Agenda

Day2	1. **Types of Machine Learning Algorithms**
	2. **Bayesian Learning:** Bayes Theorem and Concept Learning, ML and LS error Hypothesis, Naïve Bayes Classifier, Bayesian belief networks, EM Algorithm

- **Source Code**
- https://github.com/profthyagu

1. Classification of Machine Learning Algorithms

1. Based on Depth of Learning
2. Based on Type of learning

ಡಾ‖ ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

1.1 Based on Depth of Learning

1. **Shallow Learning**
   - Algorithms with Few Layers
   - Better for Less Complex and Smaller Data sets
   - *Eg: Logistic Regression and Support vector Machines*

2. **Deep Learning**
   - New technique that uses many layers of neural network (a model based on the structure of human brain)
   - Useful when **the target function is very complex and data sets are very large**.

1.2 Based on Type of Learning

1. **Supervised Learning**

   - **X and Y**

   - Given an observation **X** what is the best label for **Y**

2. **Unsupervised Learning**

   - **X**

   - Given a set of **X** cluster or summarize them

3. **Reinforcement Learning**

   - Determine what to do based on Rewards and punishments

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

1.2.1. Supervised Learning

- Here the training (labeled) data set containing ***input/predictors and output*** will be fed to the machine. The machine with the help of algorithm analyze the data set and ***generates a suitable model /function*** that best describes the input data. **i.e** it generates a function ***f(X) which makes best estimation of the output Y = f(X) for given X.***

- The generated **model / function** can be used to predict the output values for new data based on those relationships which it learned from the previous data sets.
    - When **Y is discrete (True /False, ..)** – **Classification**
    - When **X is continuous (Real Numbers)- Regression**

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
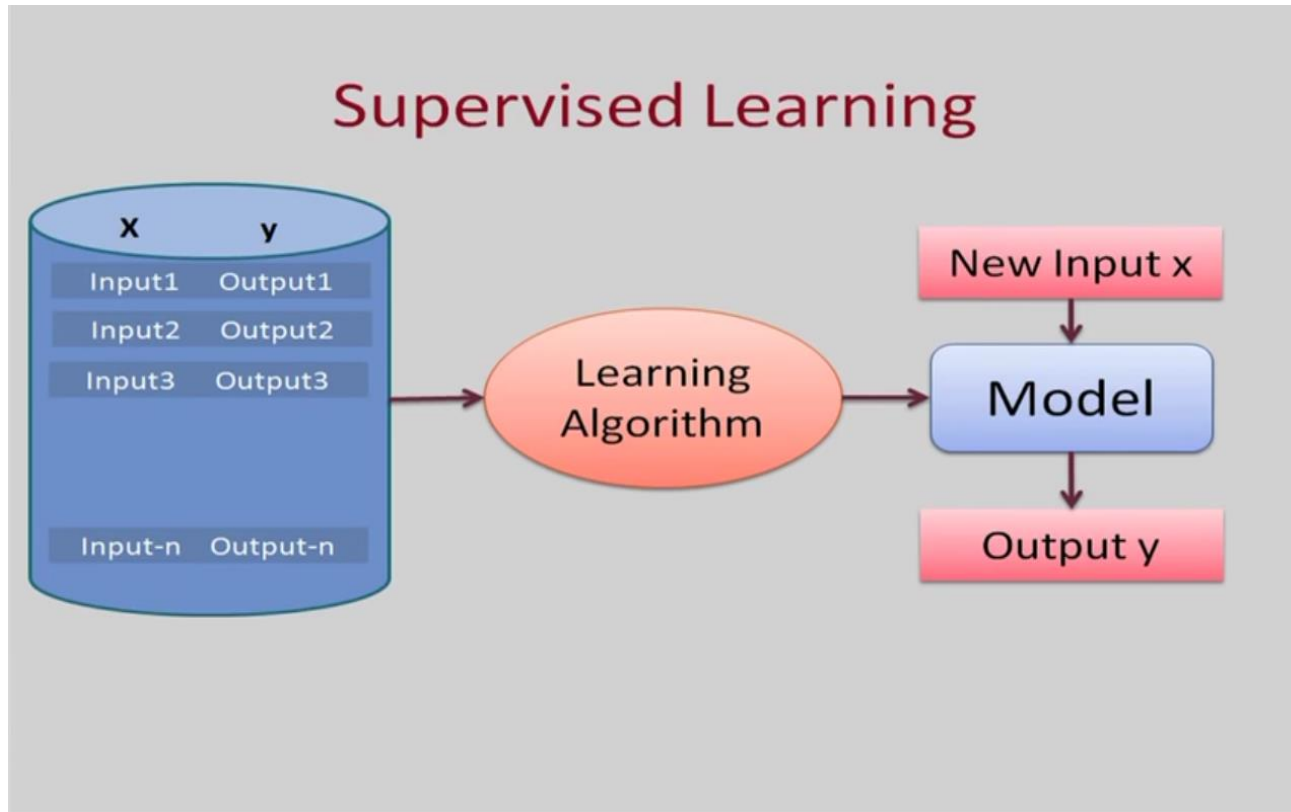
Supervised Learning
(Train Me)

Supervised Learning Use cases

Cortana or any speech automated system in your **mobile phone trains your voice and then starts** *working based on this training*. This is an application of Supervised Learning

Supervised Learning Use cases

Weather Apps

Predicts the upcoming weather by analyzing the parameters for a given time on some prior knowledge *(when its sunny, temperature is higher; when its cloudy, humidity is higher, etc.).*

Supervised Learning Use cases

Biometric Attendance

In **Biometric Attendance** you can train the machine with inputs of your biometric identity – *it can be your thumb, iris or ear-lobe, et*c. Once the machine is trained it can validate your future input and can easily identify you.

Types of Supervised Learning
(Task Driven . Develop Prediction Model based on Input and Output Data)

1. Classification (Discrete)

a) Logistic Regression
b) KNN
c) Decision Trees
d) Support Vector Machines
e) Naïve Bayesian
f) Discriminant Analysis
g) Random Forest
h) AdaBoost
i) Neural Networks

2. Regression (Continuous)

a) Linear Regression
b) SVR
c) GPR
d) Ensemble Methods

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

1.2.2: Unsupervised Learning

- This approach is **data driven** . The computer is trained with *unlabeled* input data.

- These algorithms try to use techniques on the input data to *mine for rules, detect patterns,* **and** *summarize and group the data points* which help in deriving meaningful insights and describe the data better to the users.

- **Mathematically,** Unsupervised learning is where **you only have input data (X)** and no corresponding output variables.

- The goal for unsupervised learning is to model the *underlying structure or distribution in the data in order to learn more about the data*

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
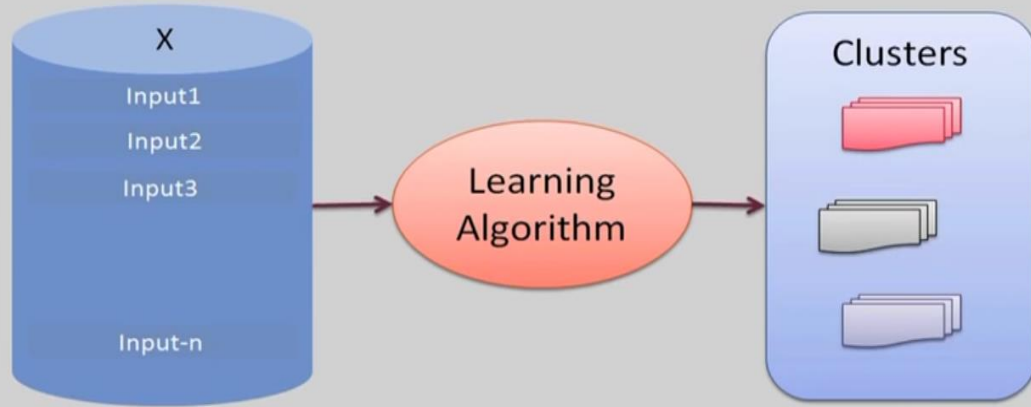
Unsupervised Learning
(I am Self Learner)

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
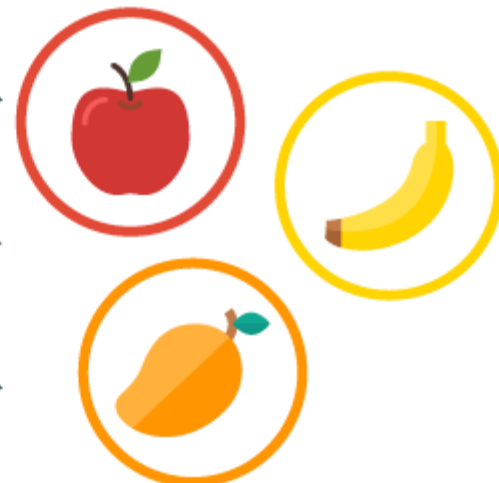
Unsupervised Learning Use cases

A friend invites you to his party where you meet **totally strangers**. Now you will classify them using unsupervised learning (no prior knowledge) and this classification can be on the basis of ***gender, age group, dressing, educational qualification or whatever way you would like.*** Since you didn't have any prior knowledge about people and so you just classified them "on-the-go".

Unsupervised Learning Use cases

Let's suppose you have never seen a Football match before and **by chance watch a video on internet, now you can classify players on the basis of different criterion like Players wearing the same sort of kits are in one class, Players of one style are in one class (players, goal keeper, referee), or on the basis of playing style(attacker or defender) or whatever way you would observe, you can classify it.**

Types of Unsupervised Learning

1. **Clustering**
    - K Means Clustering
    - Hierarchical Clustering
    - Gaussian Mixture Models
    - Genetic Algorithms
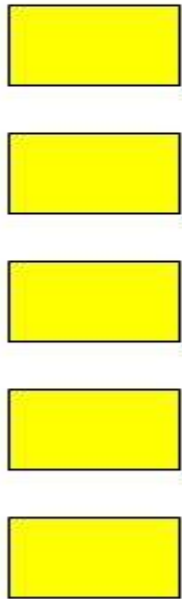   - Artificial Neural Networks

2. **Dimensionality Reduction**
    - Tensor Decomposition
    - Principal Component Analysis
    - Multidimensional statistics
    - Random Projection

3. **Association Rules**
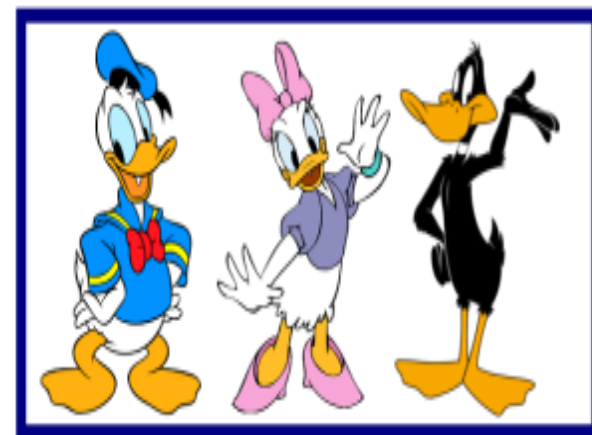
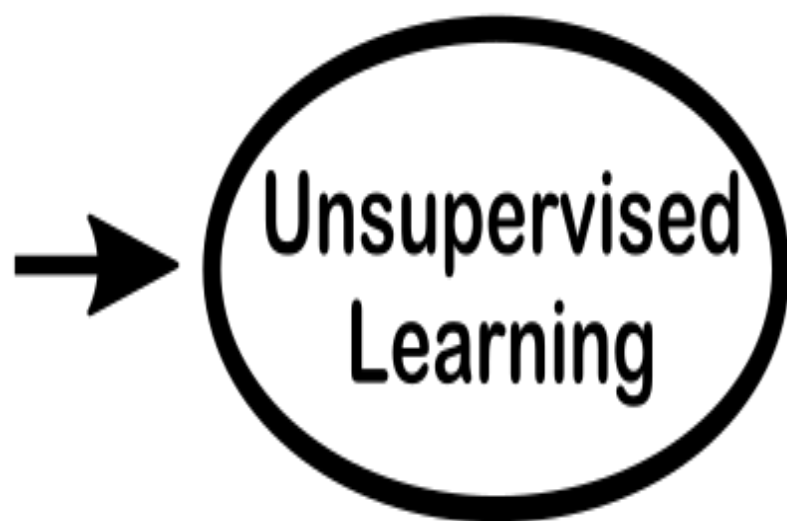# Unsupervised learning: clustering

Raw data

features

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

extract features

group into classes/clust ers

Clusters

No "supervision", we're only given data and want to find natural groupings

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

1.3 Reinforcement Learning

- This method aims at using observations gathered from the *interaction with the environment to take actions that would maximize the reward or minimize the risk.*

- Reinforcement learning algorithm *(called the agent)* continuously learns from the environment in an iterative fashion.

- **Reinforcement learning** can be thought of as a hit and trial method of learning. The machine gets a Reward or Penalty point for each action it performs. If the option is correct, the machine gains the reward point or gets a penalty point in case of a wrong response.

Reinforcement Learning

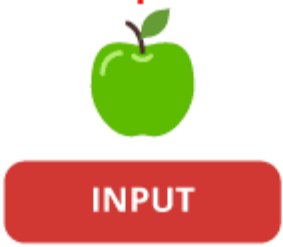**(My life, My rules!**

Rewards

(Hit & Trial))

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
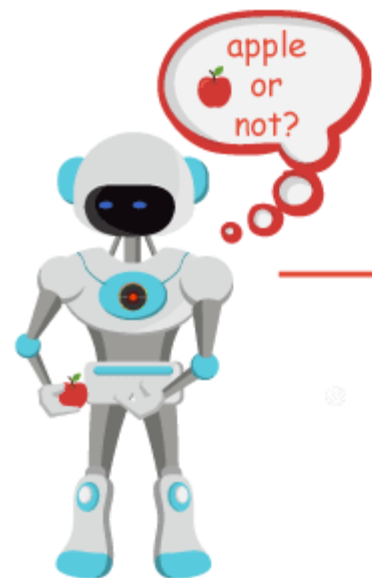
Let see how Pavlov trained his dog using reinforcement training?

Pavlov divided the training of his dog into four stages.

In the first part, *Pavlov gave meat to the dog*, and in response to the meat, the dog started **salivating.**

In the next stage he created a sound with a bell, but this time the dogs did not respond anything.

In the third stage, he tried to train his dog by using the bell and then giving them food. Seeing the food the *dog started salivating*.

Eventually, the dogs started salivating just after hearing the bell, even if the food was not given as the *dog was reinforced that whenever the master will ring the bell, he will get the food.*

1. Before Conditioning

Food → Response → Salivation

Unconditioned Stimulus → Unconditioned Response

2. Before Conditioning

Bell → Response → No Salivation

Neutral stimulus → No Conditioned Response

3. During Conditioning

Bell + Food → Response → Salivation

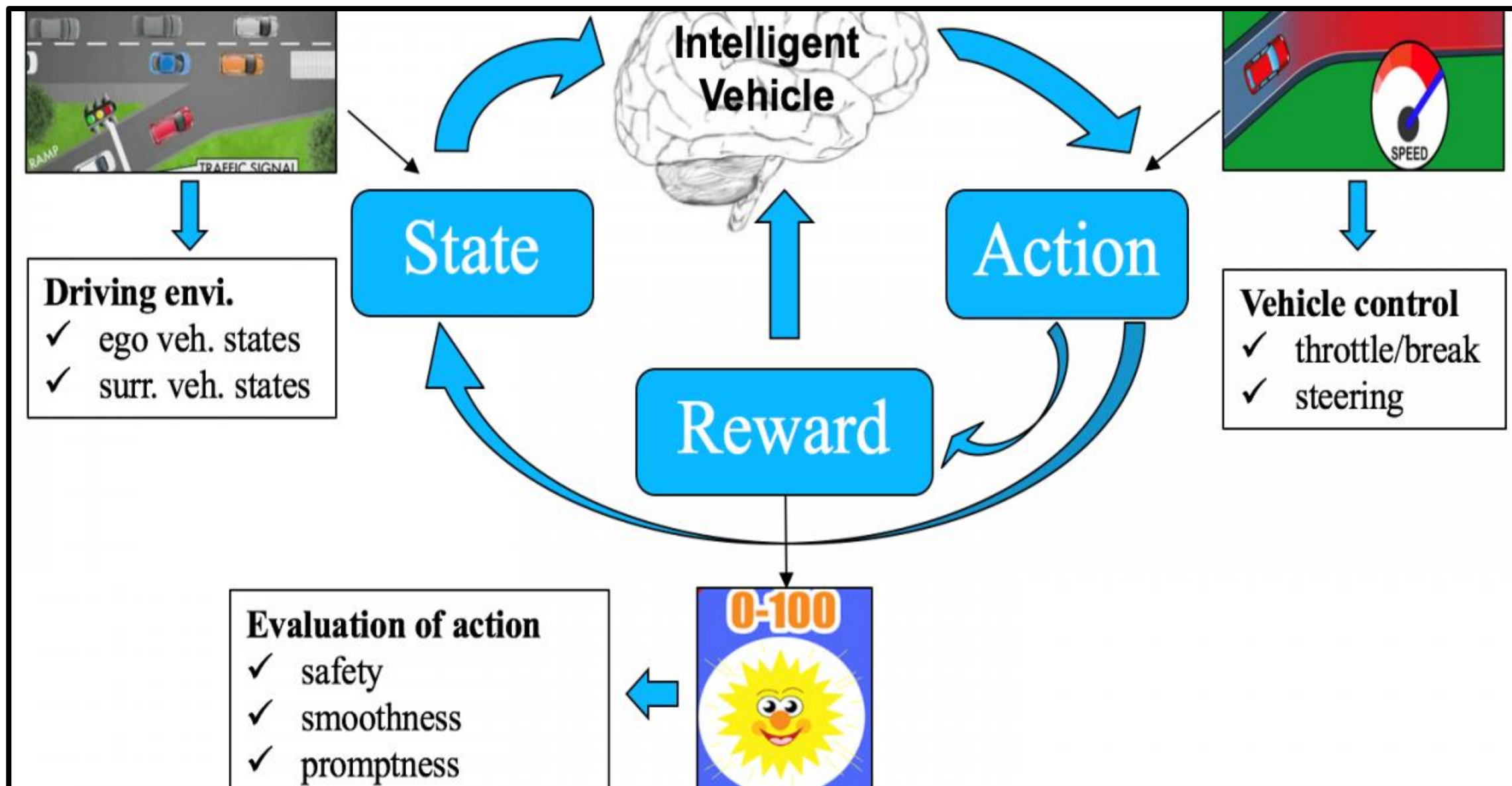Unconditioned Response

4. After Conditioning

Bell → Response → Salivation

Conditioned Stimulus → Conditioned Response

Intelligent Vehicle

State

Action

Reward

Driving envi.
✓ ego veh. states
✓ surr. veh. states

Vehicle control
✓ throttle/break
✓ steering

0-100

Evaluation of action
✓ safety
✓ smoothness
✓ promptness

Types of Reinforcement Learning Algorithms

- **Q-Learning**

- **Temporal Difference (TD)**

- **Deep Adversarial Networks**

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Table1: Examples of Types of Machine Learning Algorithms / Problem Solving Approaches

Type	Model /Algorithm or Task	Usage Examples in Business
Supervised	Neural network	■ Predicting financial results ■ Fraud detection
Supervised	Classification and/or Regression	■ Spam filtering ■ Fraud detection
Supervised	Decision tree	■ Risk assessment ■ Threat management systems ■ Any optimization problem where an exhaustive search is not feasible

Table1: Examples of Types of Machine Learning Algorithms / Problem Solving Approaches

Type	Model /Algorithm or Task	Usage Examples in Business
Unsupervised	Cluster analysis	■ Financial transactions ■ Streaming analytics in IoT ■ Underwriting in insurance
Unsupervised	Pattern recognition	■ Spam detection ■ Biometrics ■ Identity management
Unsupervised	Association rule learning	■ Security and intrusion detection ■ Bioinformatics ■Manufacturing and Assembly

2. Bayesian Learning

1. Bayes Theorem
2. MAP, ML hypotheses
3. **Relation to Concept Learning**
4. MAP learners
5. **Maximum likelihood and Least Squared Error hypothesis**
6. Naive Bayes learner
7. Example: Learning over text data
8. Bayesian belief networks
9. Expectation Maximization algorithm

Use cases

Categorizing News

BUSINESS & ECONOMY
Paying service charge at hotels not mandatory

TECHNOLOGY & SCIENCE
The 'dangers' of being admin of a WhatsApp group

ENTERTAINMENT
This actor stars in Raabta. Guess who?
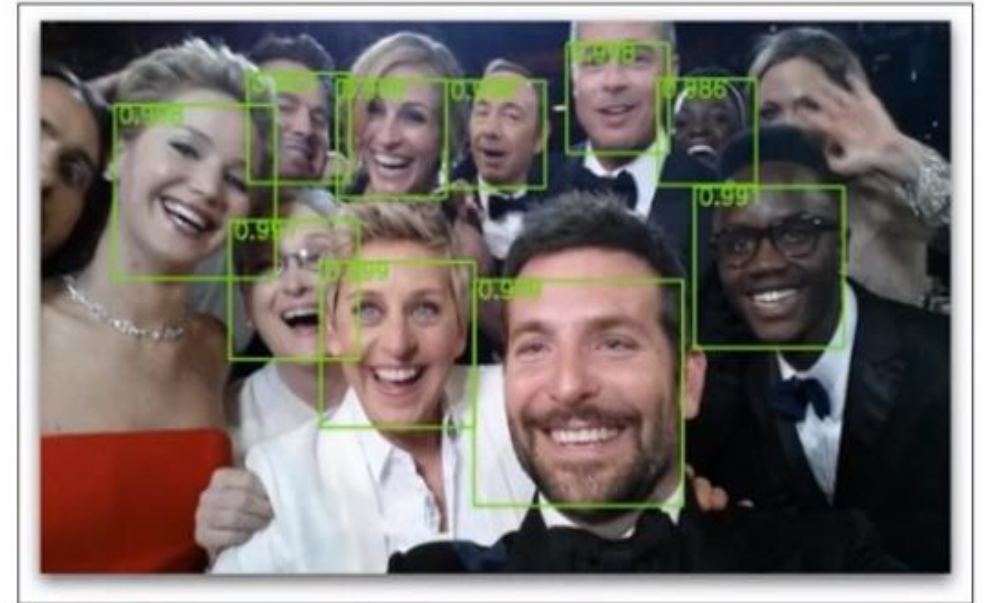
IPL 2017
Preview: Bullish KKR face depleted Lions

INDIA
Why is Aadhaar mandatory for PAN? SC asks Centre

Email Spam Detection

Email Lists
Filtering System
Good Emails   Bad Emails

Face Recognition

Sentiment Analysis

Use cases

- Bayes' theorem is stated mathematically as the following equation

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

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$ is also a conditional probability: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other;

Alternative form

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)}.$$

$P(\neg A)$ is the corresponding probability of the initial degree of belief against A, where $1 - P(A) = P(\neg A)$

$P(B \mid \neg A)$ is the conditional probability or likelihood, is the degree of belief in B, given that the proposition $\neg A$ is true.

Example of Bayes Theorem $P(A \mid B) = P(B \mid A)P(A) / P(B)$

Consider a drug test that is **99 percent sensitive**(the true positive rate) and **99 percent specific** (the true negative rate). If half a percent **(0.5 percent)** of people **use a drug**, what is the probability a random person with a positive test actually is a user?

$$P(\text{User} \mid +) = \frac{P(+ \mid \text{User})P(\text{User})}{P(+)}$$

$$= \frac{P(+ \mid \text{User})P(\text{User})}{P(+ \mid \text{User})P(\text{User}) + P(+ \mid \text{Non-user})P(\text{Non-user})}$$

$$= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995}$$

$$\approx 33.2\%$$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Bayes Theorem : For Hypothesis and Training Data

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

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

- $P(h)$ = 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.

2.2.1 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}:

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

2.2.2 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)

- In this case we can further simplify and need only consider the term **P(D|h) to find the most probable hypothesis.**

- **P(D|h)** is often called the likelihood of the data D given h and any hypothesis that maximizes **P(D|h)** is called a ***Maximum likelihood*** (ML) hypothesis h_{ML}

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

An Example : Cancer Patient Diagnosis

- To illustrate Bayes Rule , Consider a medical diagnosis problem in which there are two alternative hypotheses :

1. That the patient has a particular form of cancer and

2. That the patient does not.

The available data is from a particular laboratory test with two possible outcomes :

+ : **positive**

- : **negative**

Example : **Medical Cancer** Test Details of Patient

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

$P(cancer) =$ 0.008

$P(\neg cancer) =$ 0.992

$P(+|cancer) =$ 0.98

$P(-|cancer) =$ 0.02

$P(+|\neg cancer) =$ 0.03

$P(-|\neg cancer) =$ 0.97

Example : Does patient have cancer or not?

The Maximum a posterior hypothesis for Patient having cancer/no cancer :

$$\textbf{cancer}_{MAP} = P(+|Cancer)\ P(cancer)\ = (0.98)(0.008)\ = \textbf{0.0078}$$

$$\neg \textit{cancer}_{MAP}\ = P(\neg\ |Cancer)\ P(cancer)\ =(0.03)(0.992) = \textbf{0.0298}$$

2.3 Relation to Concept Learning

- 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 **$<<x_1,d_1><x_2,d_2>,………………..<x_m, d_m>>$** where x_i is some instance from X and where **d_i** is the target value of **x_i (i.e $d_i= c(x_i)$).**

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

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

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. d_i = $c(x_i)$)**

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)*:

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

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 consistent with D , we have

    P(h|D) = (1*1/|H|)/(|VS$_{H,D}$|/|H|)

    = 1/|VS$_{H,D}$|

- Where P(D) = |VS$_{H,D}$|/ |H| and |VS$_{H,D}$| is the subset of hypotheses from H that are consistent with **D**.

To Summarize

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

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

2.4 MAP hypothesis and Consistent Learners
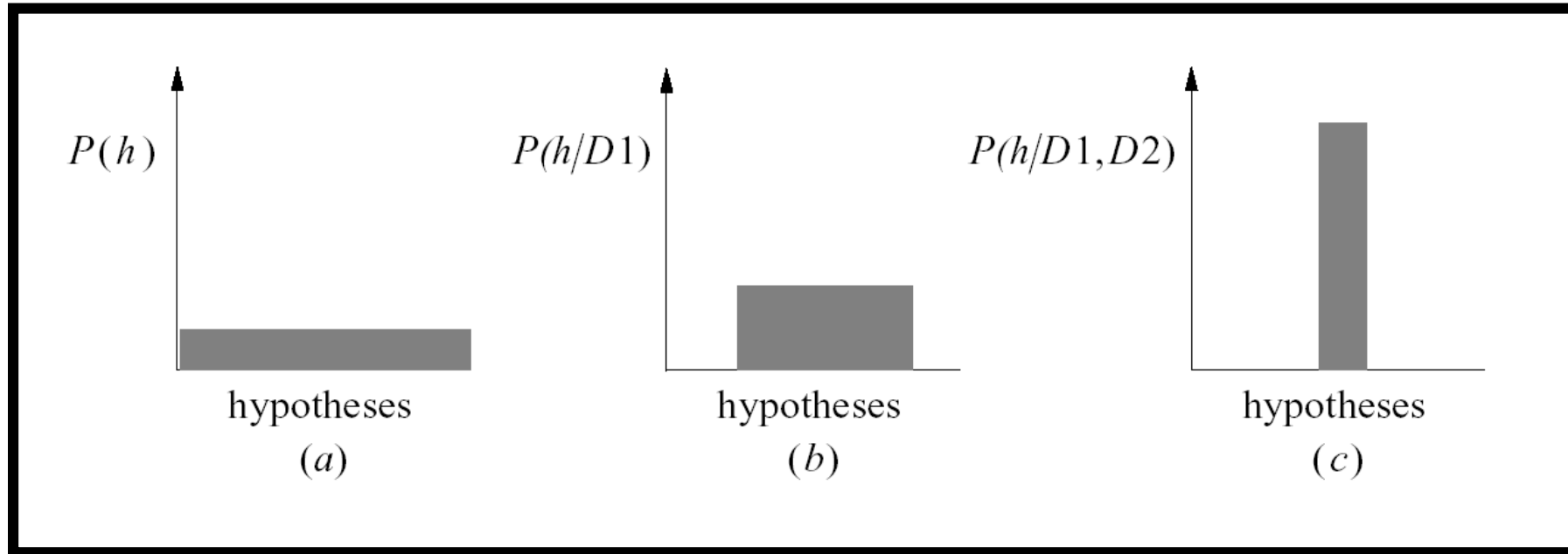
# Characterizing Learning Algorithms by Equivalent MAP Learners

Inductive system

Training examples D →

Hypothesis space H →

Candidate Elimination Algorithm

→ Output hypotheses

Equivalent Bayesian inference system

Training examples D →

Hypothesis space H →

$P(h)$ uniform
$P(D|h) = 0$ if inconsistent,
$= 1$ if consistent →

Brute force MAP learner

→ Output hypotheses
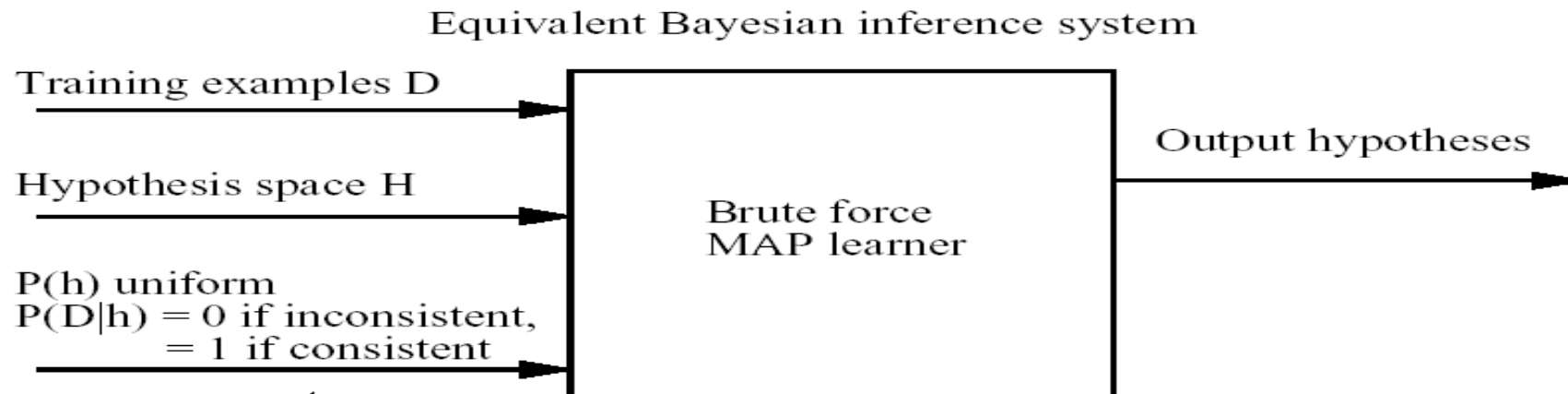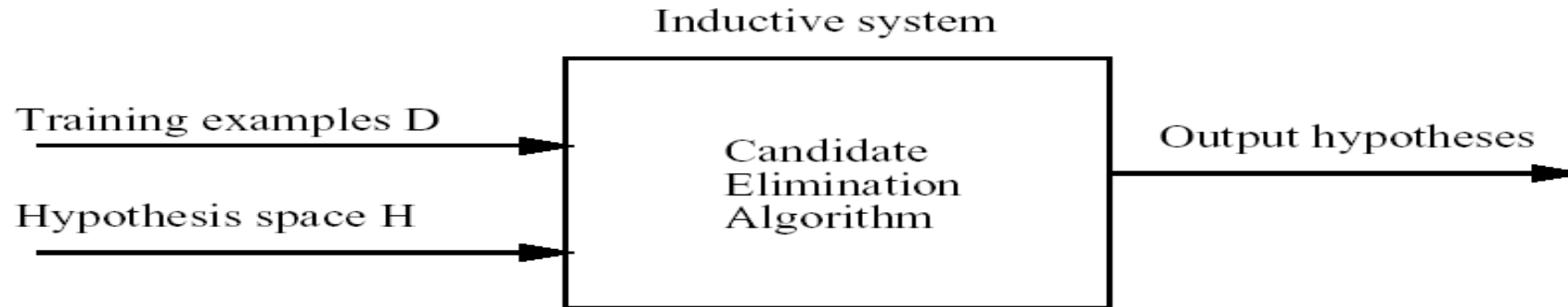
Prior assumptions made explicit

2.5 Maximum likelihood and Least Squared Error 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 hypothesis**.

Learning A Real Valued Function
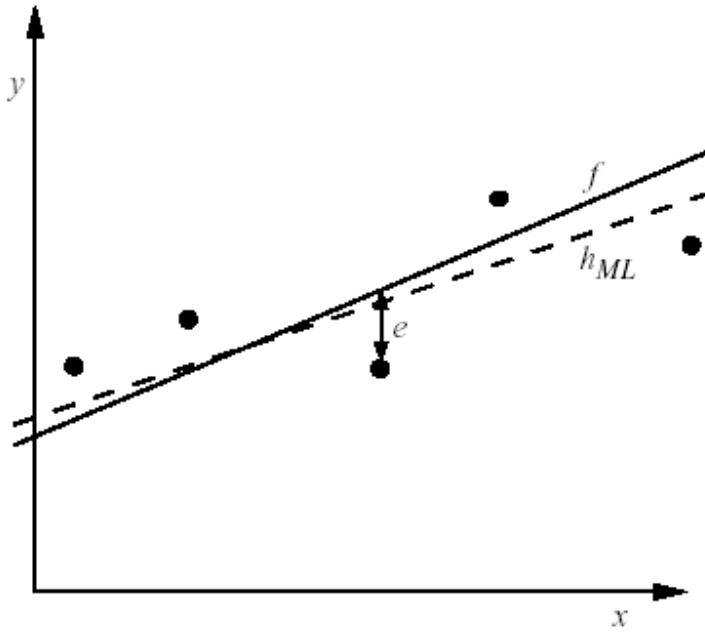
- Consider any real-valued target function f

  Training examples $<x_i, d_i>$, 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:

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

2.6 Naive Bayes classifier /Bayes Rule

Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C

The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some *k* as follows:

$$\hat{y} = \operatorname*{argmax}_{k \in \{1,\ldots,K\}} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

ಡಾ|| ತಾ₅ಗೋಾಜು ಜಿ.ಎಸ್

Example

- **Example: Play Tennis**

PlayTennis: training examples

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

PlayTennis: training examples

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(Temperature|Play)

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

PlayTennis: training examples

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

PlayTennis: training examples

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

P(Humidity|Play)

Humidity	Play=*Yes*	Play=N*o*
High	3/9	4/5
Normal	6/9	1/5

Learning Phase

PlayTennis: training examples

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

P(Wind|Play)

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

54

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=N*o*
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(Play=*Yes*) = 9/14

P(Play=*No)* = 5/14

Example

- **Test Phase**
  - Given a new instance,

    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=S*unny*|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*|**x′**): [P(*Sunny*|*Yes*)P(*Cool*|*Yes*)P(*High*|*Yes*)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

    P(*No*|**x′**): [P(*Sunny*|*No*) P(*Cool*|*No*)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

    Given the fact P(*Yes*|**x′**) < P(*No*|**x′**), we label **x′** to be *"No"*.

Event Models

- The assumptions on distributions of features are called the *event model* of the Naive Bayes classifier.

- **For discrete features** like the ones encountered in document classification (include spam filtering),   multinomial and Bernoulli distributions are popular.

- **For Continuous feature , Gaussian naive Bayes distributions is popular.**

1. Gaussian naive Bayes

- When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

- Then, the probability *distribution* of v given a class $C_k, p(x = v \mid C_k)$ can be computed by plugging V into the equation for a Normal distribution parameterized by μ_k and σ_k^2.

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

2. Multinomial naive Bayes

Its is used when we have **discrete data** (e.g. movie ratings ranging 1 and 5 as each rating will have certain **frequency** to represent). In text learning we have the count of each word to predict the class or label.
The Multinomial Naive Bayes's conditional distribution is:

The term frequencies can then be used to compute the maximum-likelihood estimate based on the training data to estimate the class-conditional probabilities in the multinomial model:

$$\hat{P}(x_i \mid \omega_j) = \frac{\sum tf(x_i, d \in \omega_j) + \alpha}{\sum N_{d \in \omega j} + \alpha \cdot V}$$

where

- x_i: A word from the feature vector $\mathbf{x}$ of a particular sample.
- $\sum tf(x_i, d \in \omega_j)$: The sum of raw term frequencies of word x_i from all documents in the training sample that belong to class ω_j.
- $\sum N_{d \in \omega j}$: The sum of all term frequencies in the training dataset for class ω_j.
- α: An additive smoothing parameter ($\alpha = 1$ for Laplace smoothing).
- V: The size of the vocabulary (number of different words in the training set).

The class-conditional probability of encountering the text $\mathbf{x}$ can be calculated as the product from the likelihoods of the individual words (under the *naive* assumption of conditional independence).

$$P(\mathbf{x} \mid \omega_j) = P(x_1 \mid \omega_j) \cdot P(x_2 \mid \omega_j) \cdot \ldots \cdot P(x_n \mid \omega_j) = \prod^{m} P(x_i \mid \omega_j)$$

3. Bernoulli naive Bayes

It assumes that all our features are binary such that they take only two values. Means **0s** can represent "word does not occur in the document" and **1s** as "word occurs in the document" .

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

3. Bernoulli naive Bayes

It assumes that all our features are binary such that they take only two values. Means **0s** can represent "word does not occur in the document" and **1s** as "word occurs in the document" .

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Lab Program 5:

- **Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.**

- **Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Libraries can be used to write the program. Calculate the accuracy, precision, and recall for your data set.**

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 wk. 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$_k$ | 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***

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

n_k ← number of times word w_k occurs in ***Text$_j$*** ಉಪನ್ಯಾಸ ಸೌಜನ್ಯ ಜಿ.ಎಸ್

S2:CLASSIFY_NAIVE_BAYES_TEXT (*Doc*)

- *positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*

- Return v_{NB} where

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

Twenty NewsGroups

- Given 1000 training documents from each group Learn to classify new documents according to which newsgroup it came from

comp.graphics	**misc.forsale**	**alt.atheism**	**sci.space**
comp.os.ms-windows.misc	**rec.autos**	**soc.religion.christian**	**sci.crypt**
comp.sys.ibm.pc.hardware	**rec.motorcycles**	**talk.religion.misc**	**sci.electronics**
comp.sys.mac.hardware	**rec.sport.baseball**	**talk.politics.mideast**	**sci.med**
comp.windows.x	**rec.sport.hockey**	**talk.politics.misc**	
		talk.politics.guns	

- Naive Bayes: 89% classification accuracy

Learning Curve for 20 Newsgroups
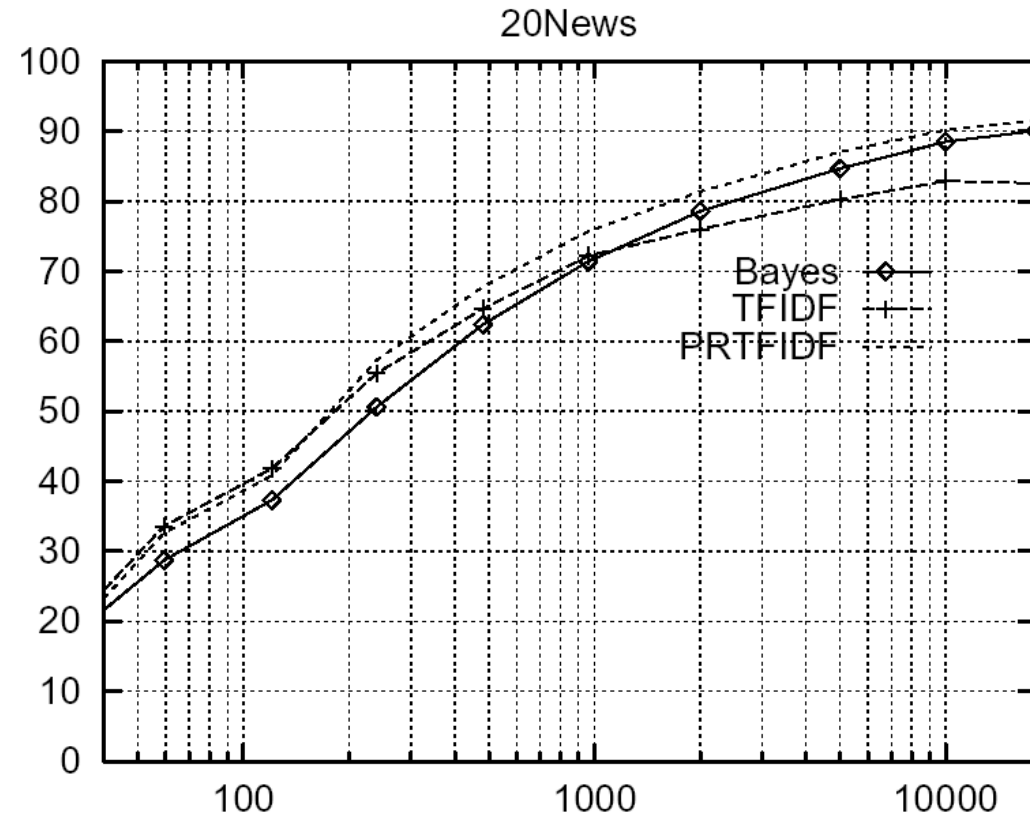
- Accuracy vs. Training set size (1/3 withheld for test)

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

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

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

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*)

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Step 2: Calculating the probabilities

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

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

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

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

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.

ಡಾ‖ ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Source Code

- Bayesian Belief networks *describe conditional independence* among *subsets* of variables

Conditional Independence

- **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

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

more compactly, we write

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

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

$$P(Thunder \mid Rain, Lightning) = P(Thunder \mid Lightning)$$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
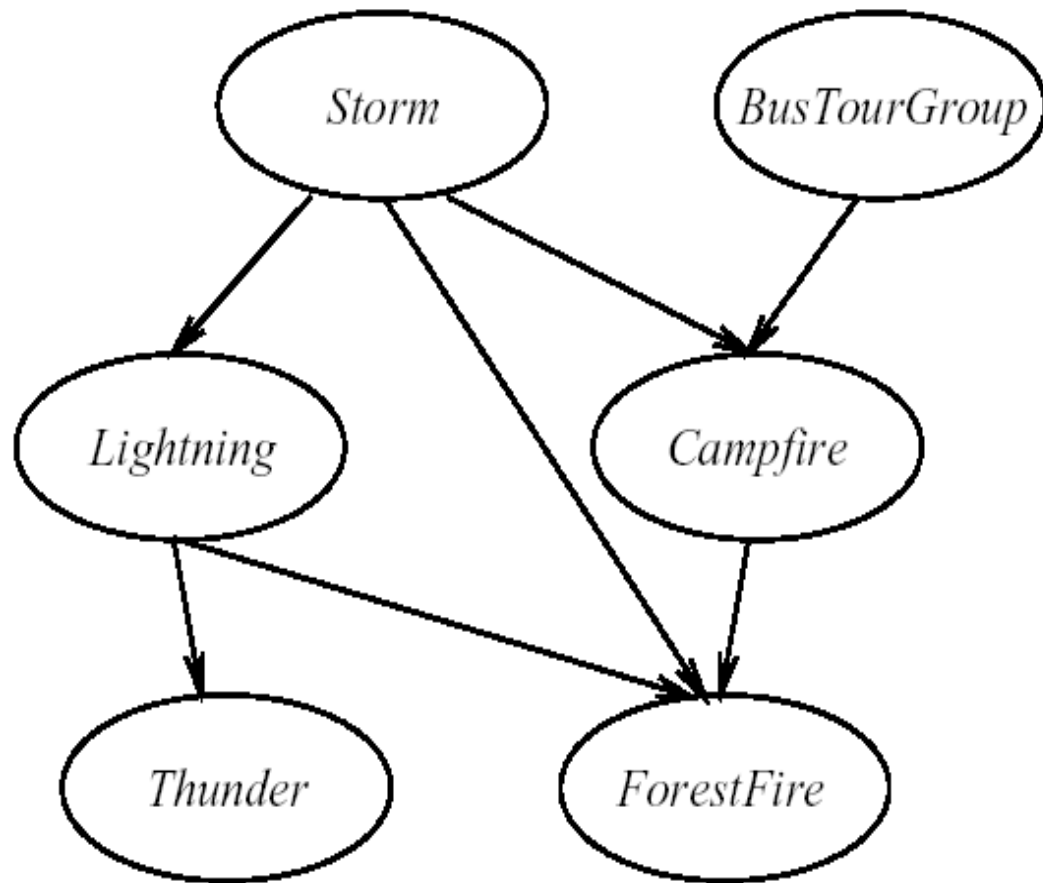
	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

Campfire

- Network represents a set of conditional independence assertions:
  - Each node is asserted to be conditionally independent of its non descendants, given its immediate predecessors.
  - Directed acyclic graph

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Bayesian Belief Network (2/2)

- Represents joint probability distribution over all variables
  - e.g., *P*(*Storm, BusTourGroup, . . . , ForestFire*)
  - in general,

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

  where *Parents*(Y_i) denotes immediate predecessors of Y_i in graph

  - so, joint distribution is fully defined by graph, plus the P(y_i|*Parents*(Y_i))

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
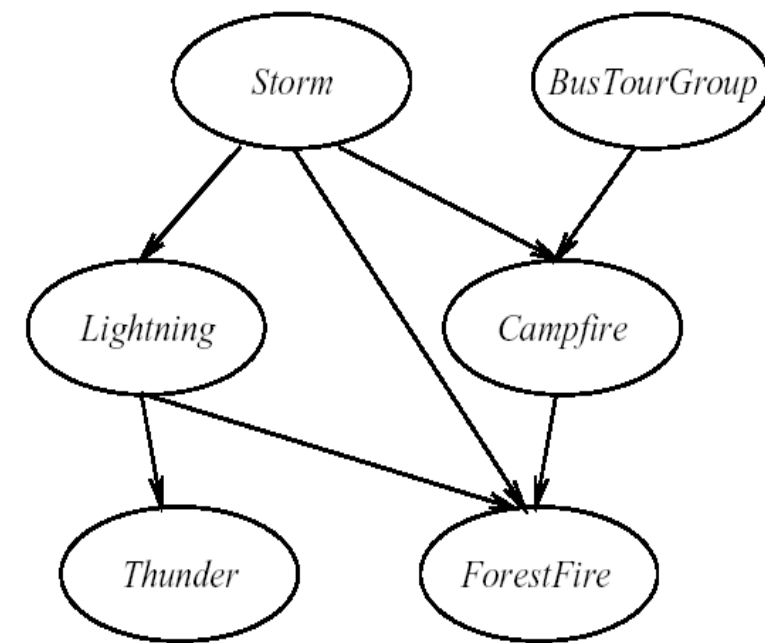
Prob	E
.4	T
.6	F

Prob	S
.15	T
.85	F

Prob	E	S	B
.45	T	T	T
.55	T	T	F
.05	T	F	T
.95	T	F	F
.95	F	T	T
.05	F	T	F
.55	F	F	T
.45	F	F	F
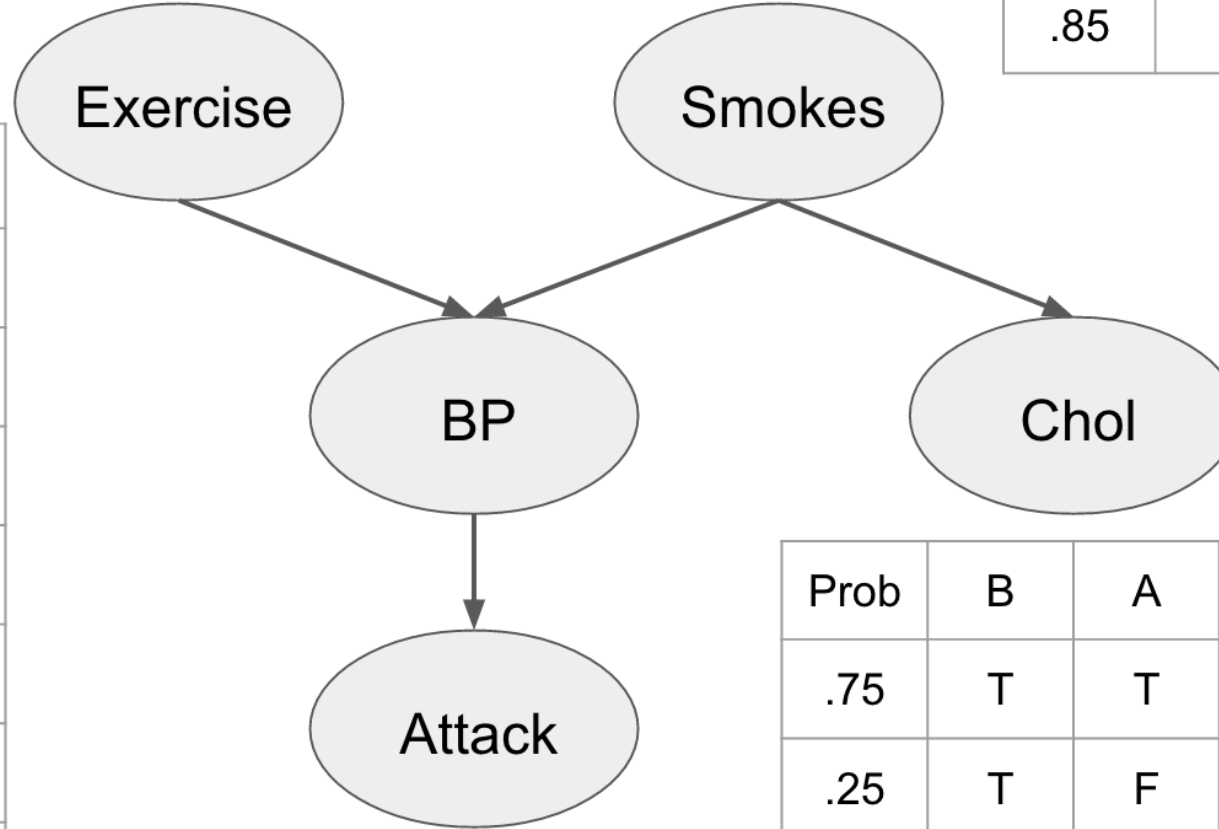
Prob	S	C
.8	T	T
.2	T	F
.4	F	T
.6	F	F

Prob	B	A
.75	T	T
.25	T	F
.05	F	T
.95	F	F

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

- **Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library API.**

Source Code

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

2.8 EM Algorithm

Generating Data from Mixture of *k* Gaussians
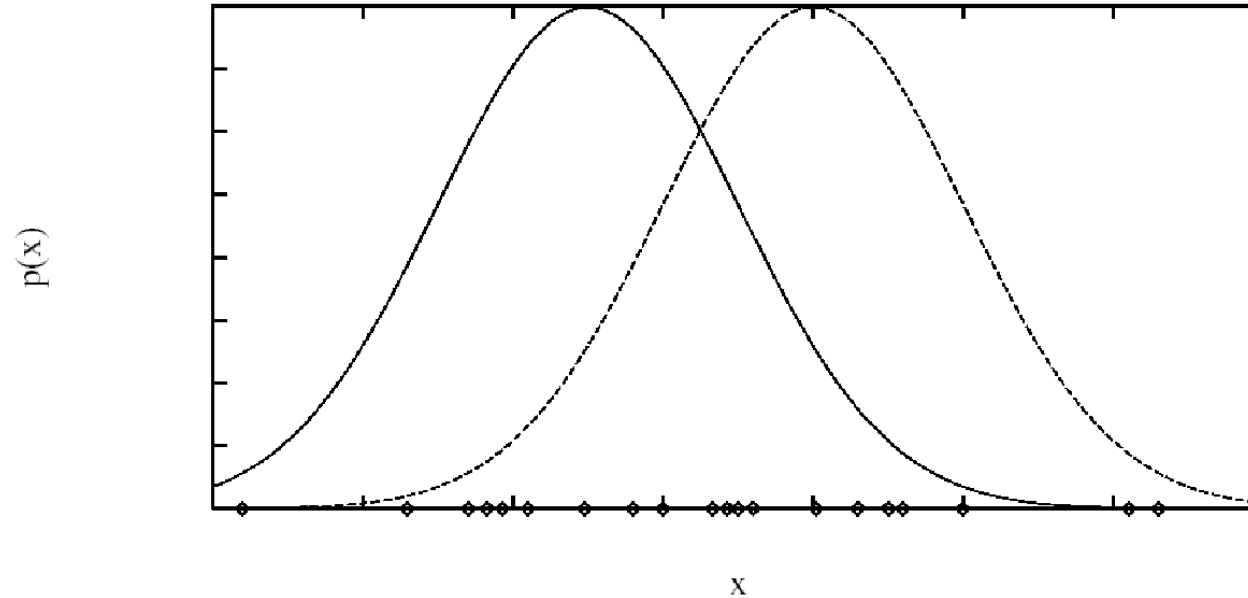
- Each instance x generated by
    1. Choosing one of the *k* Gaussians with uniform probability
    2. Generating an instance at random according to that Gaussian

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Gaussian Distribution

❏ Univariate Gaussian Distribution

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^2}{2\sigma^2}}$$

mean   **variance**

❏ Multi-Variate Gaussian Distribution

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

mean   **covariance**

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Gaussian Mixtures

❑ Linear super-position of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

Number of Gaussians

Mixing coefficient: weightage for each Gaussian dist.
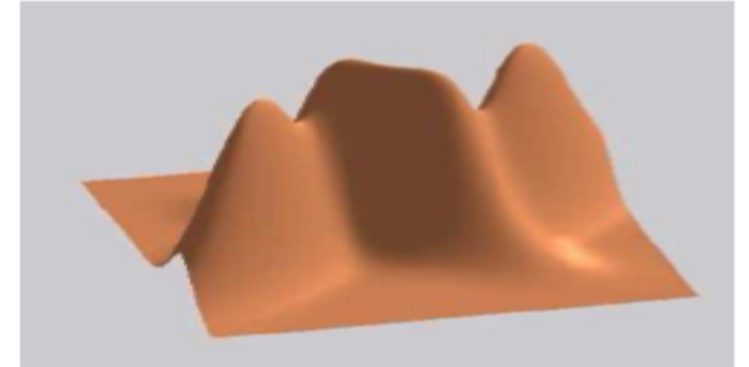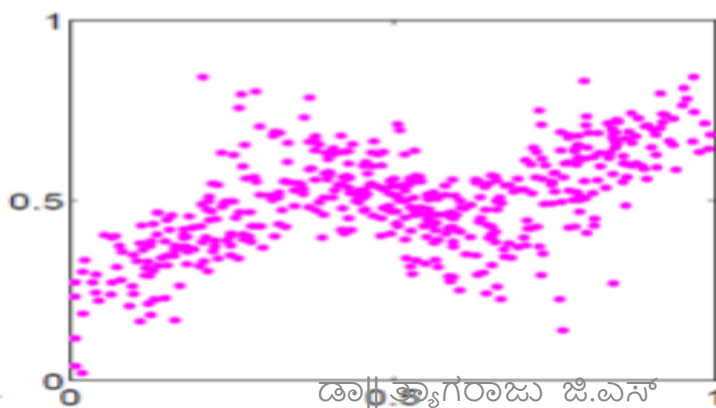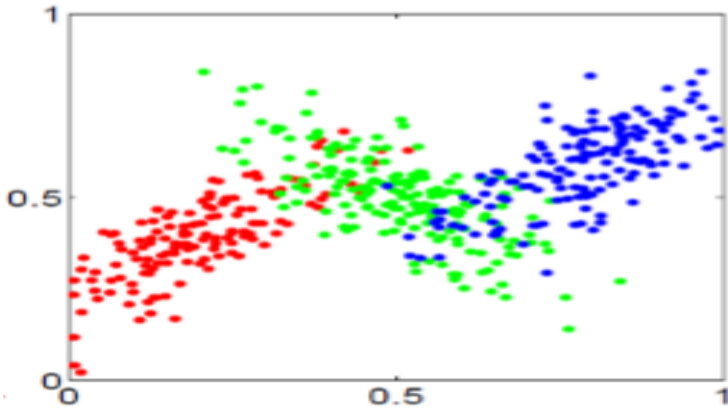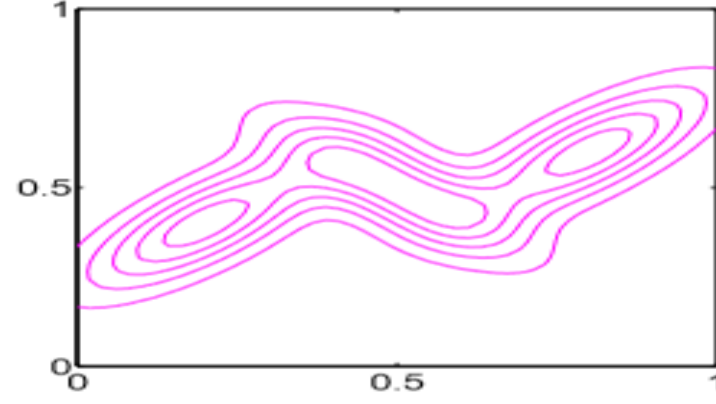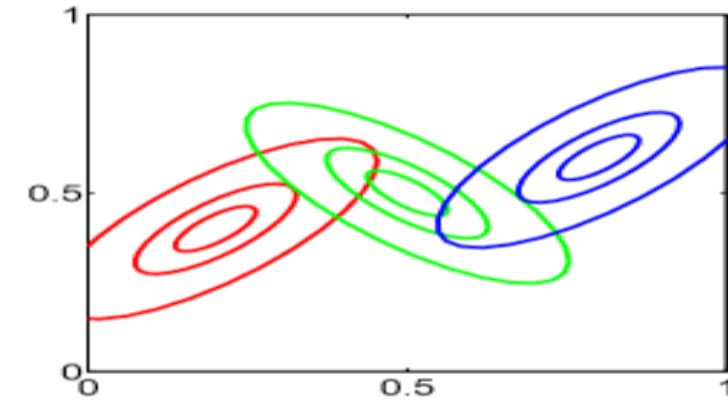
GMM : Example
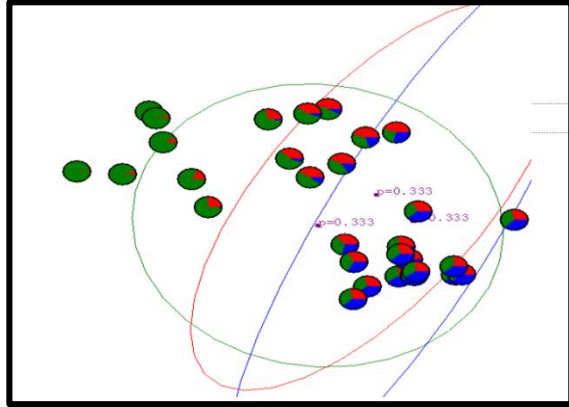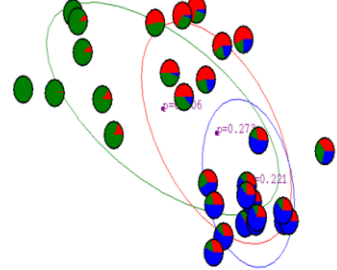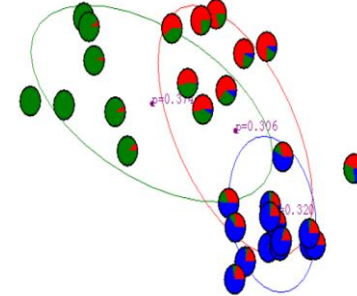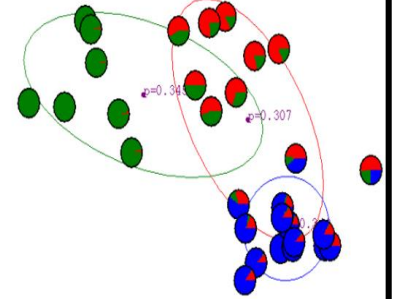
ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
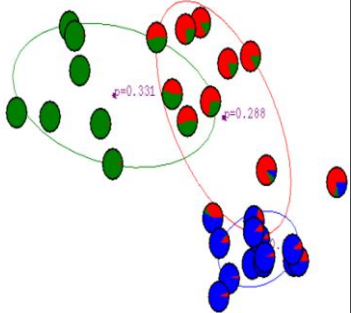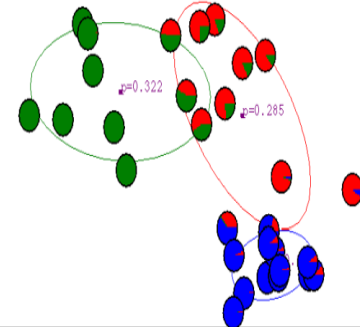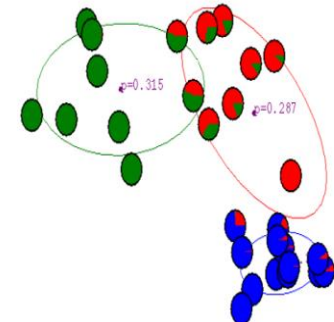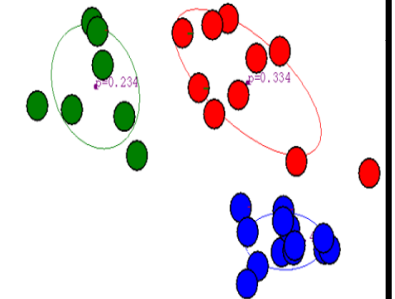
Expectation Maximization (EM) Algorithm

- When to use:
  - Filling in missing data in samples
  - Unsupervised learning of clusters
  - Semi-supervised classification and clustering

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Expectation Maximization (EM) Algorithm

- EM is typically used to compute maximum likelihood estimates given incomplete samples.

- The EM algorithm estimates the parameters of a model iteratively.
  - Starting from some initial guess, each iteration consists of
    - an E step (Expectation step)
    - an M step (Maximization step)

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

EM Algorithm

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

- 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.

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

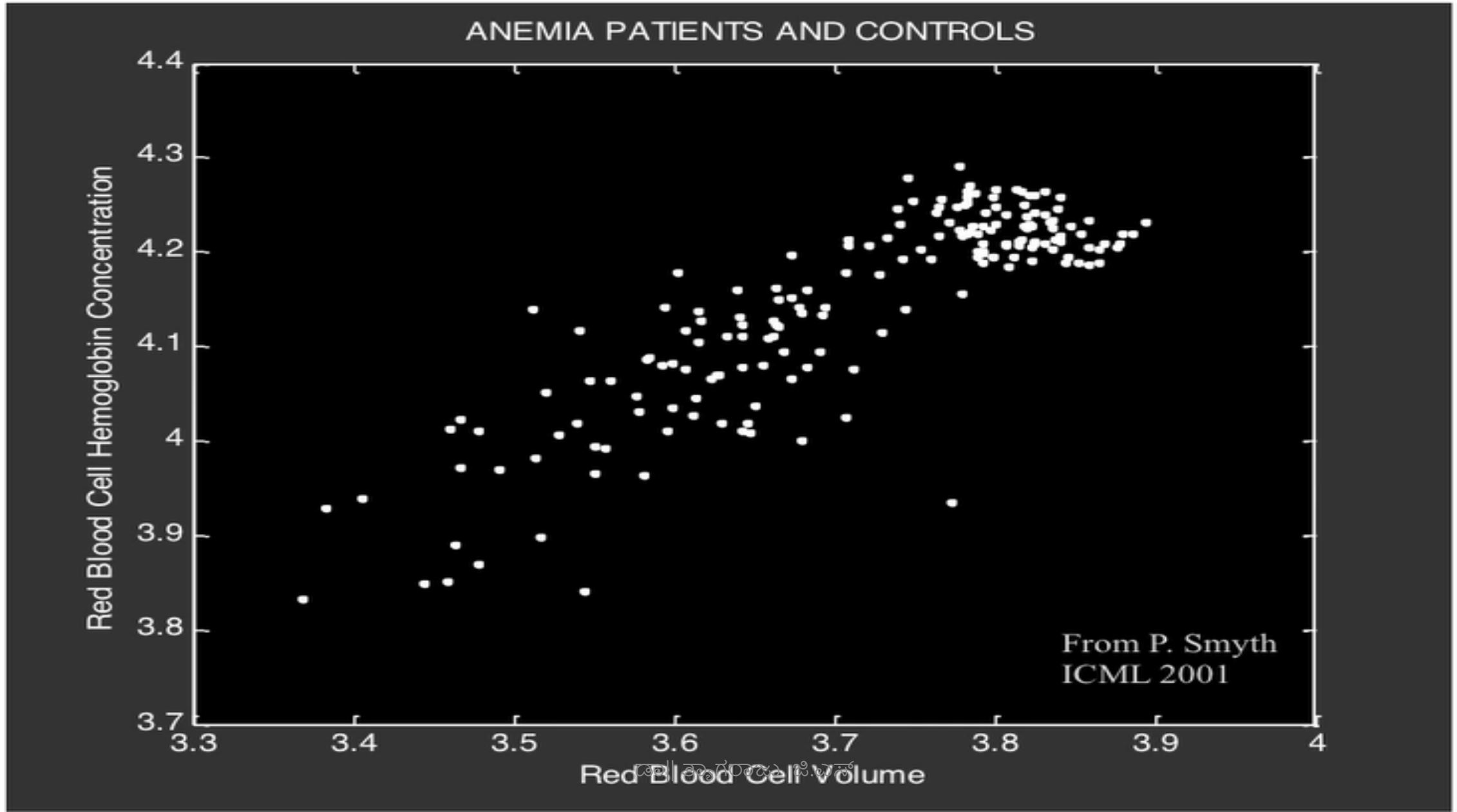$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\Sigma_{n=1}^2 e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

  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 $h' = <\mu'_1, \mu'_2>$.

$$\mu'_j \leftarrow \frac{\Sigma_{i=1}^m E[z_{ij}] \; x_i}{m}$$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

GMM :
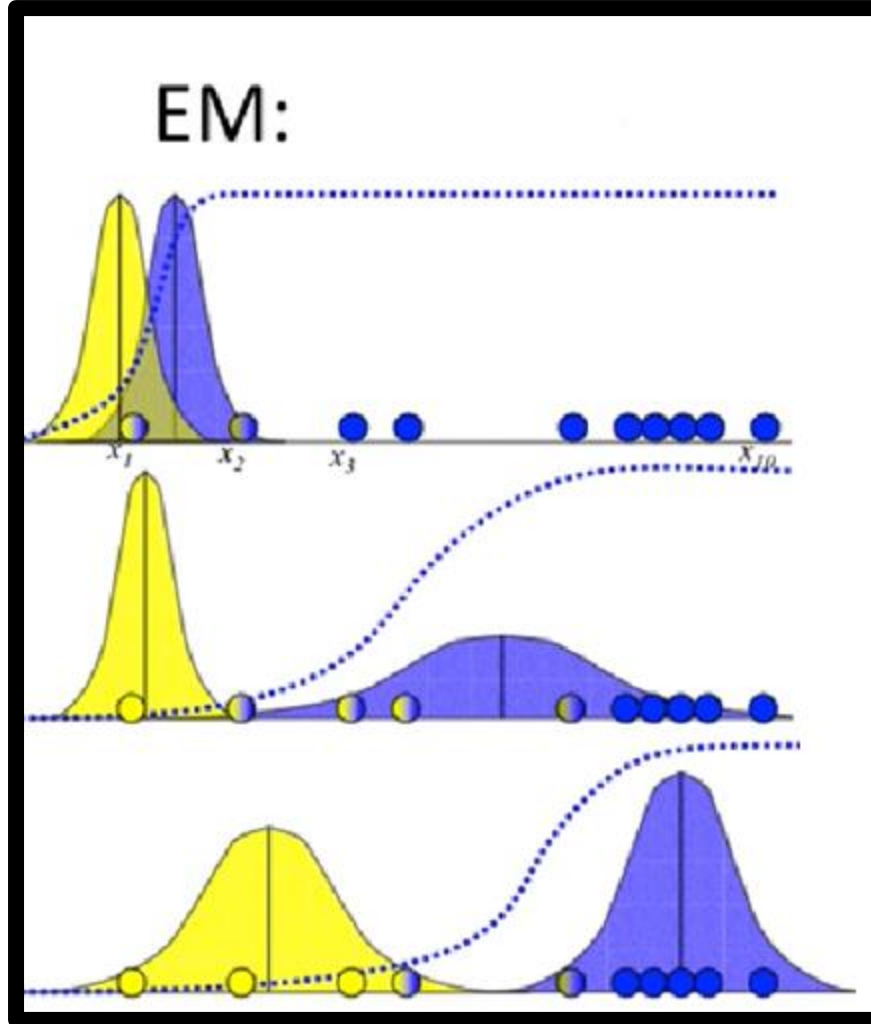
GMM : Example2

GMM : Example2

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

GMM : Example2

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

K Means Algorithm

Algorithm 1 Basic K-means Algorithm.

1: Select K points as the initial centroids.
2: **repeat**
3:   Form K clusters by assigning all points to the closest centroid.
4:   Recompute the centroid of each cluster.
5: **until** The centroids don't change
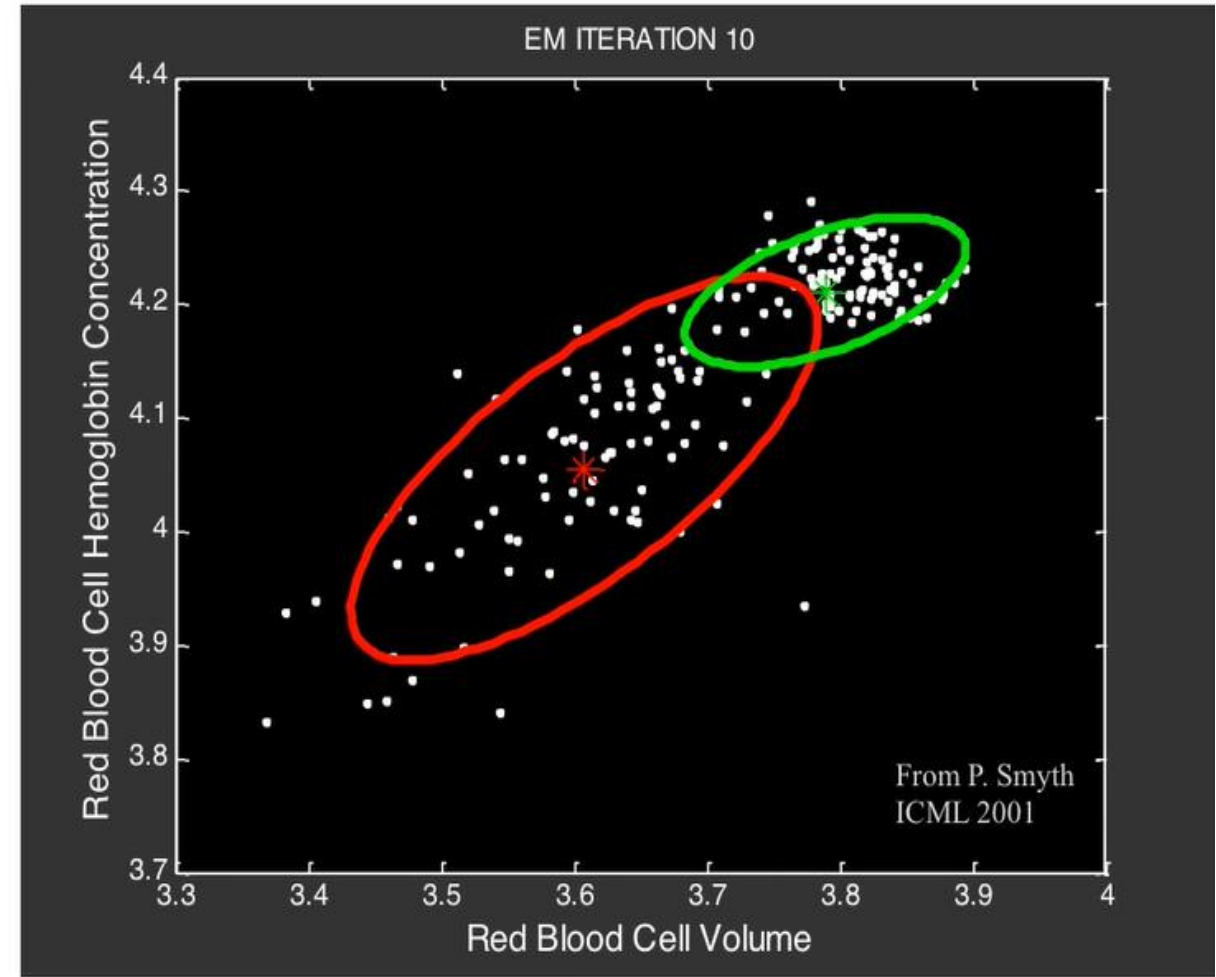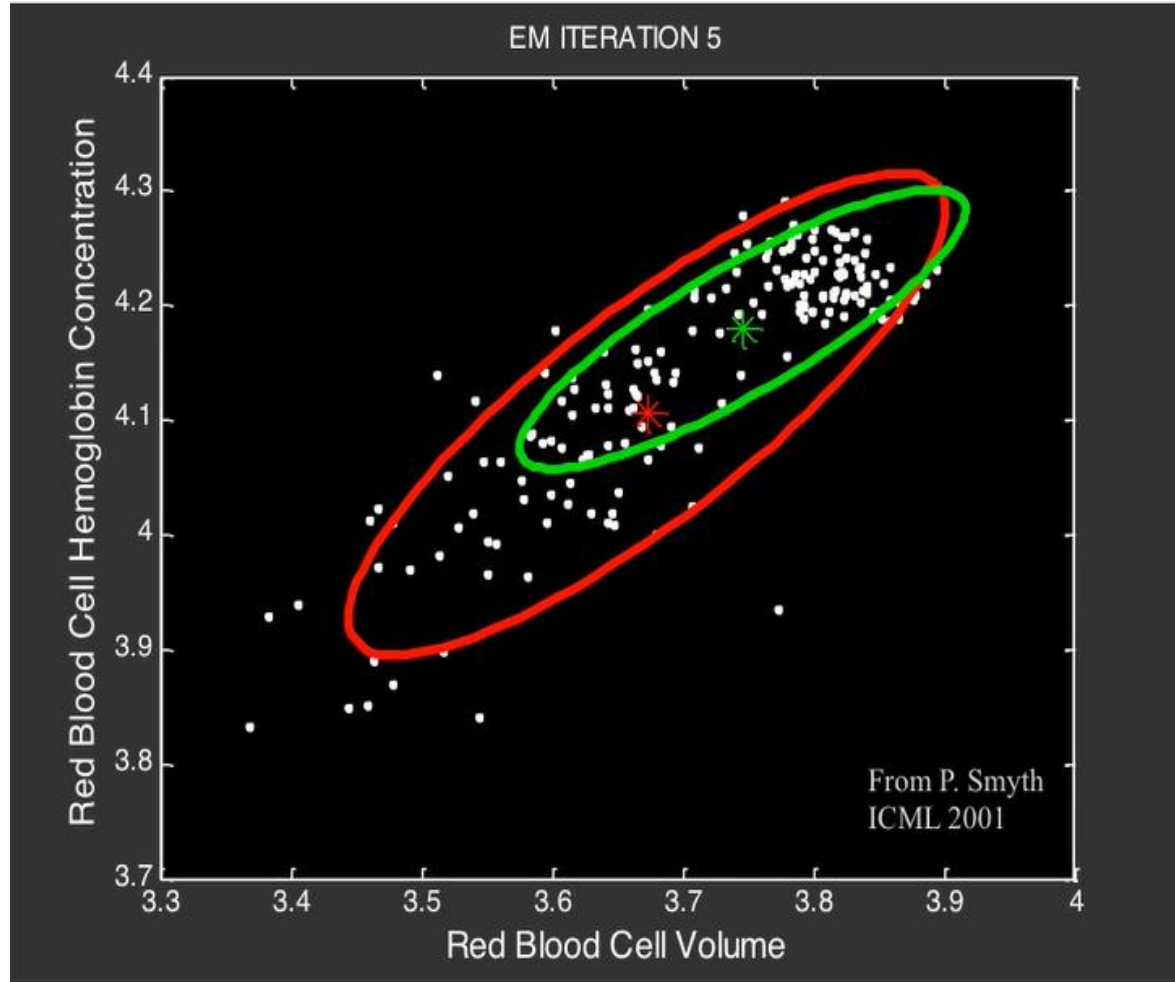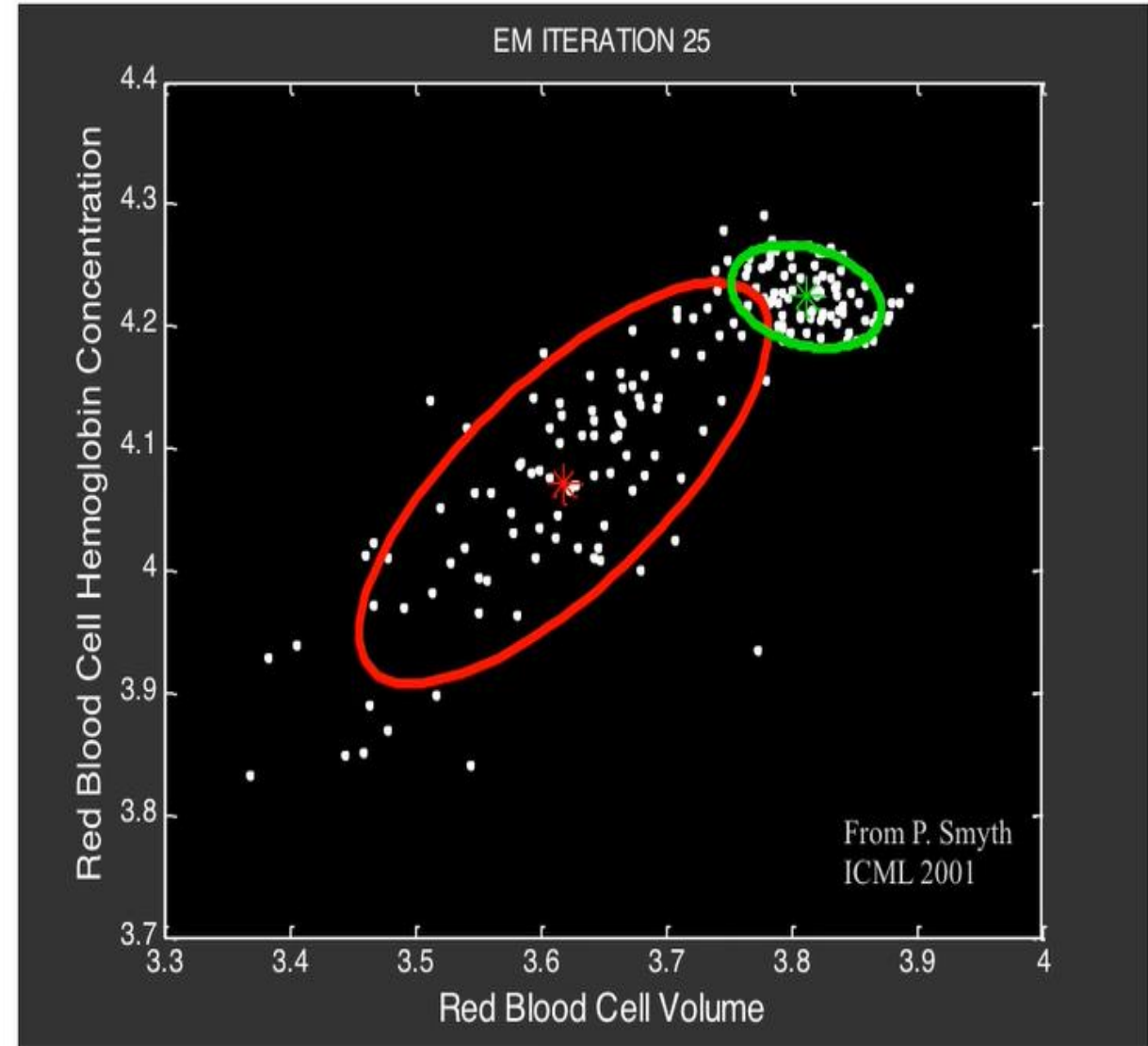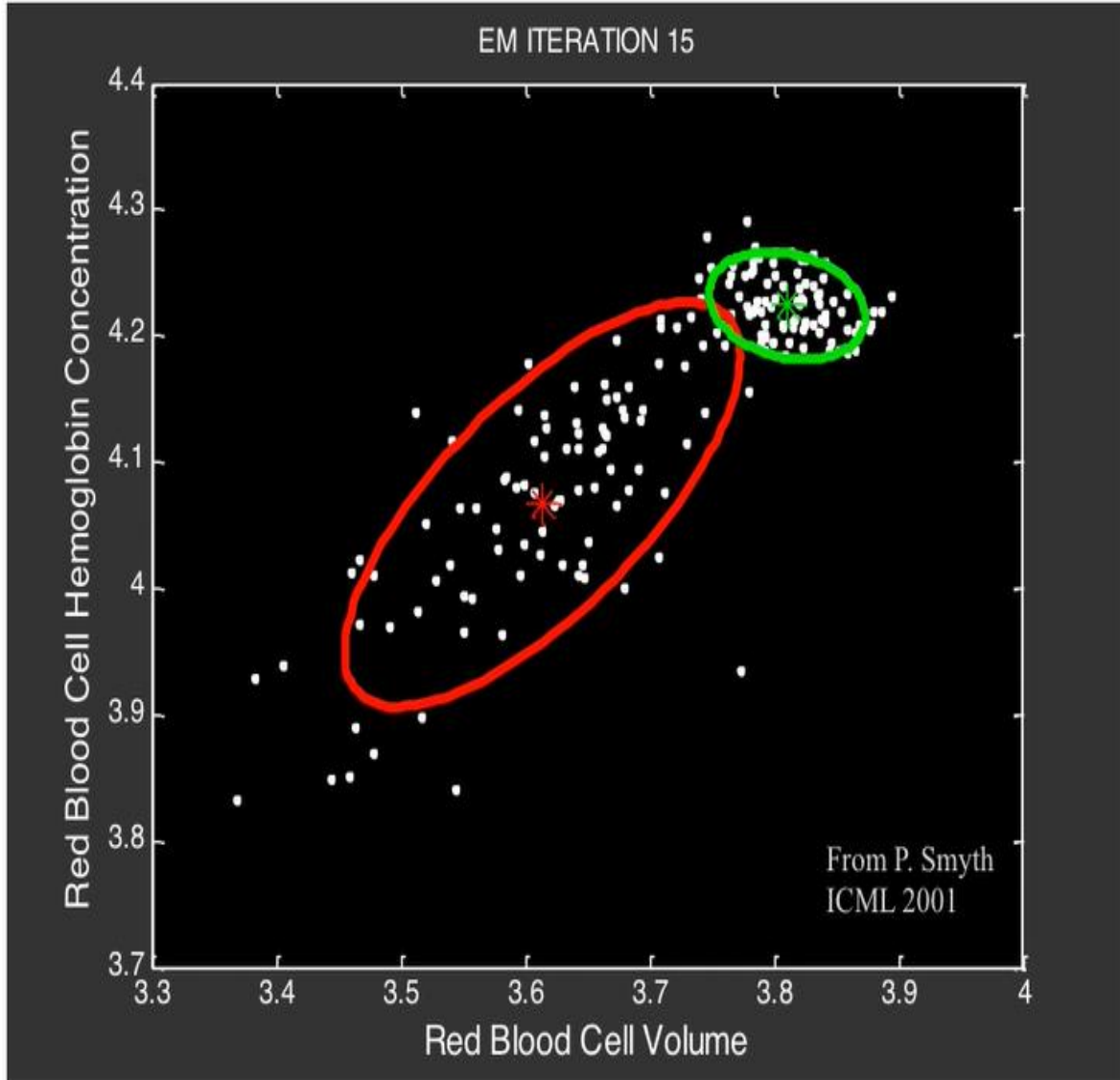
- 1**. The sample space is initially partitioned into K clusters and the observations are randomly assigned to the clusters.**

- 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**

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

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'*

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Euclidean Distance

$$d(i,j) = \sqrt{|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2}$$

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

$$d_E(O, A) = \sqrt{3^2 + 4^2} = 5$$
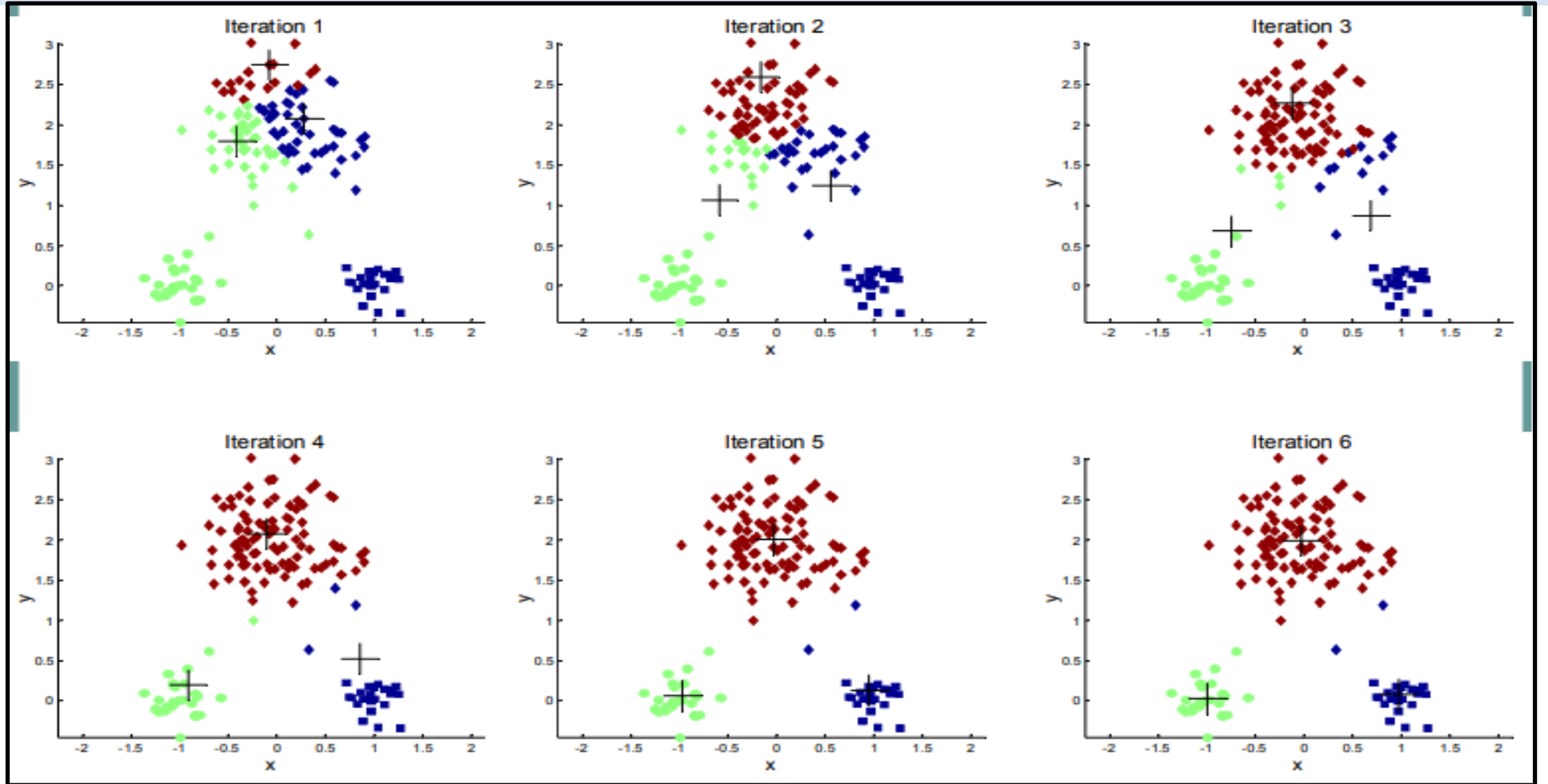
ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

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)

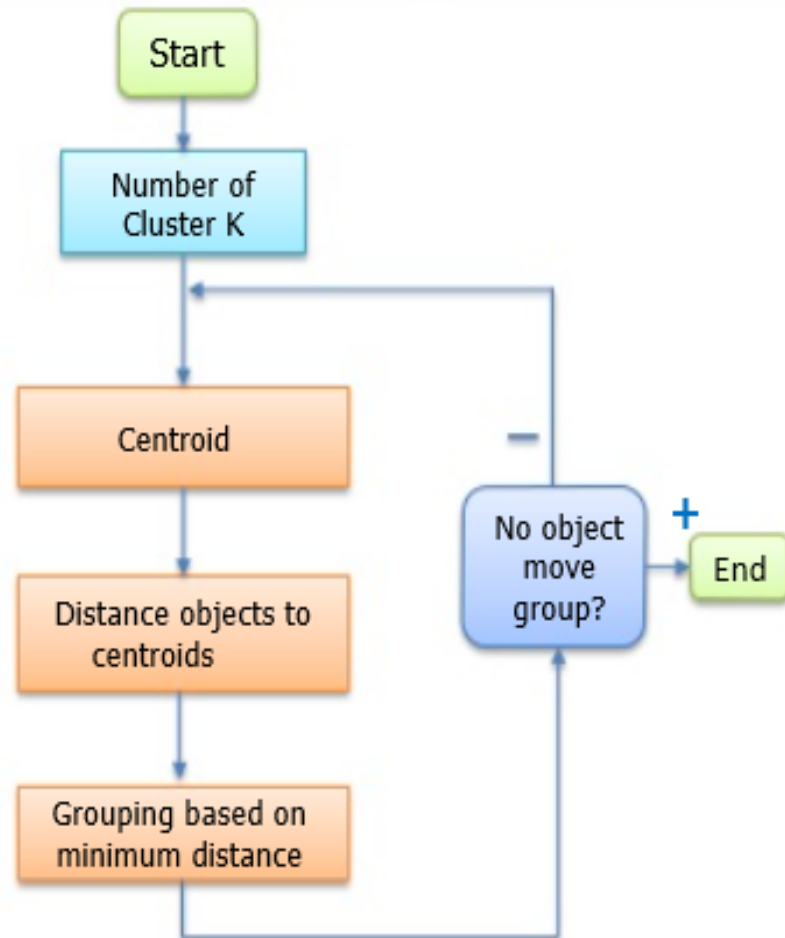$$CP = \left(\frac{2+5+8}{3}, \frac{4+2+9}{3} \right) = (5,5)$$

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

Examples of K Means

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

How the K-Mean Clustering algorithm works?

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

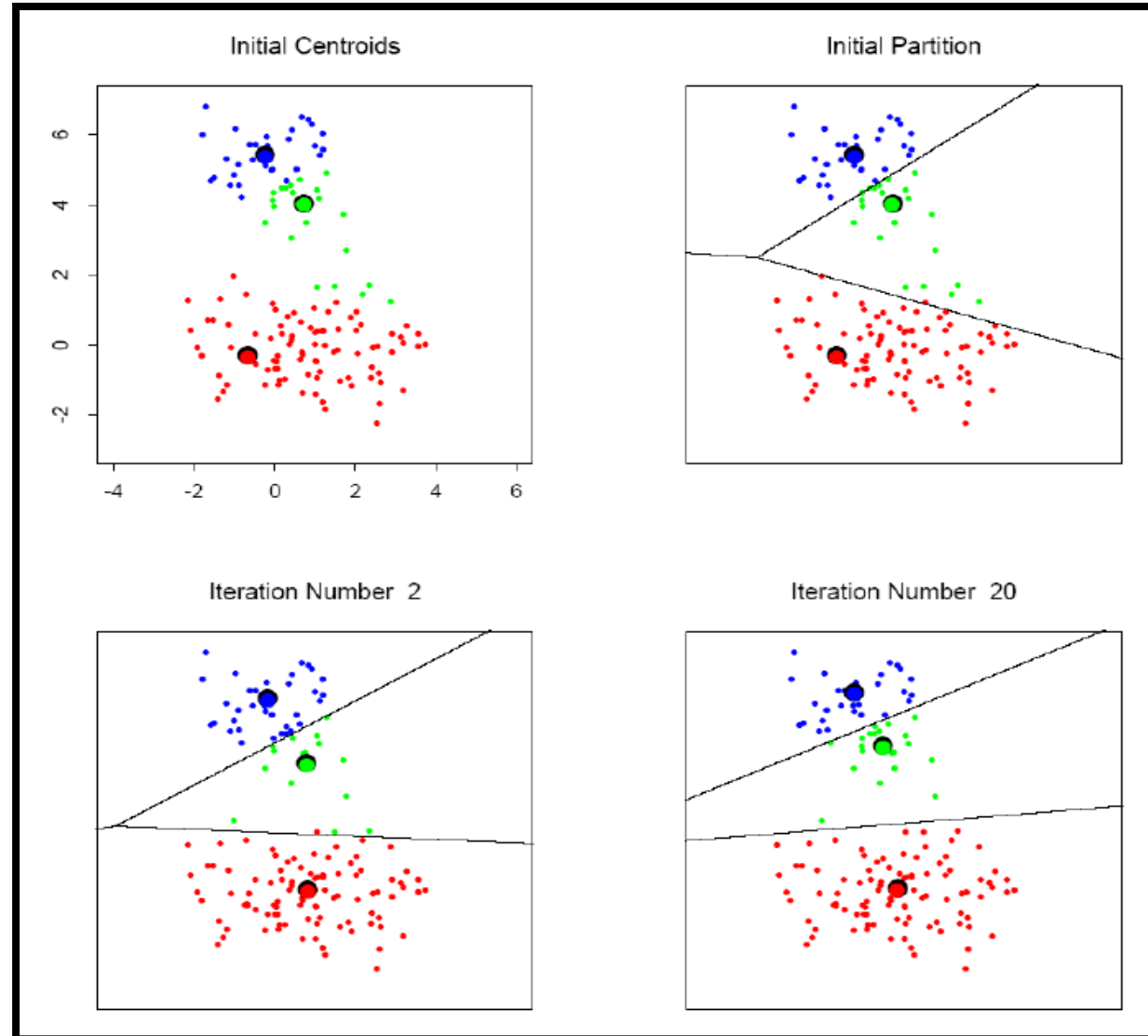ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್

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

Lab Program 8

- Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes/API in the program.

Source Code

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್