c = 3
        p = .5                  p = .5
        m = 3                   m =3
```

Looking at $P(Red|Yes)$, we have 5 cases where v_j = Yes , and in 3 of those cases a_i = Red. So for $P(Red|Yes)$, $n = 5$ and $n_c = 3$. Note that all attribute are binary (two possible values). We are assuming no other information so, $p = 1$ / (number-of-attribute-values) = 0.5 for all of our attributes. Our m value is arbitrary, (We will use $m = 3$) but consistent for all attributes. Now we simply apply eqauation (2) using the precomputed values of n , n_c, p, and m.

$$P(Red|Yes) = \frac{3+3*.5}{5+3} = .56 \qquad P(Red|No) = \frac{2+3*.5}{5+3} = .43$$

$$P(SUV|Yes) = \frac{1+3*.5}{5+3} = .31 \qquad P(SUV|No) = \frac{3+3*.5}{5+3} = .56$$

$$P(Domestic|Yes) = \frac{2+3*.5}{5+3} = .43 \qquad P(Domestic|No) = \frac{3+3*.5}{5+3} = .56$$

We have $P(Yes) = .5$ and $P(No) = .5$, so we can apply equation (2). For $v = Yes$, we have

```
P(Yes) * P(Red | Yes) * P(SUV | Yes) * P(Domestic|Yes)

    =  .5 * .56 * .31 * .43   = .037
```

and for $v = No$, we have

```
P(No) * P(Red | No) * P(SUV | No) * P (Domestic | No)

    = .5 * .43 * .56 * .56 = .069
```

Since $0.069 > 0.037$, our example gets classified as 'NO'

10. Explain how Naïve Bayes algorithm is useful for learning and classifying text?

Answer:

Learning to Classify Text – Algorithm

S1: LEARN_NAIVE_BAYES_TEXT (*Examples*, V)

S2: CLASSIFY_NAIVE_BAYES_TEXT (*Doc*)

- *Examples is a set of text documents along with their target values.*
- *V is the set of all possible target values.*
- *This function (S1) learns the probability terms* $P(w_k \mid v_j)$*, describing the probability that a randomly drawn word from a document in* ***class*** v_j *will be the English word* w_k*. It also learns the class prior probabilities* $P(v_j)$*.*

S1: LEARN_NAIVE_BAYES_TEXT (*Examples*, V)

[V: Class, W: Word, doc : Documents]

1. *collect all words and other tokens that occur in Examples*

- ***Vocabulary*** ← all distinct words and other tokens in *Examples*

2. *calculate the required* $P(v_j)$ ***and*** $P(w_k \mid v_j)$ *probability terms*

- For each target value v_j in *V* do

$$P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$$

  - $docs_j$ ← **subset of *Examples* for which the target value is** v_j
  - $Text_j$ ← **a single document created by concatenating all members of** $docs_j$
  - n ← **total number of words in** $Text_j$ **(counting duplicate words multiple times)**
  - **for each word** w_k **in *Vocabulary***

n_k ← **number of times word** w_k **occurs in** $Text_j$

S2: CLASSIFY_NAIVE_BAYES_TEXT (*Doc*)

- *positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB} where

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

Example: In the example, we are given a sentence " ***A very close game***", a training set of five sentences (as shown below), and their corresponding category (Sports or Not Sports). The goal is to build a Naive Bayes classifier that will tell us which category the sentence " ***A very close game***" belongs to. Applying a Naive Bayes classifier, thus the strategy would be calculating the probability of both "*A very close game* ***is Sports***", as well as it's ***Not Sports.*** The one with the higher probability will be the result.

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Step 1: Feature Engineering

- **word frequencies**, i.e., counting the occurrence of every word in the document.
- ***P(a very close game)*** = P(a) X P(very) X P(close) X P(game)
- ***P(a very close game | Sports)*** = P(a|Sports) X P(Very|Sports) X P(close|Sports) X P(game|Sports)
- **P(*a very close game* | *Not Sports*)** = P(*a* | *Not Sports*) x P(*very* | *Not Sports*) x P(*close* | *Not Sports*) x P(*game* | *Not Sports*)

Step 2: Calculating the probabilities

- Here , the word "close" does not exist in the category Sports, thus **P(*close* |*Sports*)** = 0, leading to **P(*a very close game* | *Sports*)**=0.
- The probabilities are calculated using multinomial probability distribution function

$$P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$$

Word	P(word | Sports)	P(word | Not Sports)
a	$\frac{2+1}{11+14}$	$\frac{1+1}{9+14}$
very	$\frac{1+1}{11+14}$	$\frac{0+1}{9+14}$
close	$\frac{0+1}{11+14}$	$\frac{1+1}{9+14}$
game	$\frac{2+1}{11+14}$	$\frac{0+1}{9+14}$

$$P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports) \times P(Sports)$$
$$= 4.61 \times 10^{-5}$$
$$= 0.0000461$$

$$\mathrm{P(a} - \text{Not Sports}) \times P(very|Not\ Sports) \times P(close|Not\ Sports) \times P(game|Not\ Sports) \times P(Not\ Sports)$$
$$= 1.43 \times 10^{-5}$$
$$= 0.0000143$$

As seen from the results shown below, P(a very close game | Sports) gives a higher probability, suggesting that the sentence belongs to the Sports category.

11. What are Bayesian Belief nets? Where are they used? Can it solve all types of problems?

Ans: Bayesian networks are a type of **Probabilistic Graphical Model** that can be used to build models from data and/or expert opinion. Bayesian networks are comprised of nodes and directed edges. Bayesian network models capture both conditionally dependent and conditionally independent relationships between random variables. Models can be prepared by experts or learned from data, then used for inference to estimate the probabilities for causal or subsequent events. Bayesian Belief networks ***describe conditional independence*** among *subsets* of variables.

Bayesian belief networks.

• Represent the full joint distribution more compactly with smaller number of parameters.

• Take advantage of conditional and marginal independences among components in the distribution

Nodes

In many Bayesian networks, each node represents a **Variable** such as someone's height, age or gender. A variable might be discrete, such as Gender = {Female, Male} or might be continuous such as someone's age.

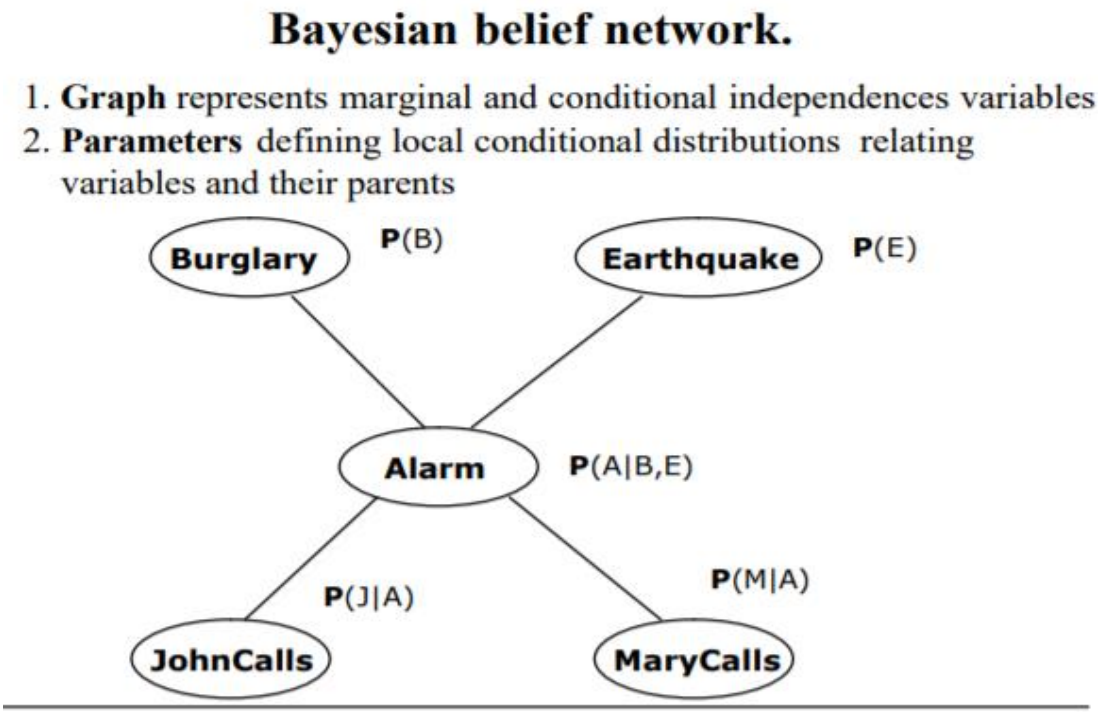
Links

Links are added between nodes to indicate that one node directly influences the other. When a link does not exist between two nodes, this does not mean that they are completely independent, as they may be connected via other nodes. They may however become dependent or independent depending on the evidence that is set on other nodes.

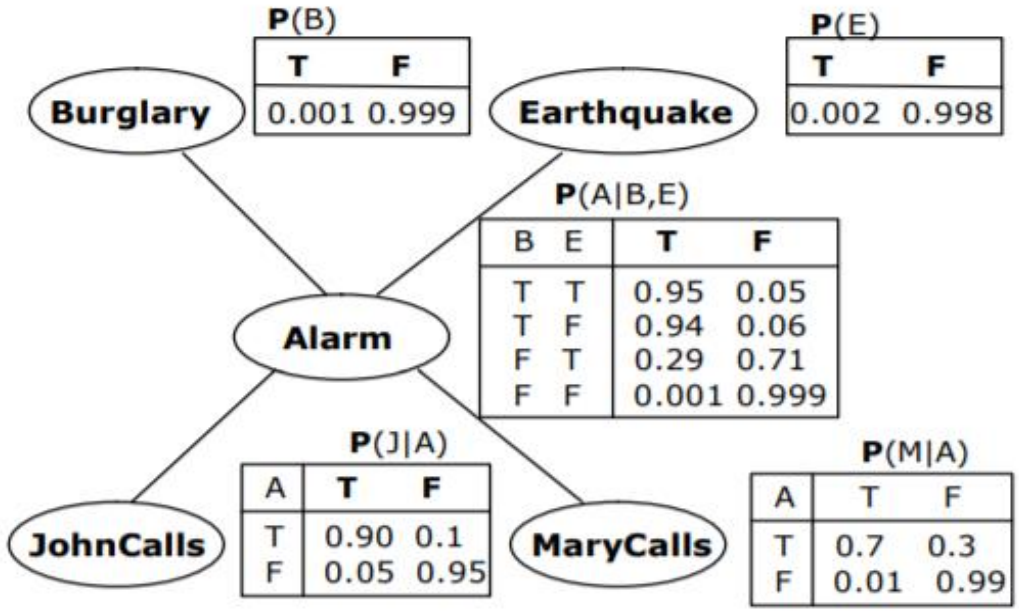
Directed Acyclic Graph (DAG)

A Bayesian network is a type of graph called a **Directed Acyclic Graph** or **DAG**. A Dag is a graph with directed links and one which contains no directed cycles.

Example :



Bayesian belief network.



Why Bayesian nets are used?

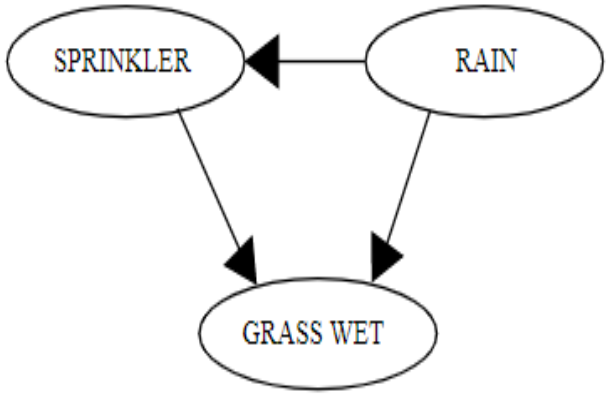
Bayesian networks are a type of Probabilistic Graphical Model that can be **used** to build models from data and/or expert opinion. They can be **used** for a wide range of tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty. They can be used for a wide range of tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty. They are also commonly referred to as **Bayes nets**, **Belief networks** and sometimes **Causal networks**.

Example :

Two events can cause grass to be wet: an active sprinkler or rain. Rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler usually is not active). This situation can be modeled with

a Bayesian network (shown to the right). Each variable has two possible values, T (for true) and F (for false).

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
T	F	
0.2	0.8	

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Limitations of Bayesian Nets :

Bayesian nets cannot do

- Combining conflicting beliefs that are based on different implicit conditions,
- Carry out inference when the premises are based different implicit conditions,

12. Write and Explain EM Algorithm. Discuss what are Gaussian Mixtures.

Ans :

- **Given:**
  - Instances from *X* generated by mixture of *k* Gaussian distributions
  - Unknown means $<\mu_1,\ldots,\mu_k>$ of the *k* Gaussians
  - Don't know which instance x_i was generated by which Gaussian
- Determine:
  - Maximum likelihood estimates of $<\mu_1,\ldots,\mu_k>$

EM Algorithm

- Pick random initial $h = <\mu_1, \mu_2>$ then iterate
- **E step:** Calculate the expected value $E[z_{ij}]$ of each ***hidden variable*** z_{ij}, assuming the current hypothesis h = $<\mu_1, \mu_2>$ holds.

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

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

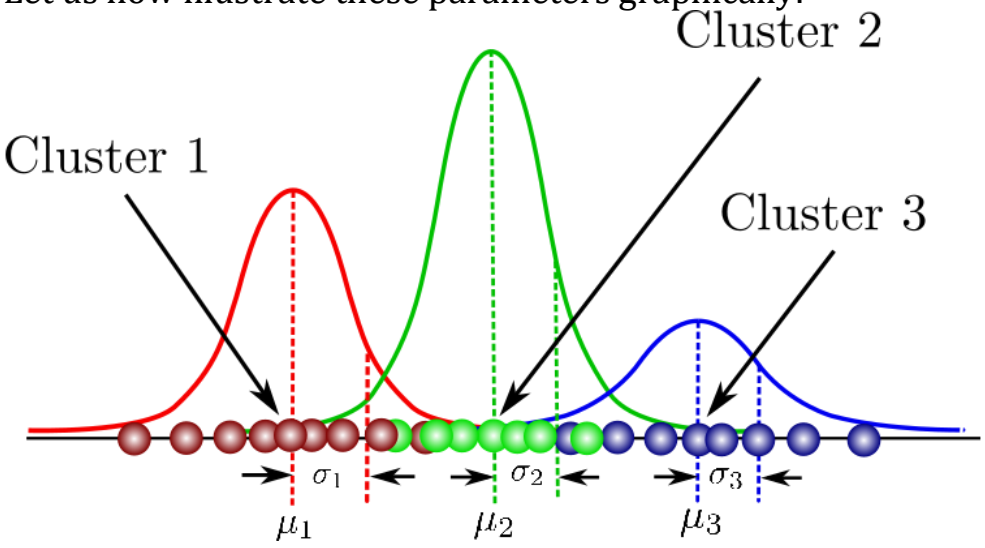
- **M step: Calculate a new maximum likelihood hypothesis $h' = <\mu'_1, \mu'_2>$, assuming the value taken on by each hidden variable z_{ij} its expected value $E[z_{ij}]$ calculated above. Replace $h = <\mu_1, \mu_2>$ by $h' = <\mu'_1, \mu'_2>$.**

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

Gaussian Mixtures:

A *Gaussian Mixture* is a function that is comprised of several Gaussians, each identified by $k \in \{1,\ldots, K\}$, where K is the number of clusters of our dataset. Each Gaussian k in the mixture is comprised of the following parameters:

- A mean μ that defines its centre.
- A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
- A mixing probability π that defines how big or small the Gaussian function will be.
  - Let us now illustrate these parameters graphically:
  -



Here, we can see that there are three Gaussian functions, hence $K = 3$. Each Gaussian explains the data contained in each of the three clusters available. The mixing coefficients are themselves probabilities and must meet this condition:

$$\sum_{k=1}^{K} \pi_k = 1 \qquad (1)$$

Now how do we determine the optimal values for these parameters? To achieve this we must ensure that each Gaussian fits the data points belonging to each cluster. This is exactly what maximum likelihood does.

In general, the Gaussian density function is given by:

$$\mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Where **x** represents our data points, *D* is the number of dimensions of each data point. μ and Σ are the mean and covariance, respectively. If we have a dataset comprised of $N = 1000$ three-dimensional points ($D = 3$), then **x** will be a 1000×3 matrix. μ will be a 1×3 vector, and Σ will be a 3×3 matrix.

13. Derive the following:

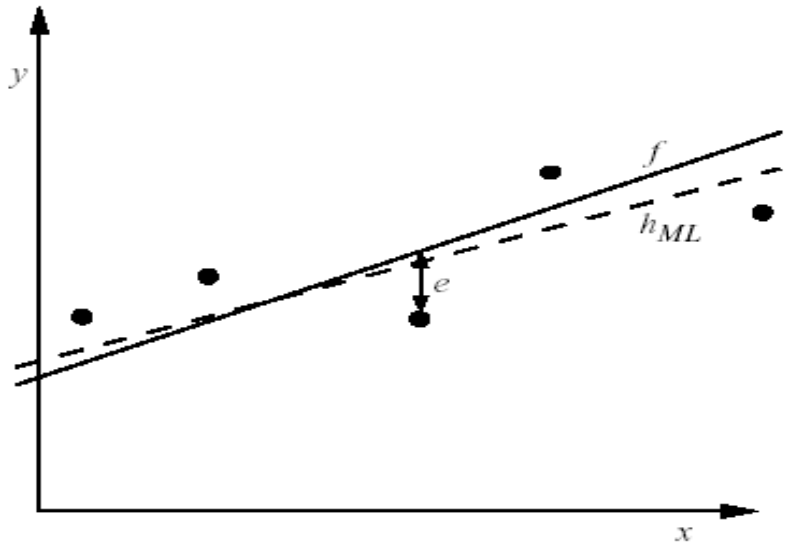
a. Maximum Likelihood and Least Square Error Hypotheses

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 hypothesis.

- Consider any real-valued target function *f*

  Training examples $\langle x_i, d_i \rangle$, where d_i is noisy training value
  - $d_i = f(x_i) + e_i$
  - e_i is random variable (noise) drawn independently for each x_i according to some Gaussian distribution with mean=0
- Then the maximum likelihood hypothesis h_{ML} is the one that minimizes the sum of squared errors:

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



$$
\begin{aligned}
h_{ML} &= \underset{h \in H}{\operatorname{argmax}} p(D|h) \\
&= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h) \\
&= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}
\end{aligned}
$$

Maximize natural log of this instead...

$$
\begin{aligned}
h_{ML} &= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2 \\
&= \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} \left(d_i - h(x_i)\right)^2
\end{aligned}
$$

b. Maximum Likelihood for Predicting Probabilities:

Consider predicting survival probability from patient data
Training examples $\langle x_i, d_i \rangle$, where d_i is 1 or 0
Want to train neural network to output a *probability* given x_i (not a 0 or 1)
In this case can show

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

- **Weight update rule for a sigmoid unit**
- $$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$
- where

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

For Complete Derivation Refer the text book " Machine Learning " Tom M Mitchell : **Page No 168 to 171**

c. Minimum Description Length (MDL) Principle

Occam's razor: prefer the shortest hypothesis
MDL: prefer the hypothesis *h* that minimizes
where $L_C(x)$ is the description length of x under encoding C
Example:

- *H* = decision trees, *D* = training data labels
- $L_{C1}(h)$ is # bits to describe tree *h*
- $L_{C2}(D|h)$ is #bits to describe *D* given *h*
  - Note $L_{C2}(D|h) = 0$ if examples classified perfectly by *h*. Need only describe exceptions

- Hence h_{MDL} trades off tree size for training errors

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h)P(h)$$
$$= \arg\max_{h \in H} \log_2 P(D \mid h) + \log_2 P(h)$$
$$= \arg\min_{h \in H} -\log_2 P(D \mid h) - \log_2 P(h) \quad (1)$$

Interesting fact from information theory:

The optimal (shortest expected length) code for an event with probability *p* is $\log_2 p$ bits.

So interpret (1):

$-\log_2 P(h)$ is the length of *h* under optimal code

$-\log_2 P(D|h)$ is length of *D* given *h* in optimal code

$\rightarrow$ prefer the hypothesis that minimizes

length(h)+length(misclassifications)

d. K Means Algorithm

Algorithm

1. The sample space is initially partitioned into K clusters and the observations are randomly assigned to the clusters.
2. For each sample:

   Calculate the distance from the observation to the centroid of the cluster.

   IF the sample is closest to its own cluster THEN leave it ELSE select another cluster.
3. Repeat steps 1 and 2 untill no observations are moved from one cluster to another

Details of K-means

1. Initial centroids are often chosen randomly.
   - *Clusters produced vary from one run to another*
2. The centroid is (typically) the mean of the points in the cluster.
3. 'Closeness' is measured by **Euclidean distance**, cosine similarity, correlation, etc.
4. K-means will converge for common similarity measures mentioned above.
5. Most of the convergence happens in the first few iterations.
   - *Often the stopping condition is changed to 'Until relatively few points change clusters'*

Euclidean Distance

$$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\ldots+|x_{ip}-x_{jp}|^2}$$

A simple example: Find the distance between two points, the original and the point (3,4)

$$d_E(O,A)=\sqrt{3^2+4^2}=5$$

Update Centroid

We use the following equation to calculate the n dimensional centroid point amid k n-dimensional points

$$CP(x_1,x_2,\ldots,x_k) = \left(\frac{\sum_{i=1}^{k} x1st_i}{k}, \frac{\sum_{i=1}^{k} x2nd_i}{k}, \ldots, \frac{\sum_{i=1}^{k} xnth_i}{k}\right)$$

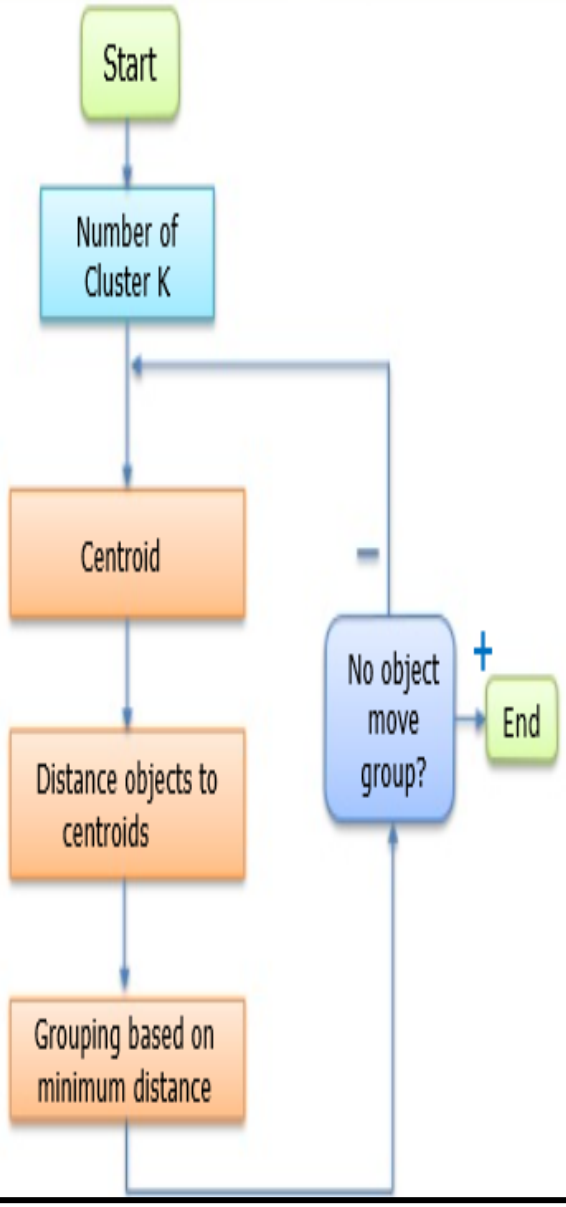
Example: Find the centroid of 3 2D points, (2,4), (5,2) and (8,9)

$$CP = \left(\frac{2+5+8}{3}, \frac{4+2+9}{3}\right) = (5,5)$$

How the K-Mean Clustering algorithm works?

$$\left[\frac{x_1 + x_2 + x_3}{3} \;,\; \frac{y_1 + y_2 + y_3}{3}\right]$$

Process Flow of K-means



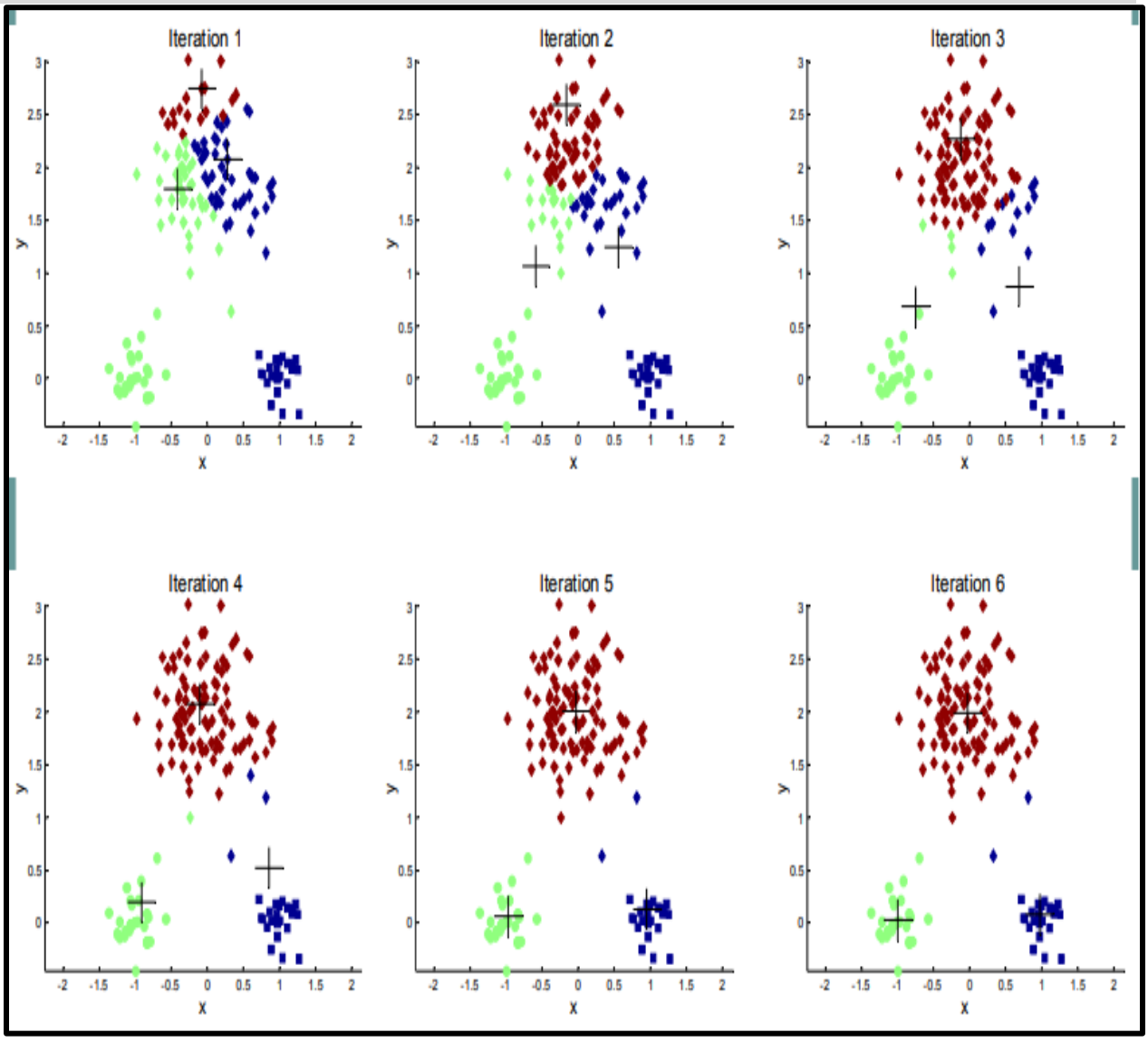
Iterate until *stable* (cluster centers converge):

1. Determine the centroid coordinate.

2. Determine the distance of each object to the centroids.

3. Group the object based on minimum distance (find the closest centroid)

K-means clustering example



Derivation of the k –means Algorithm

- **Refer the text book " Machine Learning " Tom M Mitchell : Page No 195 to 196.**

Note: All Students are hereby instructed to study the VTU Prescribed Textbook: Machine Learning Tom M. Mitchell in addition to this notes for the main examination .