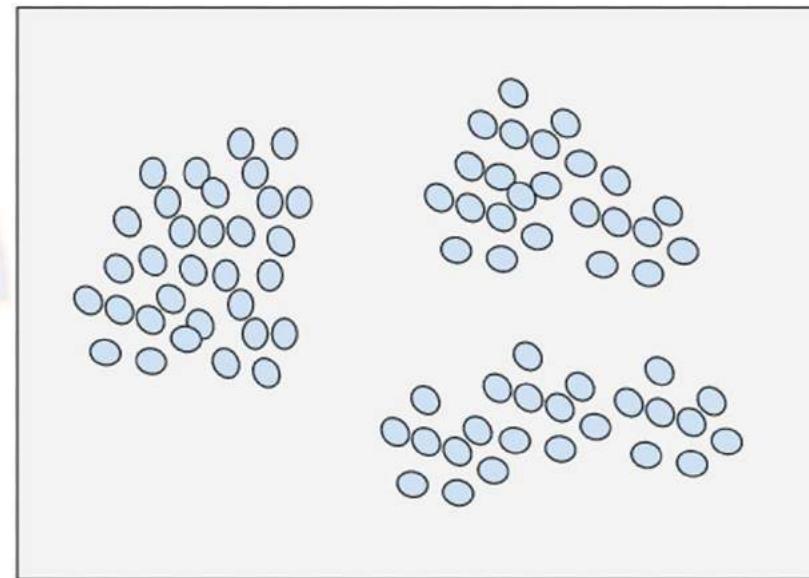


Introduction

Unsupervised Learning

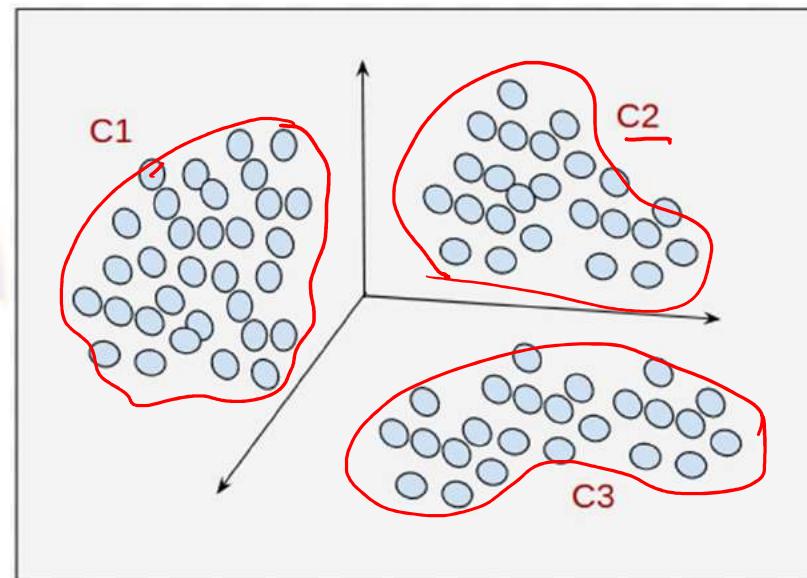
- Learning from unlabelled data
- Let $X = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(N)}\}$
 - The points does not carry labels



Introduction

Unsupervised Learning

- Learning from unlabelled data
- Let $X = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(N)}\}$
 - The points does not carry labels
 - Supervised vs. Unsupervised Learning

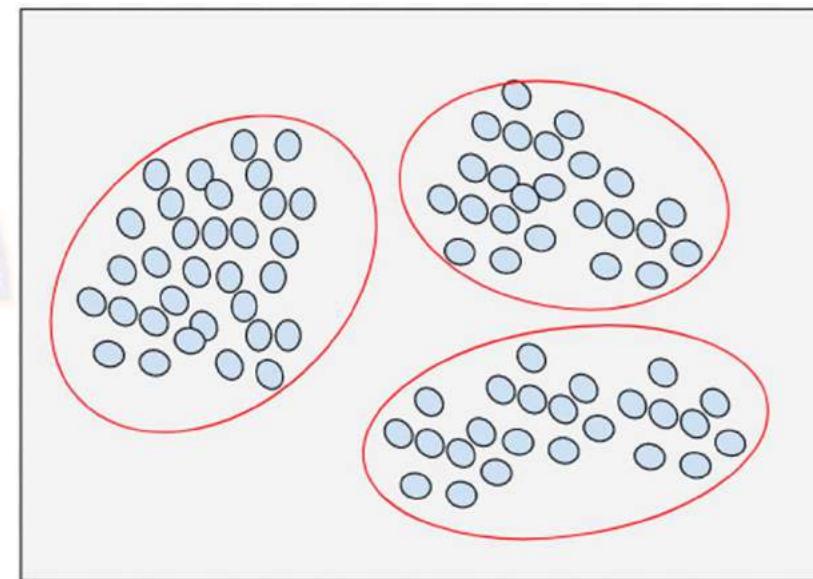


Supervised Learning - Learn the categories using class labels

Introduction

Unsupervised Learning

- Learning from unlabelled data
- Let $X = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(N)}\}$
 - The points does not carry labels
 - Supervised vs. Unsupervised Learning
 - Objective:
 - Find patterns / sub-groups among the the data points using data similarity
 - Visualizing higher dimension data (PCA)
 - [Covered Earlier]

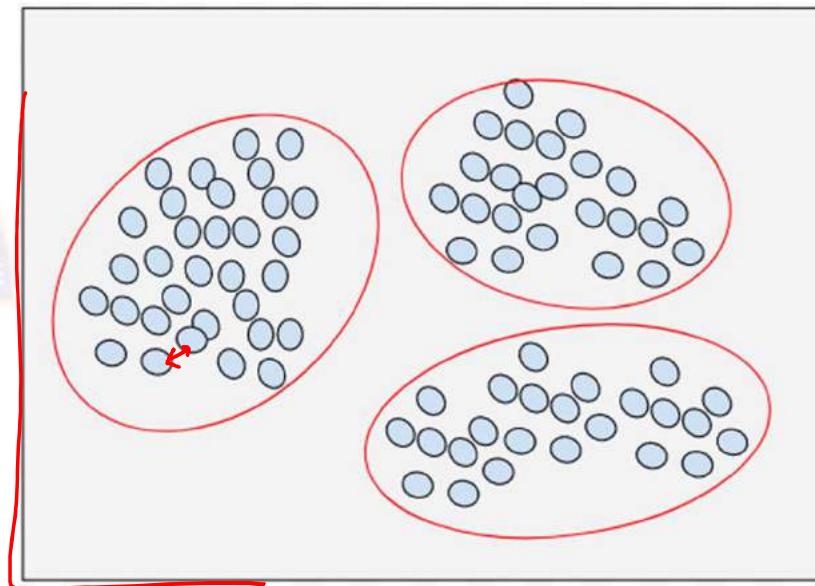
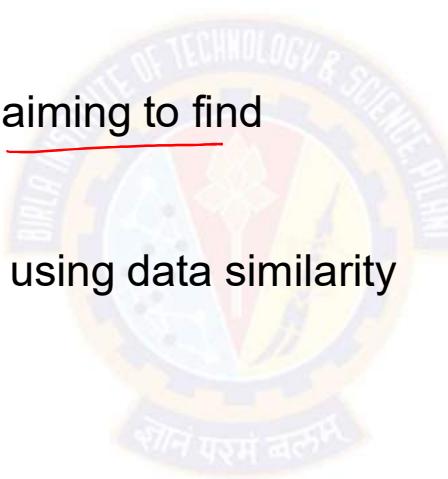


Unsupervised Learning - Find grouping based using data similarity

Introduction

Clustering

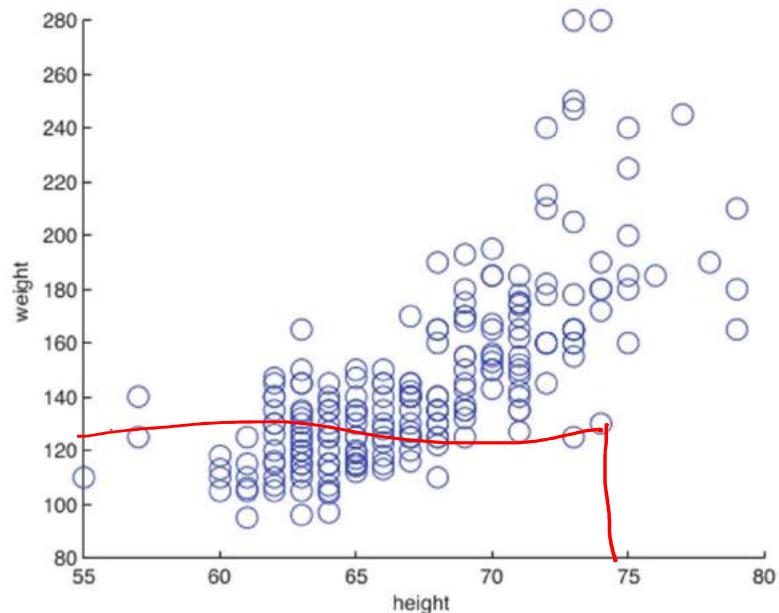
- An unsupervised Learning task aiming to find groupings in data
 - Given a X , find K clusters using data similarity



Unsupervised Learning - Find grouping based using data similarity

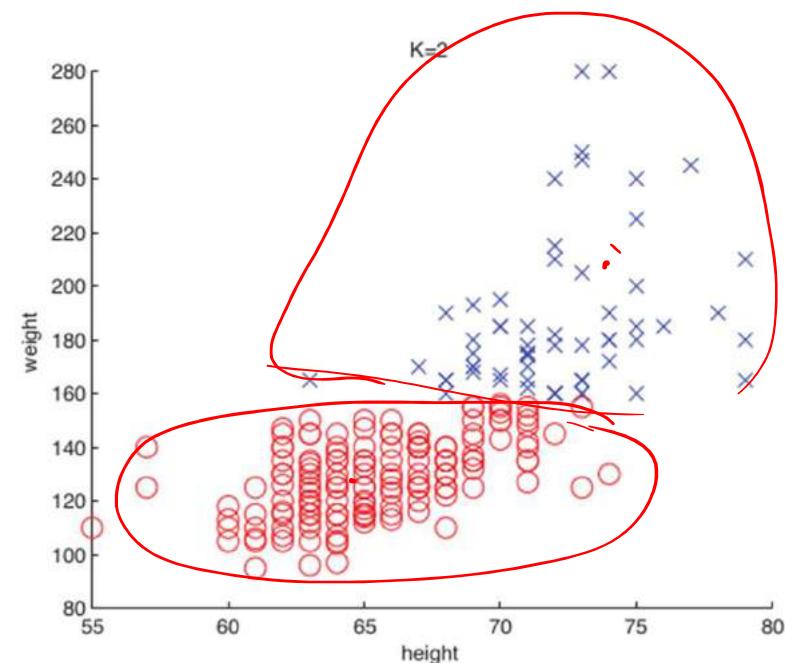
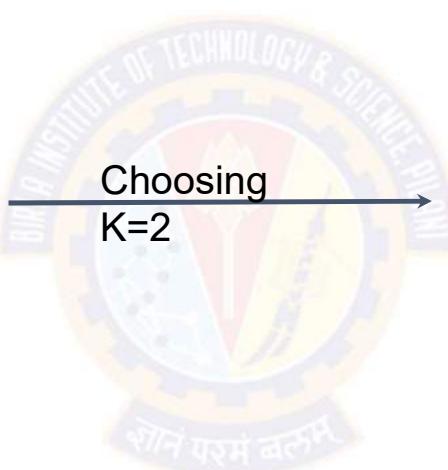
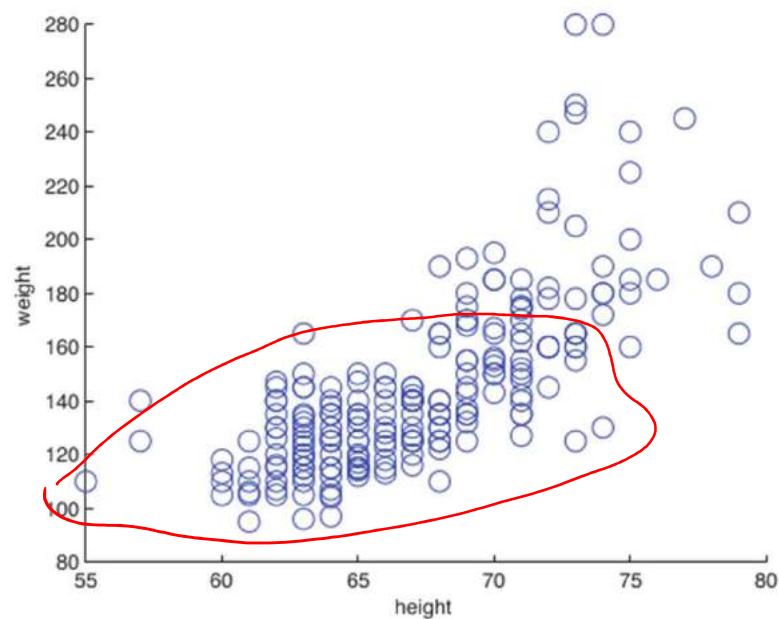
Introduction

Clustering



Introduction

Clustering



Introduction

Clustering



Introduction

Clustering

Politics and The Political in the "Berkeley School" of Political Theory

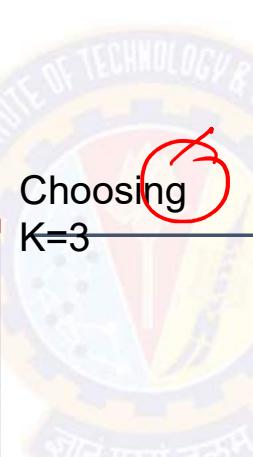
JUDGEOF S. CHIMA, Psychology and Scientific Research

THE LIGHT FANTASTIC MIND READY, MIND SET? IF NOT WHY NOT?

NORTHANTS GROWING THE GAME

Flight of the Ballhawks

Choosing K=3



Flight of the Ballhawks

D SET? NOT?

THE LIGHT FANTASTIC

Domestication of Food Plants in the Old World

Politics and The Political in the "Berkeley School" of Political Theory

JUDGEOF S. CHIMA, Psychology and Scientific Research

THE LIGHT FANTASTIC MIND READY, MIND SET? IF NOT WHY NOT?

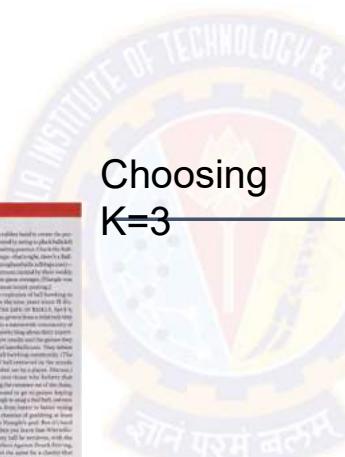
Introduction

Clustering



Introduction

Clustering



Choosing $K=3$



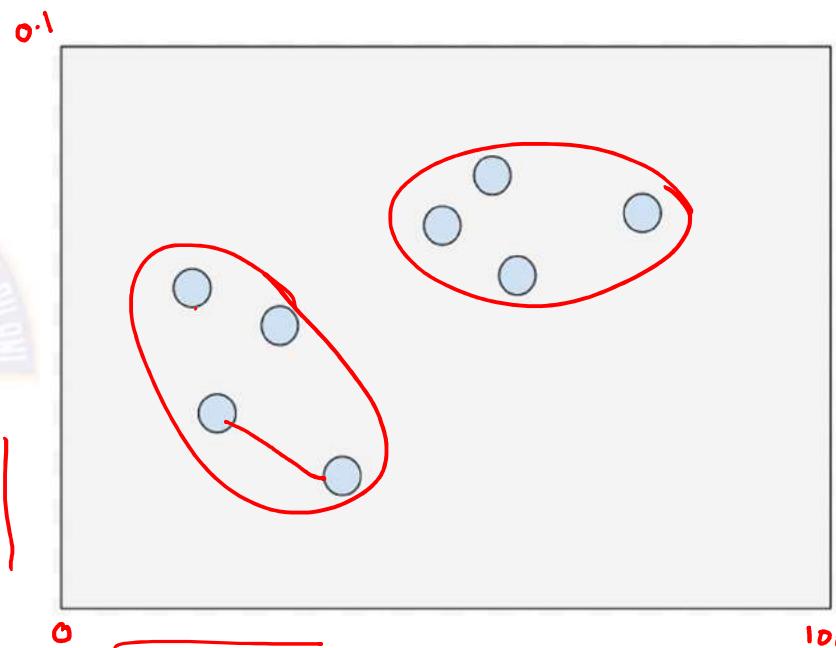
Politics and The Political in the
"Berkeley School" of Political Theory



Approaches to Clustering

Partitioning Methods

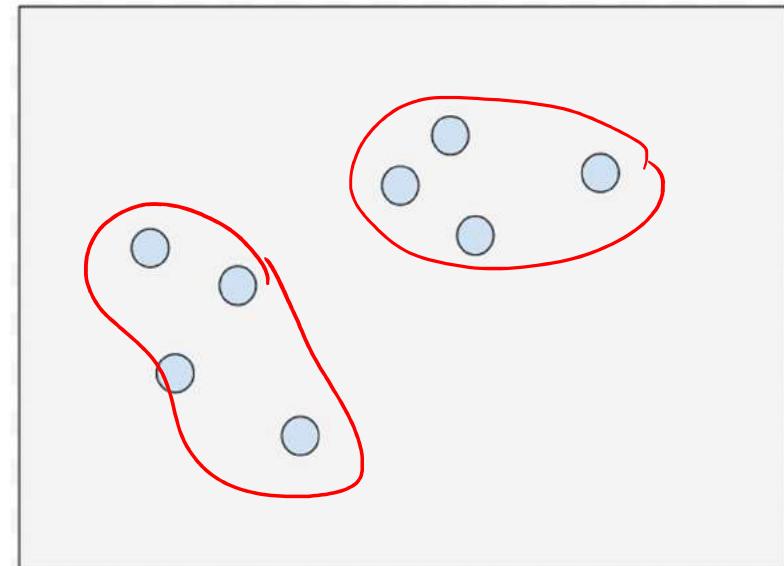
- Given X and K , construct k partitions in one level
- Uses distance measures
 - Each point is assigned exclusively assigned to one cluster / learn a
 - Use of normalization on numerical data
- K -means, K -Medoids [Module #2]



Approaches to Clustering

Model based

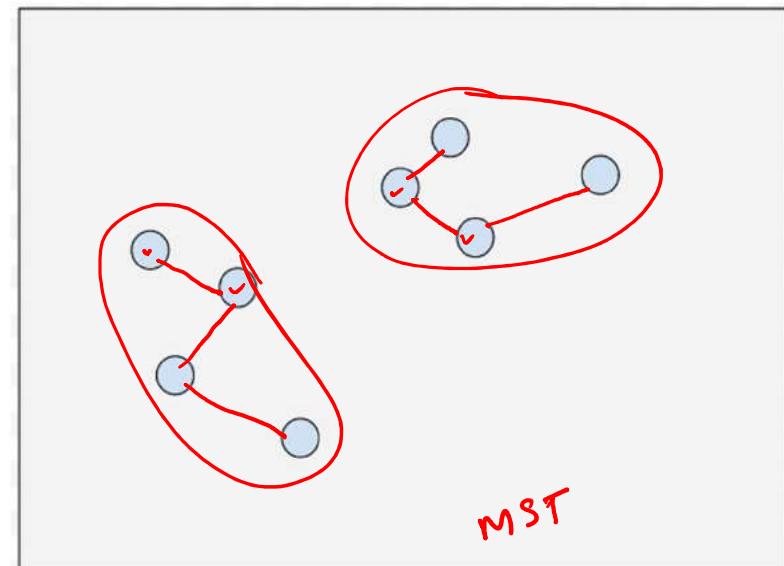
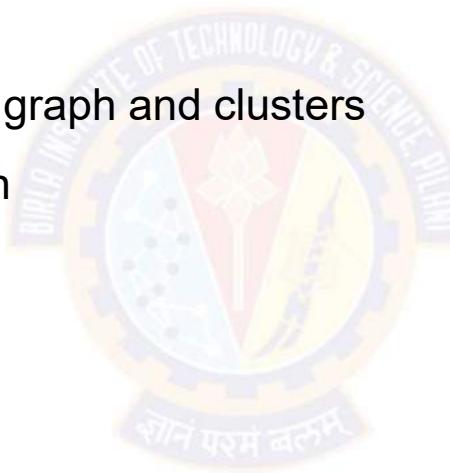
- Model is chosen for each cluster and model parameters are learned
 - EM Algorithm ✓



Approaches to Clustering

Graph theoretic based

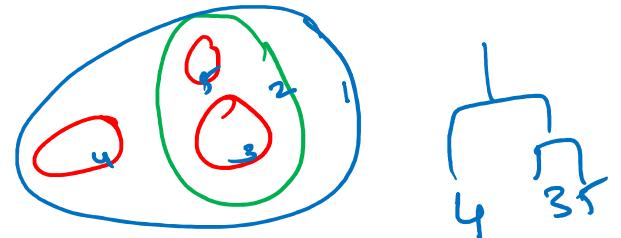
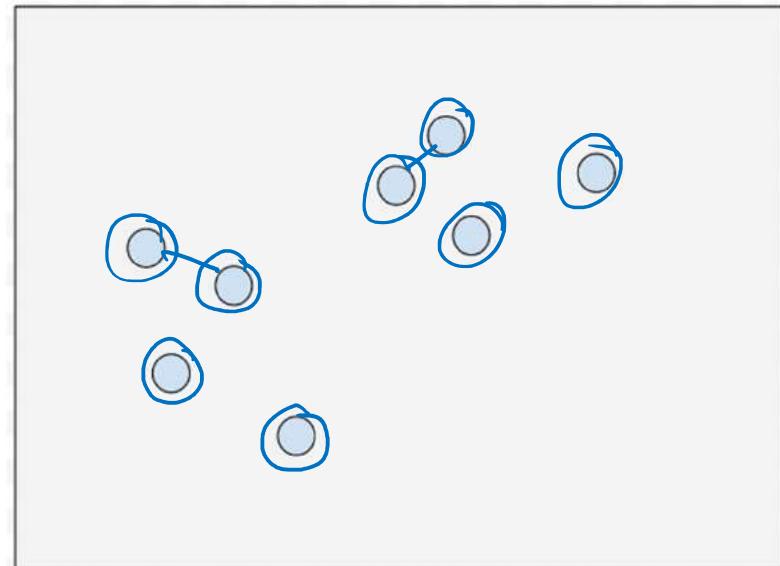
- Data points represented as a graph and clusters form components in the graph



Approaches to Clustering

Hierarchical Algorithm

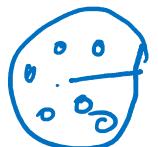
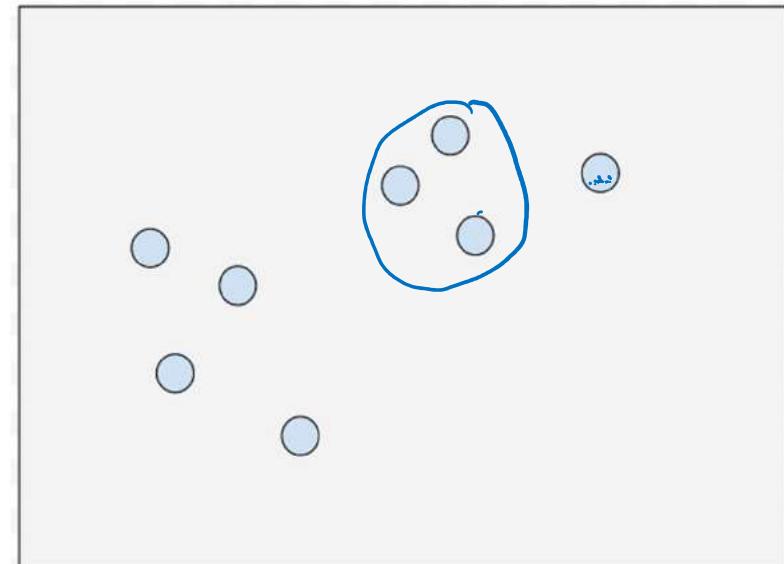
- Visualize clusters forming hierarchical structure
 - Agglomerative, Divisive
 - Single Link , Complete Link



Approaches to Clustering

Density Based

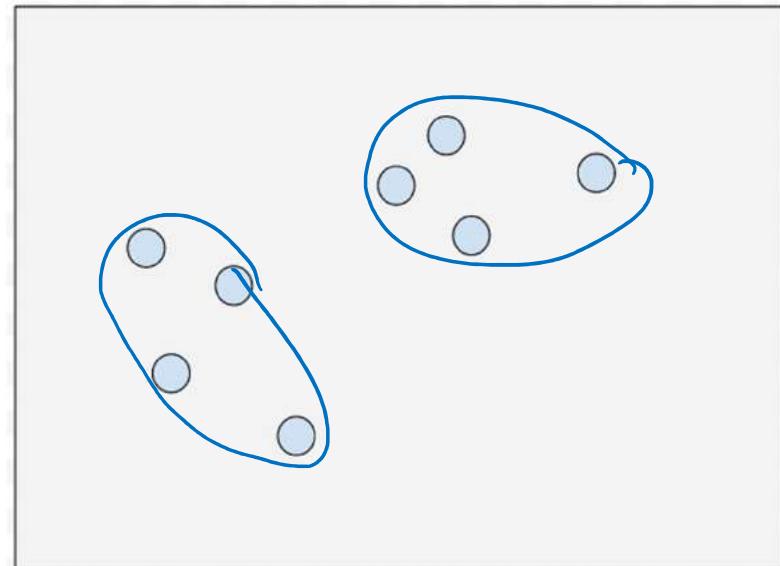
- Uses the notion of density of a neighbourhood in the data set and uses this to identify cluster
- DBSCAN



Few more aspects of clustering

Quality of Clustering

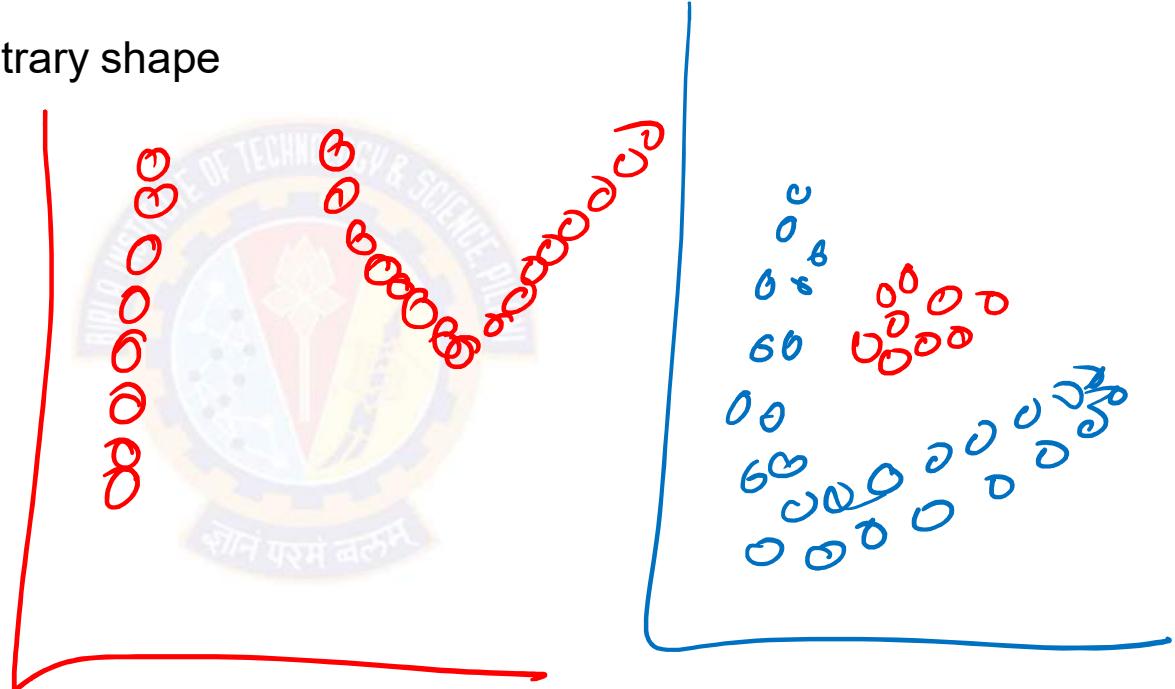
- A good clustering method will produce high quality clusters
 - high intra-class similarity
 - low inter-class similarity
- The quality of a clustering method depends on
 - The similarity measure used / strategy used
 - Various chosen parameters, and domain knowledge used
 - Able to find the structures/ regularities in the data



Few more aspects of clustering

(Some of the) Challenges

- Ability to find clusters of arbitrary shape
- Noise in the data
- Huge dimensions
- Scale



Introduction

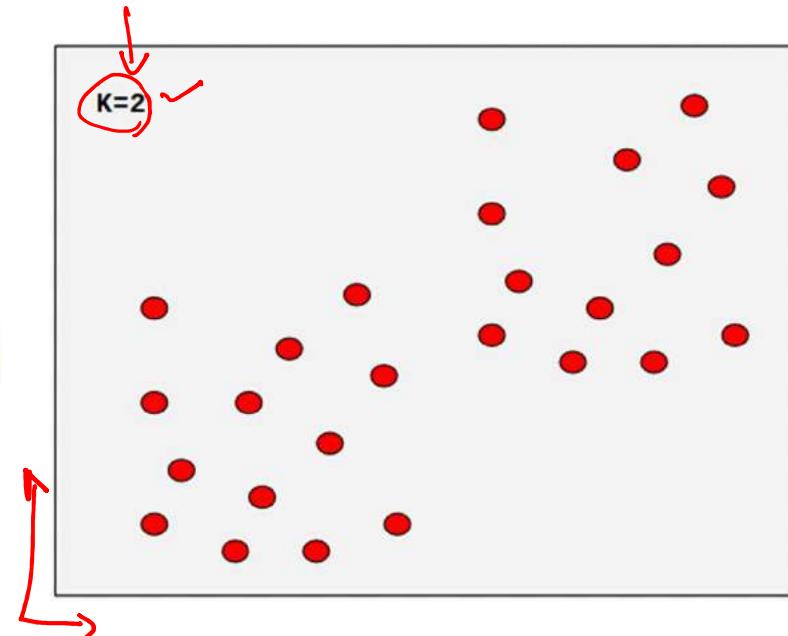
Note on demonstrations used in the course

- Data Sets, descriptions of datasets will be posted on the course page with detailed annotation
- Demonstration will be shared as python notebook files, with explanations for each steps
 - Queries will be answered proactively
- For programming assignments, mini-projects
 - Template of submission will be shared with detailed rubrics
 - Submissions will be accepted only in the templated form
- Reach us through canvas with any of your queries

Introduction

What is it about?

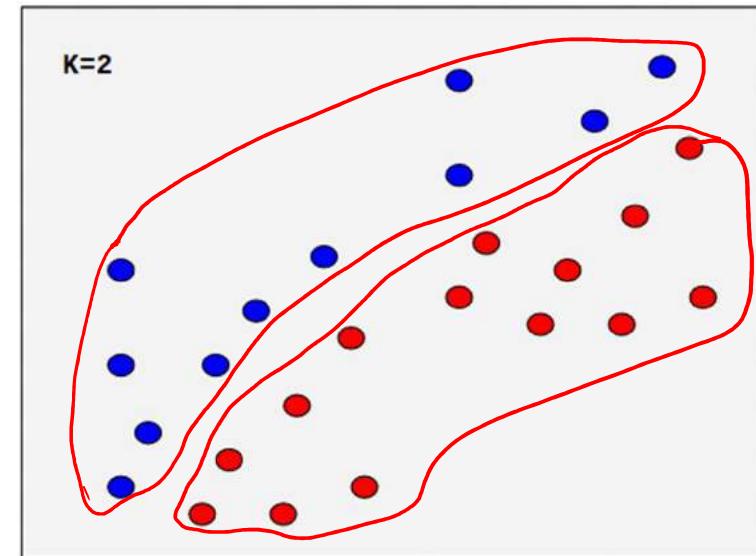
- Simple non-probabilistic algorithm for identifying clusters or groups (of data points) in a multidimensional space
- Input :
 - A data set $\{x_1, x_2, x_3, \dots, x_N\}$
 - Each x_i is a random point in a D dimensional space
 - K - number of clusters form
 - An unsupervised learning task



Introduction

What is it about?

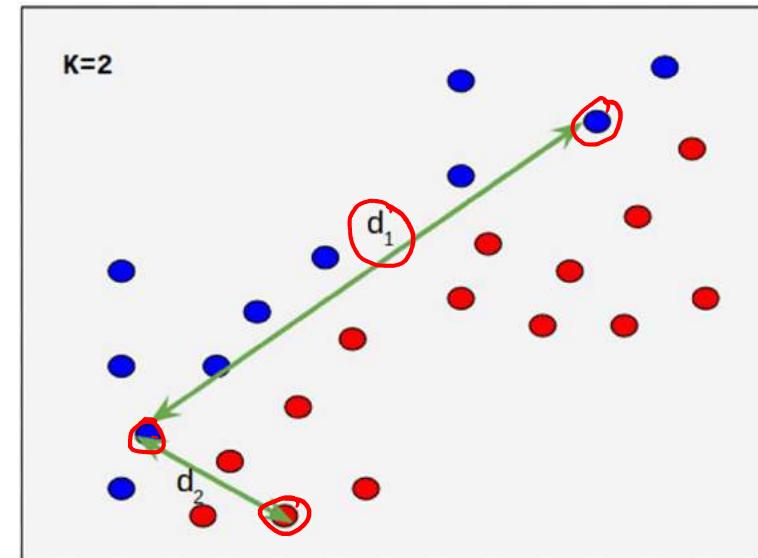
- k-means partitions points in K clusters such that
 - Distance between points inside the clusters is smaller than distance between points across clusters



Introduction

What is it about?

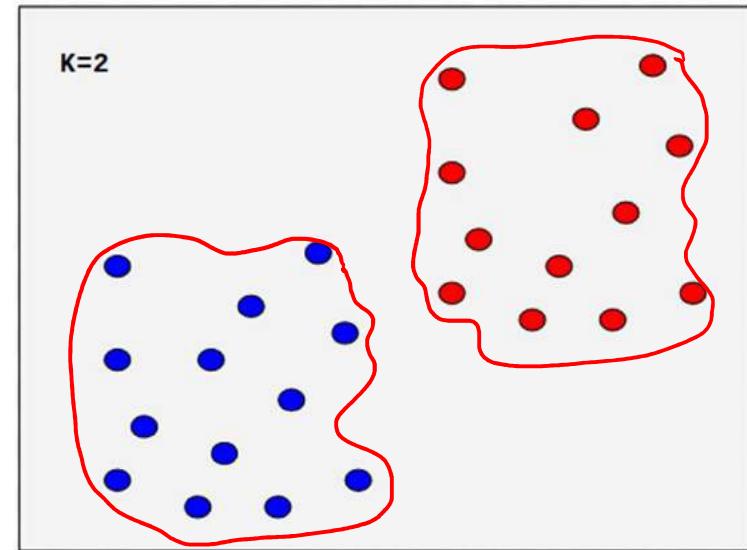
- k-means partitions points in K clusters such that
 - Distance between points inside the clusters is smaller than distance between points across clusters



Introduction

What is it about?

- k-means partitions points in K clusters such that
 - Distance between points inside the clusters is smaller than distance between points across clusters

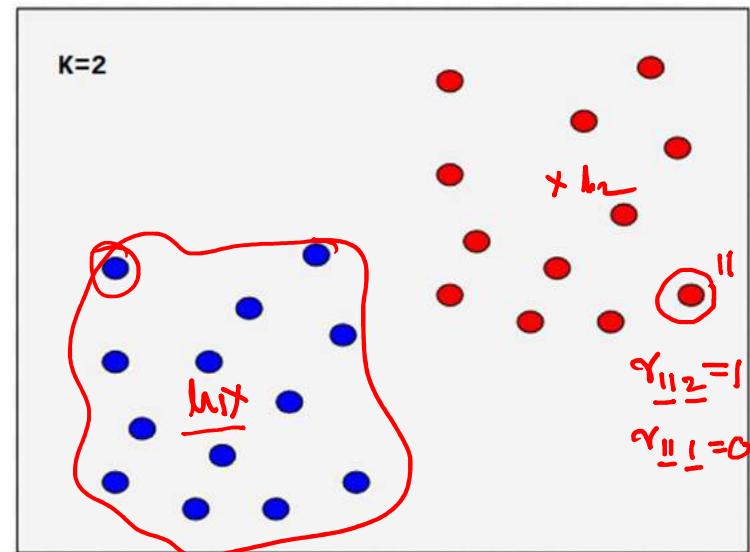


Better One !!!

Introduction

What is it about?

- k-means partitions points in K clusters such that
 - Distance between points inside the clusters is smaller than distance between points across clusters
- Let
 - μ_k Mean of all points in cluster k
 - $r_{nk} \in \{0, 1\}$
 - $r_{nk} = 1$, point x_n is assigned to cluster k
 - $r_{nk} = 0$, otherwise

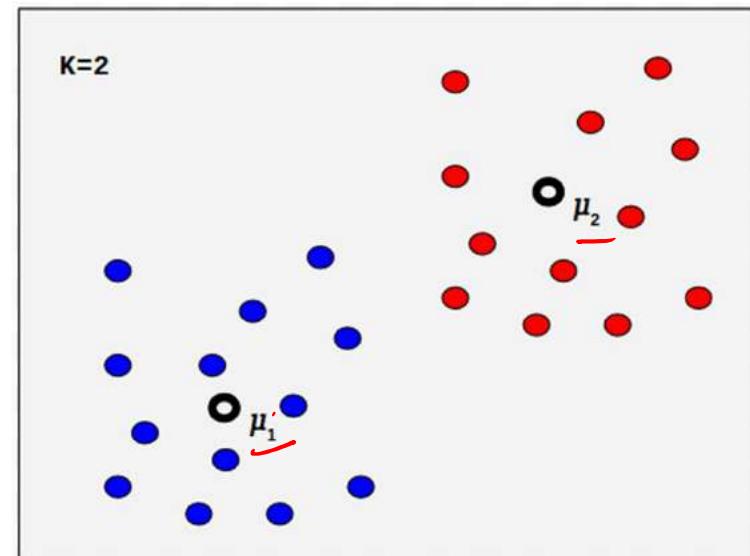
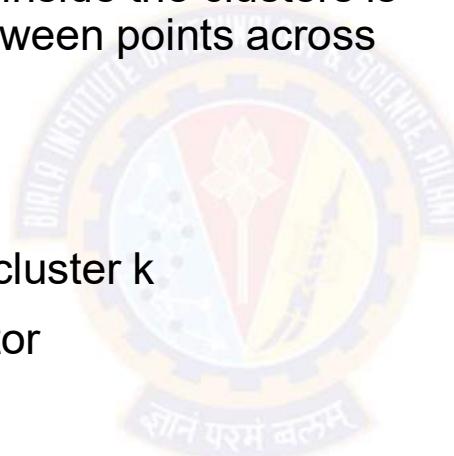


Better One !!!

Introduction

What is it about?

- k-means partitions points in K clusters such that
 - Distance between points inside the clusters is smaller than distance between points across clusters
- Let
 - μ_k - Mean of all points in cluster k
 - a d-dimensional vector
 - $r_{nk} \in \{0,1\}$
 - $r_{nk} = 1$, point x_n is assigned to cluster k
 - $r_{nk} = 0$, otherwise



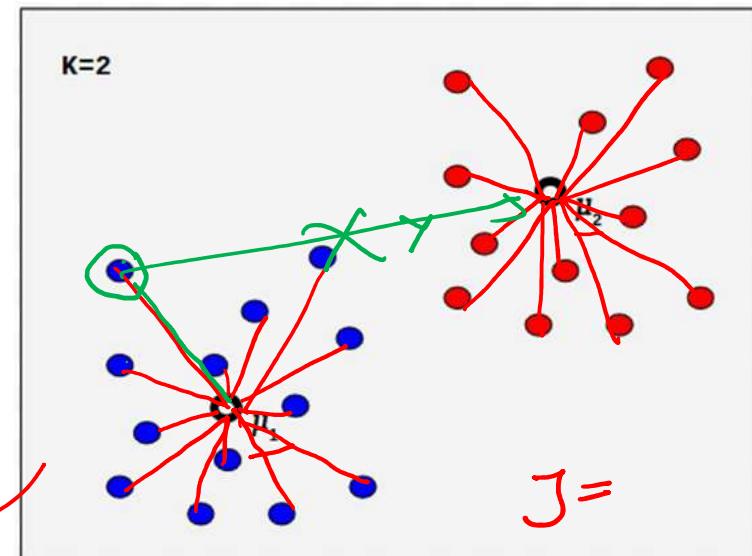
Introduction

Distortion Measure ✓

- k-means identifies K clusters such that the sum of the squares of the distances of each data point to its center (represented by its mean) is minimized.

Objective function:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$



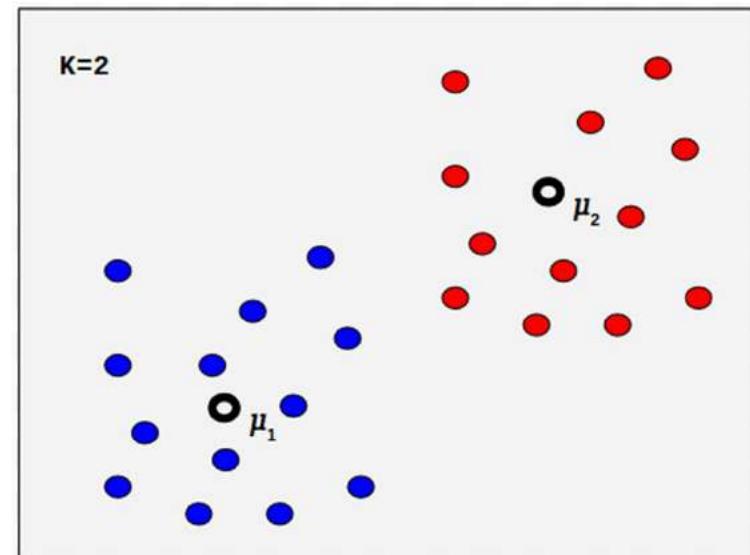
Introduction

Distortion Measure

- k-means identifies K clusters such that the sum of the squares of the distances of each data point to its center (represented by its mean) is minimized.

Objective function:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



- Objective of Learning: Find $\{\boldsymbol{\mu}_k\}$ and $\{r_{nk}\}$ such that J is minimized for the given X and K
- K-Means is an iterative algorithm to find such $\{\boldsymbol{\mu}_k\}$ and $\{r_{nk}\}$

K-Means Algorithm

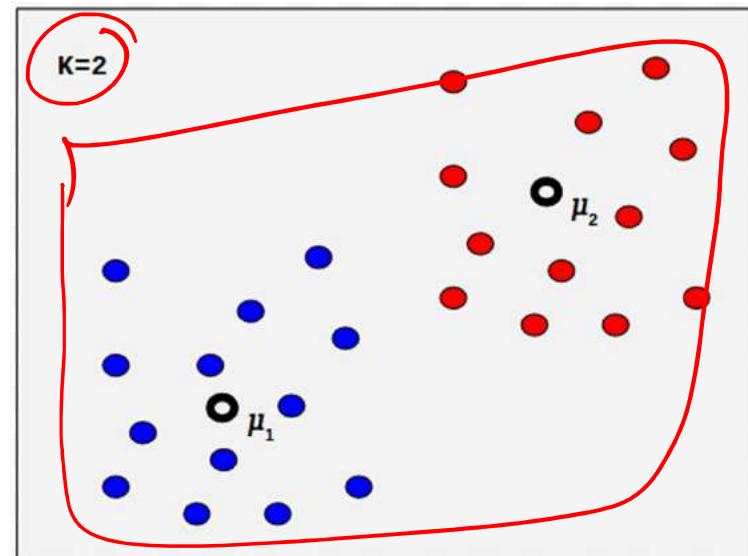
- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Iteration involves two key steps

- (1) Find $\{r_{nk}\}$, fixing $\{\mu_k\}$ to minimize J
- (2) Find $\{\mu_k\}$, fixing $\{r_{nk}\}$ to minimize J

Let us look at each of these steps



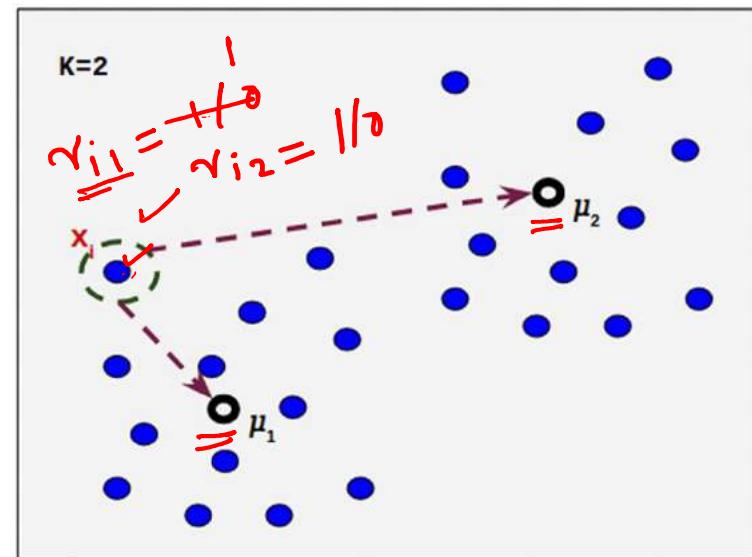
K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Let us determine r_{nk} :

Assume μ_1 and μ_2 are fixed.



K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

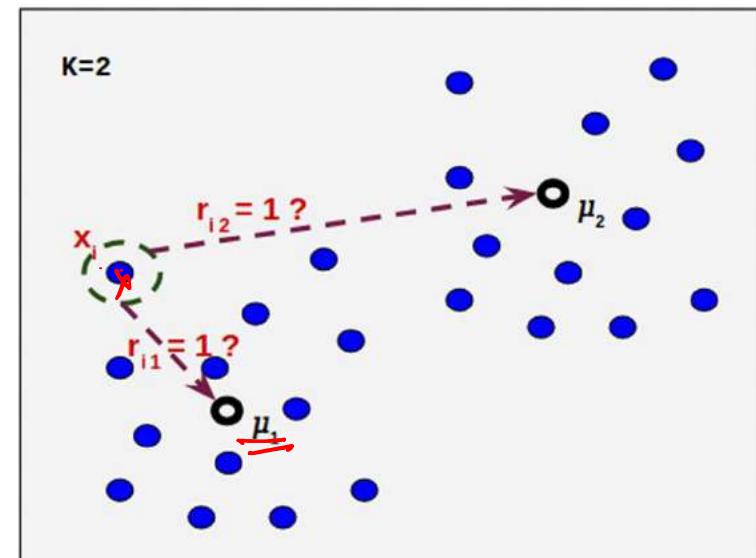
$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Let us determine r_{nk} :

Assume μ_1 and μ_2 are fixed.

J is a linear combination of r_{nk} .

Each r_{ik} can be optimized independently.



K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Let us determine r_{nk} :

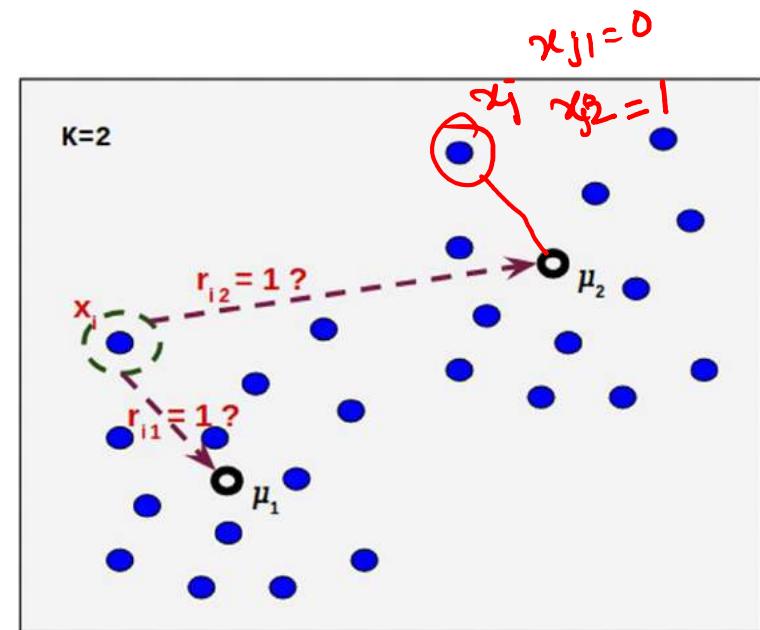
Assume μ_1 and μ_2 are fixed.

J is a linear combination of r_{nk} .

Each r_{ik} can be optimized independently.

Assigning r_{ik} to the closest μ_j minimizes J .

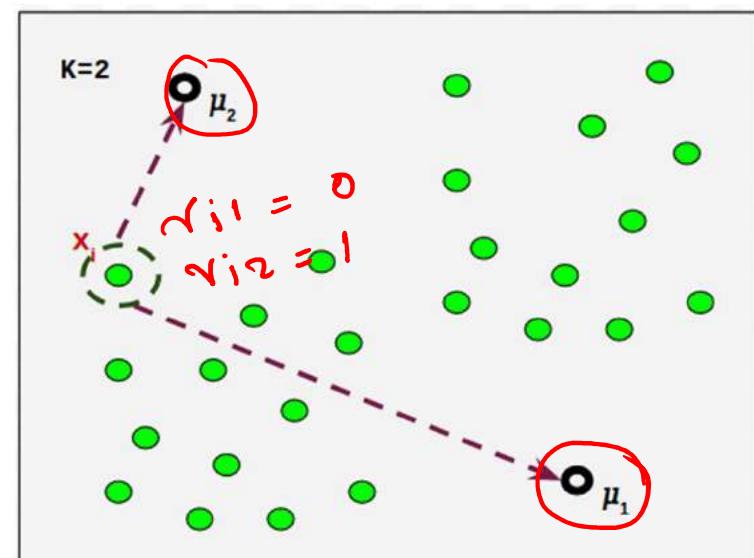
[Called as E-Step]



$$\underline{r_{nk}} = \begin{cases} 1 & \text{if } k = \arg \min_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

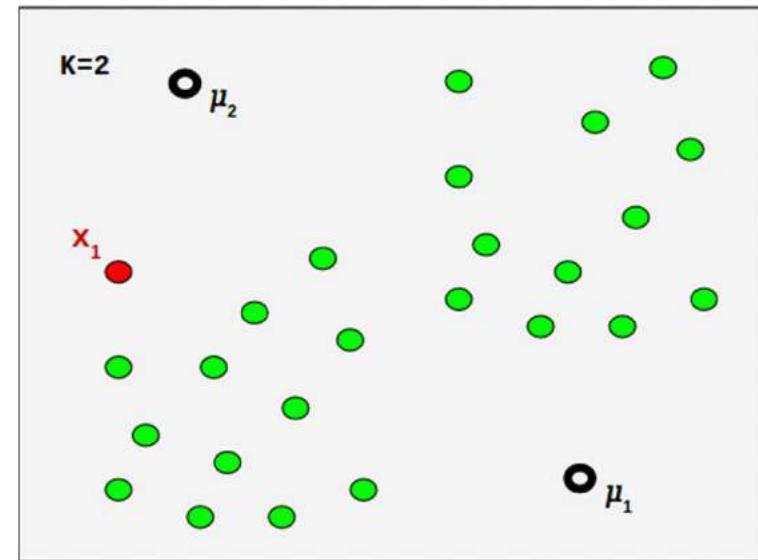
K-Means Algorithm

A sample E-Step



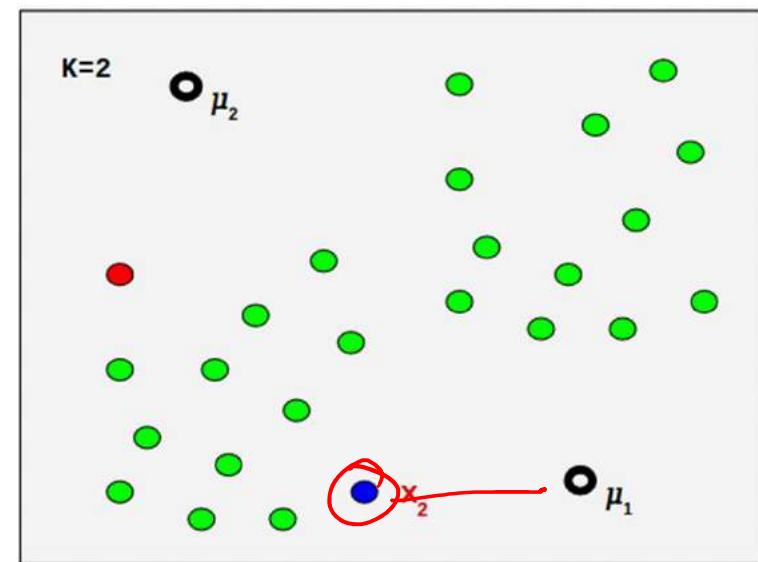
K-Means Algorithm

A sample E-Step



K-Means Algorithm

A sample E-Step

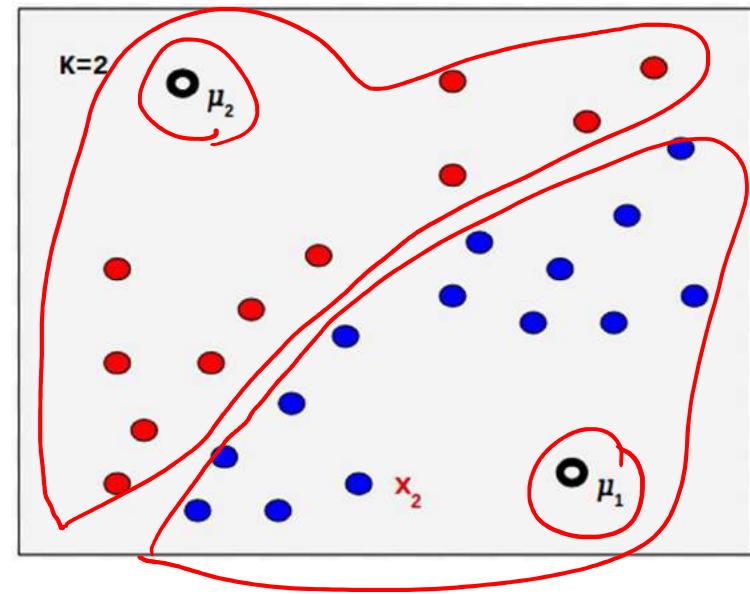


K-Means Algorithm

E-Step:

For all $x_t \in X$:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$



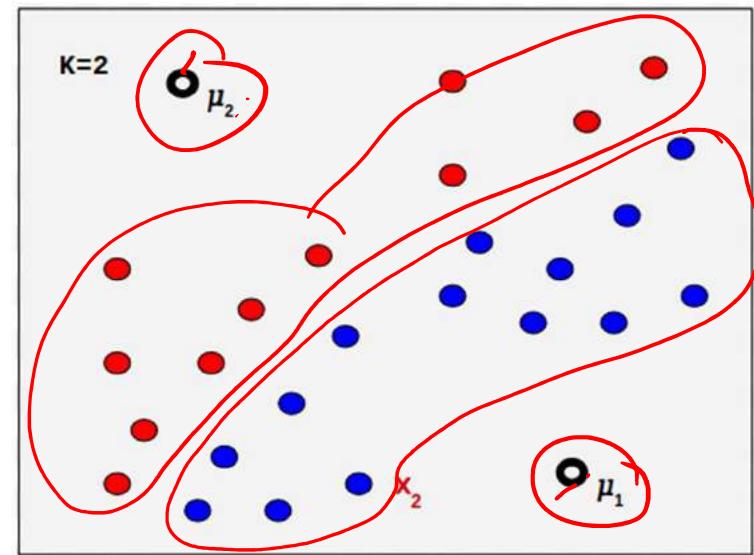
K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

Let us determine $\boldsymbol{\mu}_k$:

Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.



K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Let us determine μ_k :

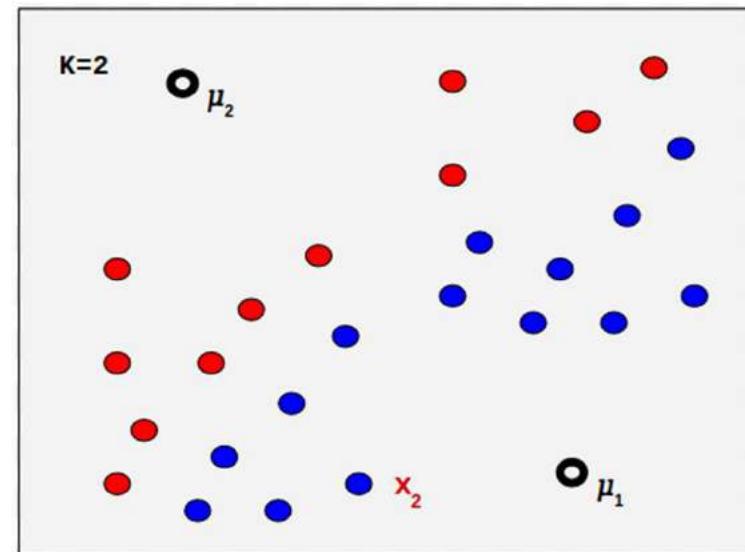
Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.

J is a quadratic function of μ_k .

To optimize,

Take the derivative of J w.r.t μ_k , & set it to 0

Solve for μ_k , we get $\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$



K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

Let us determine μ_k :

Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.

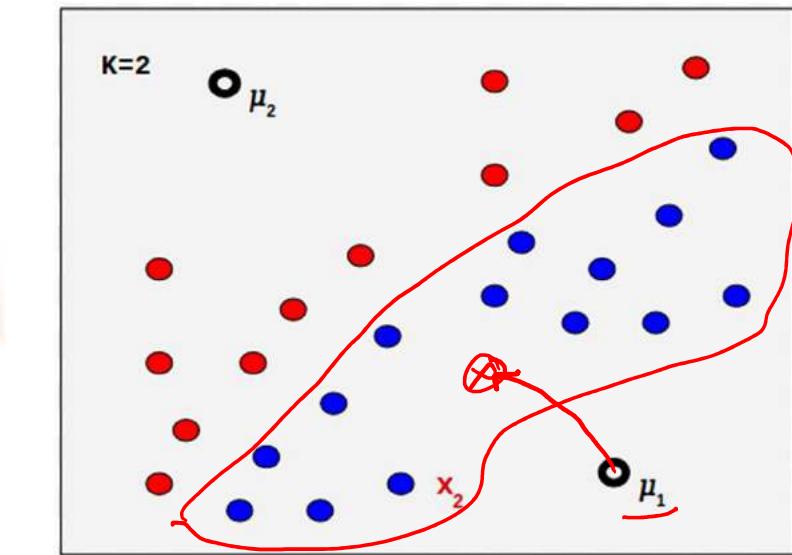
J is a quadratic function of μ_k .

To optimize,

Take the derivative of J w.r.t μ_k , & set it to 0

Solve for μ_k , we get

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$



Sum of all the points in a cluster
Number of points in a cluster

K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

Let us determine $\boldsymbol{\mu}_k$:

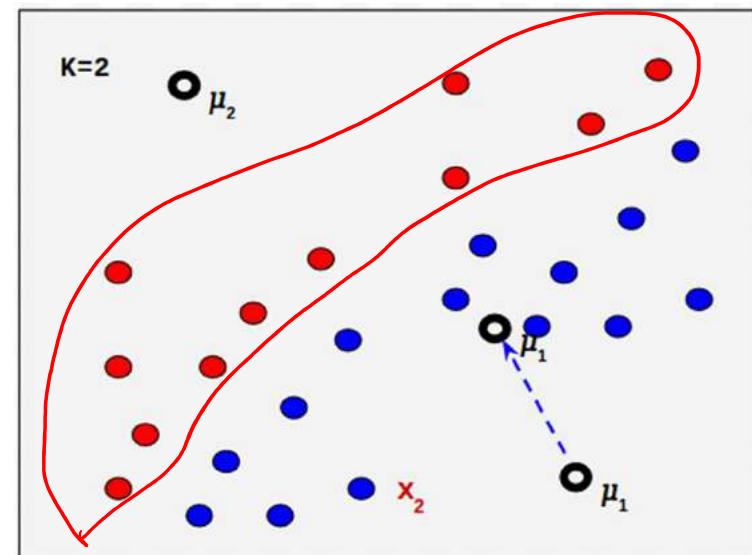
Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.

J is a quadratic function of $\boldsymbol{\mu}_k$.

To optimize,

Take the derivative of J w.r.t $\boldsymbol{\mu}_k$, & set it to 0

Solve for $\boldsymbol{\mu}_k$, we get
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

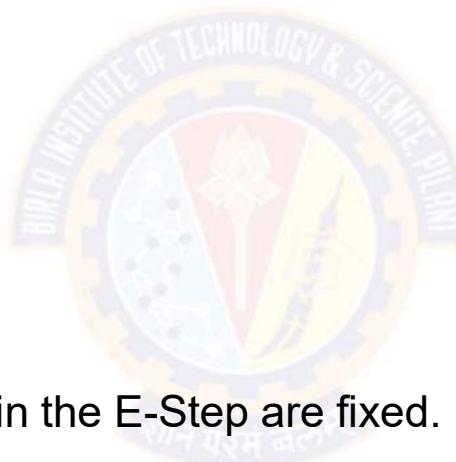


Sum of all the points in a cluster
Number of points in a cluster

K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



Let us determine $\boldsymbol{\mu}_k$:

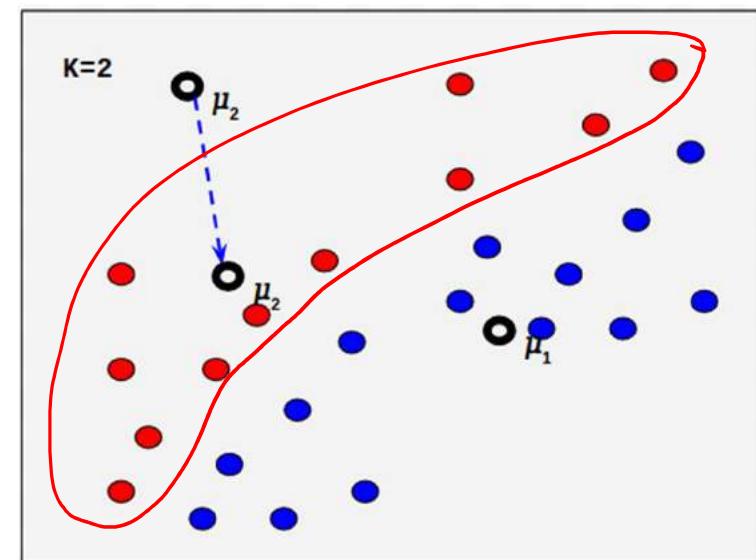
Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.

J is a quadratic function of $\boldsymbol{\mu}_k$.

To optimize,

Take the derivative of J w.r.t $\boldsymbol{\mu}_k$, & set it to 0

Solve for $\boldsymbol{\mu}_k$, we get
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$



Sum of all the points in a cluster
Number of points in a cluster

K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Let us determine μ_k :

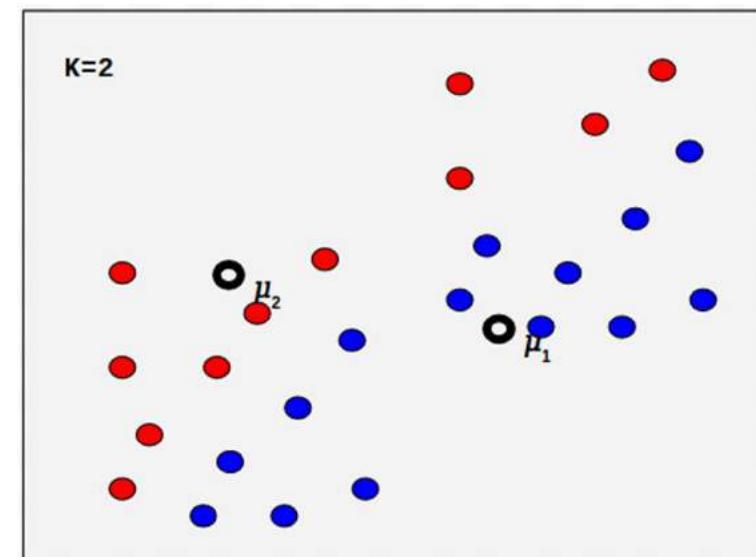
Assume $\{r_{nk}\}$ are determined in the E-Step are fixed.

J is a quadratic function of μ_k .

To optimize,

Take the derivative of J w.r.t μ_k , & set it to 0

Solve for μ_k , we get
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$



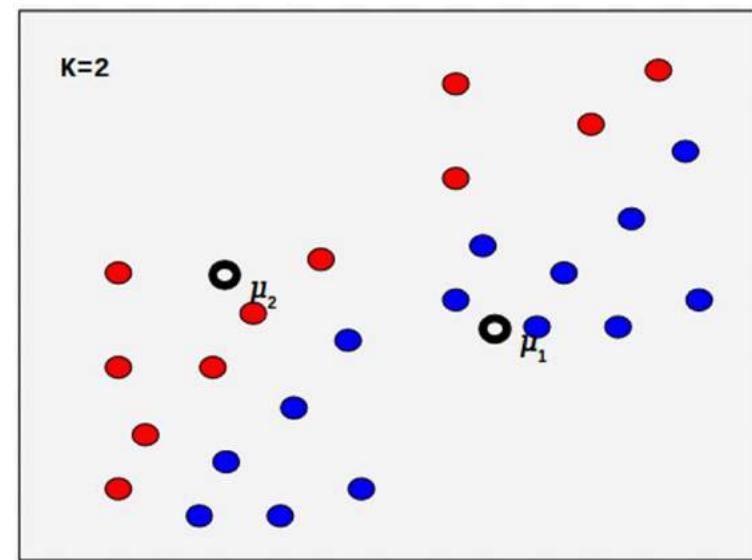
Sum of all the points in a cluster
Number of points in a cluster

K-Means Algorithm

M-Step:

For all μ_k [where $k = 1, 2, \dots, K$] :

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}} \quad \checkmark$$



K-Means Algorithm



Algorithm:

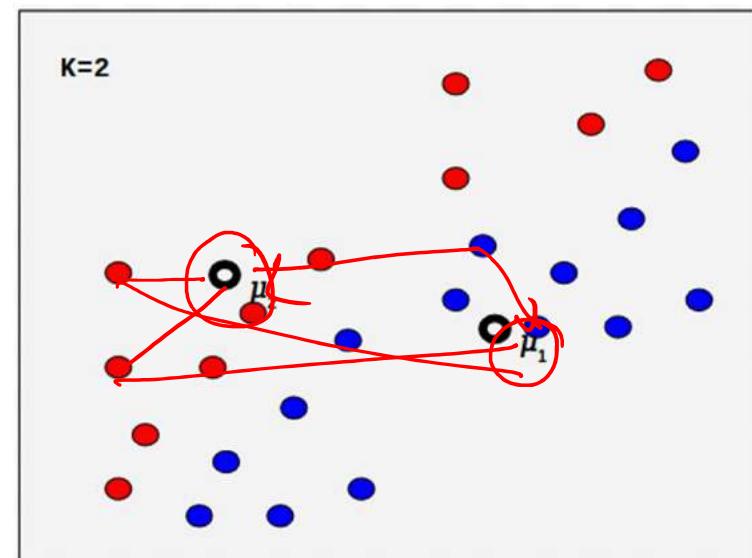
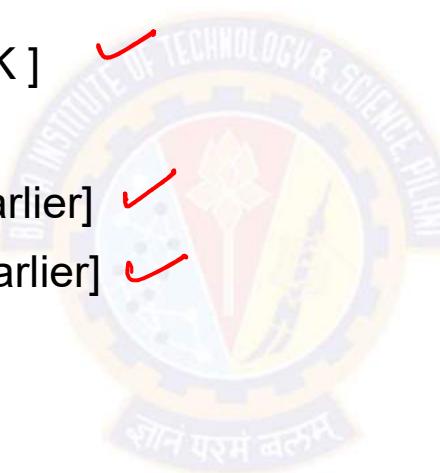
Initialize μ_k [where $k = 1, 2, \dots, K$]

Repeat

E-Step [as defined earlier]

M-Step [as defined earlier]

Until convergence of μ_k .



K-Means Algorithm

Algorithm:

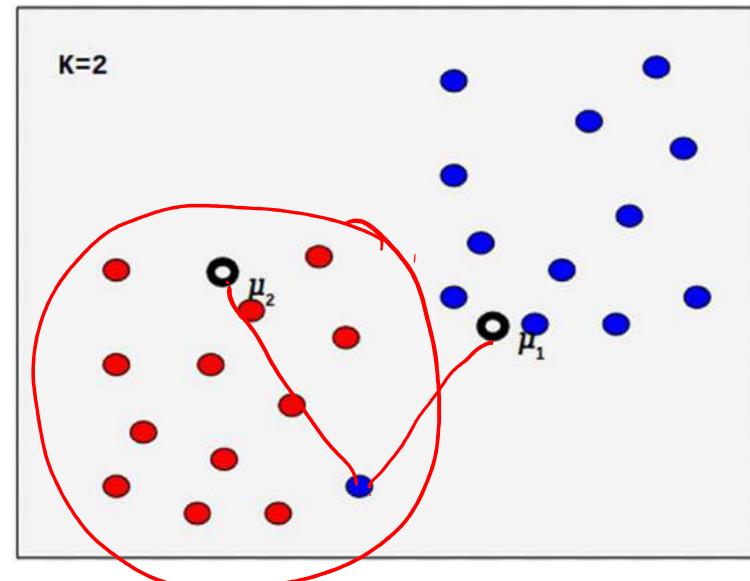
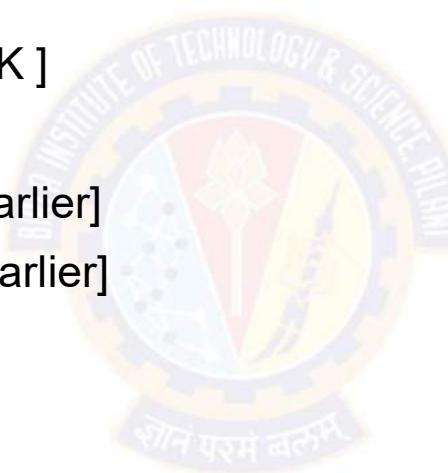
Initialize μ_k [where $k = 1, 2, \dots, K$]

Repeat

E-Step [as defined earlier]

M-Step [as defined earlier]

Until convergence of μ_k .



E-Step in the second iteration

K-Means Algorithm

Algorithm:

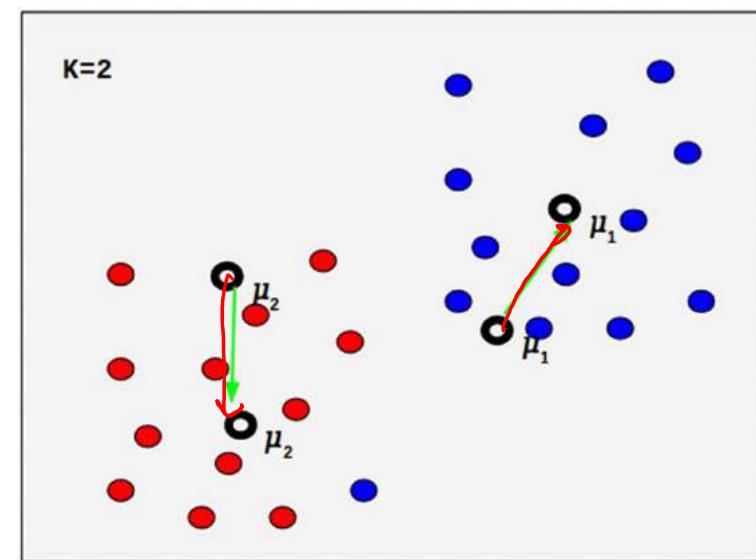
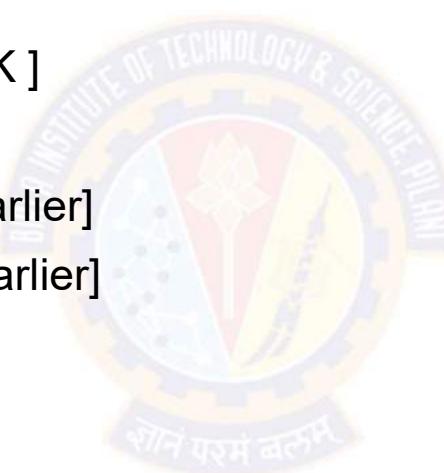
Initialize μ_k [where $k = 1, 2, \dots, K$]

Repeat

E-Step [as defined earlier]

M-Step [as defined earlier]

Until convergence of μ_k .



M-Step in the second iteration

K-Means Algorithm

Algorithm:

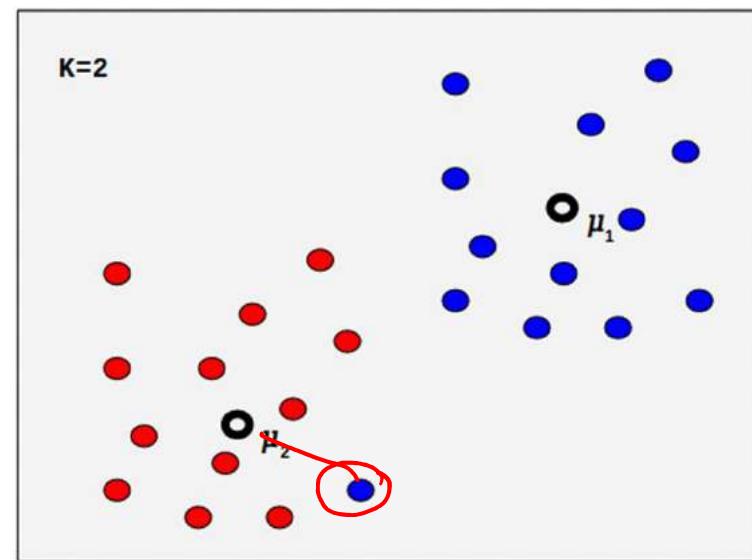
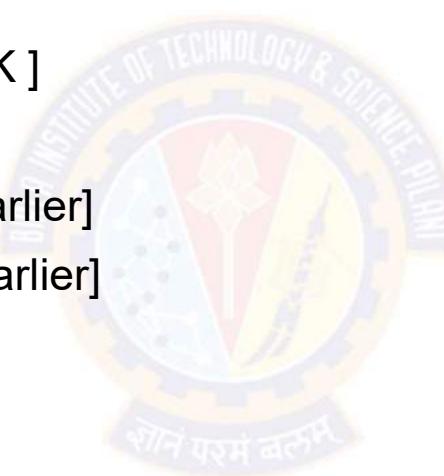
Initialize μ_k [where $k = 1, 2, \dots, K$]

Repeat

E-Step [as defined earlier]

M-Step [as defined earlier]

Until convergence of μ_k .

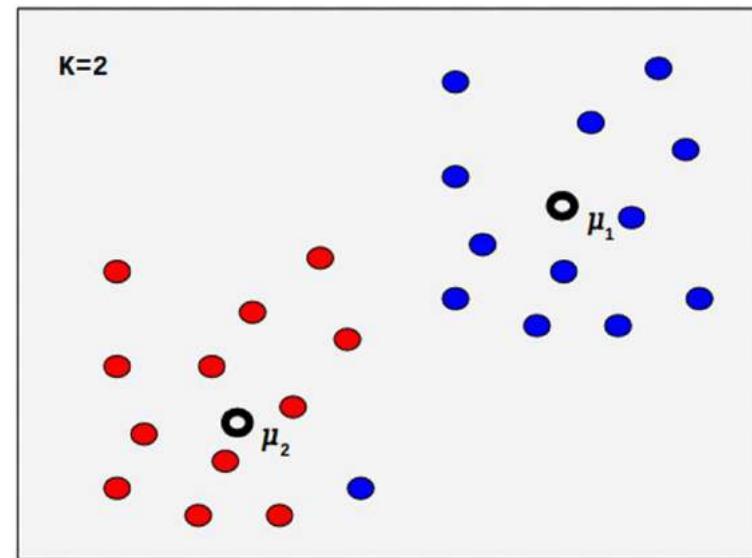


M-Step in the second iteration

K-Means Algorithm

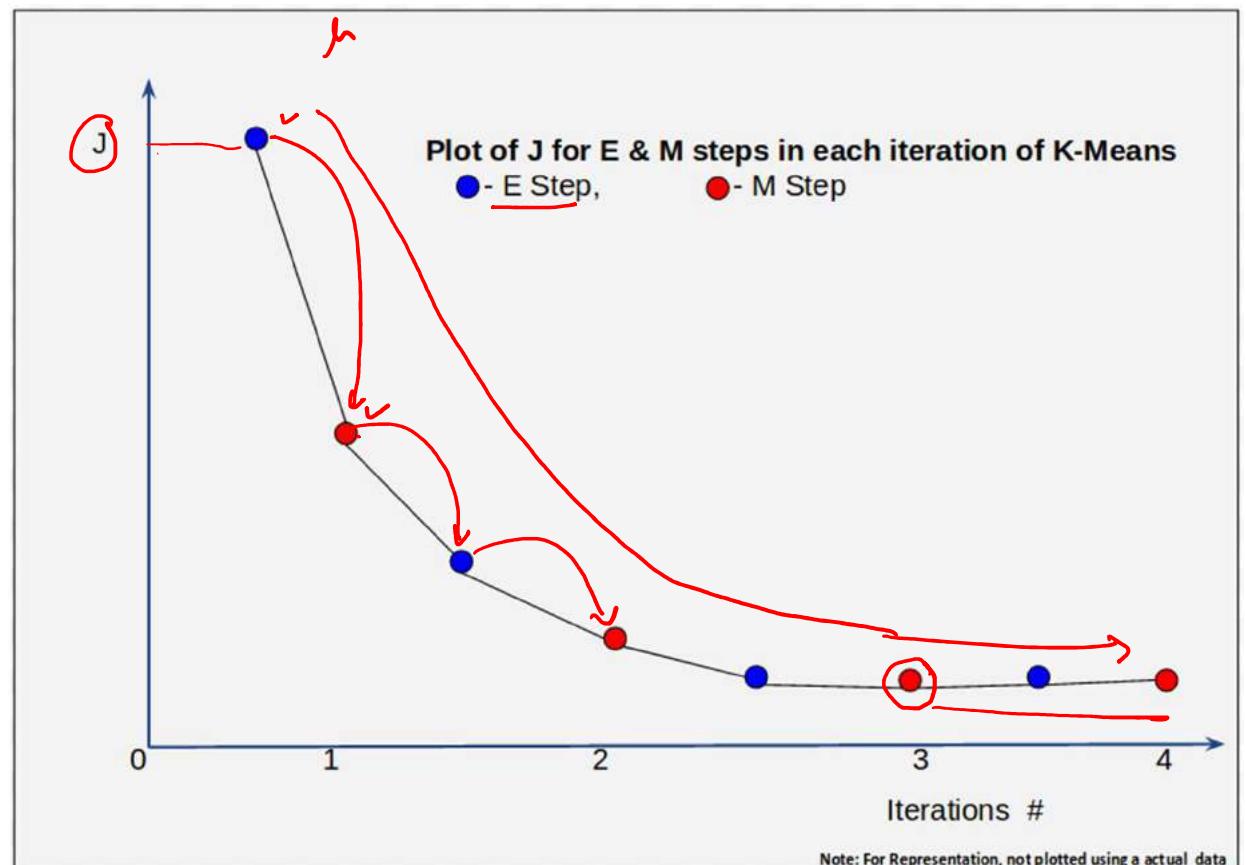
Some Questions?

- Repeat E & M for one more iteration
 - How many more iterations required? [Convergence]



K-Means Algorithm - Convergence

- E and M step of each iteration optimizes J. Stop E- Means when
 - J no longer changes or changes to J is not significant
 - There may be an upper limit on the number of iterations set.
- Choice of initial values have an impact on the convergence



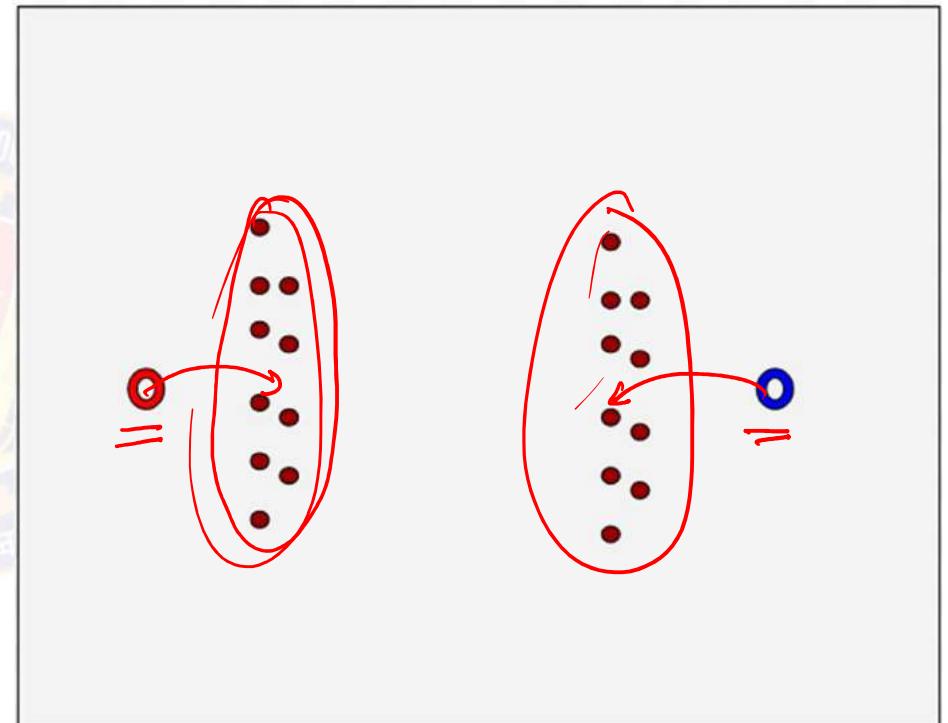
K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.



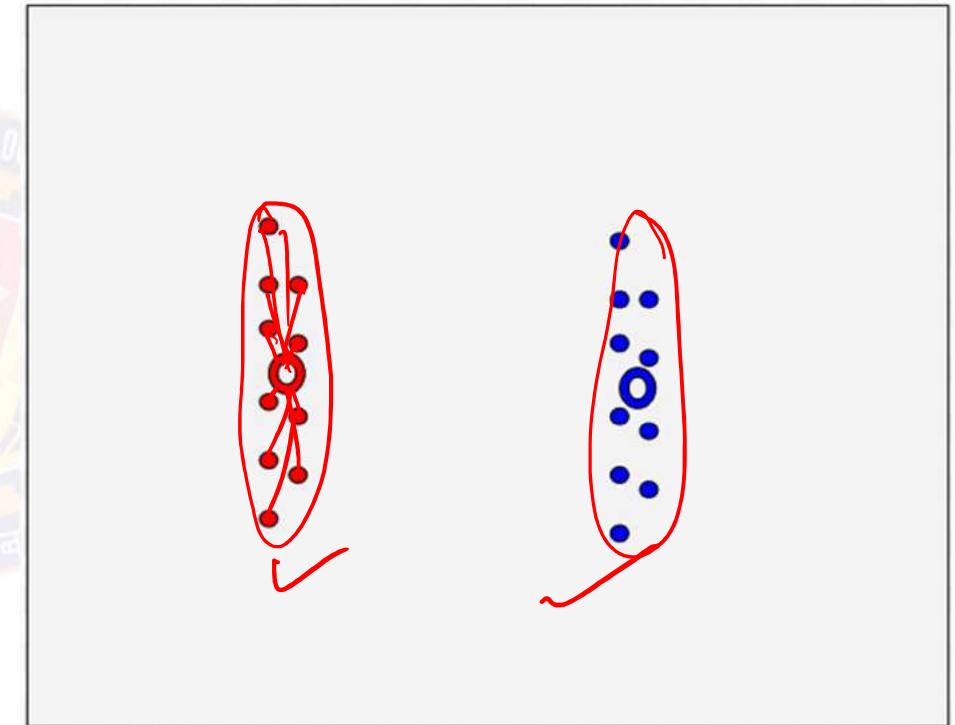
K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.



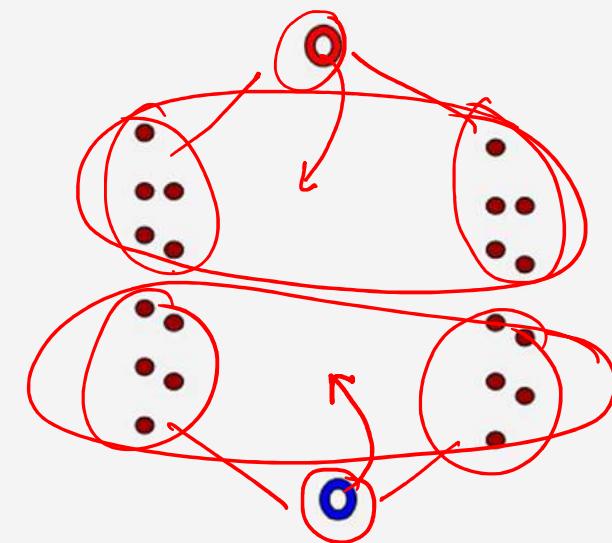
K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.



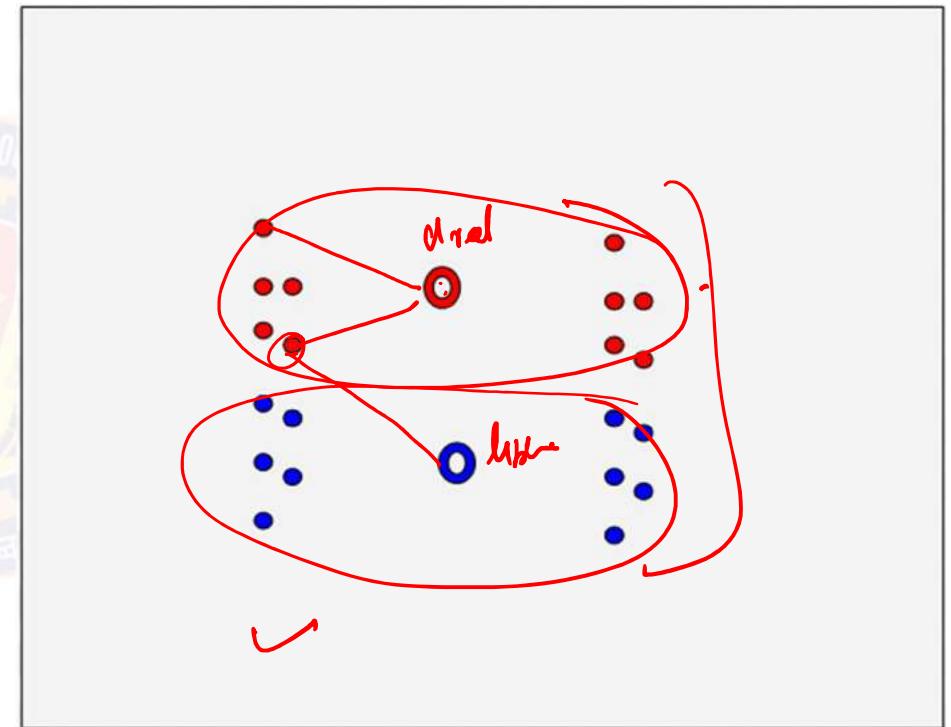
K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.



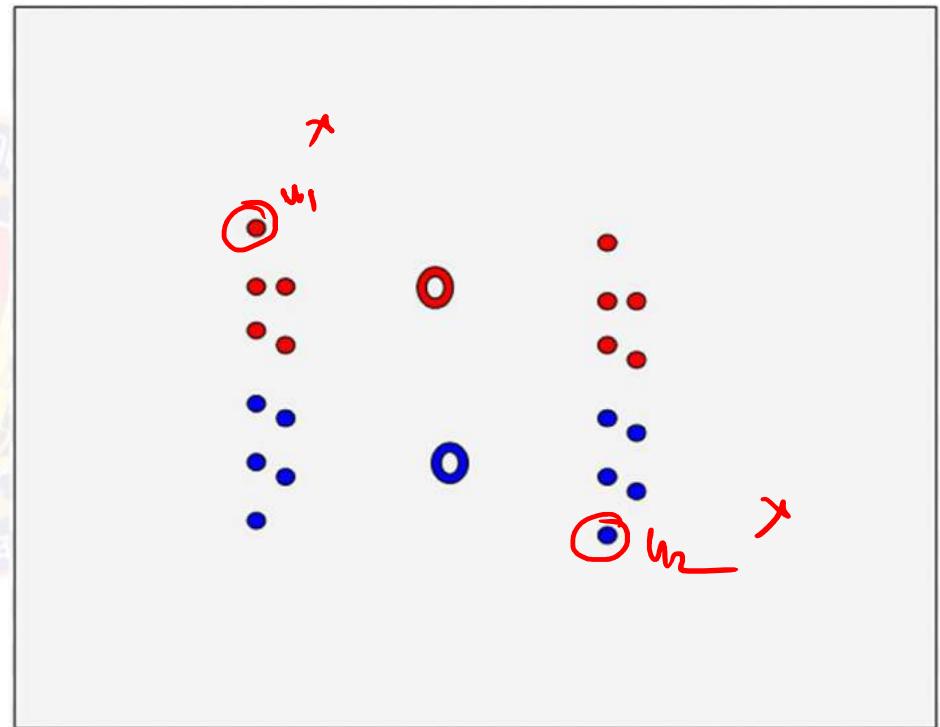
K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.



K-Means Algorithm - Convergence

- K-Means is a local search procedure, converges to a local optimum.
- Choice of initial $\{\mu_k\}$ impacts both the number of iterations and the nature of clusters obtained.
 - Select k-randomly chosen instances from X as initial $\{\mu_k\}$
 - Compute mean of all the points, add small random vectors to obtain initial $\{\mu_k\}$



K-Means Algorithm - Complexity

In each E-Step:

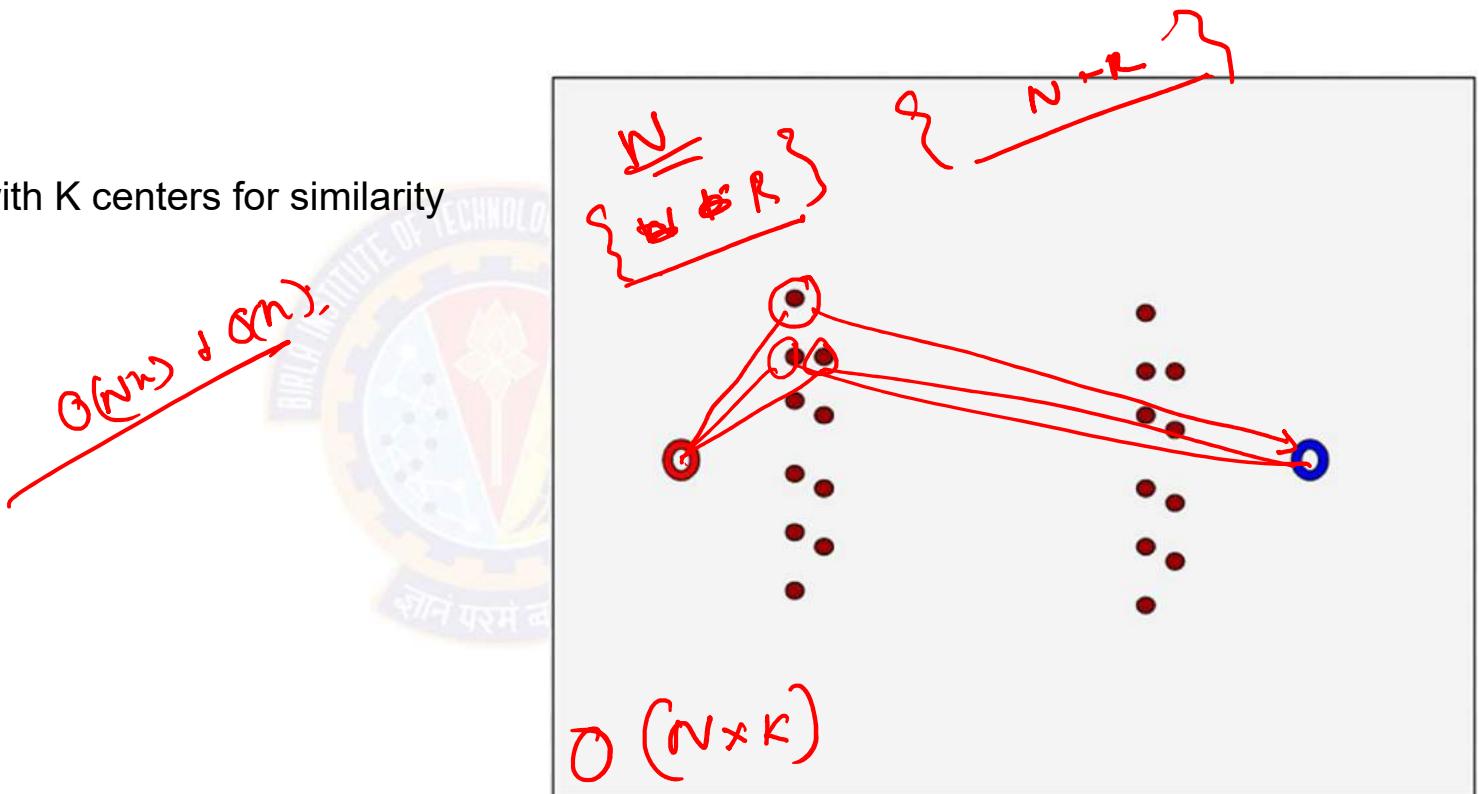
- Compare each point with K centers for similarity
- $O(NK)$ ✓

In each M-Step:

- We recompute $\{\mu_k\}$
- $O(N)$ ✓

K-Means, iterating t times:

$$O(tNK)$$



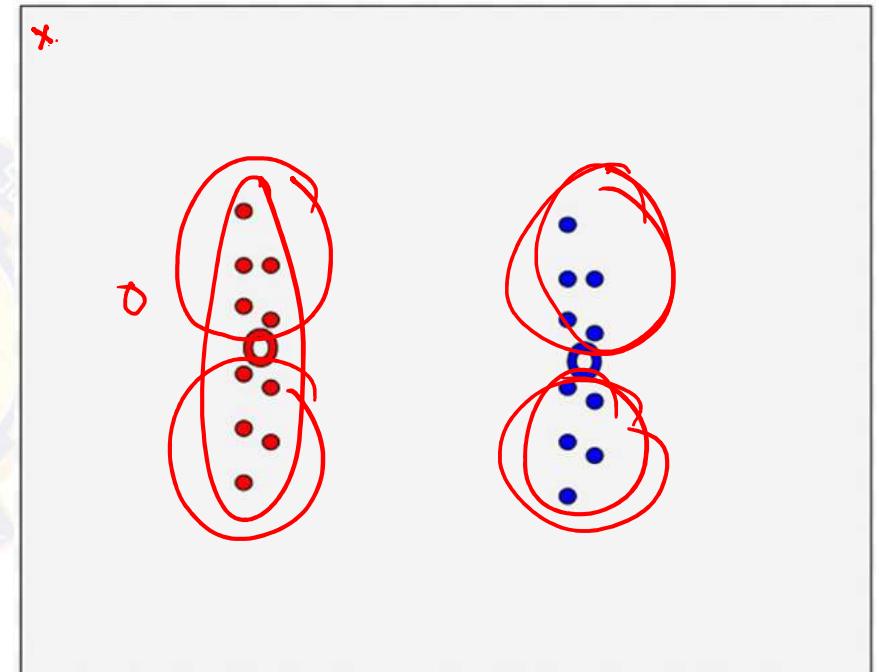
K-Means Algorithm

Some more points:

K-Means is applicable when

- Mean is defined for X
- K is provided as input

K-Means is extremely sensitive to noise/outlier points



K-Means Algorithm

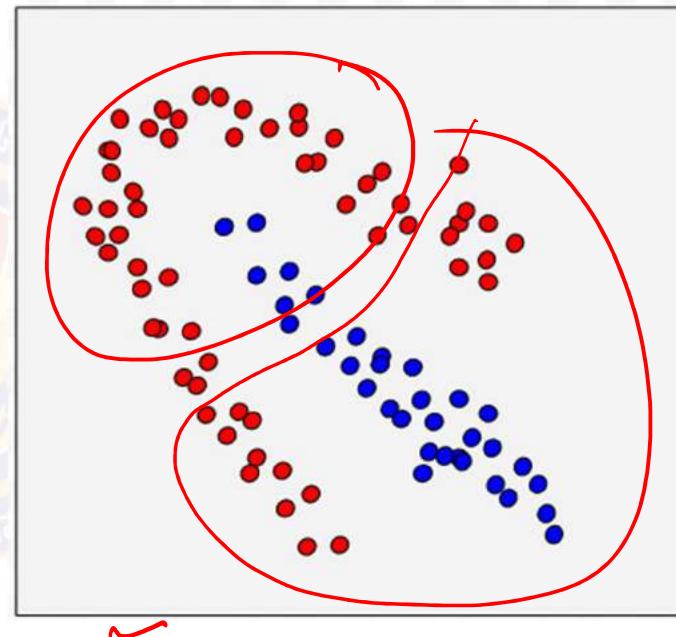
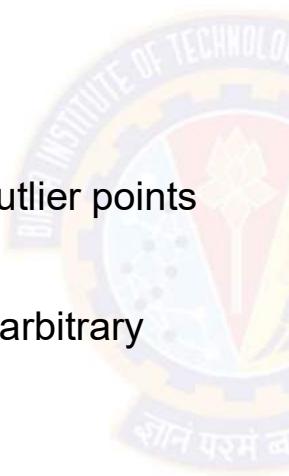
Some more points:

K-Means is applicable when

- Mean is defined for X
- K is provided as input

K-Means is extremely sensitive to noise/outlier points

K-Means is not suitable to find clusters of arbitrary shape

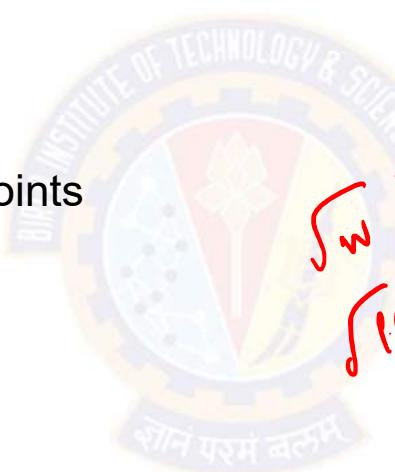


Number of Clusters

Basic Approaches

Empirical method:

of clusters $\approx \sqrt{n}/2$ for a dataset of n points



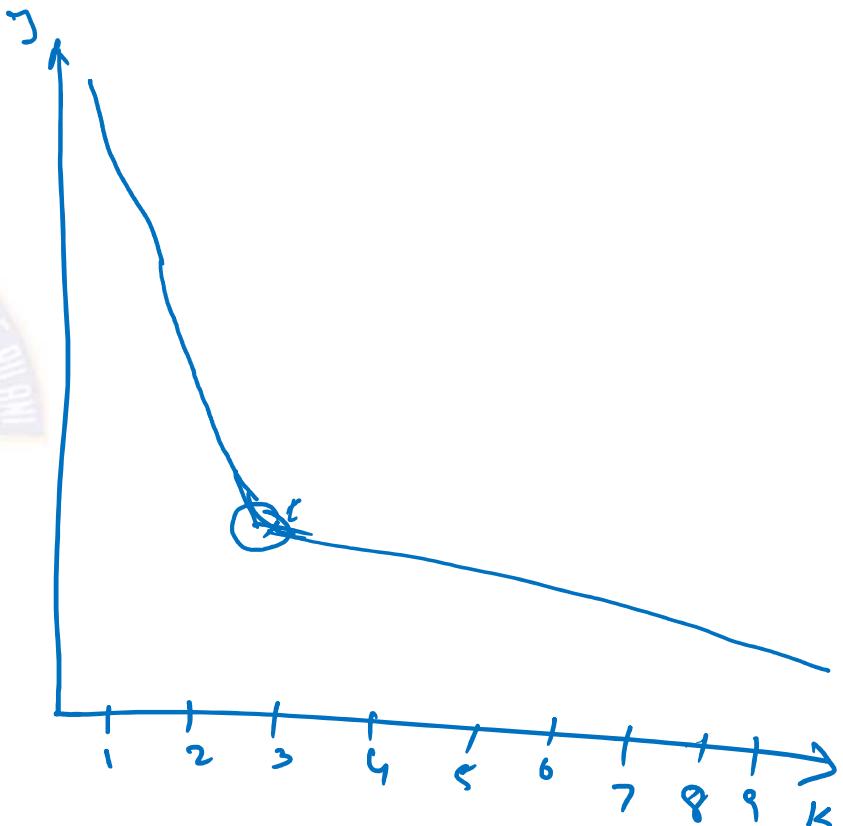
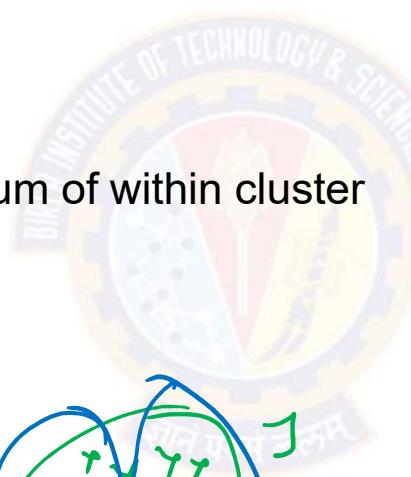
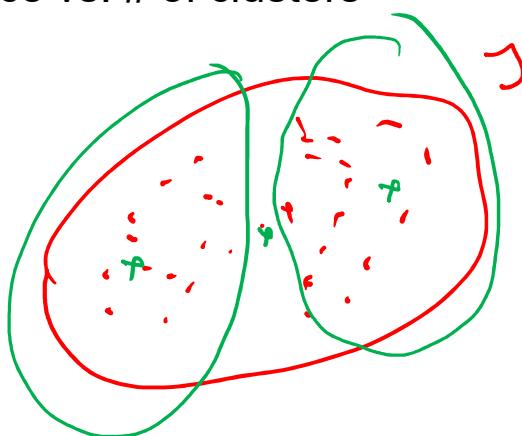
$$\begin{aligned}\sqrt{w}/2 \\ \sqrt{100}/2 = 5\end{aligned}$$

Number of Clusters

Basic Approaches

Elbow method:

Use the turning point in the curve of sum of within cluster variance vs. # of clusters

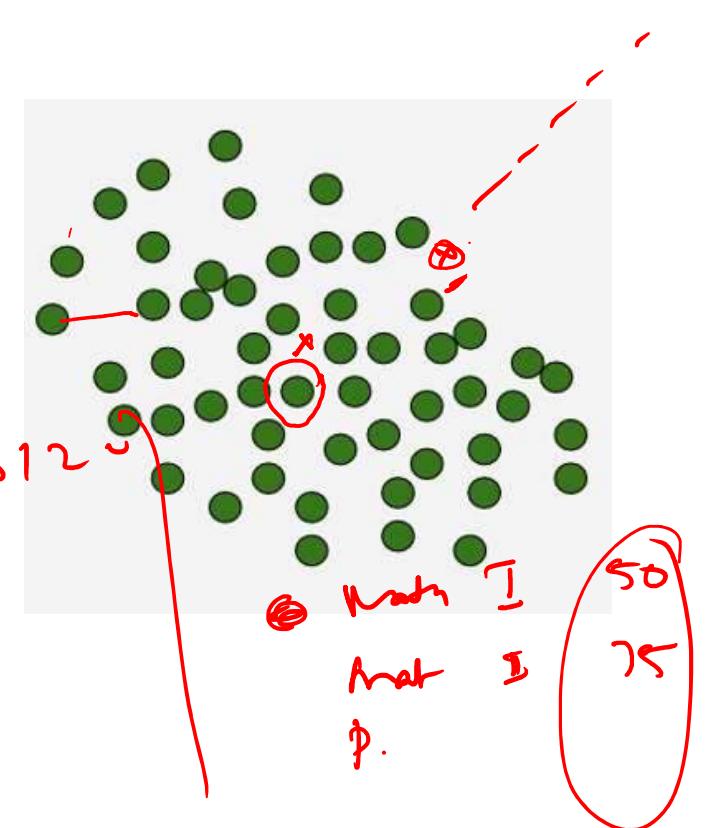
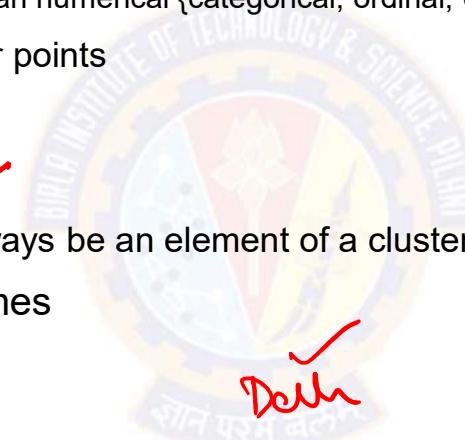


K-Means Algorithm - An enhancement



- Use of Euclidean distance in K-Means has the following issues
 - Restricts X to points only from \mathcal{R}^N .
 - Features could have types other than numerical {categorical, ordinal, etc. }
 - Algorithm is quite sensitive to outlier points
- Make two changes
 - Abstract dissimilarity to $\mathcal{V}(x, x')$
 - Ensure the cluster prototypes to always be an element of a cluster
- That is, the distortion function becomes

$$\tilde{J} = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathcal{V}(x_n, \mu_k)$$

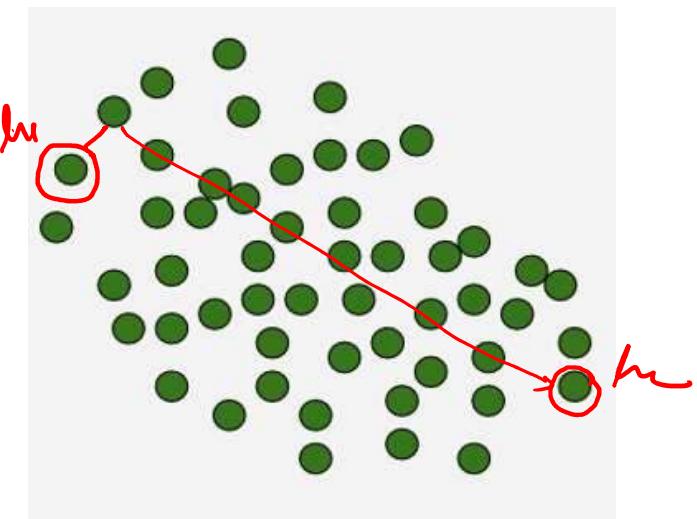


K-Means Algorithm - An enhancement

- Use of Euclidean distance in K-Means has the following issues
 - Restricts X to points only from \mathcal{R}^N .
 - Features could have types other than numerical {categorical, ordinal, etc. }
 - Algorithm is quite sensitive to outliers.
- Make two changes
 - Abstract dissimilarity to \mathcal{V} (Recall proximity measures for binary, categorical, continuous attributes covered in feature engineering.)
 - Ensure the cluster prototype
- That is, the distortion function becomes

$$\tilde{J} = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

Recall proximity
measures for
binary, categorical,
continuous
attributes covered
in feature
engineering.
cluster



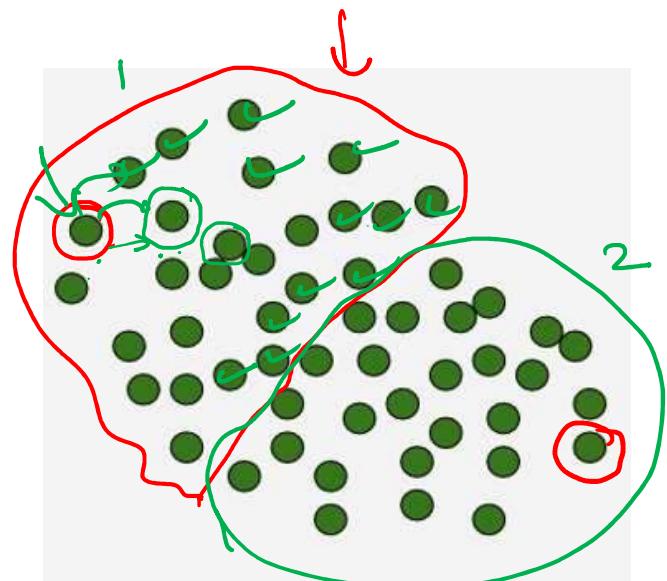
K-Means Algorithm - An enhancement

- Use of Euclidean distance in K-Means has the following issues
 - Restricts X to points only from \mathcal{R}^N .
 - Features could have types other than numerical {categorical, ordinal, etc. }
 - Algorithm is quite sensitive to outlier points
- Make two changes
 - Abstract dissimilarity to \mathcal{V} ()
 - Ensure the cluster prototypes are well-defined
- That is, the distortion function becomes

$$\tilde{J} = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

E-Step is straight forward with those measures

cluster



K-Means Algorithm - An enhancement

M-Step:

- 1: For each non representative objects O_x in a cluster represented by O_m ,
 - 1.1: Verify if swapping the representative from O_m to O_x improves the cost
 - 1.1.1: If yes, perform the swap, else
 - 1.2: Repeat 1.1 for all the remaining non-representative objects in the cluster

- Let N_k be the number of objects in kth cluster :

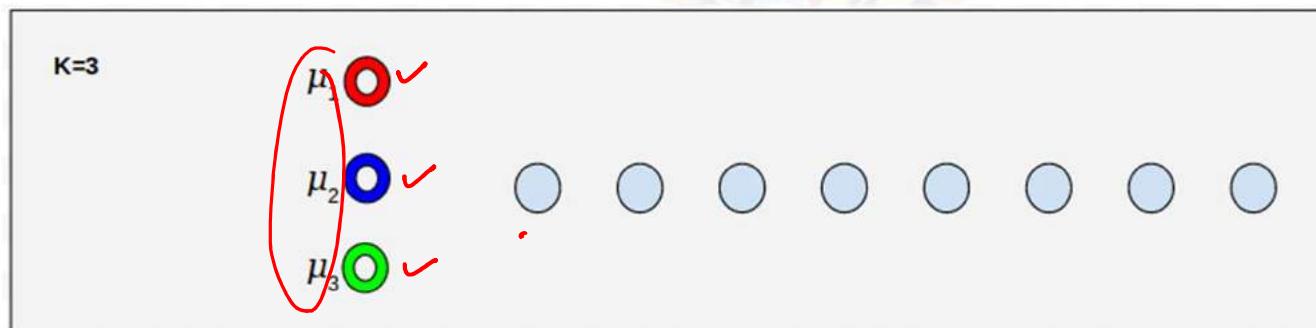
M-Step involves $O(N_k^2)$ evaluations of $\nu(\cdot)$

K-Medoids

- More robust to outliers, works with non-numeric data at a higher computational cost.

K-Means Algorithm - Sequential Update

- The algorithm discussed so far is a batch algorithm
 - Needs processing the entire data set together
 - Does not scale for large data sets, data streams
- Use sequential / On-line version of learning algorithms
 - Consider data points one at a time
 - Update model parameters each time you see a data point.

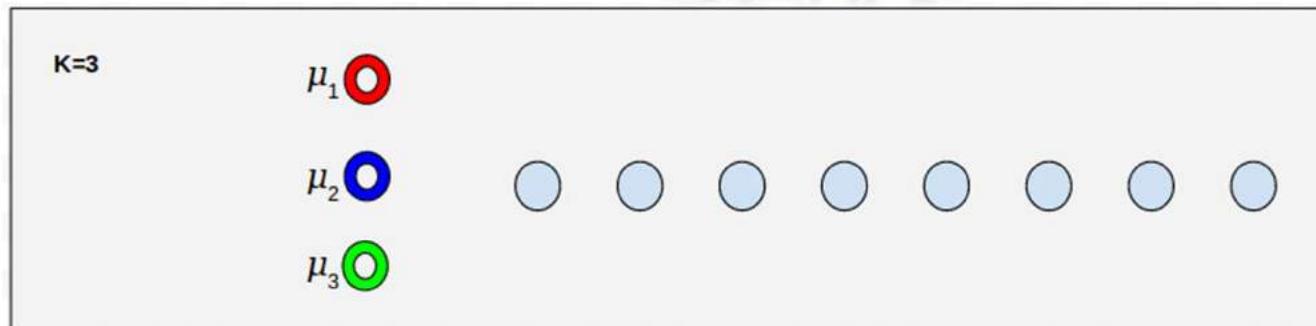


K-Means Algorithm - Sequential Update

- Uses stochastic gradient descent / sequential gradient descent to update the model parameters
 - Let w be the model parameter , η being learning rate, ΔE_n being directional error
 - $w^{(0)}$ will be an appropriate initial vector

$$w^{(t+1)} = w^{(t)} - \eta \Delta \eta_n E_n$$

$$\frac{1}{N}$$

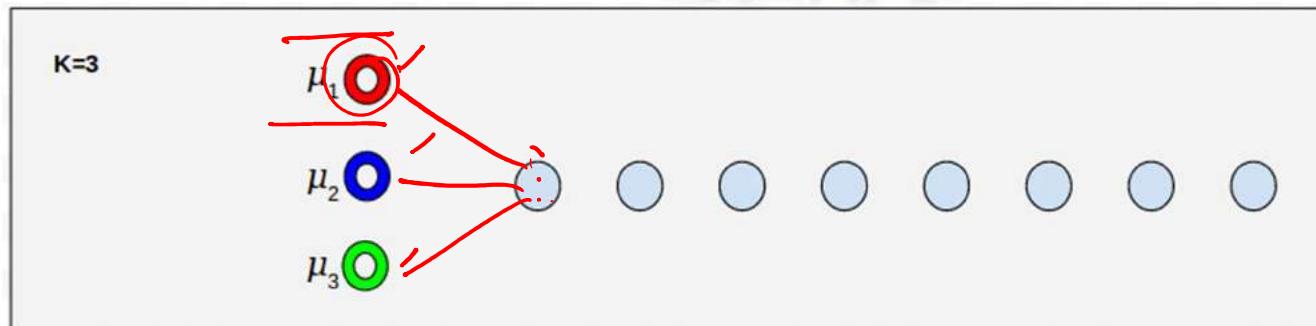


K-Means Algorithm - Sequential Update

- Uses stochastic gradient descent / sequential gradient descent to update the model parameters
 - Sequential update for k-means

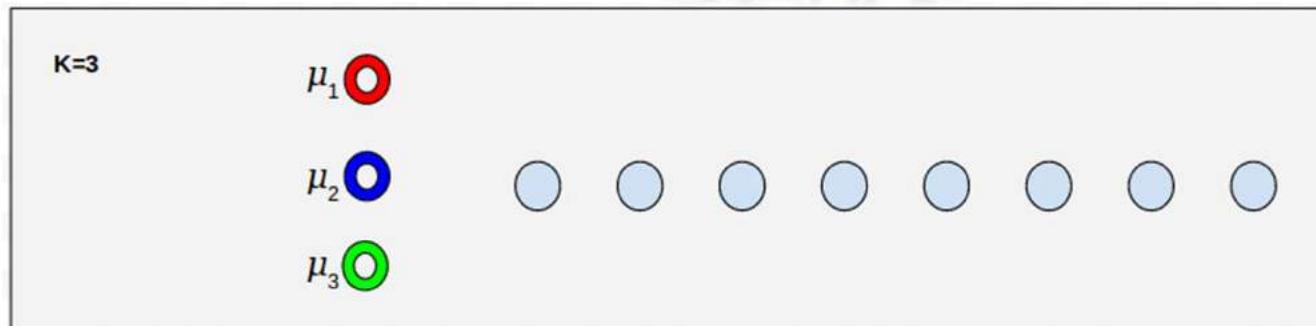
$$\mu_k^{(\text{new})} = \mu_k^{(\text{old})} - \eta \eta_n (\mu_k^{(\text{old})} - x_n)$$

[for each x_n , η_n is decreased]



K-Means Algorithm - Sequential Update

- Uses stochastic gradient descent / sequential gradient descent to update the model parameters
 - Sequential update for k-means
$$\mu_k^{(\text{new})} = \mu_k^{(\text{old})} - \eta \eta_n (\mu_k^{(\text{old})} - x_n)$$
[for each x_n , η_n is decreased]

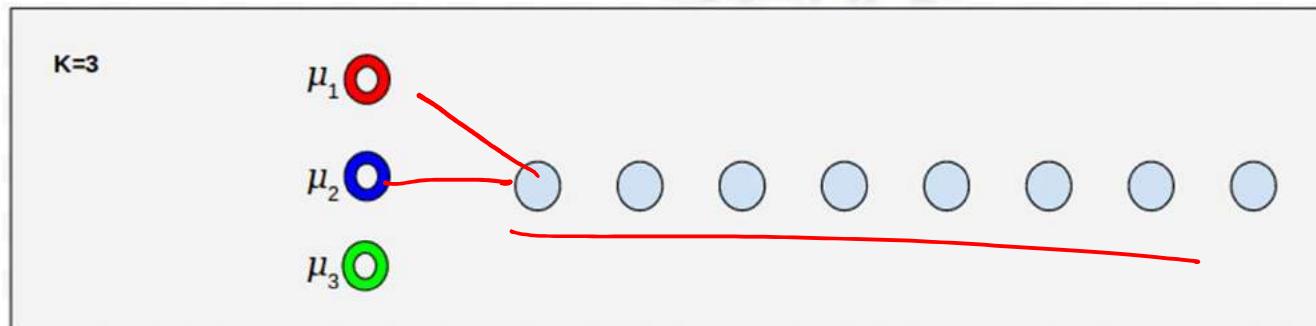


K-Means Algorithm - Sequential Update

Algorithm :

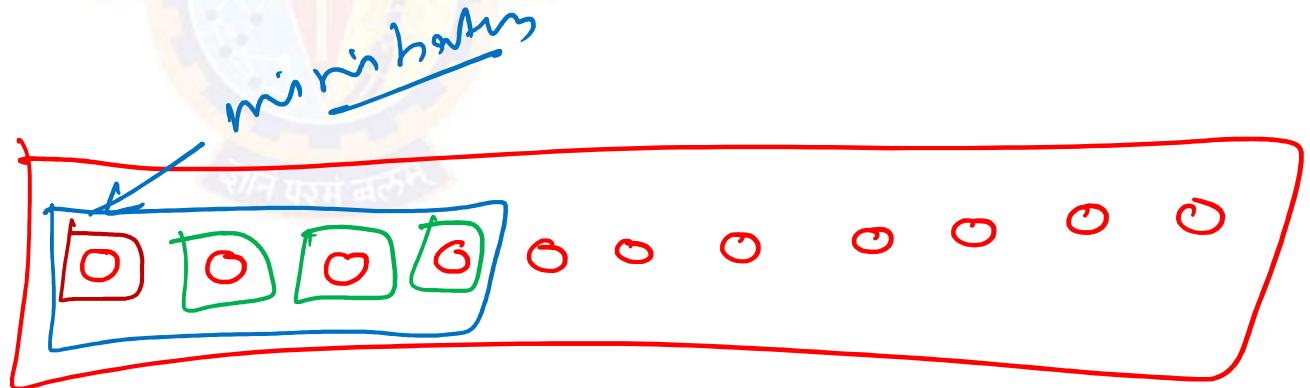
1. Make Initial guesses for $\{\mu_k\}$
2. For each new example, x_n :
 - a. Find the closest μ_k for x_n
 - b. Replace μ_k as

$$\mu_k^{(\text{new})} = \mu_k^{(\text{old})} - \eta_n (\mu_k^{(\text{old})} - x_n)$$



Mini-batch K-Means

- The batch algorithm is too slow to scale for large data sets
- The online version presented, though scales, produces poor quality of clustering
- Mini-batch K-Means* provides better runtime performances over classic k-means and also produces better quality clusters than the online version.
 - MBK-Means is more robust to random noises in the data points than the online version

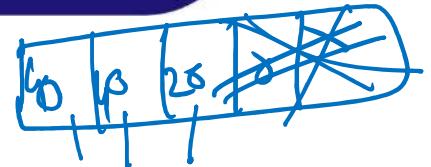


Mini-batch K-Means

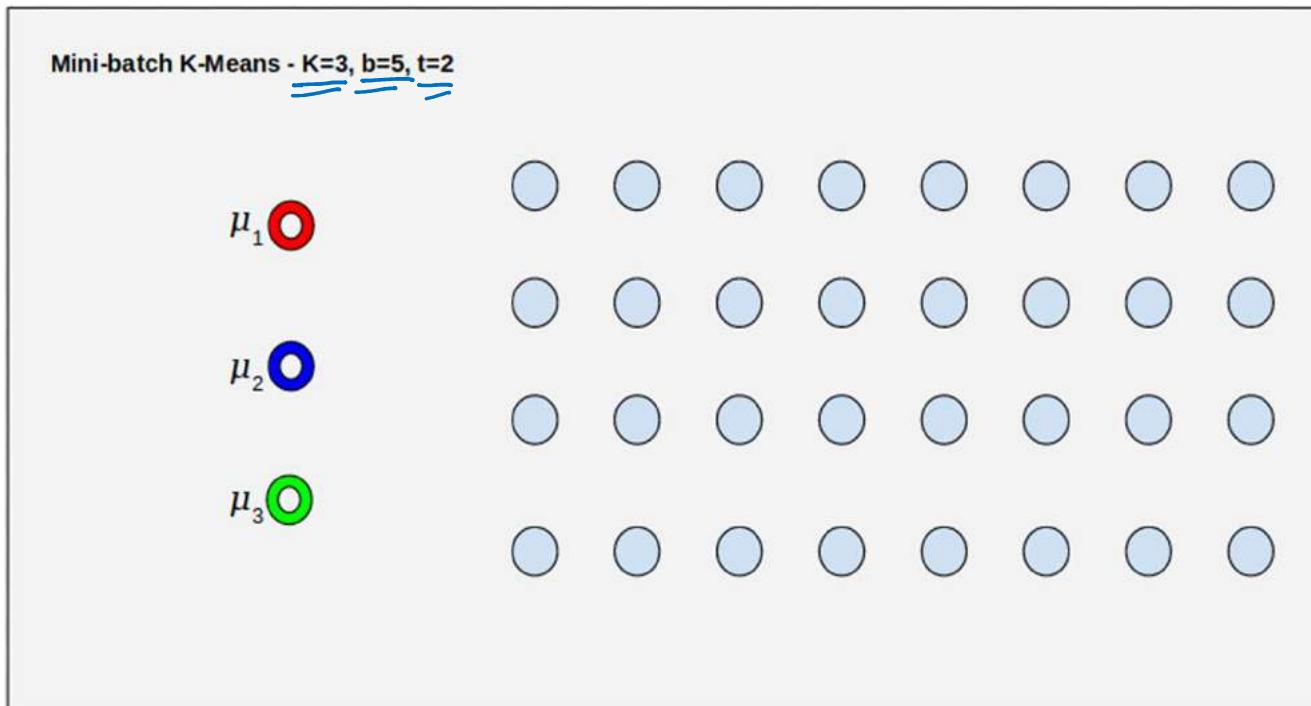
- The batch algorithm is too slow to scale for large data sets
- The online version presented, though scales, produces poor quality of clustering
- Mini-batch K-Means* provides better runtime performances over classic k-means and also produces better quality clusters than the online version.
 - MBK-Means is more robust to random noises in the data points than the online version
- Algorithm takes as input
 - X, K, b (mini batch size), t (iterations)

Mini-batch K-Means - Algorithm

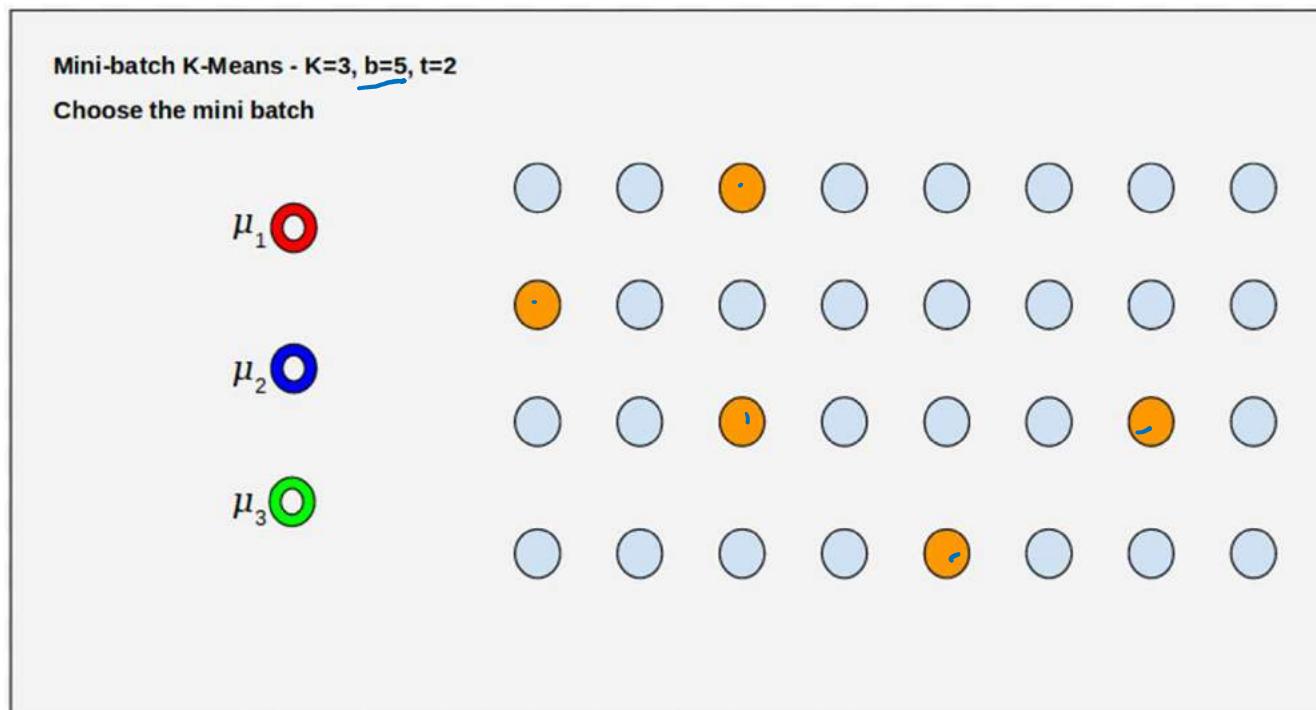
1. Initialize $\{\mu_k\}$ by randomly picking k instances from X ✓
2. Initialize $v[1 \dots k]$ to 0's
3. For $i = 1 \dots t$ do
 - a. $M \leftarrow$ pick b examples randomly from X [choosing the mini-batch]
 - b. For each $x \in M$ do
 - i. $d[x] \leftarrow f(\{\mu_k\}, x)$ [f returns μ_k closest to x]
 - c. For each $x \in M$ do
 - i. $\mu_k \leftarrow d[x]$ [Get the μ_k closest to x]
 - ii. $v[\mu_k] \leftarrow v[\mu_k] + 1$ [maintain the count of x 's closer to μ_k]
 - iii. $\eta = 1 / v[\mu_k]$ [per-center learning rate]
 - iv. $\mu_k^{(new)} = \mu_k^{(old)} - \eta (\mu_k^{(old)} - x_n)$ [gradient step - same as prev. alg]
4. End For



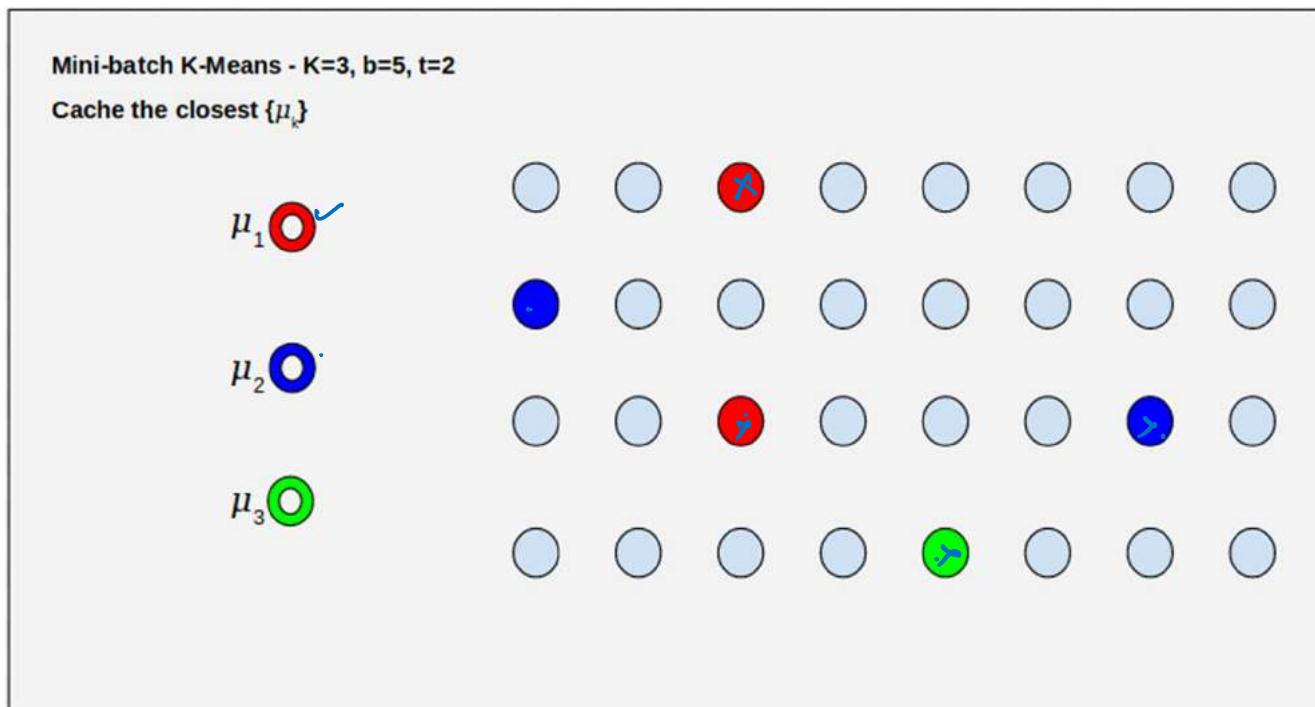
Mini-batch K-Means - Algorithm



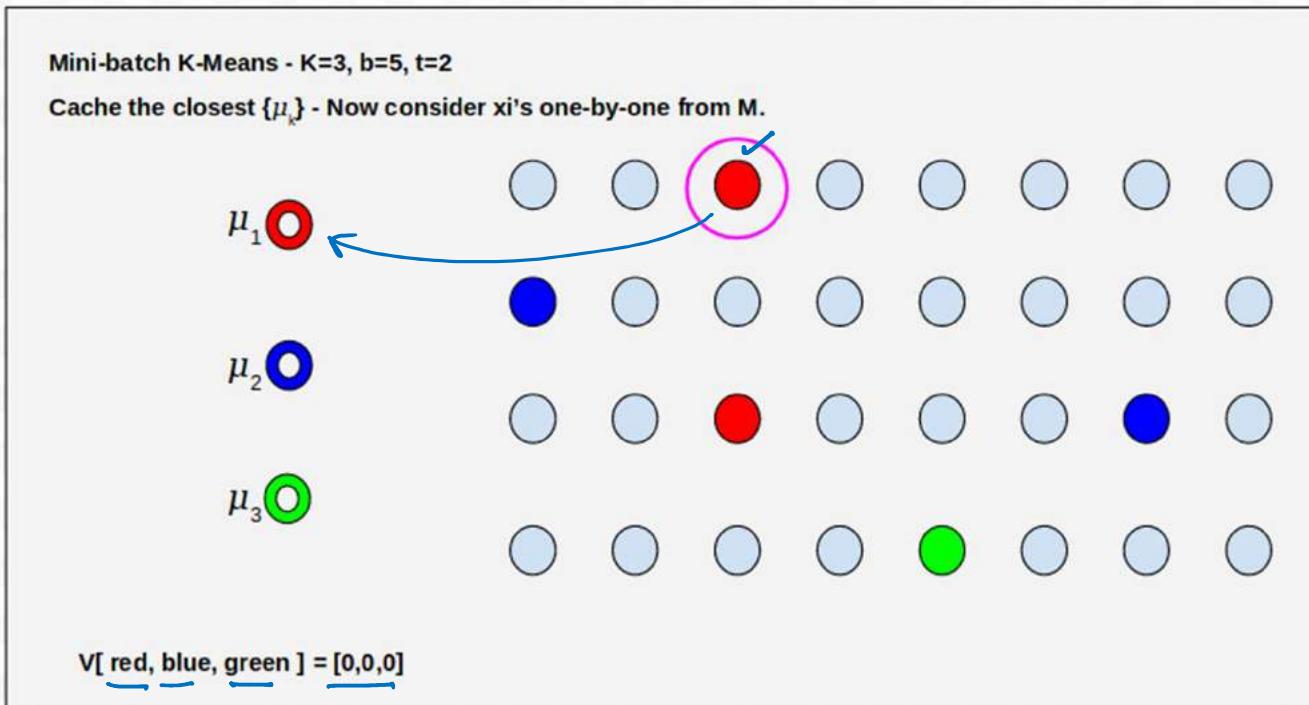
Mini-batch K-Means - Algorithm



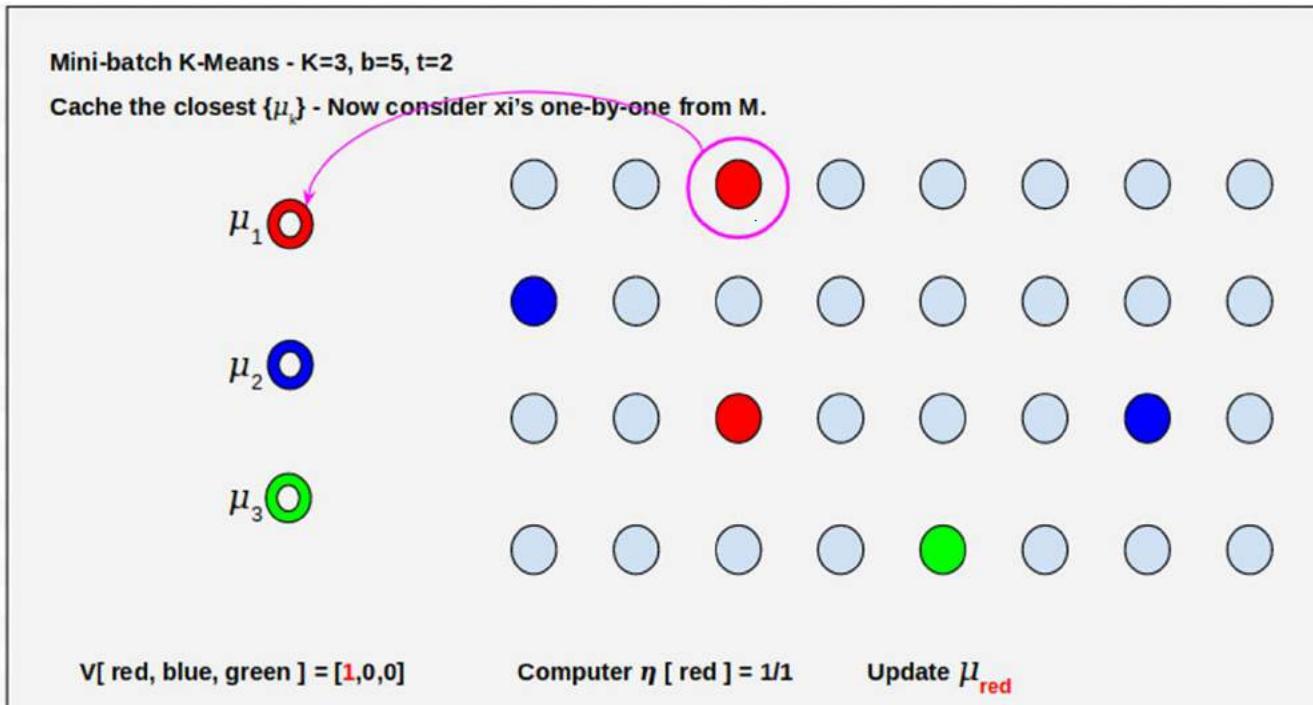
Mini-batch K-Means - Algorithm



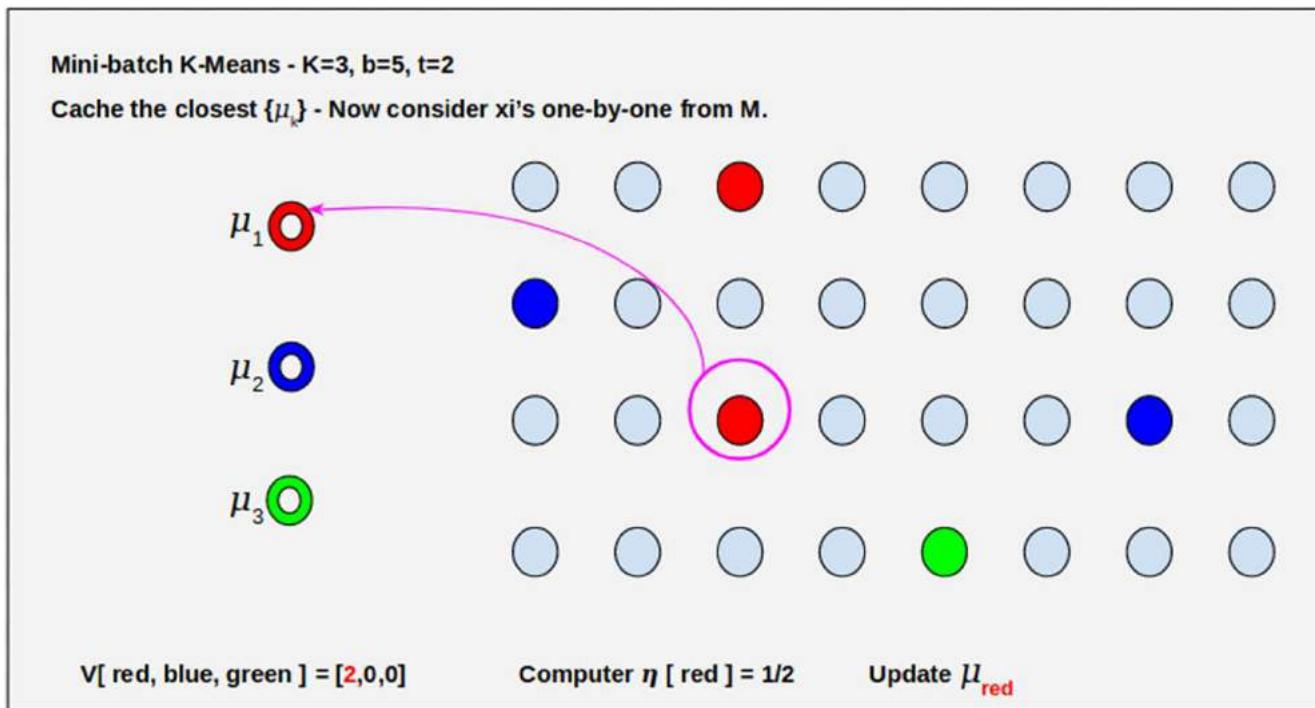
Mini-batch K-Means - Algorithm



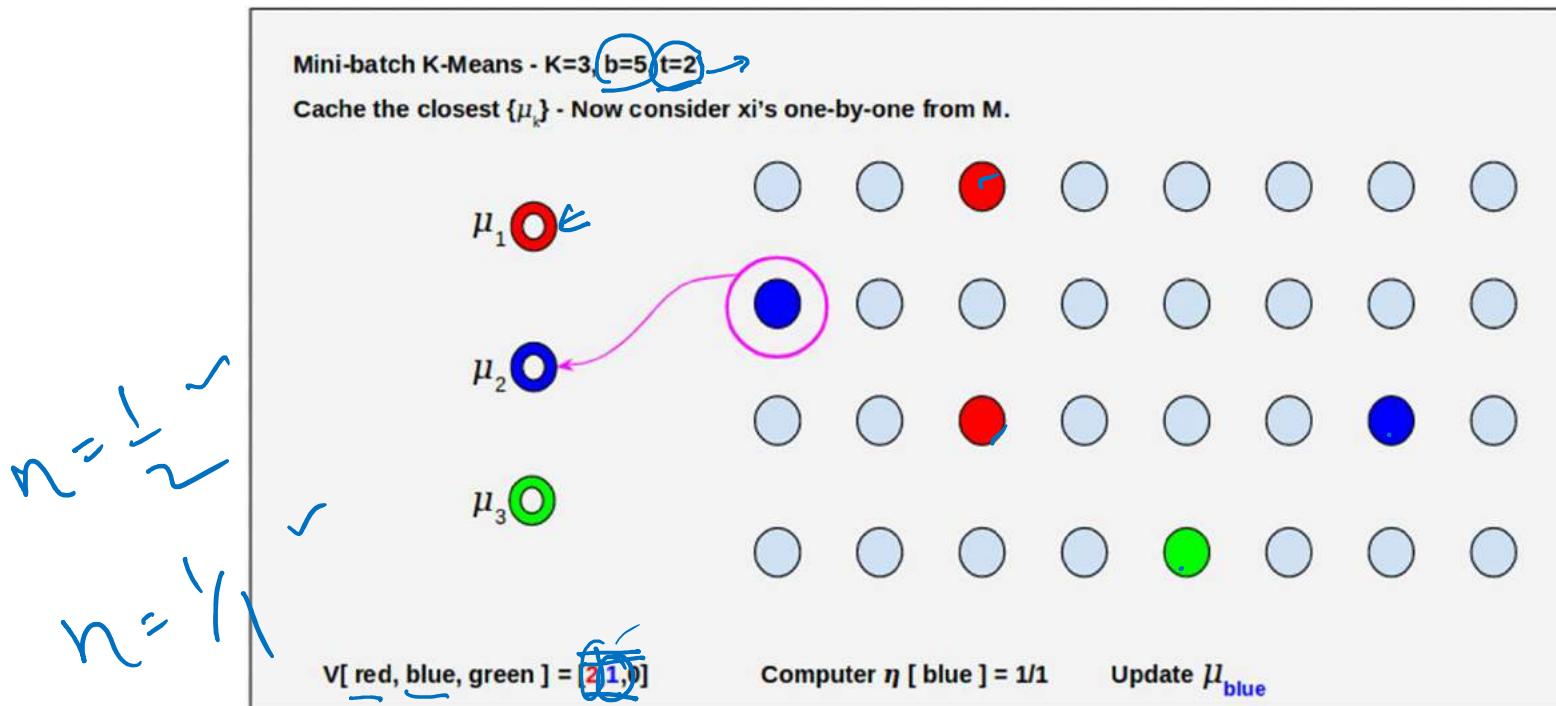
Mini-batch K-Means - Algorithm



Mini-batch K-Means - Algorithm



Mini-batch K-Means - Algorithm



Repeat this for all the elements in chosen M, and proceed to the next t.

Note, V is not reset between iterations, leading to learning rate decreases monotonically with examples seen for each centers.

Mini-batch K-Means - Algorithm

1. Initialize $\{\mu_k\}$ by randomly picking k instances from X
2. Initialize $v[1\dots k]$ to 0's
3. For $i = 1 \dots t$ do
 - a. $M \leftarrow$ pick b examples randomly from X [choosing the mini-batch] ✓
 - b. For each $x \in M$ do
 - i. $d[x] \leftarrow f(\{\mu_k\}, x)$ [f returns μ_k closest to x]
 - c. For each $x \in M$ do
 - i. $\mu_k \leftarrow d[x]$ [Get the μ_k closest to x]
 - ii. $v[\mu_k] \leftarrow v[\mu_k] + 1$ [maintain the count of x 's closer to μ_k]
 - iii. $\eta = 1 / v[\mu_k]$ [per-center learning rate]
 - iv. $\mu_k^{(new)} = \mu_k^{(old)} - \eta (\mu_k^{(old)} - x_n)$ [gradient step - same as prev. alg]
4. End For

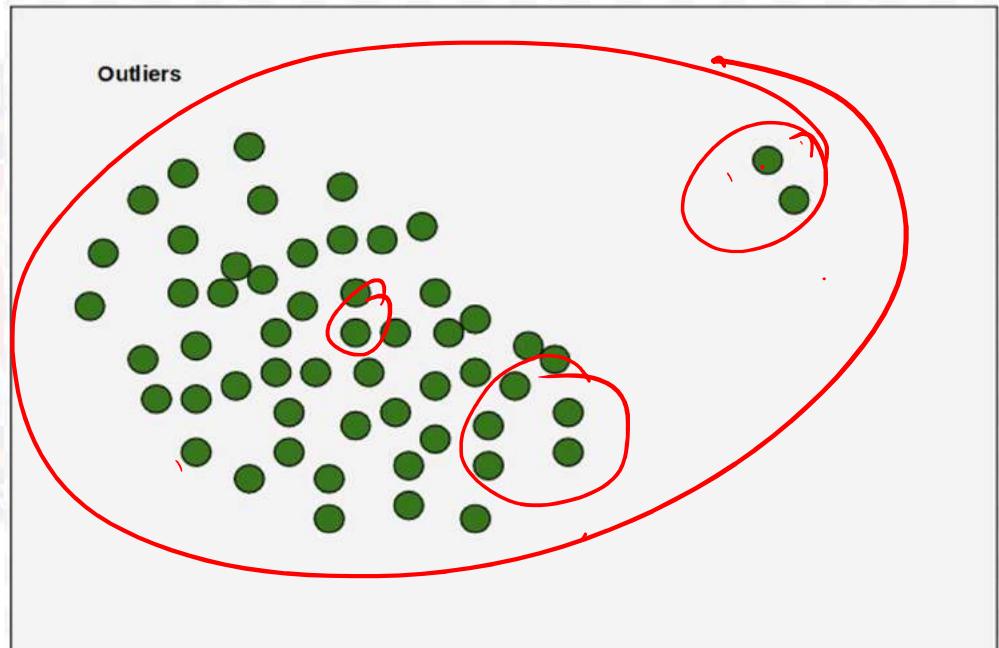
In this segment

- What are outliers ?
- K-means (/ cluster based approaches) for outliers detection



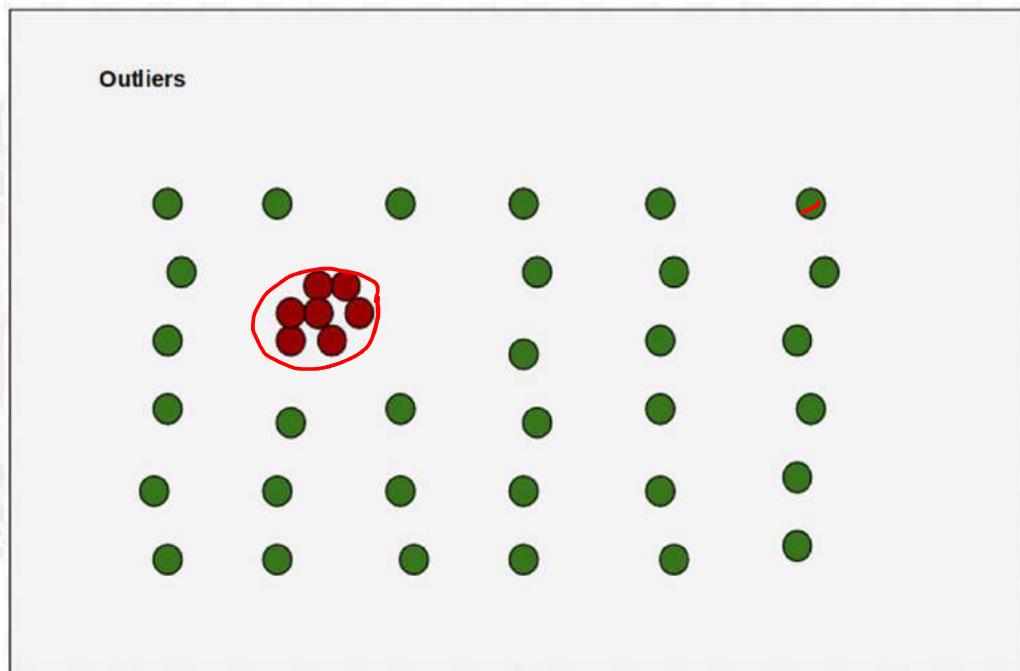
What is an Outlier?

- An outlier is a data point which deviate significantly from other points
 - Interesting as these points are suspected not being generated by the same mechanisms as the rest of the data
- Noise present in data impacts outlier detection
- Ex:
 - Suspected fraudulent credit card transaction
 - Intrusion detection in a network



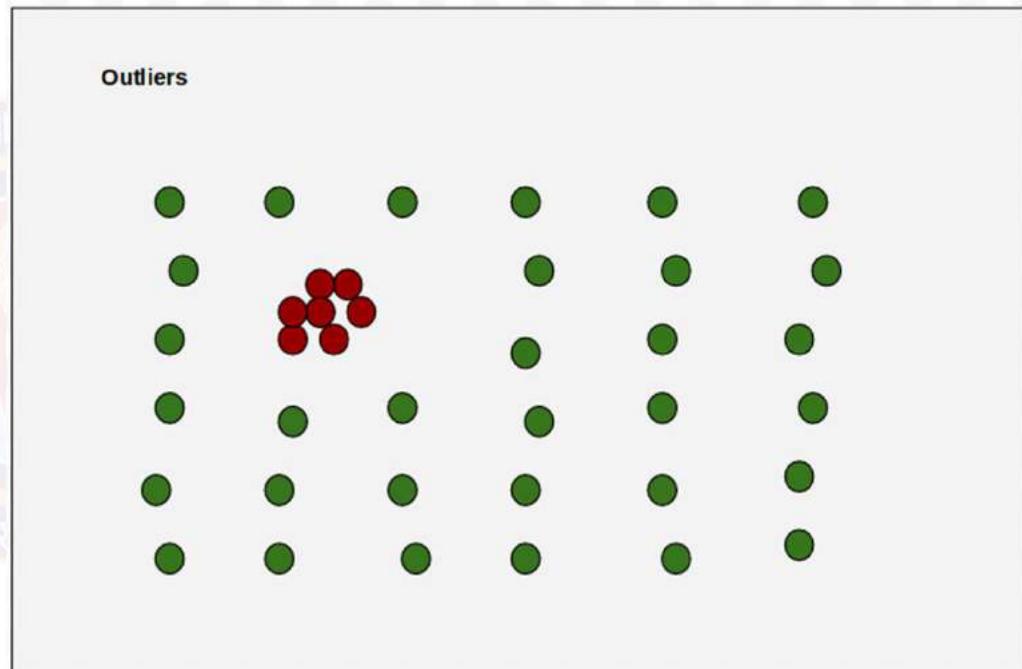
What is an Outlier?

- An outlier is a data point which deviate significantly from other points
 - Interesting as these points are suspected not being generated by the same mechanisms as the rest of the data
- Noise present in data impacts outlier detection
- Ex:
 - Suspected fraudulent credit card transaction
 - Intrusion detection in a network



Approaches

- Supervised ✓
 - Outlier detection as a two-class classification problem
- Unsupervised
 - Normal objects are somewhat clustered
 - Assumption
 - not true all the time
- Semi supervised approaches ✓



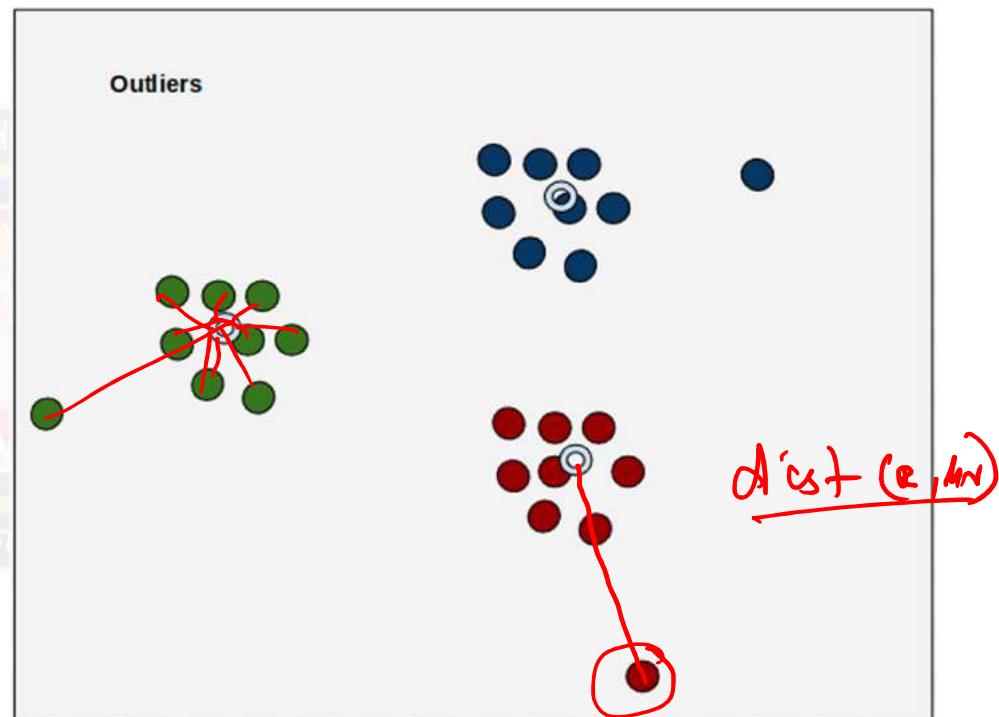
K-Means for detecting outliers

Let

- $\text{dist}(x, \mu_k)$ be the distance of a point x , assigned to cluster k to its center μ_k .
- $L\mu_k$ be the average distance of all the points assigned to cluster k with its center

The ratio $\frac{\text{dist}(x, \mu_k)}{L\mu_k}$ for each point is the outlier score for each point.

- Higher the ratio for a point x , more likely x is a outlier



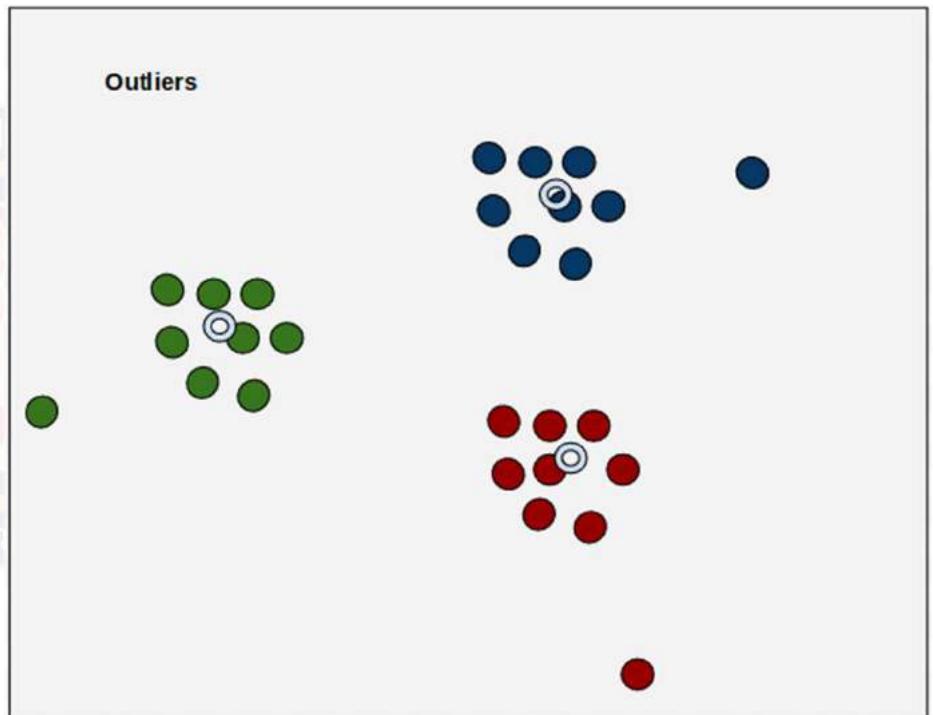
Other clustering approaches to outlier detection

Let

- $\text{dist}(x, \mu_k)$ be the distance of a point x , assigned to cluster k to its center μ_k .
- $L\mu_k$ be the average distance of all the points assigned to cluster k with its center

The ratio $\text{dist}(x, \mu_k) / L\mu_k$ for each point is the outlier score for each point.

- Higher the ratio for a point x , more likely x is a outlier



Agenda

- Multivariate Gaussian
- What is Mixtures of Gaussian (MoG)?
- Why?

Multivariate Gaussian

- Gaussian for a single variable x :

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

- Given the x is D-dimensional vector & x is Gaussian distributed :

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

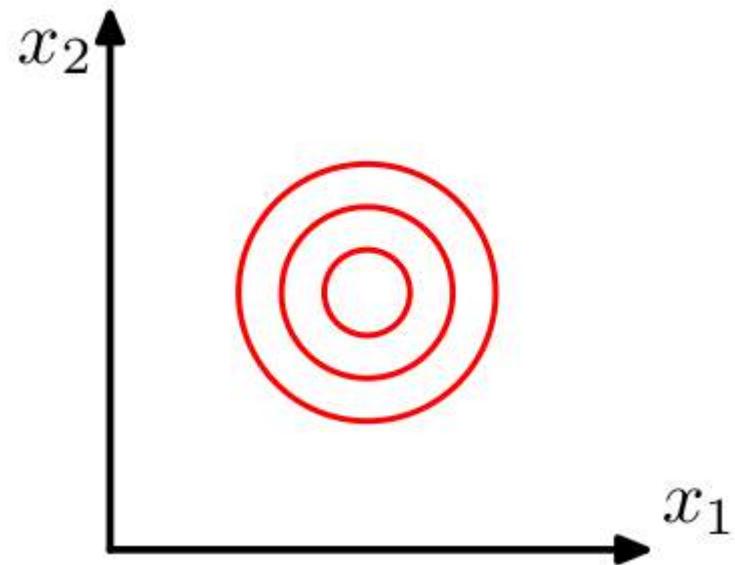
Where

$\boldsymbol{\mu}$: D-dimensional mean vector

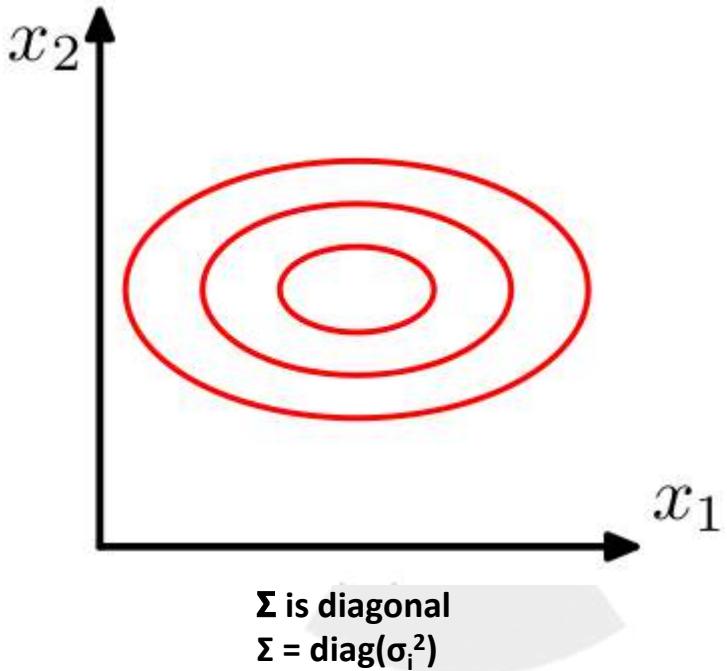
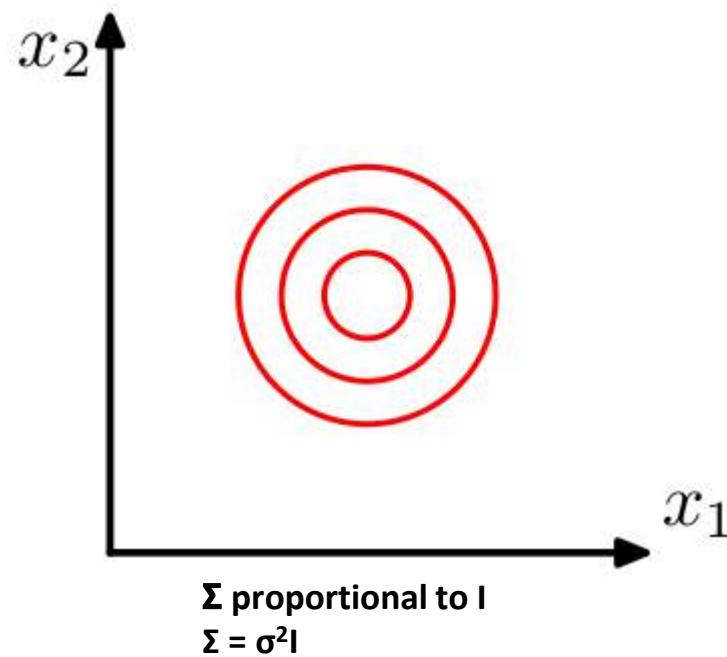
$\boldsymbol{\Sigma}$: $D \times D$ covariance matrix

$|\boldsymbol{\Sigma}|$: Determinant of $\boldsymbol{\Sigma}$

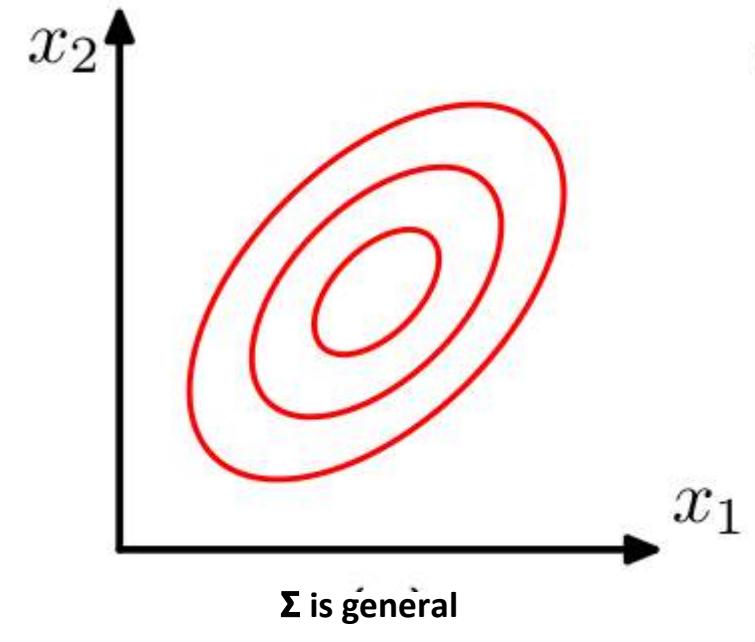
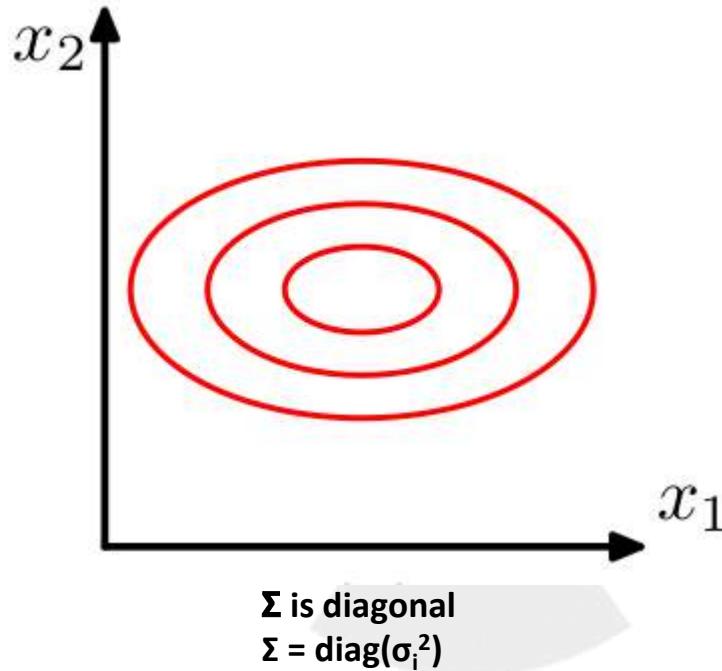
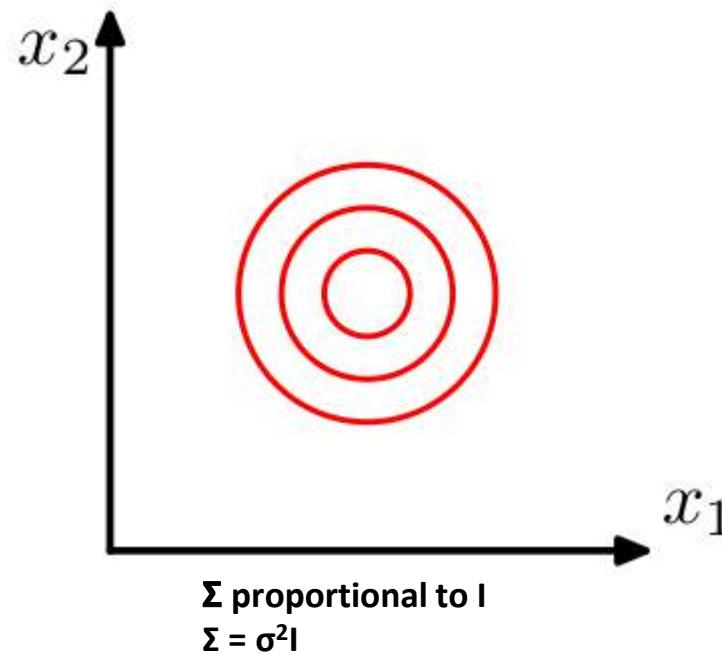
Multivariate Gaussian



Multivariate Gaussian

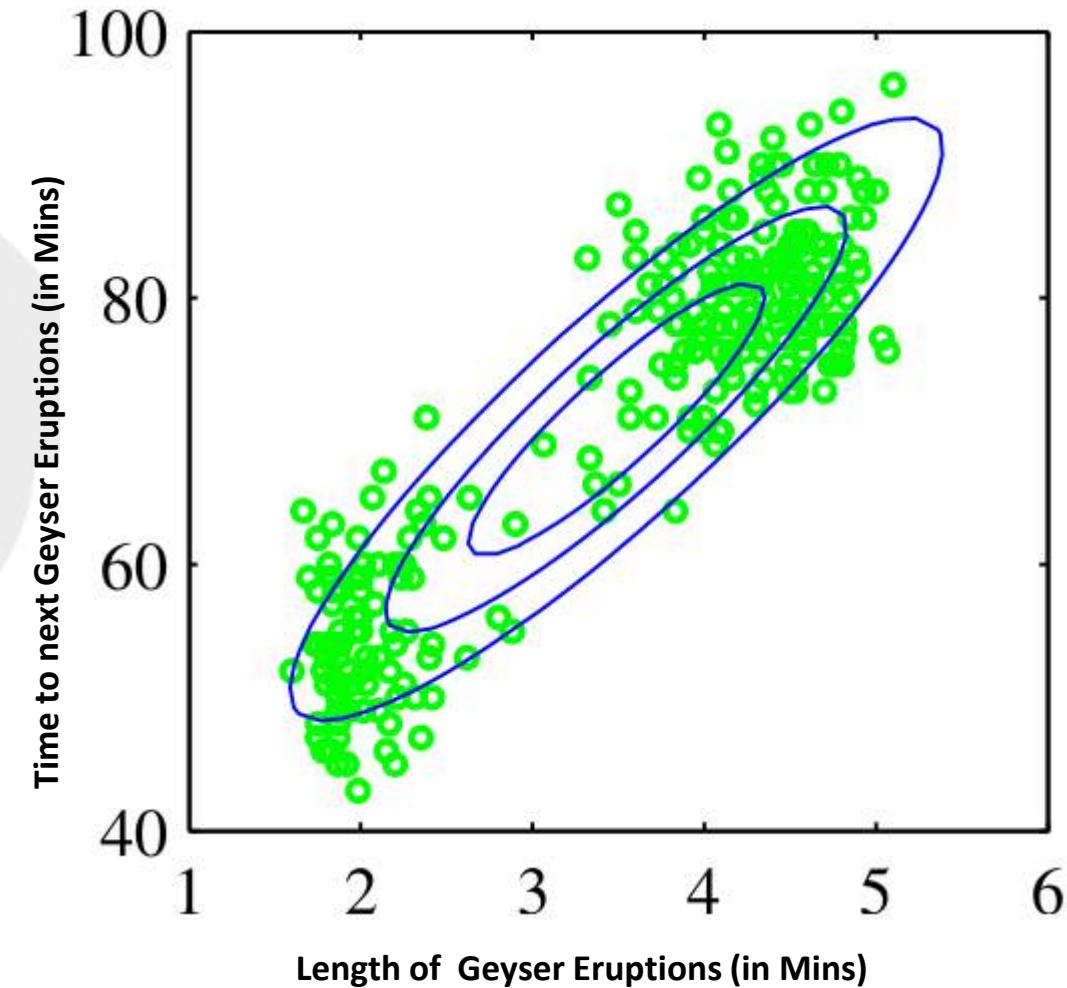
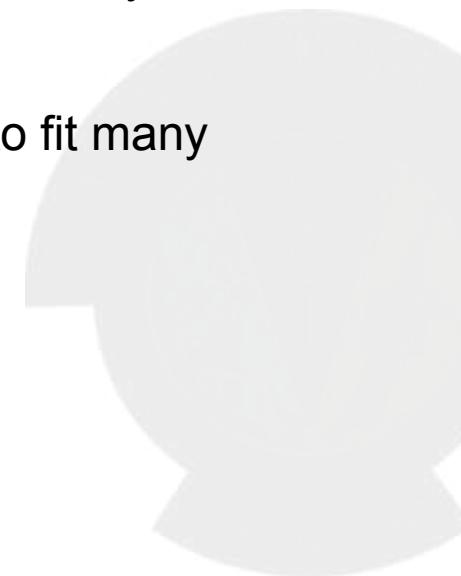


Multivariate Gaussian



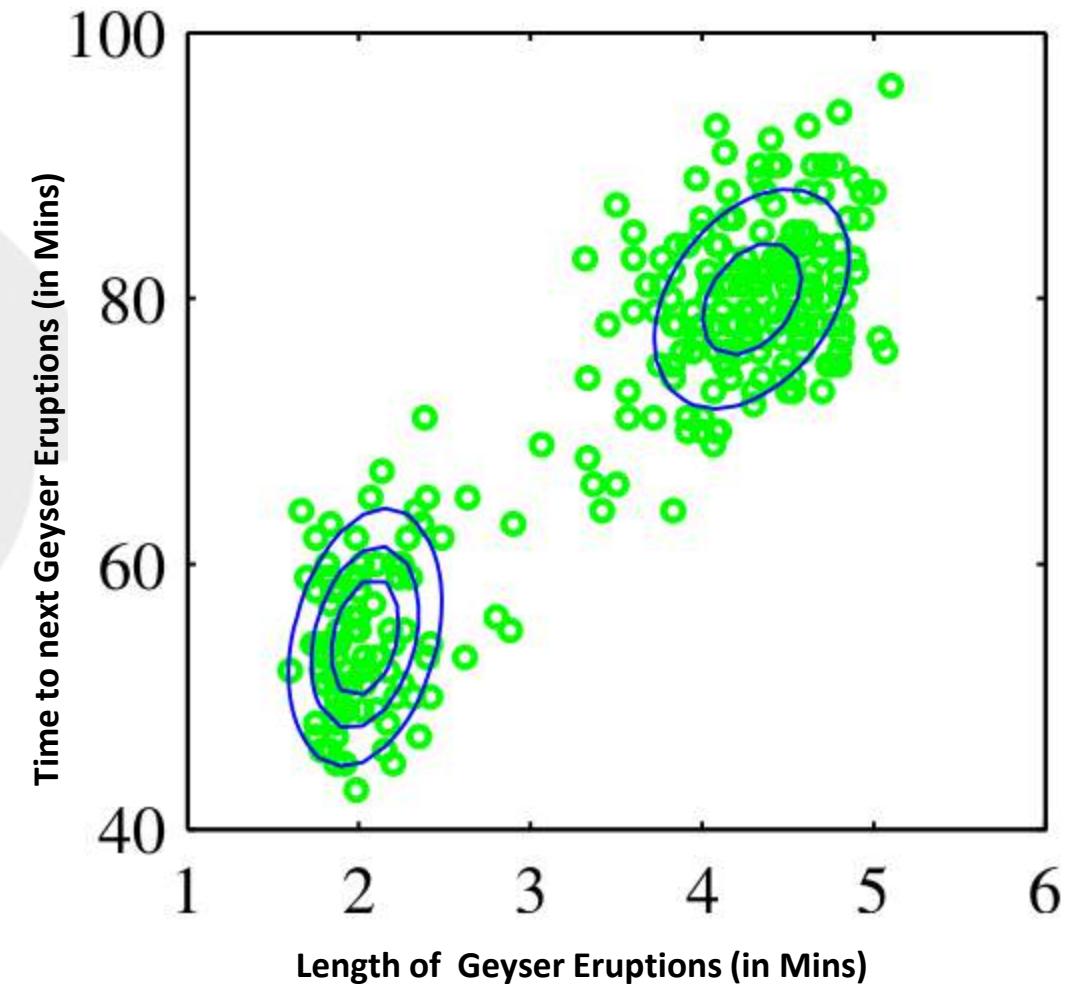
Multimodal Data

- Gaussian distribution is that it is intrinsically unimodal
- A single distribution is not sufficient to fit many real data sets which is multimodal



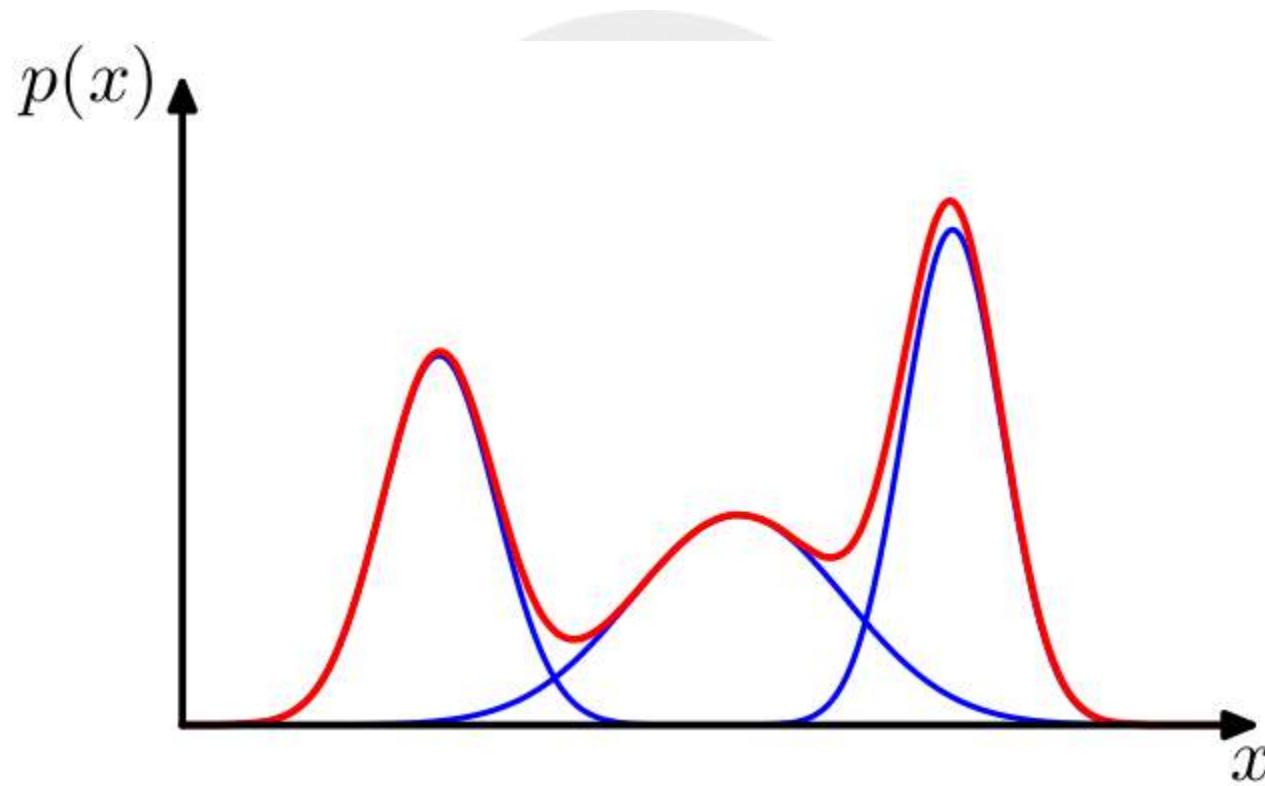
Multimodal Data

- A single distribution is not sufficient to fit many real data sets



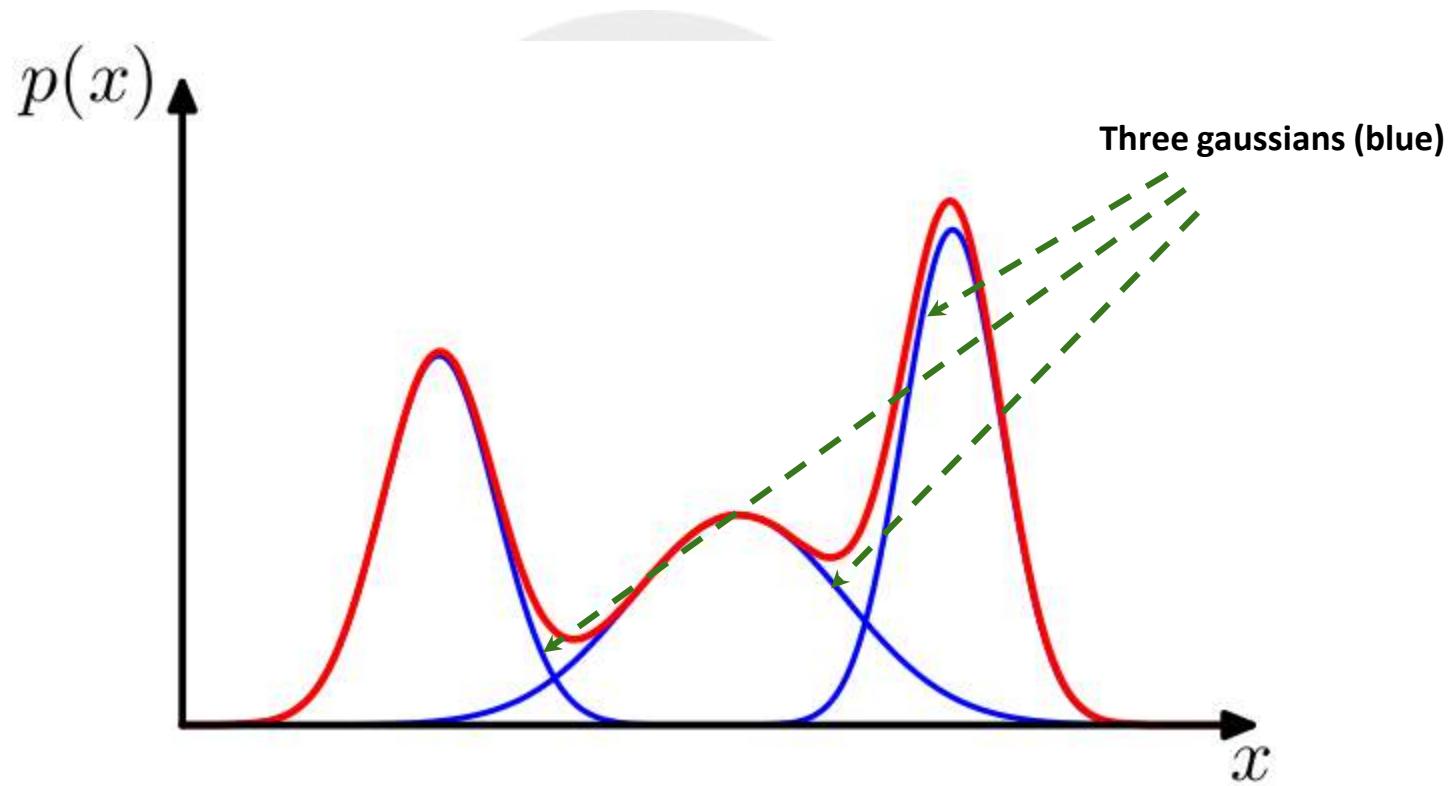
Mixture Distributions

- Mixture Distributions: Linear combinations of basic distributions (such as Gaussian) to approximate complex density



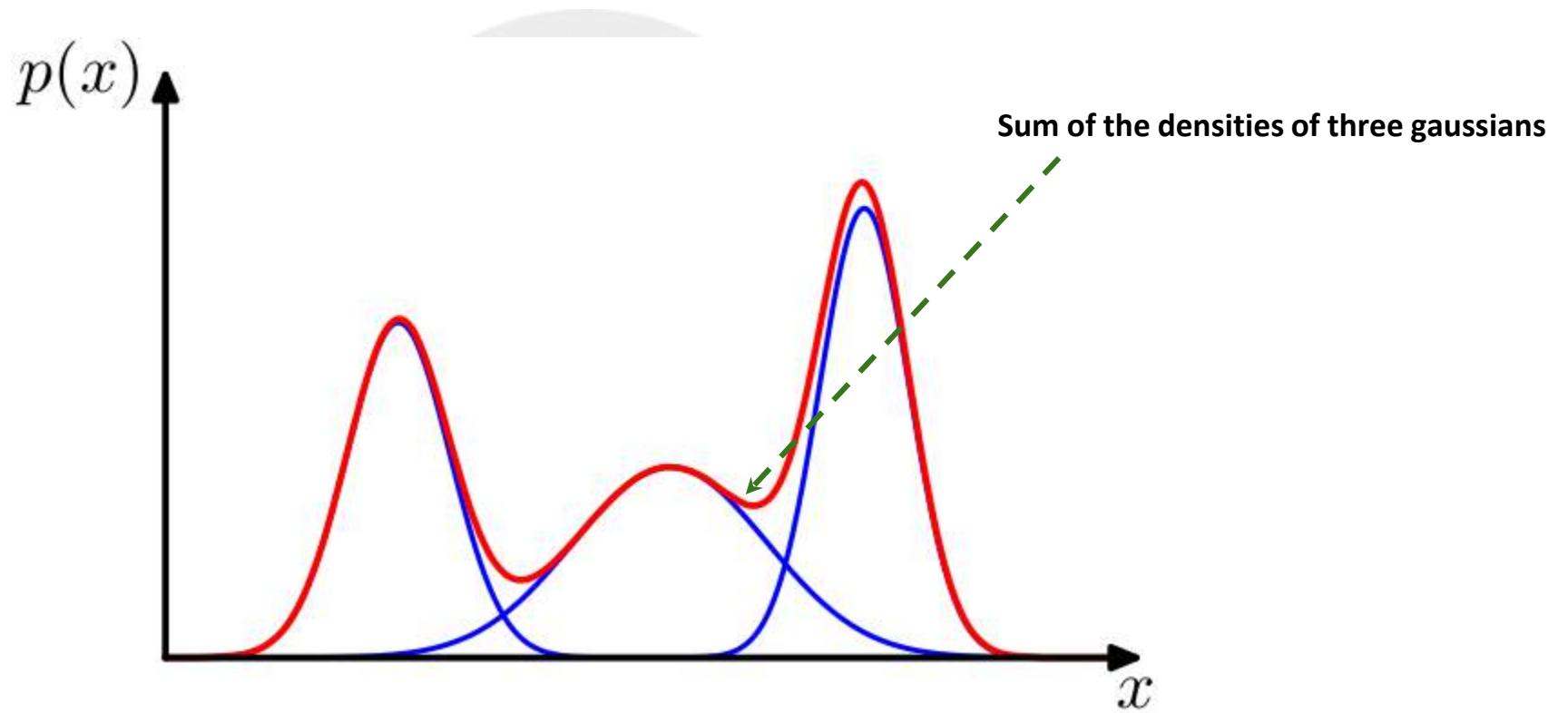
Mixture Distributions

- Mixture Distributions: Linear combinations of basic distributions (such as Gaussian) to approximate complex density



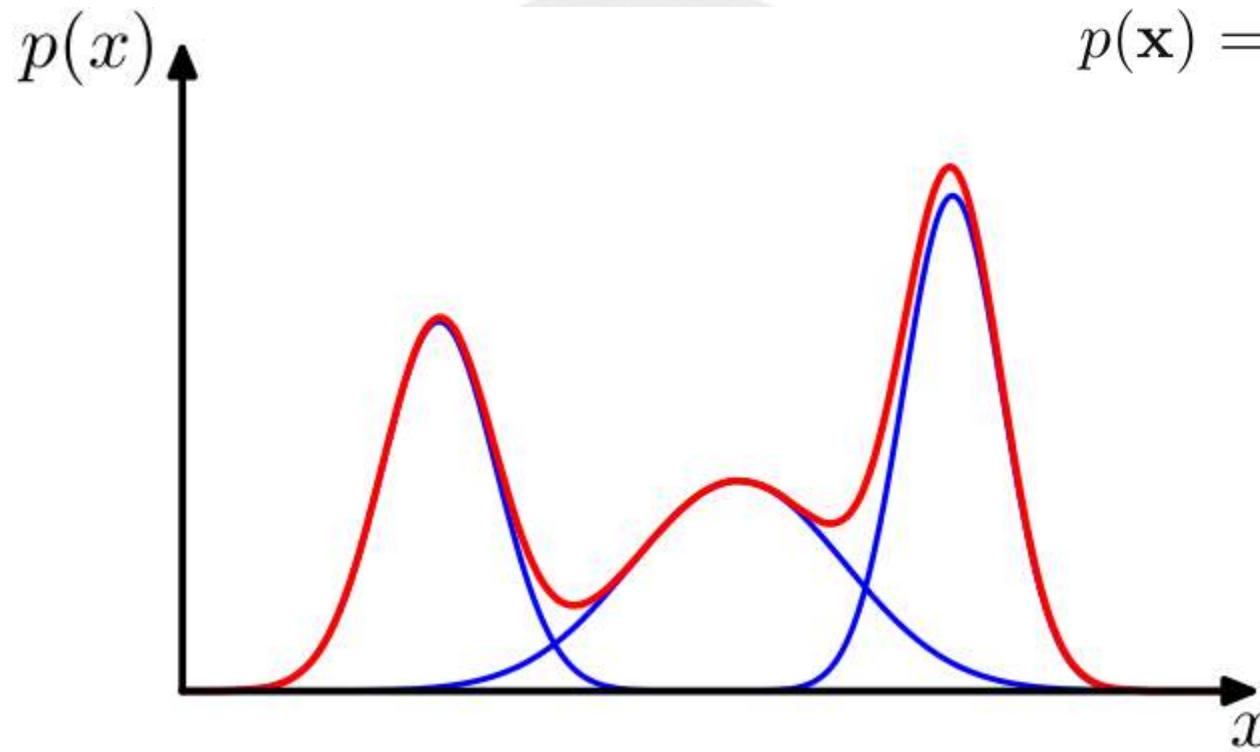
Mixture Distributions

- Mixture Distributions: Linear combinations of basic distributions (such as Gaussian) to approximate complex density



Mixture of Gaussians

- Mixture of Gaussians: Component densities are Gaussian.



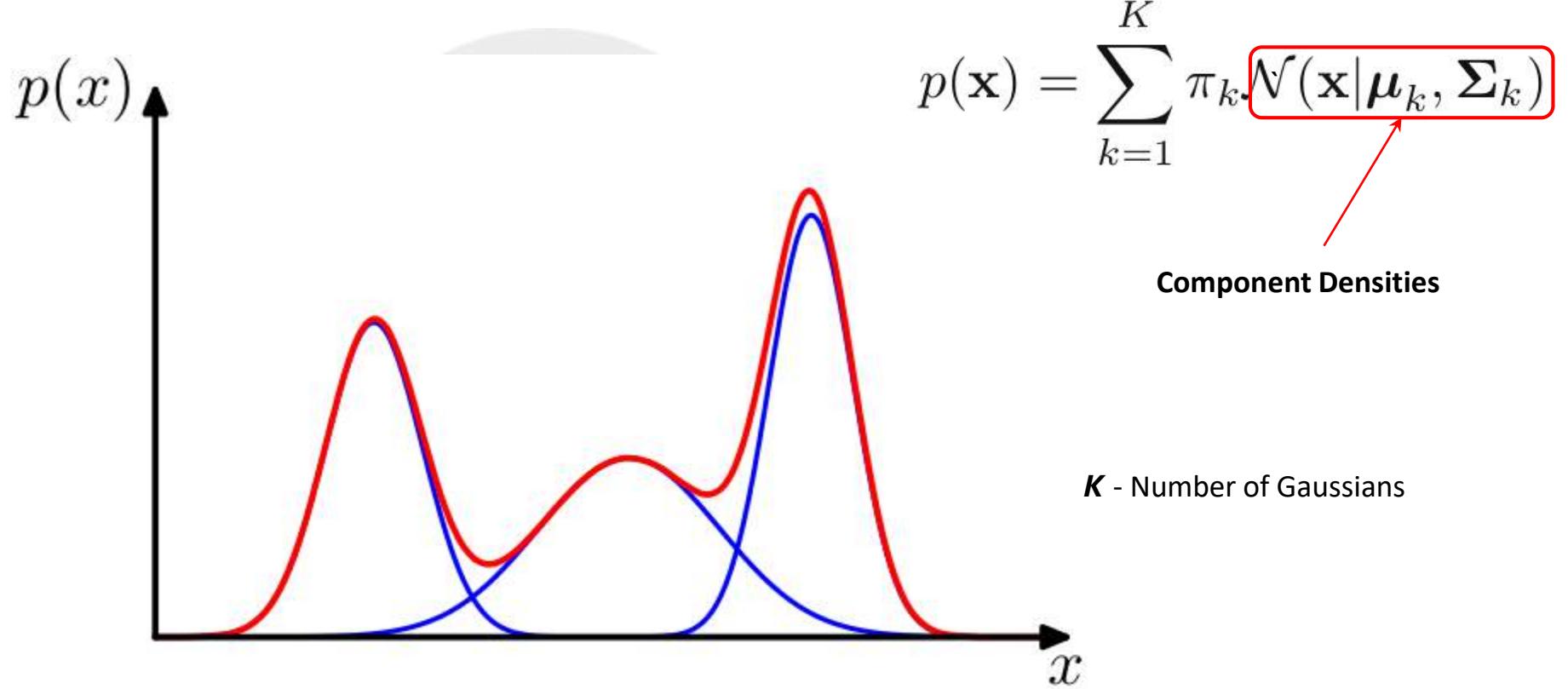
Mixing Coefficients

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

K - Number of Gaussians

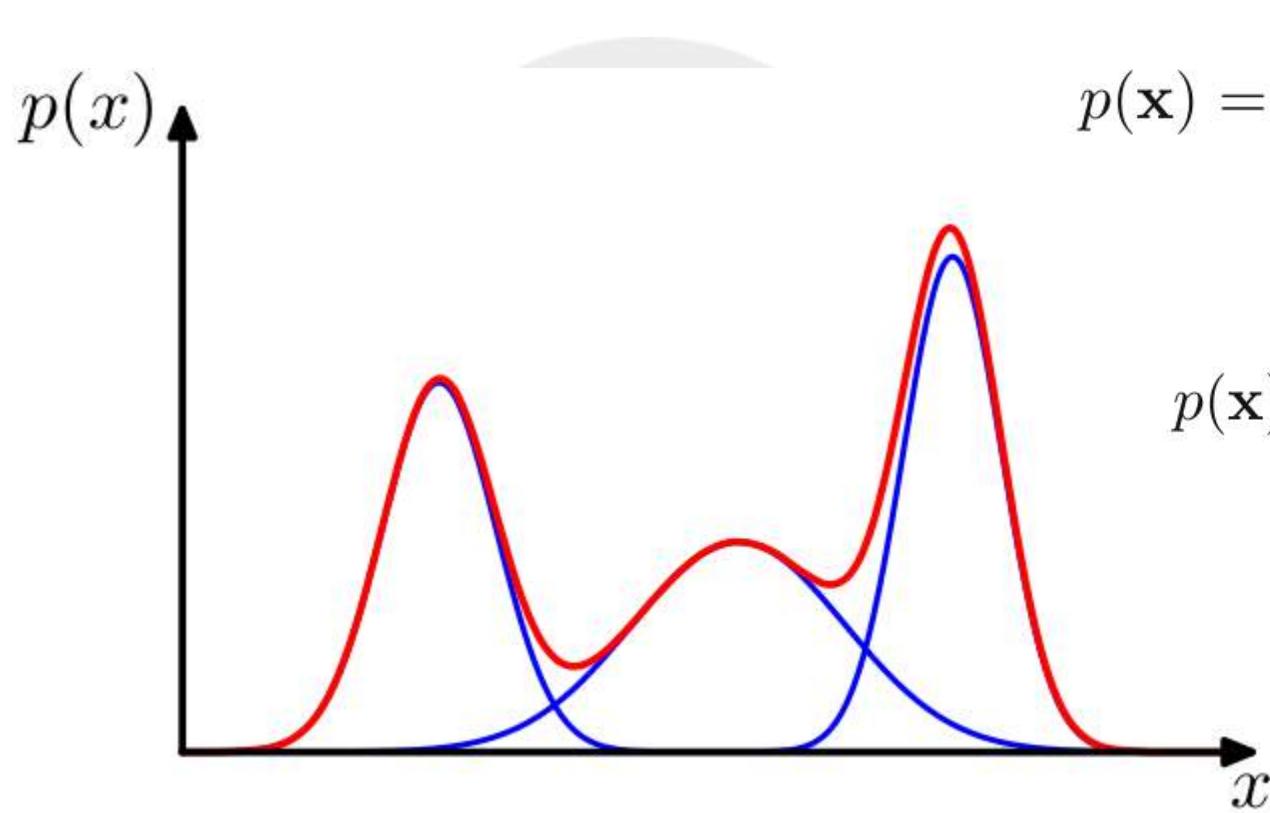
Mixture of Gaussians

- Mixture of Gaussians: Component densities are Gaussian.



Mixture of Gaussians

- Mixture of Gaussians: Component densities are Gaussian.

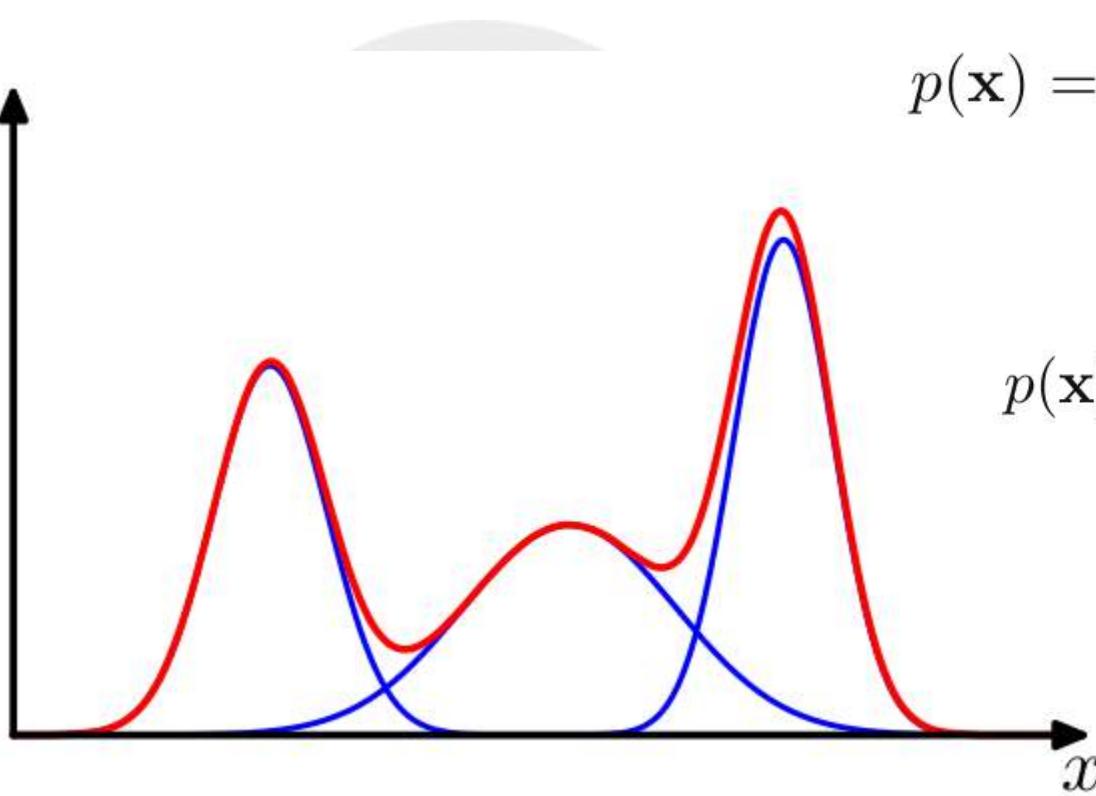


Mixture of Gaussians

- Mixture of Gaussians: Component densities are Gaussian.

Parameters of MoG:

$$\begin{aligned}\pi &: \{\pi_1, \dots, \pi_K\}, \\ \mu &: \{\mu_1, \dots, \mu_K\} \\ \Sigma &: \{\Sigma_1, \dots, \Sigma_K\}\end{aligned}$$



$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x}|k)$$

Mixture Density

Parameters of MoG:

$$\begin{aligned}\pi &: \{\pi_1, \dots, \pi_K\}, \\ \mu &: \{\mu_1, \dots, \mu_K\} \\ \Sigma &: \{\Sigma_1, \dots, \Sigma_K\}\end{aligned}$$

X	Assuming Latent Variable z for 2 mixture components.	
	$z_{(1)}$	$z_{(2)}$
$x^{(1)}$	0	1
$x^{(2)}$	1	0
...		
$x^{(N)}$	1	0

$$P(z_k=1) = \pi_k$$



$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$$

$$0 \leq \pi_k \leq 1 \quad \text{and} \quad \sum_k \pi_k = 1$$

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x}|k)$$

Component Density

Parameters of MoG:

$\pi : \{\pi_1, \dots, \pi_K\}$,
 $\mu : \{\mu_1, \dots, \mu_K\}$
 $\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$

Given a particular z_k :

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x}|k)$$

Marginal Distribution of \mathbf{x}

Parameters of MoG:

$\pi : \{\pi_1, \dots, \pi_K\}$,
 $\mu : \{\mu_1, \dots, \mu_K\}$
 $\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$$
$$= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x}|k)$$

$\gamma(z_k)$

Parameters of MoG:

$\pi : \{\pi_1, \dots, \pi_K\}$,
 $\mu : \{\mu_1, \dots, \mu_K\}$
 $\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$

$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x} | k)$$

Log Likelihood

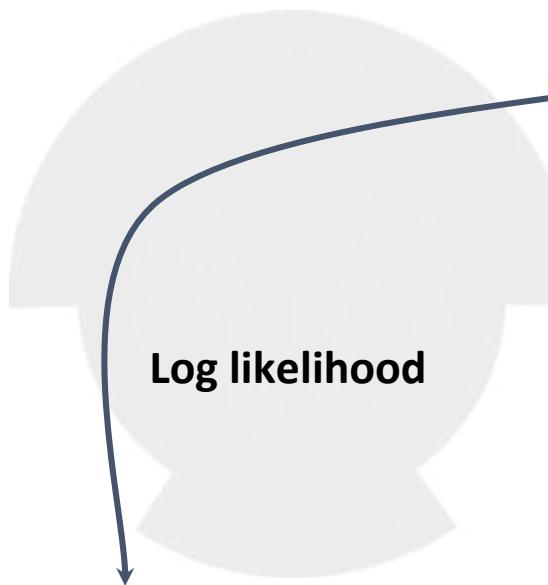
- Mixture of Gaussians: Component densities are Gaussian.

Parameters of MoG:

$$\pi : \{\pi_1, \dots, \pi_K\},$$

$$\mu : \{\mu_1, \dots, \mu_K\}$$

$$\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$$



$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

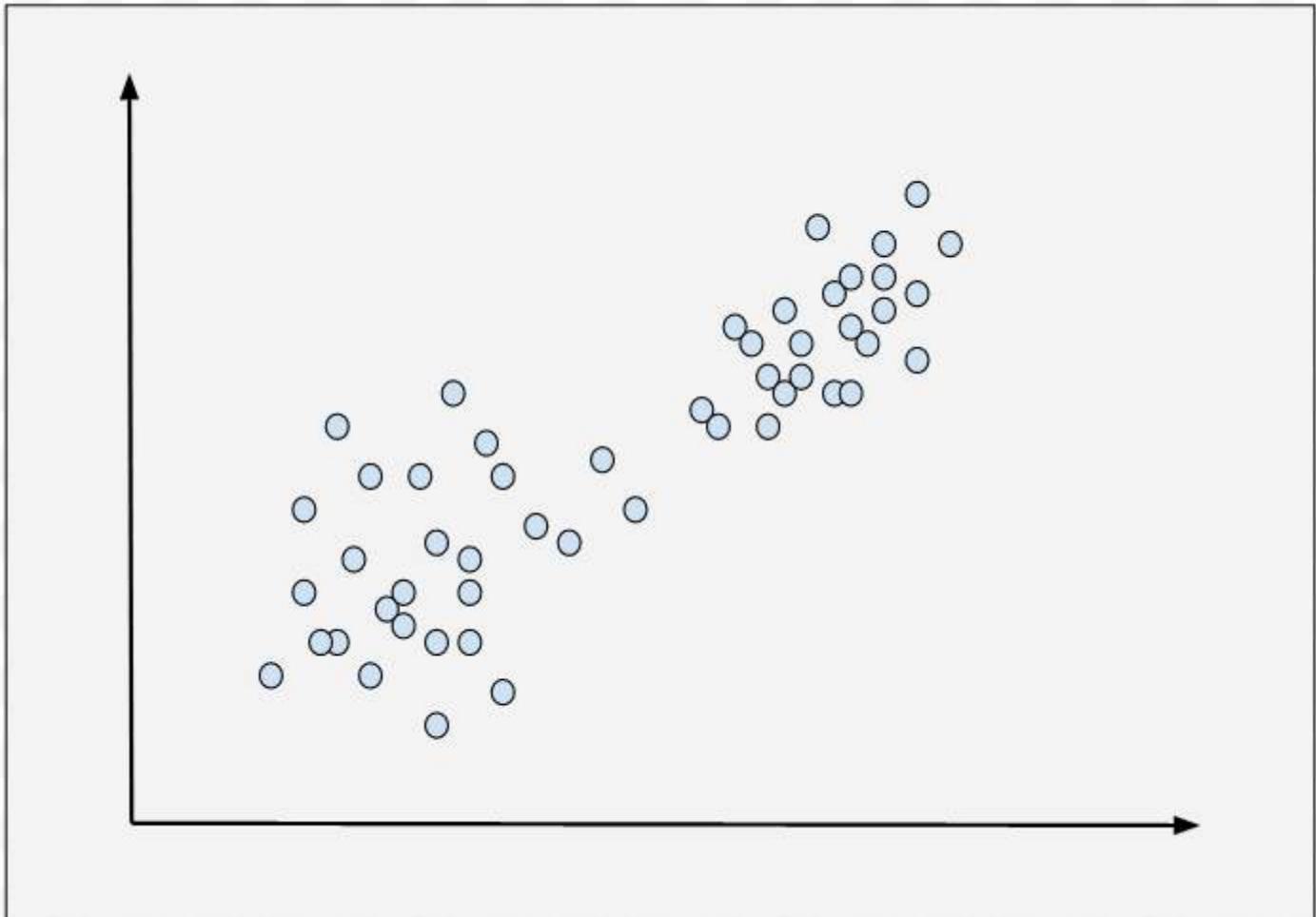
$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x}|k)$$

$$\ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

MoG for Clustering

To determine:

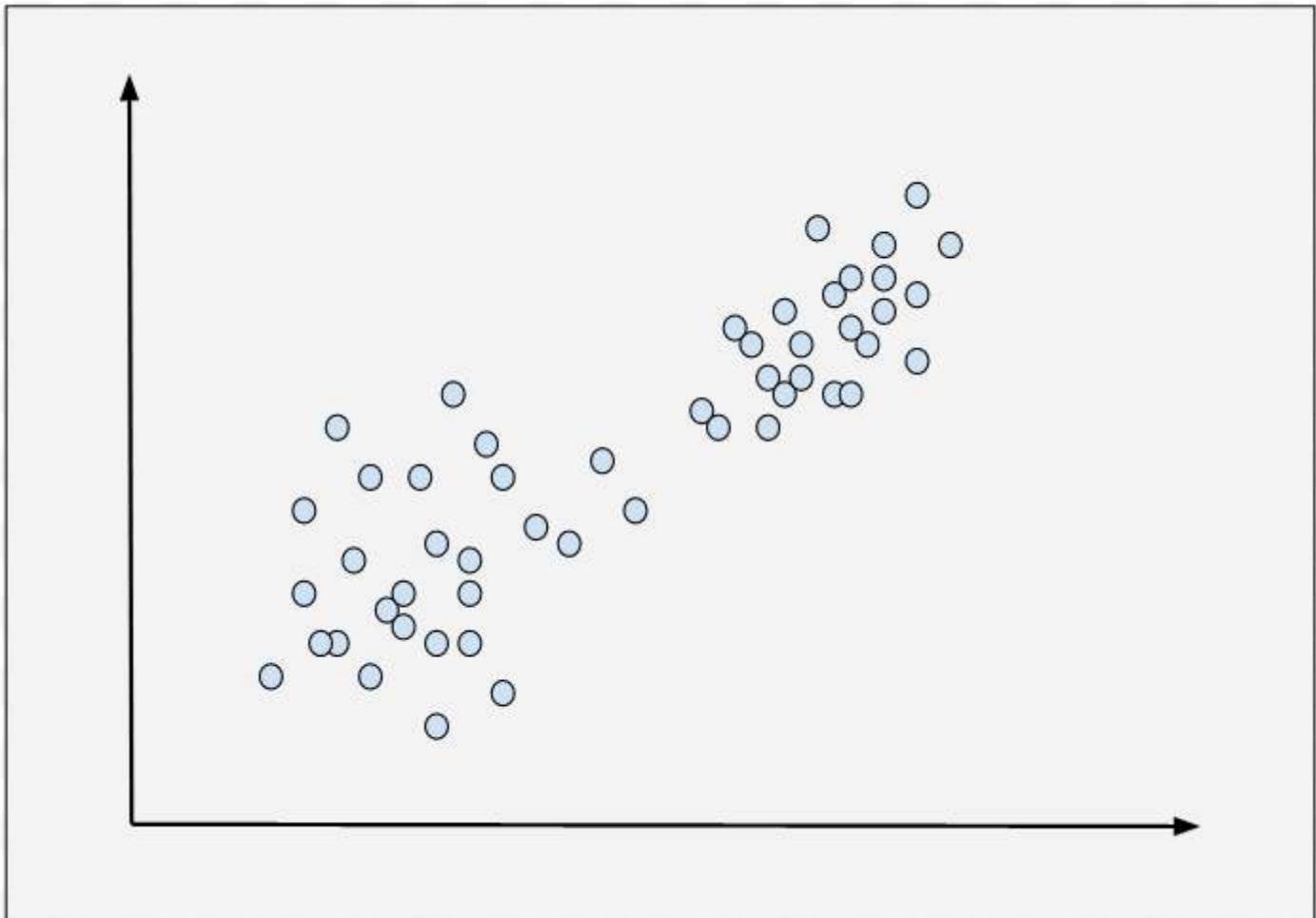
Cluster Memberships (K) for each
 $x^{(i)}$ that fits the data best.



MoG for Clustering

MoG for Clustering:

No. of Mixture Densities (Gaussian) : K



MoG for Clustering

MoG for Clustering:

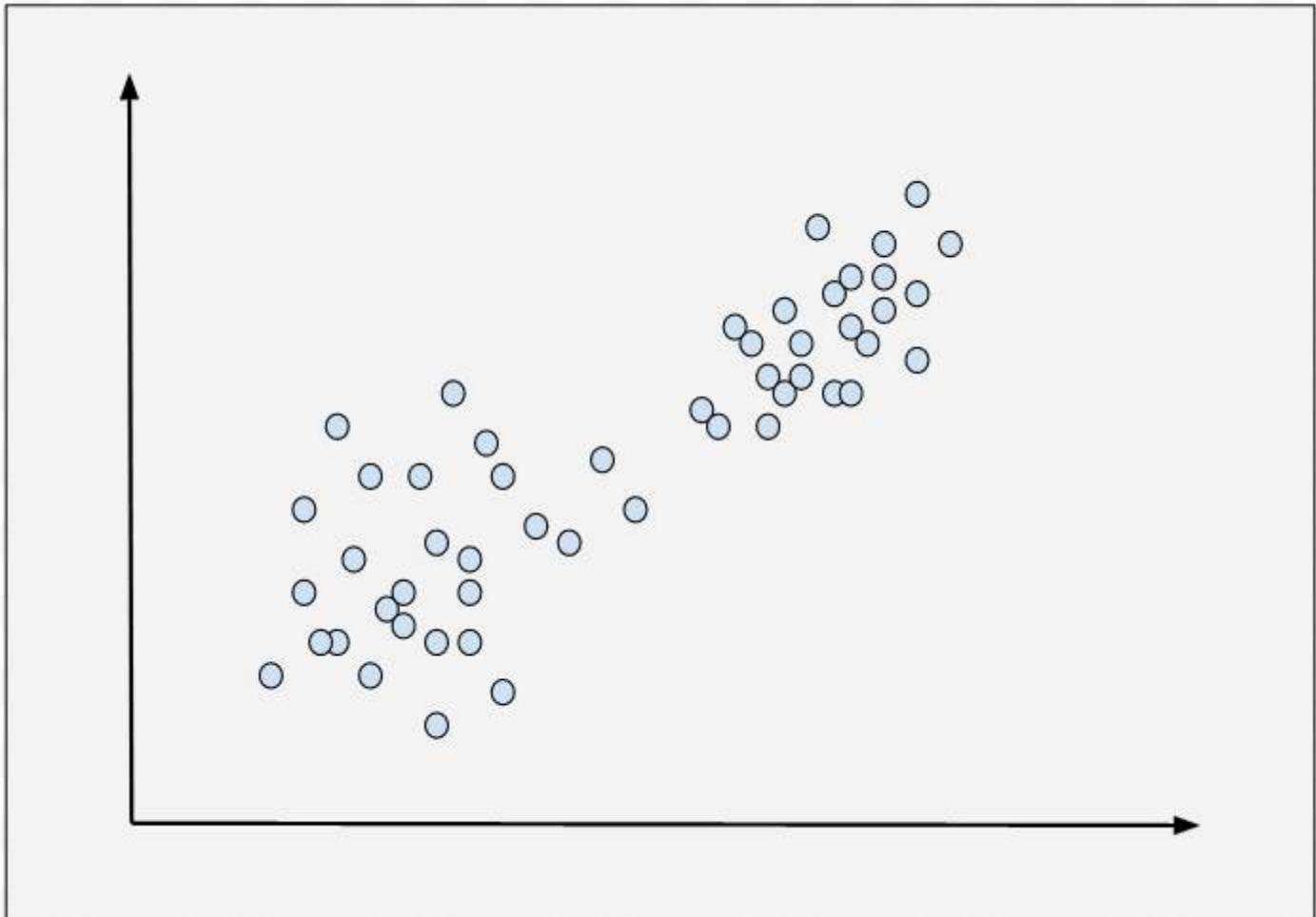
No. of Mixture Densities (Gaussian) : K

Parameters of MoG:

$\pi : \{\pi_1, \dots, \pi_K\}$,

$\mu : \{\mu_1, \dots, \mu_K\}$

$\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$



MoG for Clustering

MoG for Clustering:

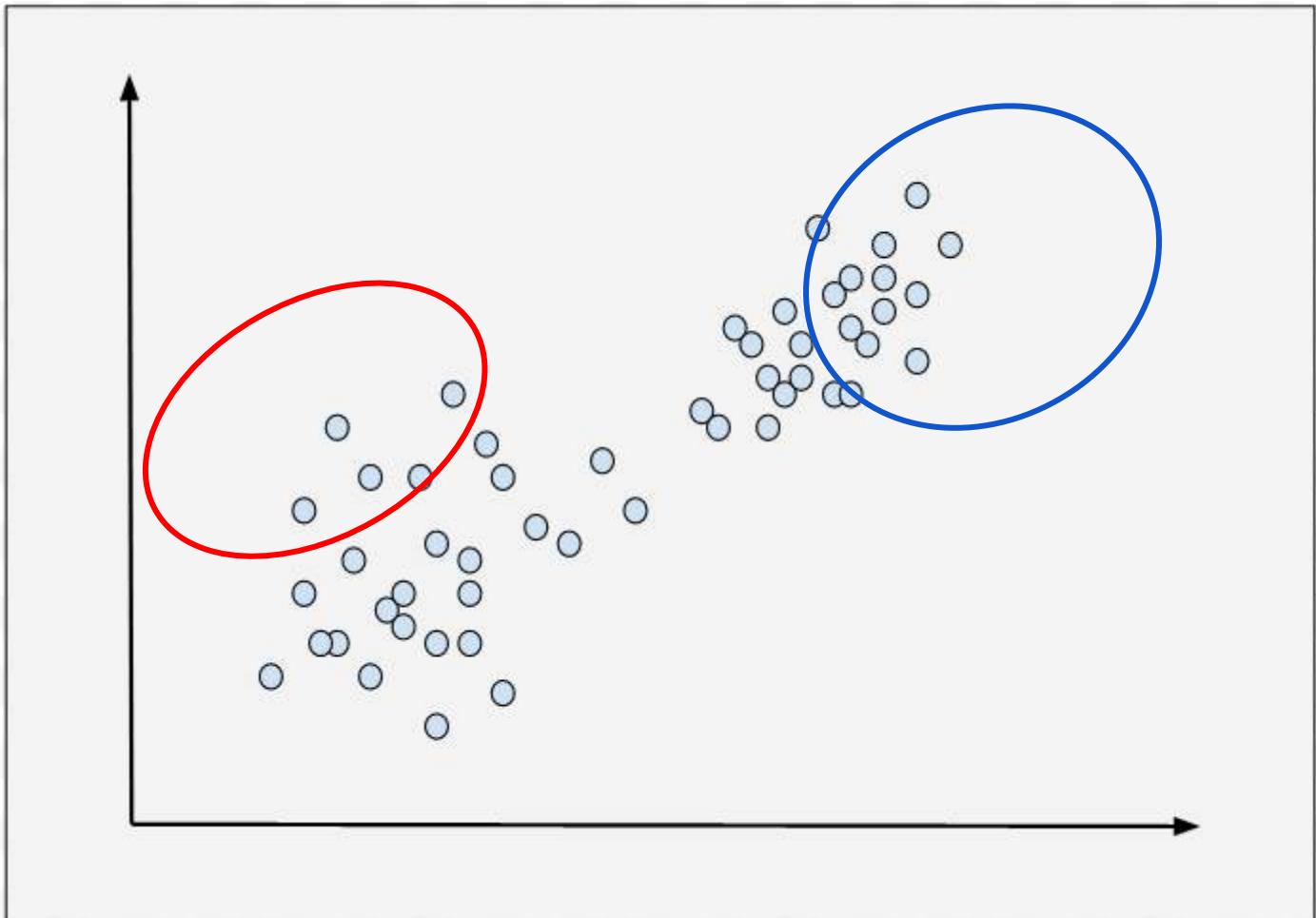
No. of Mixture Densities (Gaussian) : K

Parameters of MoG:

$\pi : \{\pi_1, \dots, \pi_K\}$,

$\mu : \{\mu_1, \dots, \mu_K\}$

$\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$



MoG for Clustering

MoG for Clustering:

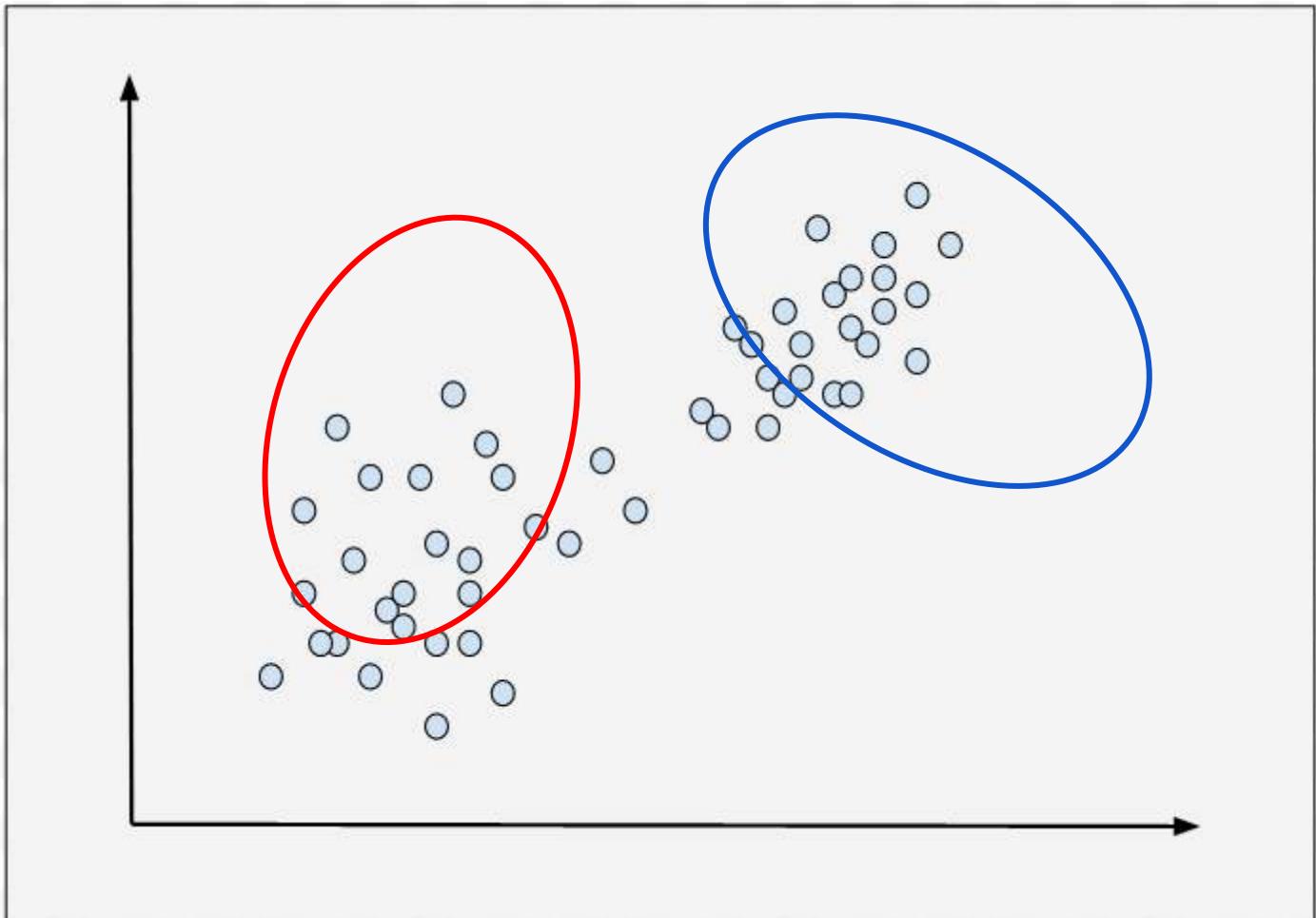
No. of Mixture Densities (Gaussian) : K

Parameters of MoG:

$$\pi : \{\pi_1, \dots, \pi_K\},$$

$$\mu : \{\mu_1, \dots, \mu_K\}$$

$$\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$$



MoG for Clustering

MoG for Clustering:

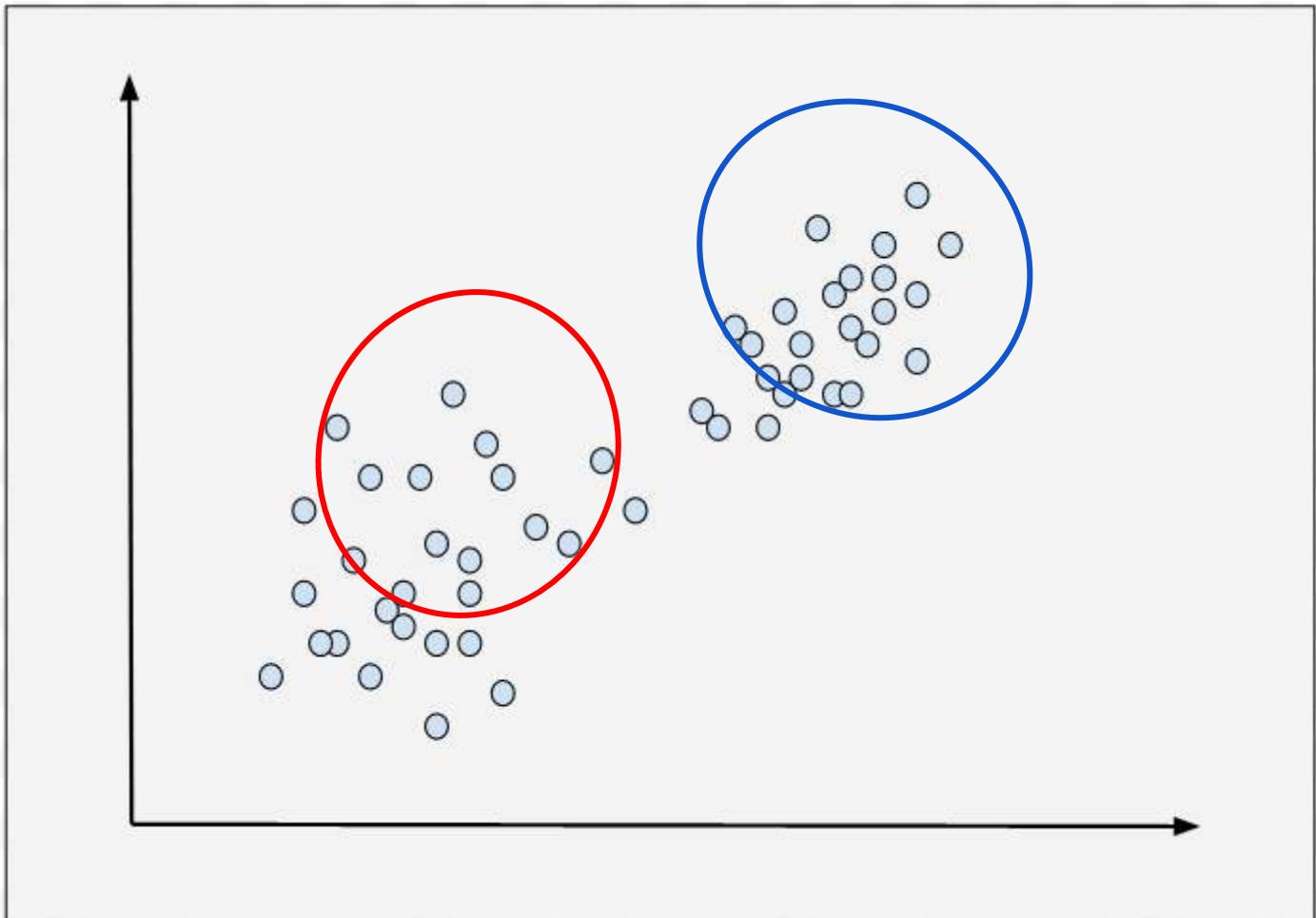
No. of Mixture Densities (Gaussian) : K

Parameters of MoG:

$$\pi : \{\pi_1, \dots, \pi_K\},$$

$$\mu : \{\mu_1, \dots, \mu_K\}$$

$$\Sigma : \{\Sigma_1, \dots, \Sigma_K\}$$

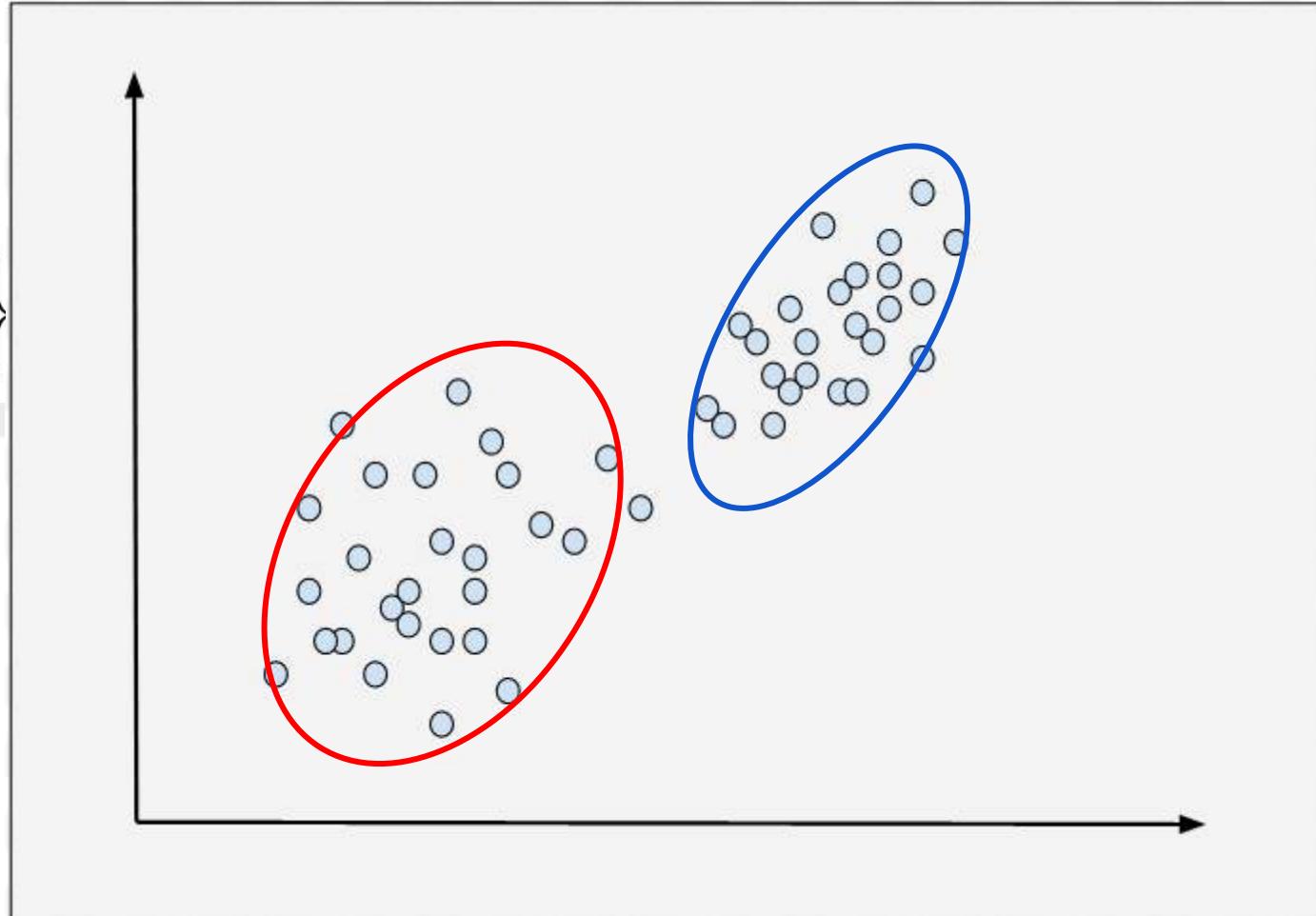


MoG for Clustering

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Parameters of MoG:

$$\begin{aligned}\boldsymbol{\pi} &: \{\pi_1, \dots, \pi_K\}, \\ \boldsymbol{\mu} &: \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\} \\ \boldsymbol{\Sigma} &: \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}\end{aligned}$$

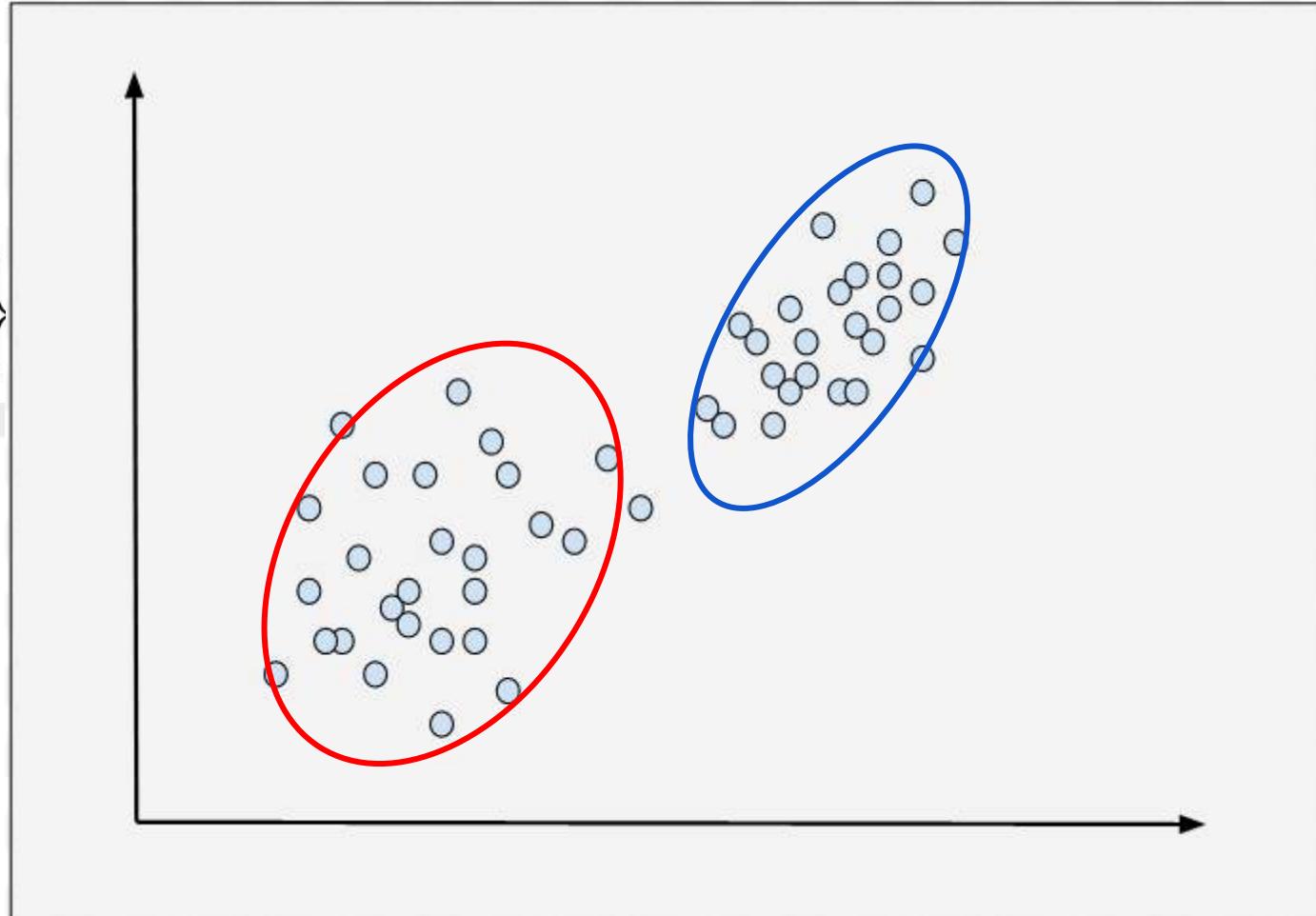


MoG for Clustering

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Parameters of MoG:

$$\begin{aligned}\boldsymbol{\pi} &: \{\pi_1, \dots, \pi_K\}, \\ \boldsymbol{\mu} &: \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\} \\ \boldsymbol{\Sigma} &: \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}\end{aligned}$$



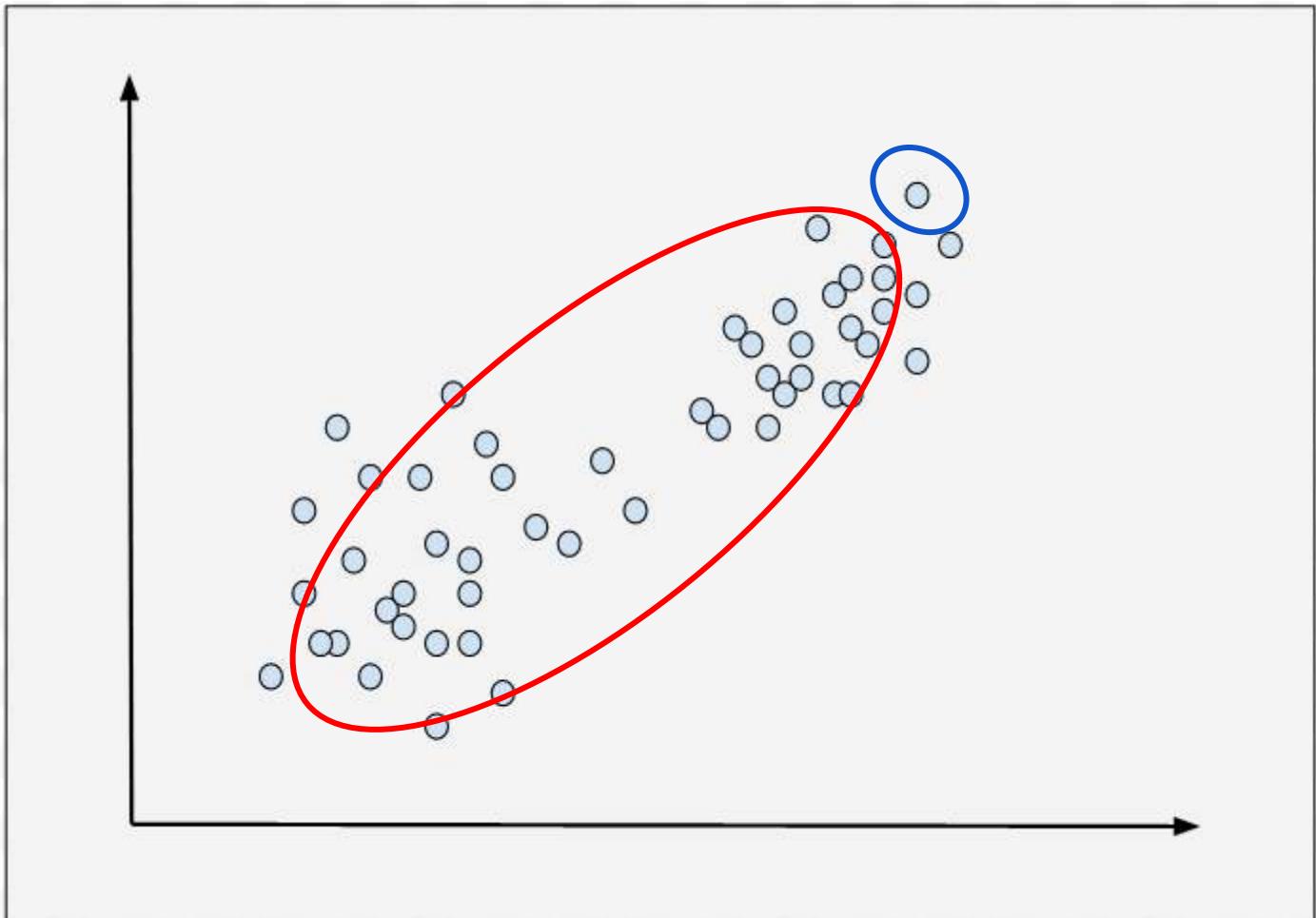
It turns out that there are no closed form solution to the parameters while optimizing for the parameters.

MoG for Clustering

Singularity :

A component ‘collapses’ onto a data point

In likelihood function when a component ‘collapses’ onto a data point:

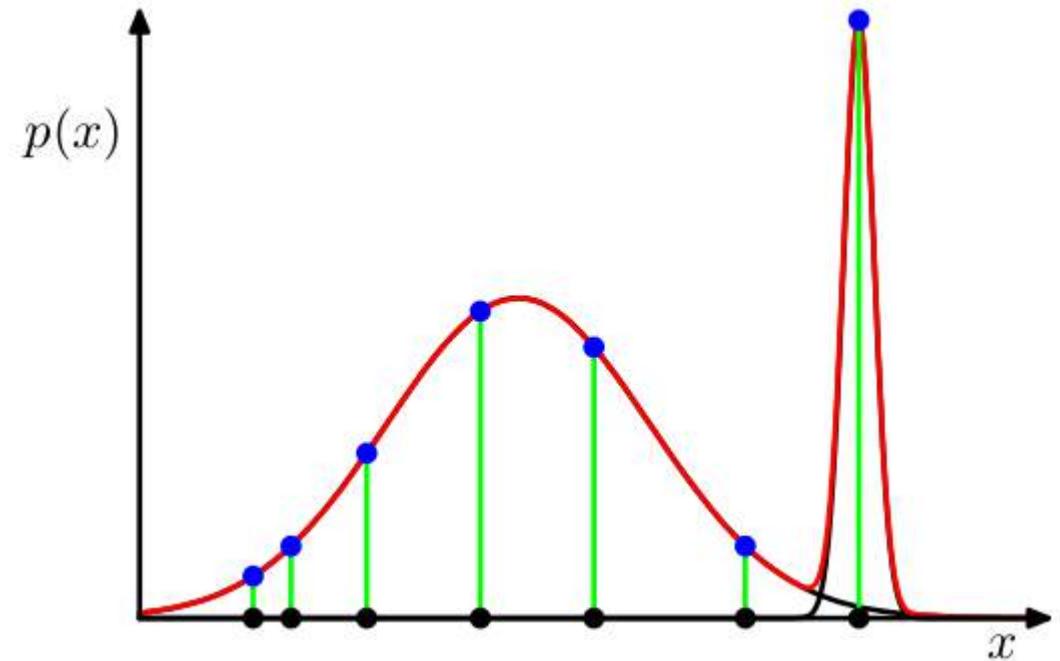


MoG for Clustering

Singularity :

A component ‘collapses’ onto a data point

In likelihood function when a component ‘collapses’ onto a data point:



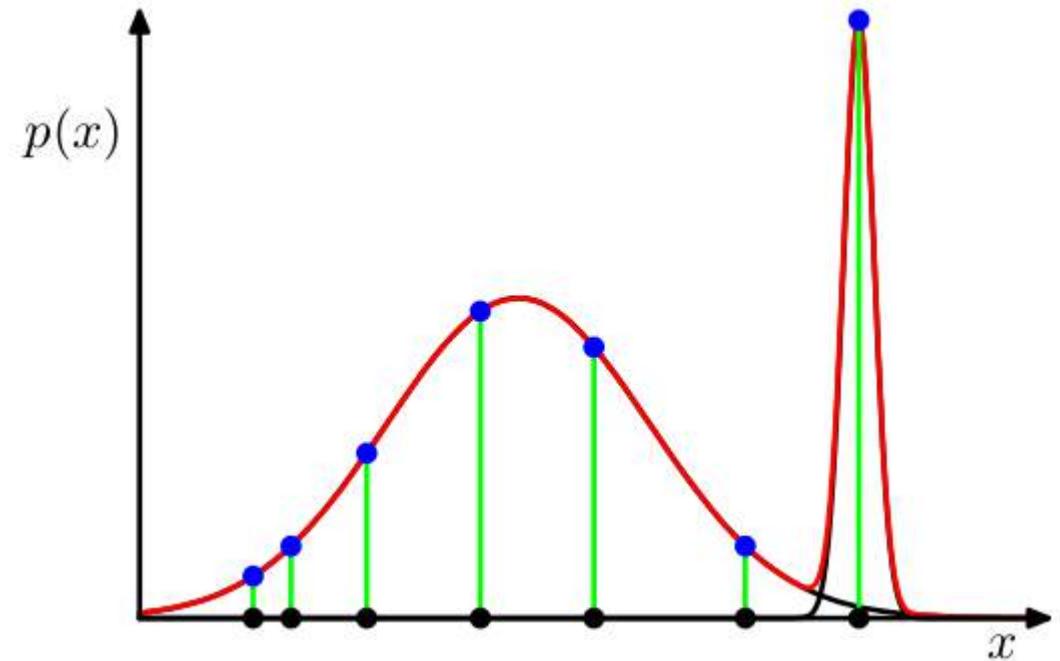
MoG for Clustering

Singularity :

A component ‘collapses’ onto a data point

In likelihood function when a component ‘collapses’ onto a data point:

$$\mathcal{N}(\mathbf{x}_n | \mathbf{x}_n, \sigma_j^2 \mathbf{I}) = \frac{1}{(2\pi)^{1/2}} \frac{1}{\sigma_j}$$



Agenda

- EM Algorithm for MoG



Expectation-Maximization Algorithm

- Method for finding ML solutions for models with Latent Variables
 - Broad applicability for estimating parameters for various models
 - Estimating parameters of a MoG is one application
- Task : To Find ML parameters of MoG

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Expectation-Maximization Algorithm

- Method for finding ML solutions for models with Latent Variables
 - Broad applicability for estimating parameters for various models
 - Estimating parameters of a MoG is one application
- Task : To Find ML parameters of MoG

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Set the derivative of $\ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$
 - w.r.t $\boldsymbol{\mu}_k$ to 0 & solve it for $\boldsymbol{\mu}_k$
 - w.r.t $\boldsymbol{\Sigma}_k$ to 0 & solve it for $\boldsymbol{\Sigma}_k$
 - w.r.t $\boldsymbol{\pi}_k$ to 0 & solve it for $\boldsymbol{\pi}_k$ - [Constrained optimization using Lagrangian Multipliers]

Expectation-Maximization Algorithm

- We get μ_k as

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

Where

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$



$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

$$p(z_k = 1) = \pi_k$$

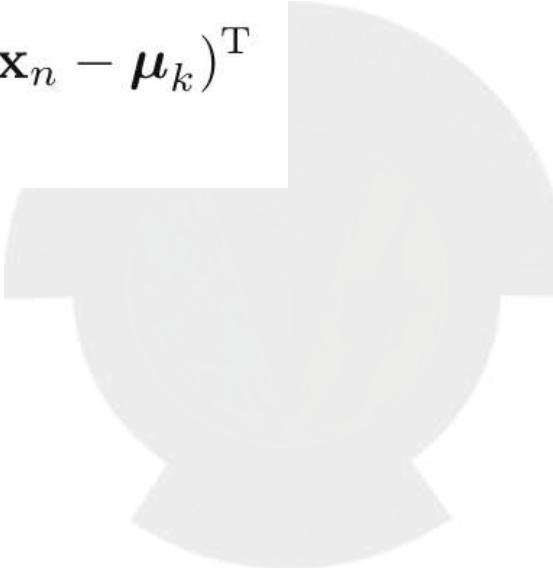
Expectation-Maximization Algorithm

- We get Σ_k as

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

Where

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$



$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \Sigma_j)}.\end{aligned}$$

$$p(z_k = 1) = \pi_k$$

Expectation-Maximization Algorithm

- We get π_k as

$$\pi_k = \frac{N_k}{N}$$

Where

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$



$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

$$p(z_k = 1) = \pi_k$$

Expectation-Maximization Algorithm

To Estimate:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$

$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

$$p(z_k = 1) = \pi_k$$

Not closed form solutions !!!

Expectation-Maximization Algorithm

M-Step

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$

E-Step

$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\&= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\&= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

$$p(z_k = 1) = \pi_k$$

Expectation-Maximization Algorithm

To Estimate:

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$

M-Step

Initialize $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ and
also evaluate the log likelihood

Perform E-Step Given $\gamma(z_k)$

E-Step

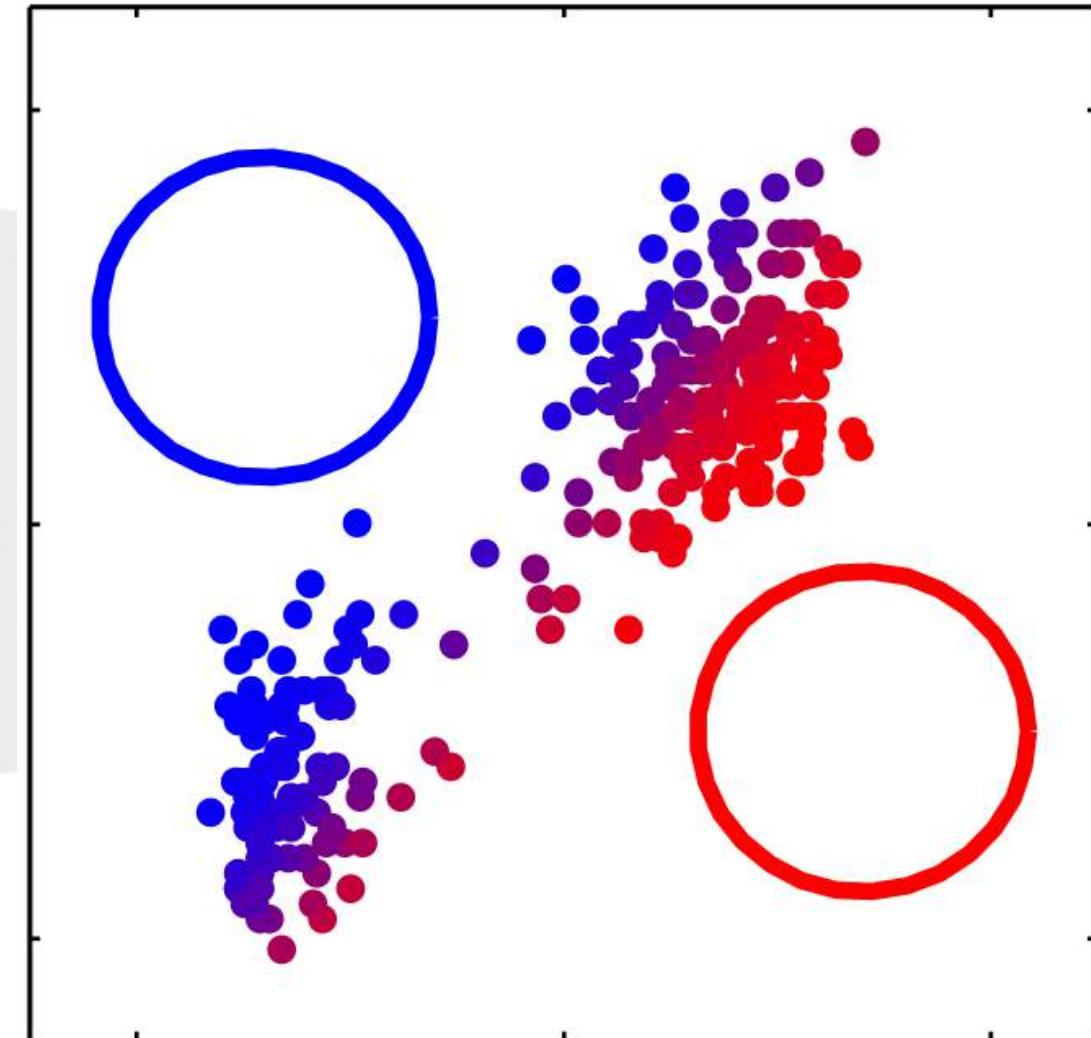
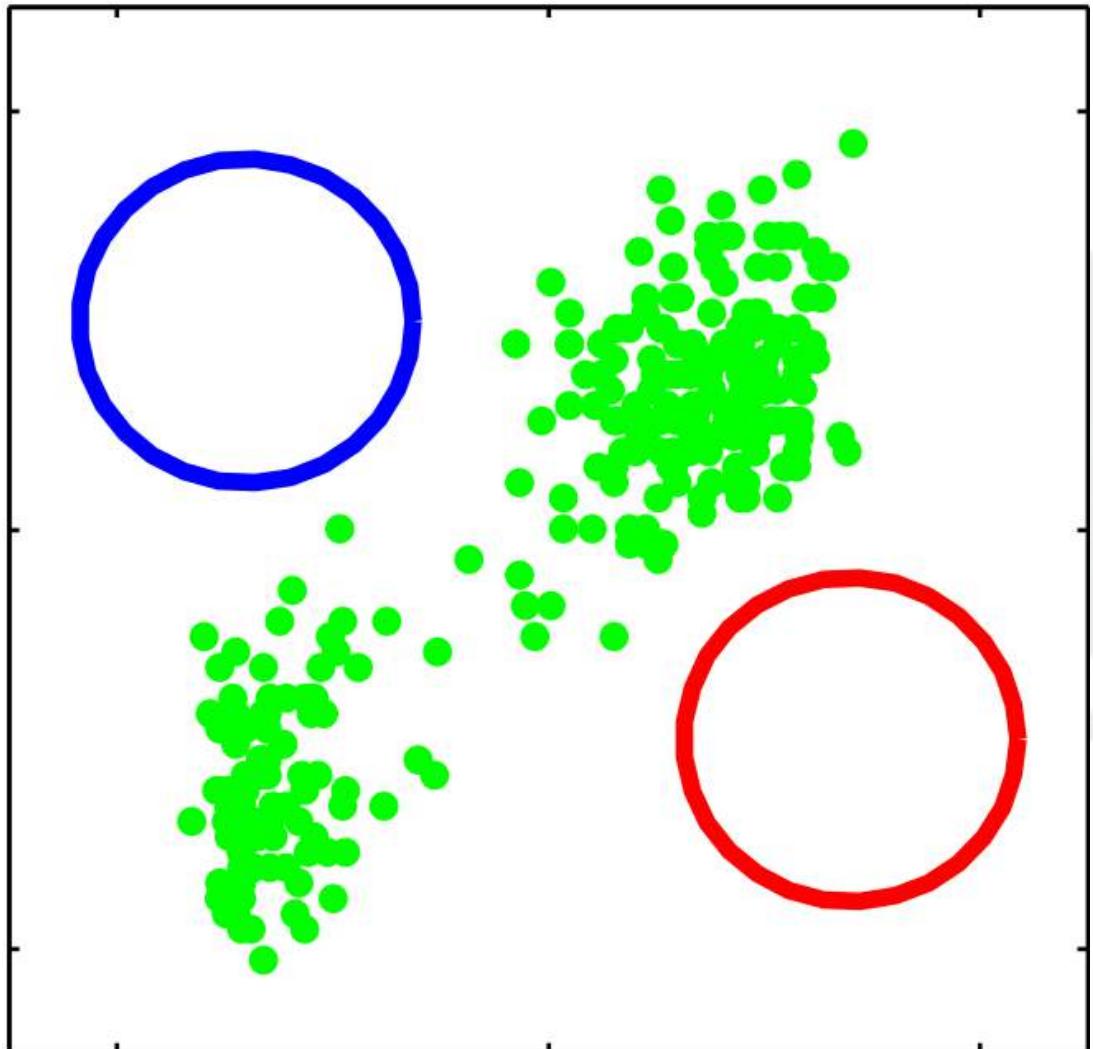
$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\ &= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

Perform M-Step Given $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

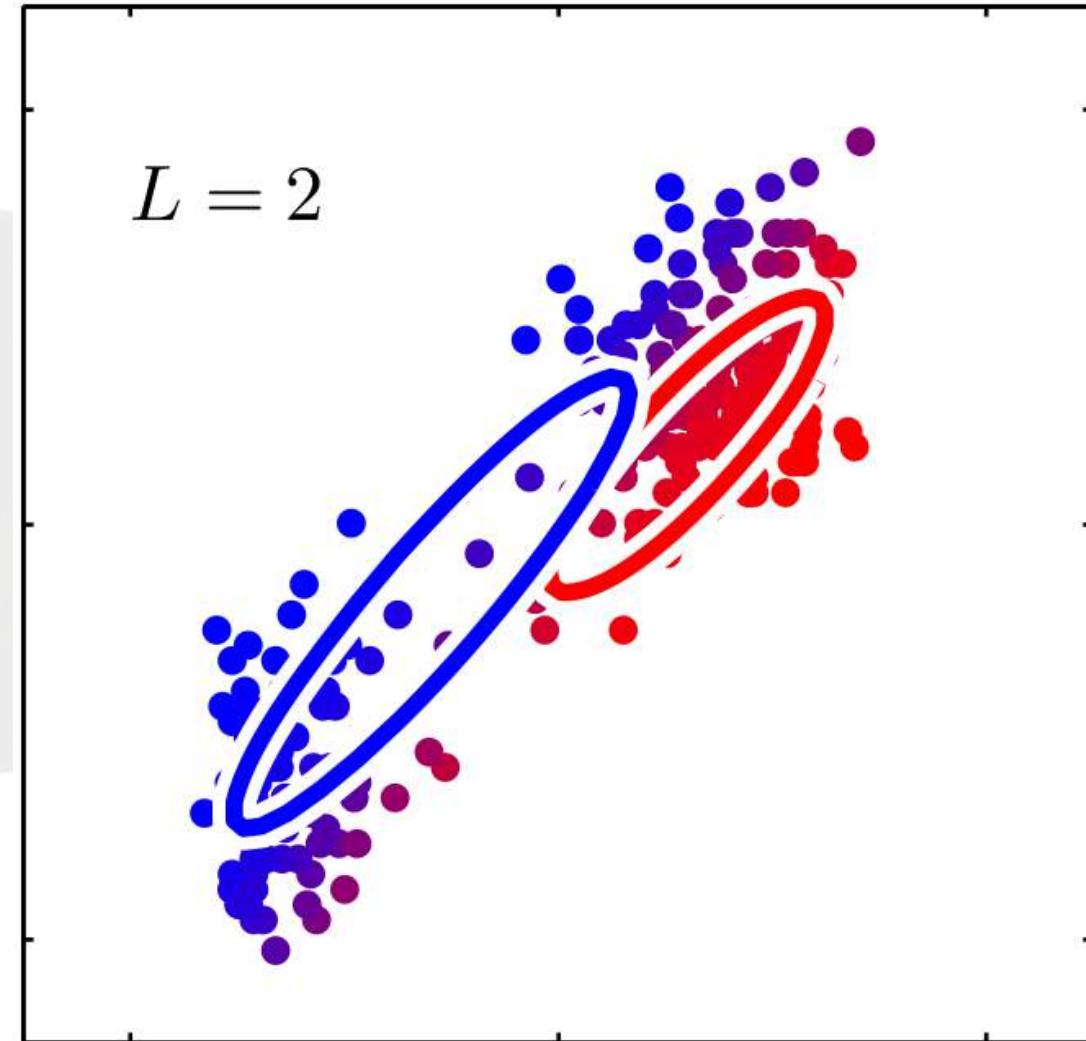
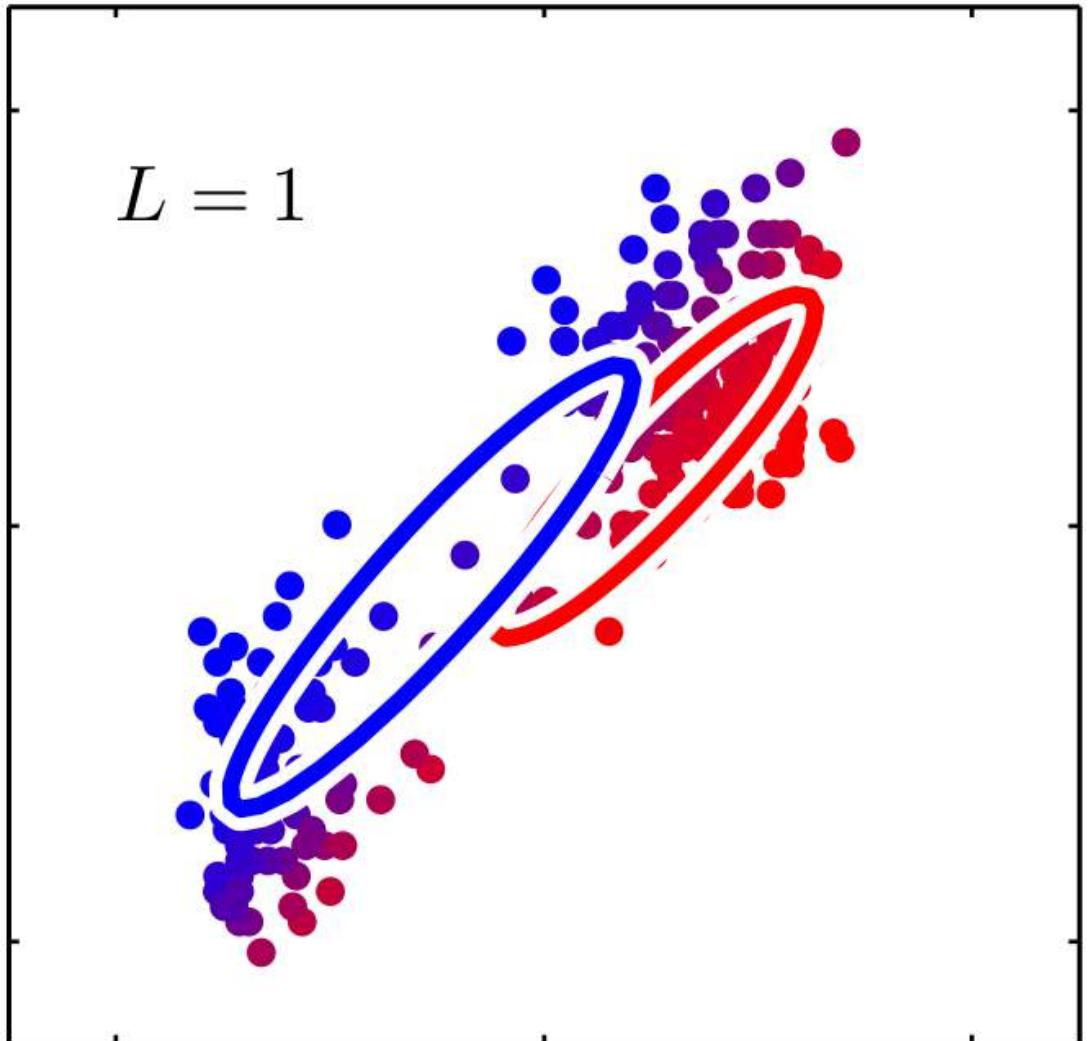
$$p(z_k = 1) = \pi_k$$

Repeat Until Convergence - [Use log likelihood / parameters to decide this]

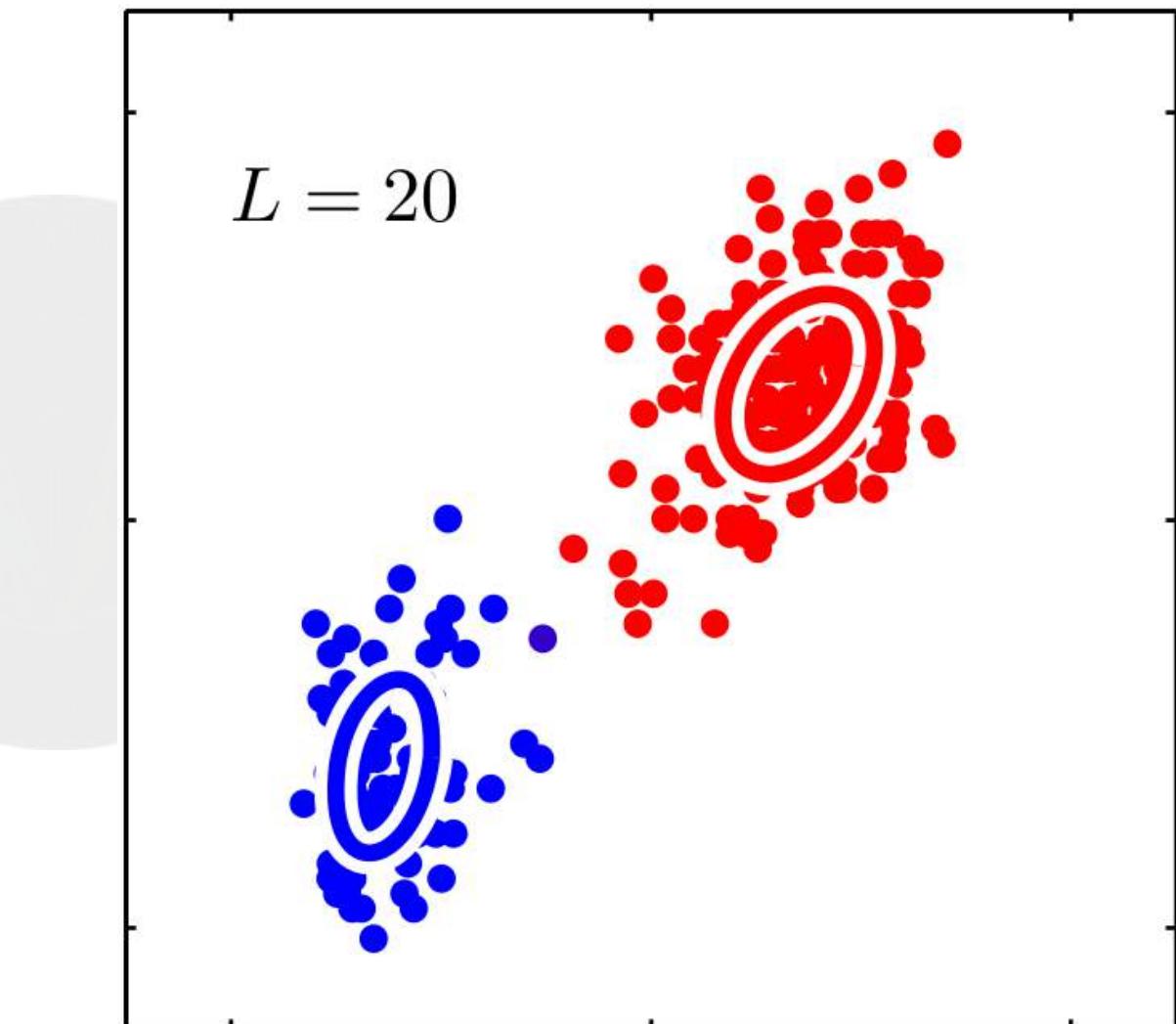
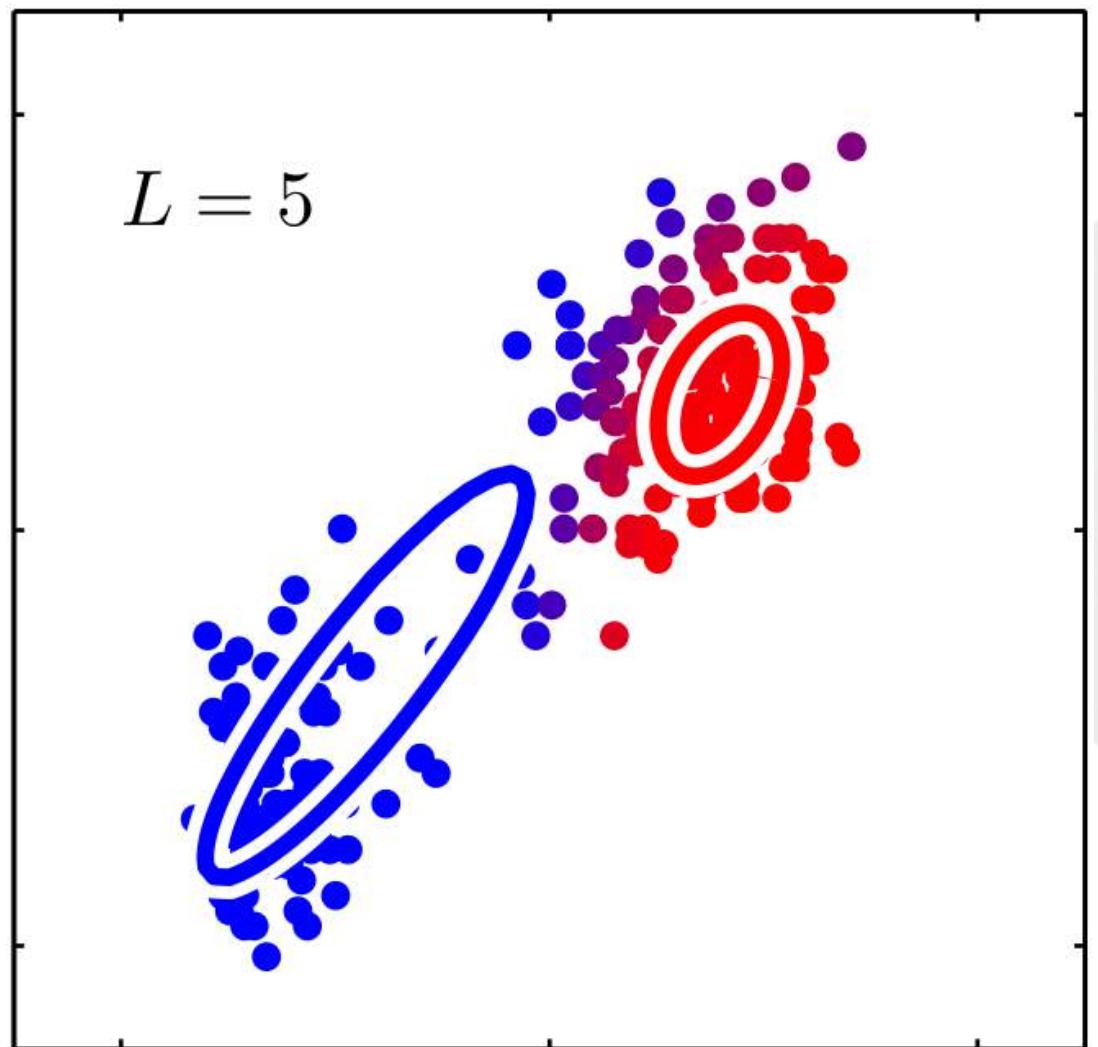
Expectation-Maximization Algorithm



Expectation-Maximization Algorithm



Expectation-Maximization Algorithm





Thank You!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

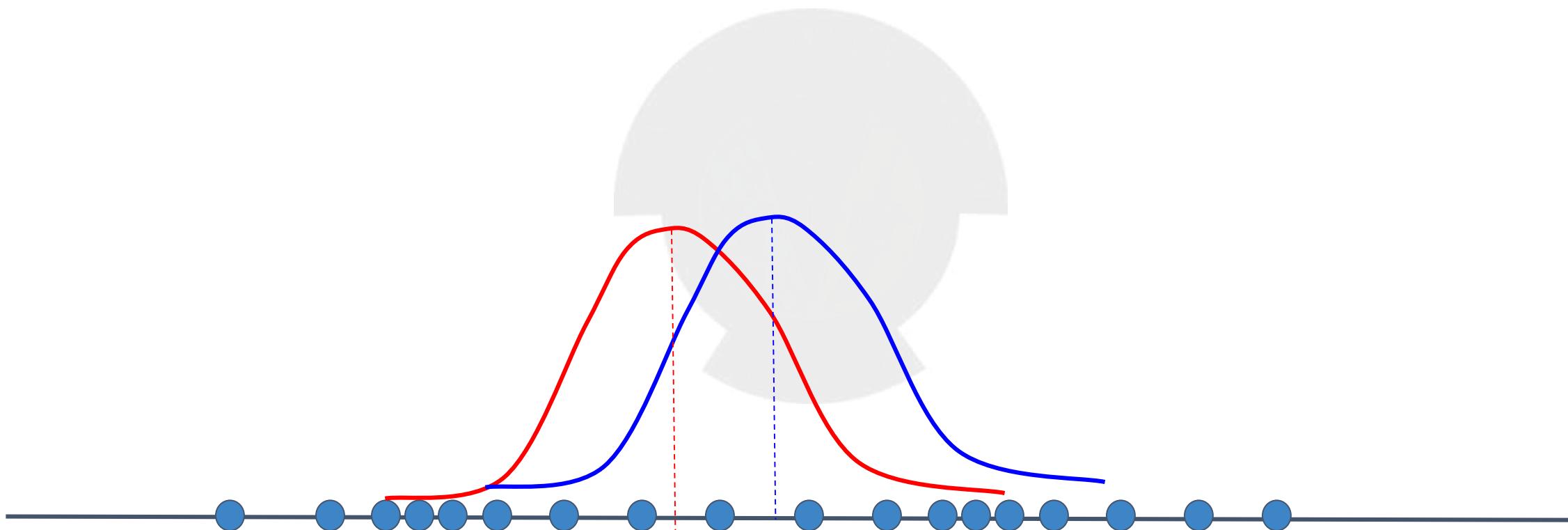
Expectation Maximization Algorithm -A General Formulation

S.P.Vimal

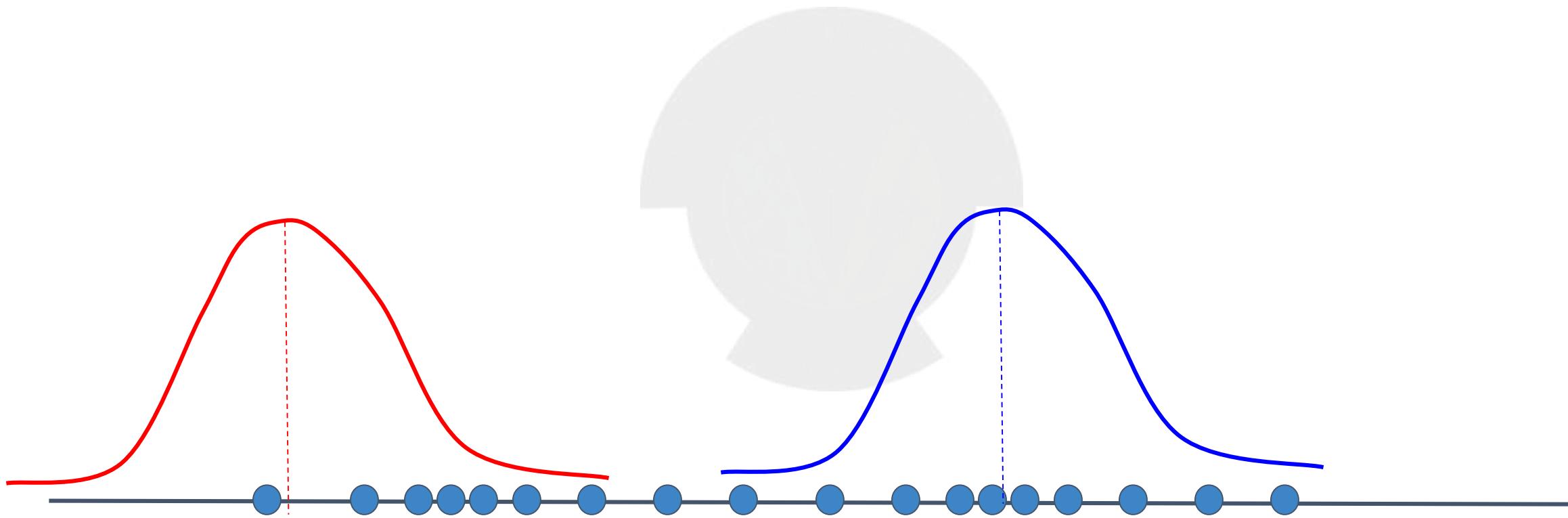
Agenda

- An Alternative formulation of EM Algorithm (more commonly used formulation)
 - Discussion on MoG

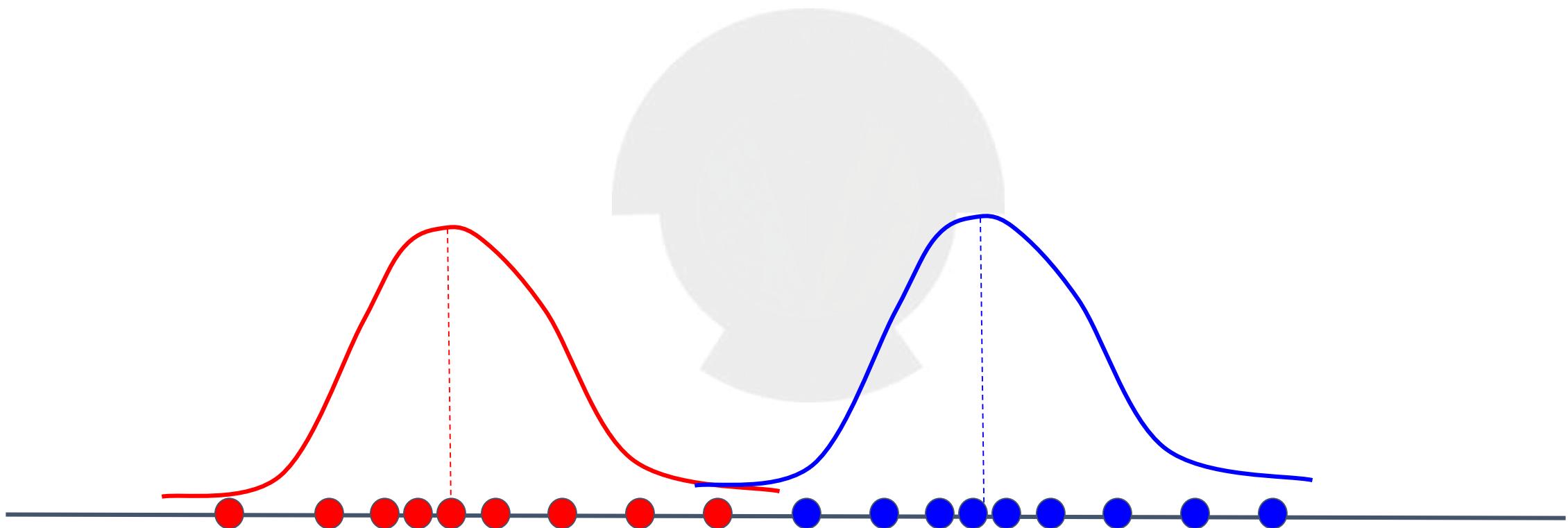
Introduction



Introduction



Introduction



Introduction

Observed Data

: $X [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

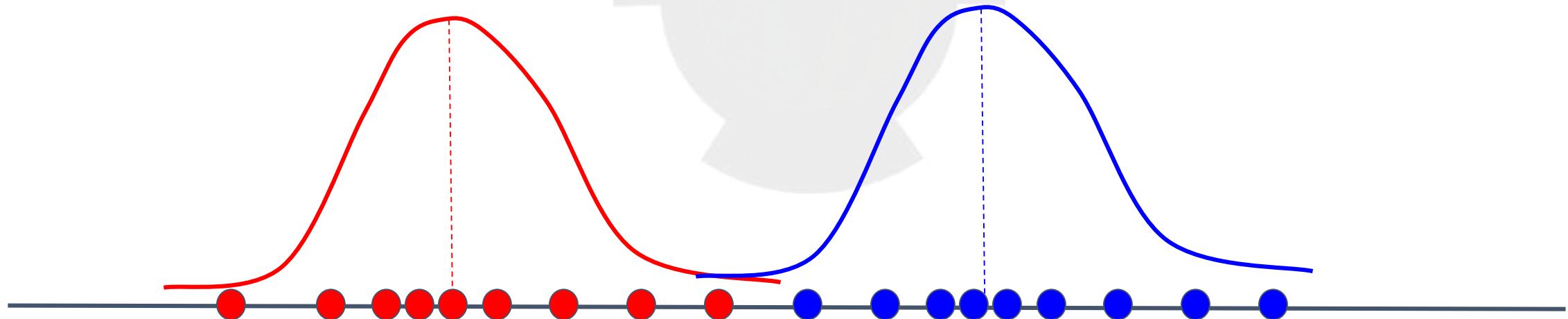
Latent Variables

: $Z [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal

: Find ML solution to model parameters

	X			Z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}



Introduction

Observed Data

: $\mathbf{X} [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

Latent Variables

: $\mathbf{Z} [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal

: Find ML solution to model parameters

Let the set of model parameters be θ :

We worked to obtain θ by maximizing

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right\}$$

	X			Z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Introduction

Observed Data : $\mathbf{X} [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

Latent Variables : $\mathbf{Z} [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal : Find ML solution to model parameters

	X			Z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Let the set of model parameters be θ :

We worked to obtain θ by maximizing

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right\}$$



$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Observed Data

: $\mathbf{X} [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

Latent Variables

: $\mathbf{Z} [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal

: Find ML solution to model parameters

	\mathbf{X}			\mathbf{Z}	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Let the set of model parameters be θ :

We worked to obtain θ by maximizing

$$\ln p(\mathbf{X}|\theta) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \right\}$$

Summation inside the logarithm → Complicated expressions for ML solution
→ Couldn't get closed form solutions to ML parameters.

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Introduction

Observed Data

: $\mathbf{X} [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

Latent Variables

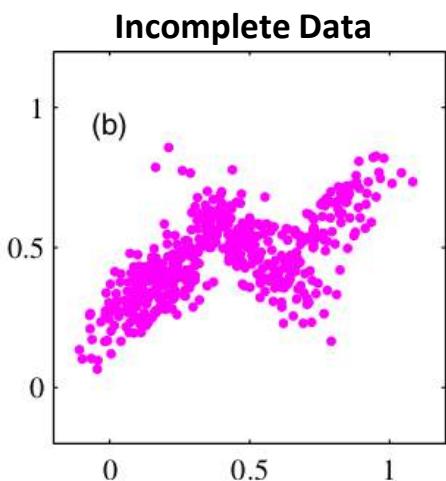
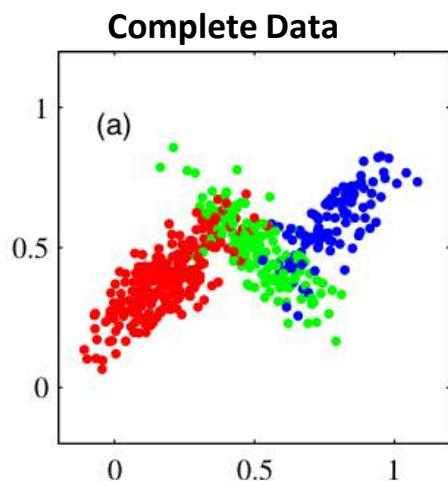
: $\mathbf{Z} [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal

: Find ML solution to model parameters

	X			Z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

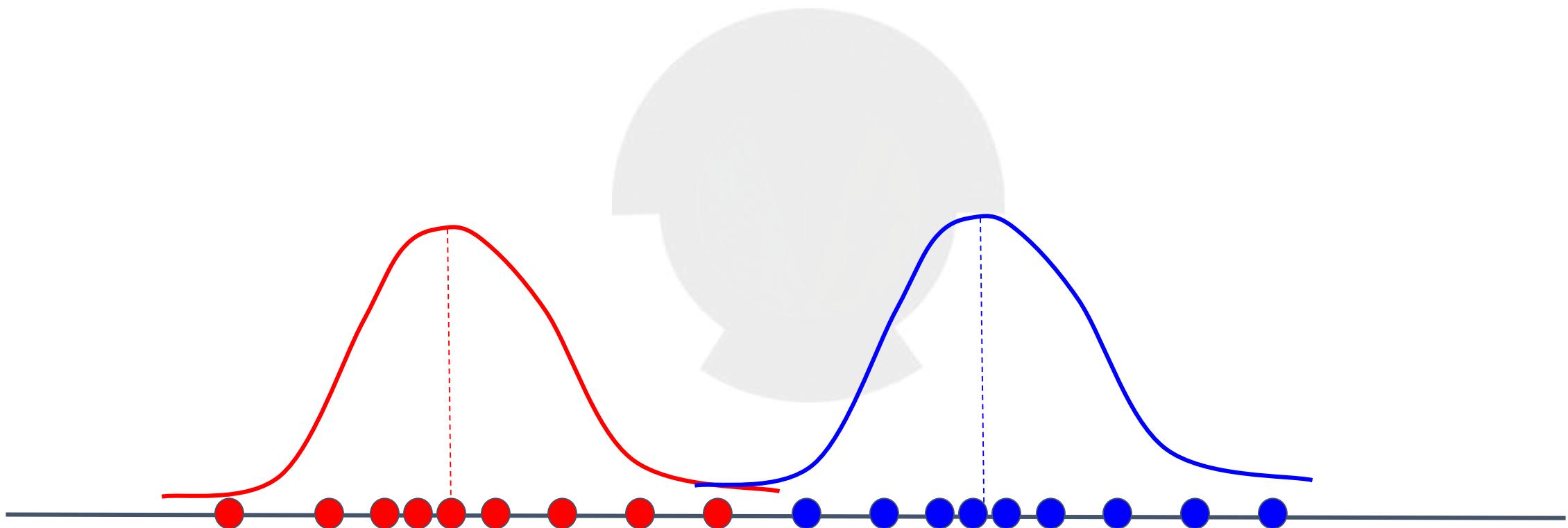
Let us suppose Z is given :



Summation inside the logarithm → Complicated expressions for ML solution
→ Couldn't get closed form solutions to ML parameters.

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Introduction



Introduction

Observed Data

: $\mathbf{X} [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$

Latent Variables

: $\mathbf{Z} [z^{(1)}, z^{(2)}, \dots, z^{(N)}]$

Goal

: Find ML solution to model parameters

	X			Z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Let us suppose Z is given :

The form of the complete data log likelihood becomes

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right\}$$

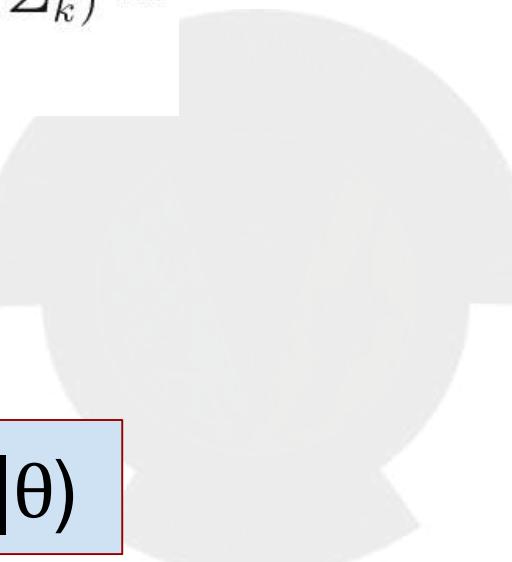
$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$



	x			z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...					...
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

$\ln p(\mathbf{X}, \mathbf{Z} | \theta)$

$$\ln p(\mathbf{X} | \theta) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \theta) \right\}$$

$$\ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \}$$

↗ ln $p(\mathbf{X}, \mathbf{Z} | \theta)$

$$\ln p(\mathbf{X} | \boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \right\}$$

$$\ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

	x			z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...			
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \}$$

We know that, z_{ik} is 1 only for one of the components for ith data.

	x			z	
1	x_{11}	...	x_{1D}	z_{11}	z_{12}
2	x_{21}	...	x_{2D}	z_{21}	z_{22}
...			
N	x_{N1}	...	x_{ND}	z_{N1}	z_{N2}

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \}$$

We know that, z_{ik} is 1 only for one of the components for ith data.

Let us suppose we need the ML estimate for $\boldsymbol{\mu}_k$.

	x				z	
1	x_{11}	...	x_{1D}	0	1	
2	x_{21}	...	x_{2D}	1	0	
...				
N	x_{N1}	...	x_{ND}	1	0	

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \}$$

	x				z	
1	x_{11}	...	x_{1D}	0	1	
2	x_{21}	...	x_{2D}	1	0	
...				
N	x_{N1}	...	x_{ND}	1	0	

We know that, z_{ik} is 1 only for one of the components for ith data.

Let us suppose we need the ML estimate for $\boldsymbol{\mu}_k$.

ML estimate for parameters is exactly for the parameters for a single gaussian for the fraction of examples with their $z_{ik}=1$

Introduction

Form of $p(\mathbf{X}, \mathbf{Z} | \theta)$ [For Mixture of Gaussians]:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \}$$

The ML estimates takes the same form as earlier:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad \pi_k = \frac{N_k}{N}$$

Note: Here γ takes the same form as earlier
Since we have \mathbf{z} , the solution is still closed form

	\mathbf{x}				\mathbf{z}
1	x_{11}	...	x_{1D}	0	1
2	x_{21}	...	x_{2D}	1	0
...				...	
N	x_{N1}	...	x_{ND}	1	0

Introduction

We have actually taken the discussion assuming, Z is given

- It is not given indeed !!!
 - Our knowledge of Z is only from $P(Z|X, \theta)$

	x				z	
1	x_{11}	...	x_{1D}	0	1	
2	x_{21}	...	x_{2D}	1	0	
...					...	
N	x_{N1}	...	x_{ND}	1	0	

Introduction

We have actually taken the discussion assuming, Z is given

- It is not given indeed !!!
 - Our knowledge of Z is only from $P(Z|X, \theta)$
 - Instead maximizing $\ln p(X, Z | \theta)$, do the following
 - Maximize its expectation under $P(Z|X, \theta)$!!!
 - That is $E_z [\ln p(X, Z | \theta)]$ denoted as $Q(\theta, \theta^{\text{old}})$

$$Q(\theta, \theta^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\theta)$$

	x				z	
1	x_{11}	...	x_{1D}	0	1	
2	x_{21}	...	x_{2D}	1	0	
...					...	
N	x_{N1}	...	x_{ND}	1	0	

General EM Algorithm

1. Choose an initial setting for the parameters θ^{old} .
2. E-Step: Evaluate $p(Z|X, \theta^{\text{old}})$
3. M-Step:

Evaluate θ^{new} given by

$$\theta^{\text{new}} = \arg \max_{\theta} Q(\theta, \theta^{\text{old}})$$

Where

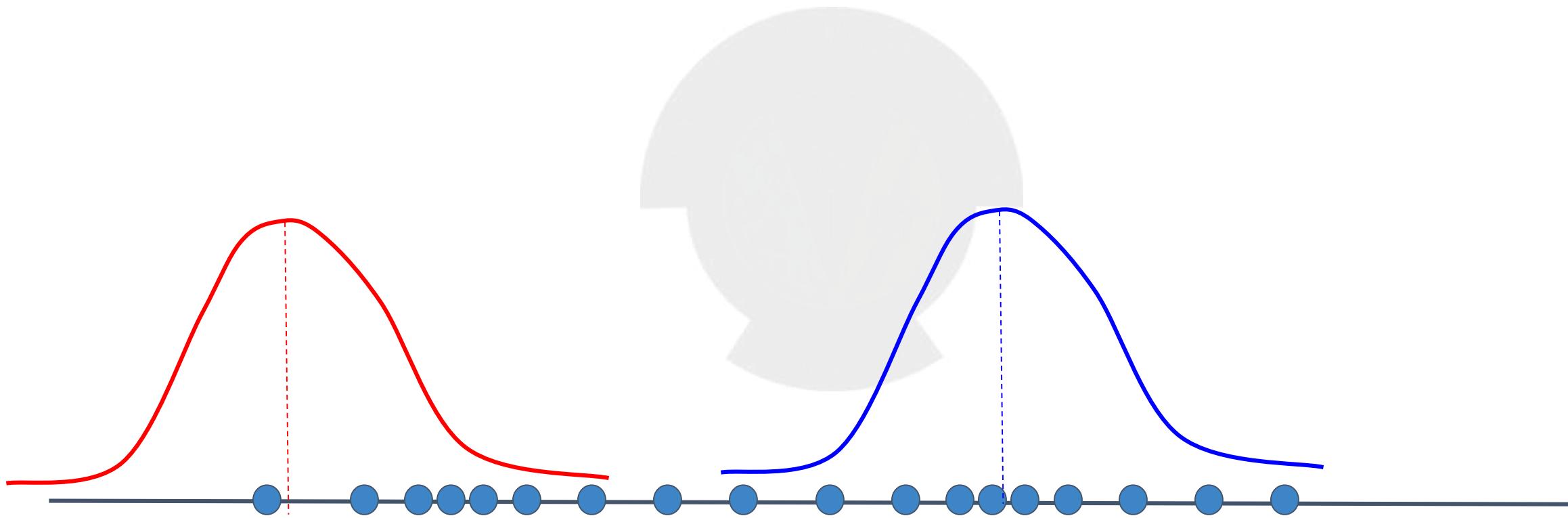
$$Q(\theta, \theta^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\theta)$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\theta^{\text{old}} \leftarrow \theta^{\text{new}}$$

and return to step-2

Introduction



General EM Algorithm - for MoG

1. Choose initial values for parameters μ^{old} , Σ^{old} and π^{old}
2. **E-Step** : Compute $\gamma(z_{nk})$
3. **M-Step** : Keep $\gamma(z_{nk})$ fixed, and maximize $E_z[\ln p(X,Z|\theta)]$ for μ_k , Σ_k and π_k , to get μ^{new} , Σ^{new} and π^{new} ,
4. Repeat E & M until convergence

Expectation-Maximization Algorithm

To Estimate:

M-Step

Initialize π, μ, Σ and
also evaluate the log likelihood

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$

Perform E-Step Given $\gamma(z_k)$

E-Step

$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) = \\ &= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

Perform M-Step Given π, μ, Σ

$$p(z_k = 1) = \pi_k$$

Repeat Until Convergence

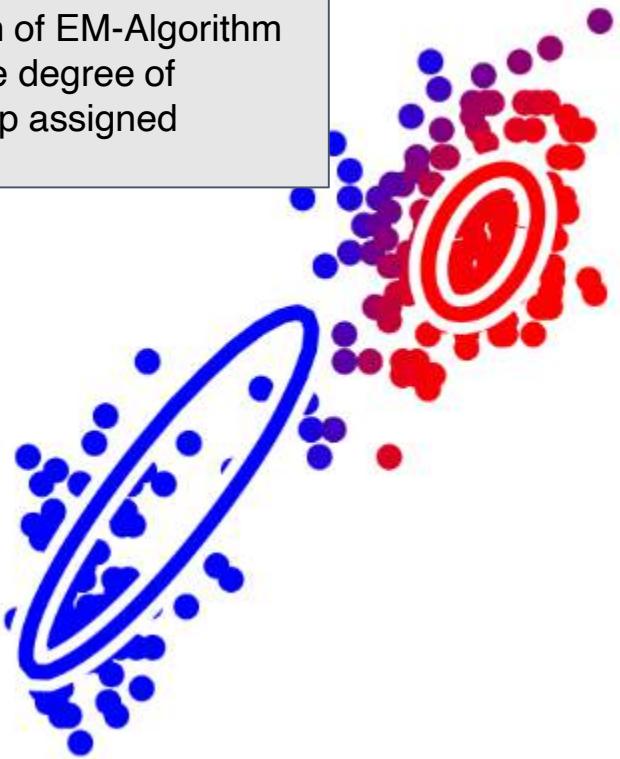
Agenda

- Relationship between K-Means and EM Algorithm

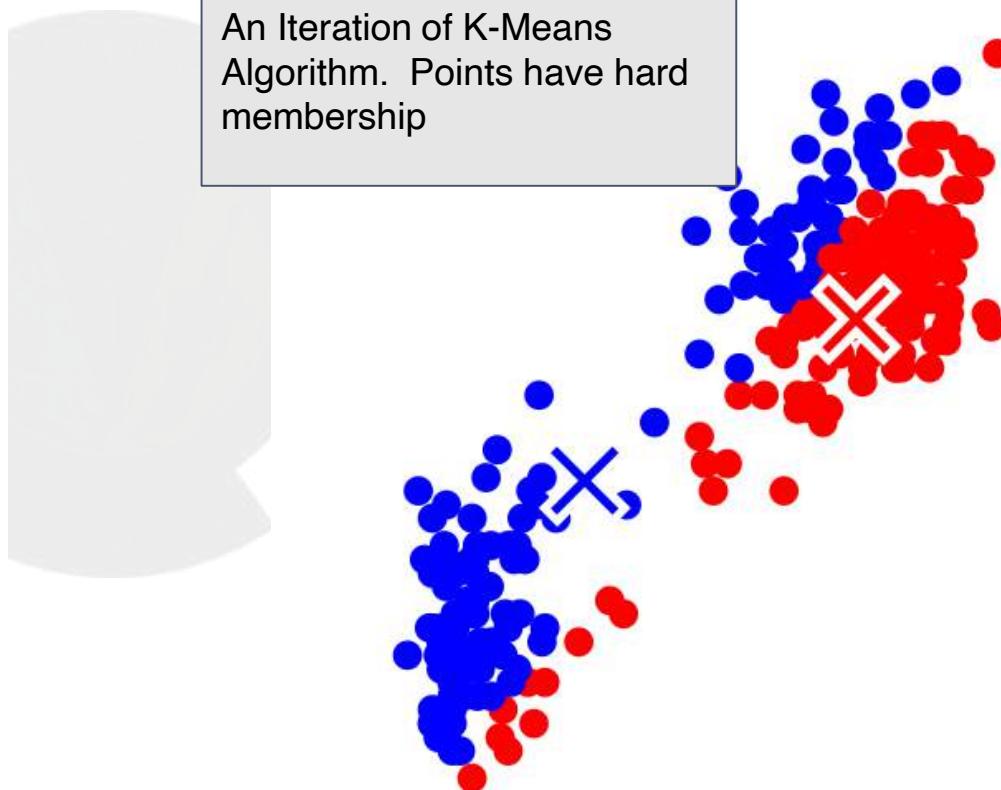
Introduction

- Hard Assignment Vs. Soft Assignment

An Iteration of EM-Algorithm
Points have degree of membership assigned



An Iteration of K-Means Algorithm. Points have hard membership



Introduction

- K-Means does not have covariance parameter for its clusters.
 - Let us try to find conditions that approximate k-means from EM-algorithm

Introduction

- K-Means does not have covariance parameter for its clusters.
 - Let us try to find conditions that approximate k-means from EM-algorithm
 - Assume all components of EM Algorithm uses same variance

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp \left\{ -\frac{1}{2\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \right\}$$

Introduction

- K-Means does not have covariance parameter for its clusters.
 - Let us try to find conditions that approximate k-means from EM-algorithm
 - Assume all components of EM Algorithm uses same variance

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp \left\{ -\frac{1}{2\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \right\}$$

- But in k-means, we do not estimate !!!
 - Fine! Let us keep this as a constant across all iterations

Introduction

- K-Means does not have covariance parameter for its clusters.
 - Let us try to find conditions that approximate k-means from EM-algorithm
 - Assume all components of EM Algorithm uses same variance

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp \left\{ -\frac{1}{2\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \right\}$$

- But in k-means, we do not estimate !!!
 - Fine! Let us keep this as a constant across all iterations
- For a particular point \mathbf{x}_n , its posterior is given by

$$\gamma(z_{nk}) = \frac{\pi_k \exp \{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\}}{\sum_j \pi_j \exp \{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\}}$$

Introduction

- So What about γ ?
 - This behaves in a particular fashion as $\epsilon \rightarrow 0$
 - Let x_i is closer to cluster represented by μ_p , that is $k = p$.
 - Every other terms becomes zero except for $k = p$.
 - That is for $k = p$, $\gamma(z_{np}) = 1$
 - That is hard assignment !!!

$$\gamma(z_{nk}) = \frac{\pi_k \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\}}{\sum_j \pi_j \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\}}$$

Introduction

- So What about γ ?
 - This behaves in a particular fashion as $\gamma \rightarrow 0$
 - Let x_i is closer to cluster represented by μ_p , that is $k = p$.
 - Every other terms becomes zero except for $k = p$.
 - That is for $k = p$, $\gamma(z_{np}) = 1$
 - That is hard assignment !!!
 - Also for $\gamma = 0$, the form of distortion function J used in K-Means and EM's log likelihood tends to be same

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] \rightarrow -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \text{const}$$

Introduction

- EM Algorithm breaks down the problem of estimating the ML parameters of a Mixture Model into E & M Steps and provides an iterative way of computing this
- EM Algorithm converges slowly
 - Use K-Means in the initialization
- EM Algorithm converges to local optimum
 - Sensitive to initialization
 - Each iteration increases the Log likelihood until convergence
 - Many restarts
- What is a good k?
- Issues due to singularity

Some Applications

Parameter Estimation for a Partially Observed Data

- Clustering
 - Image segmentation
 - Document Clustering
 - Object Tracking - [With the help of distribution of the object of interest, we can track its distributions in the next frames]
 - Generating summaries of document collection / detecting salient objects in image collection
- Estimating ML parameters for many kind of machine learning models [particularly widely used for graphical models]



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Hierarchical Clustering Algorithms

S.P.Vimal

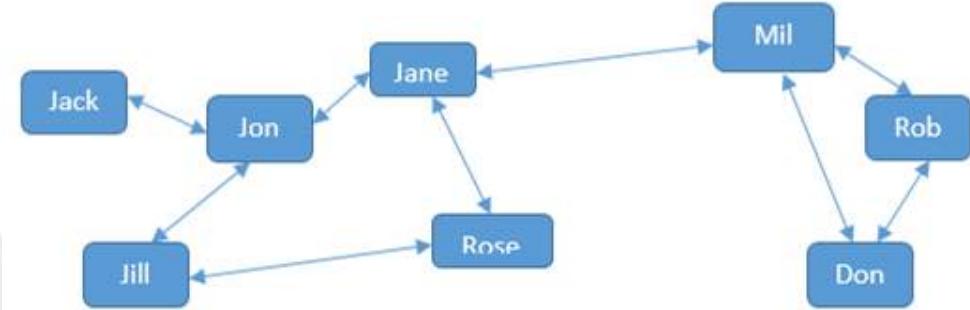
Agenda

Hierarchical Clustering Algorithms



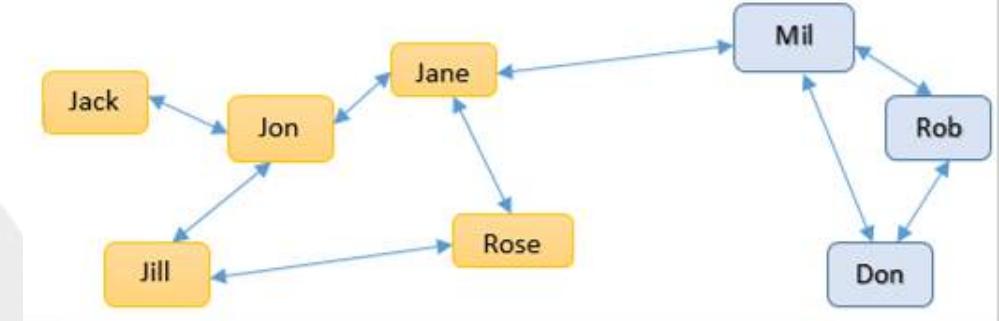
Hierarchical Clustering

- Helps discovering hierarchical structure in the data
 - Data from certain domains expected to possess hierarchical structure due to the nature of process generating the data
 - Social Networks



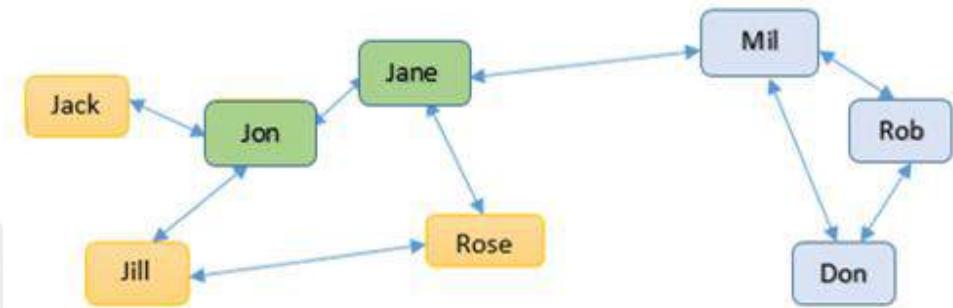
Hierarchical Clustering

- Helps discovering hierarchical structure in the data
 - Data from certain domains expected to possess hierarchical structure due to the nature of process generating the data
 - Social Networks



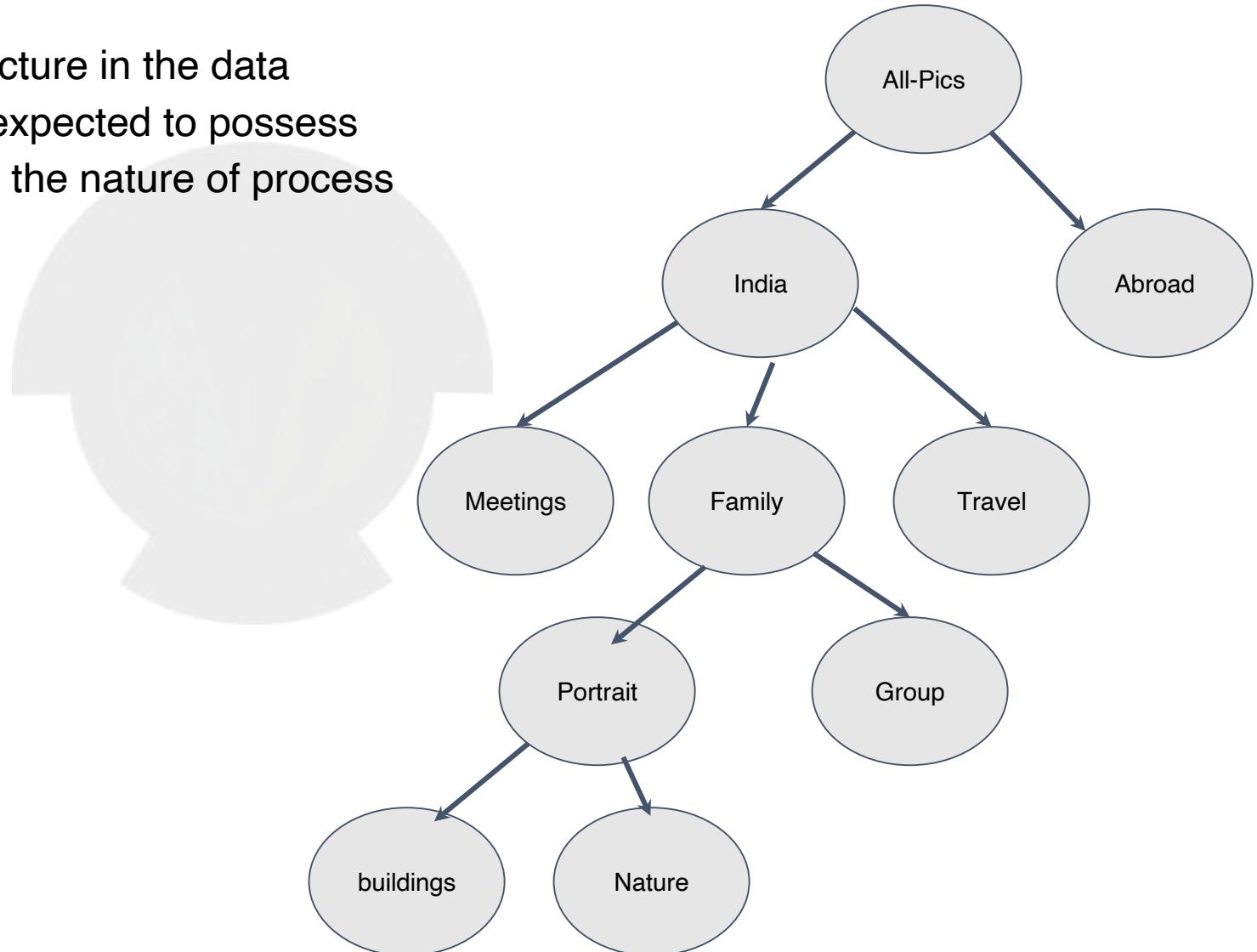
Hierarchical Clustering

- Helps discovering hierarchical structure in the data
 - Data from certain domains expected to possess hierarchical structure due to the nature of process generating the data
 - Social Networks



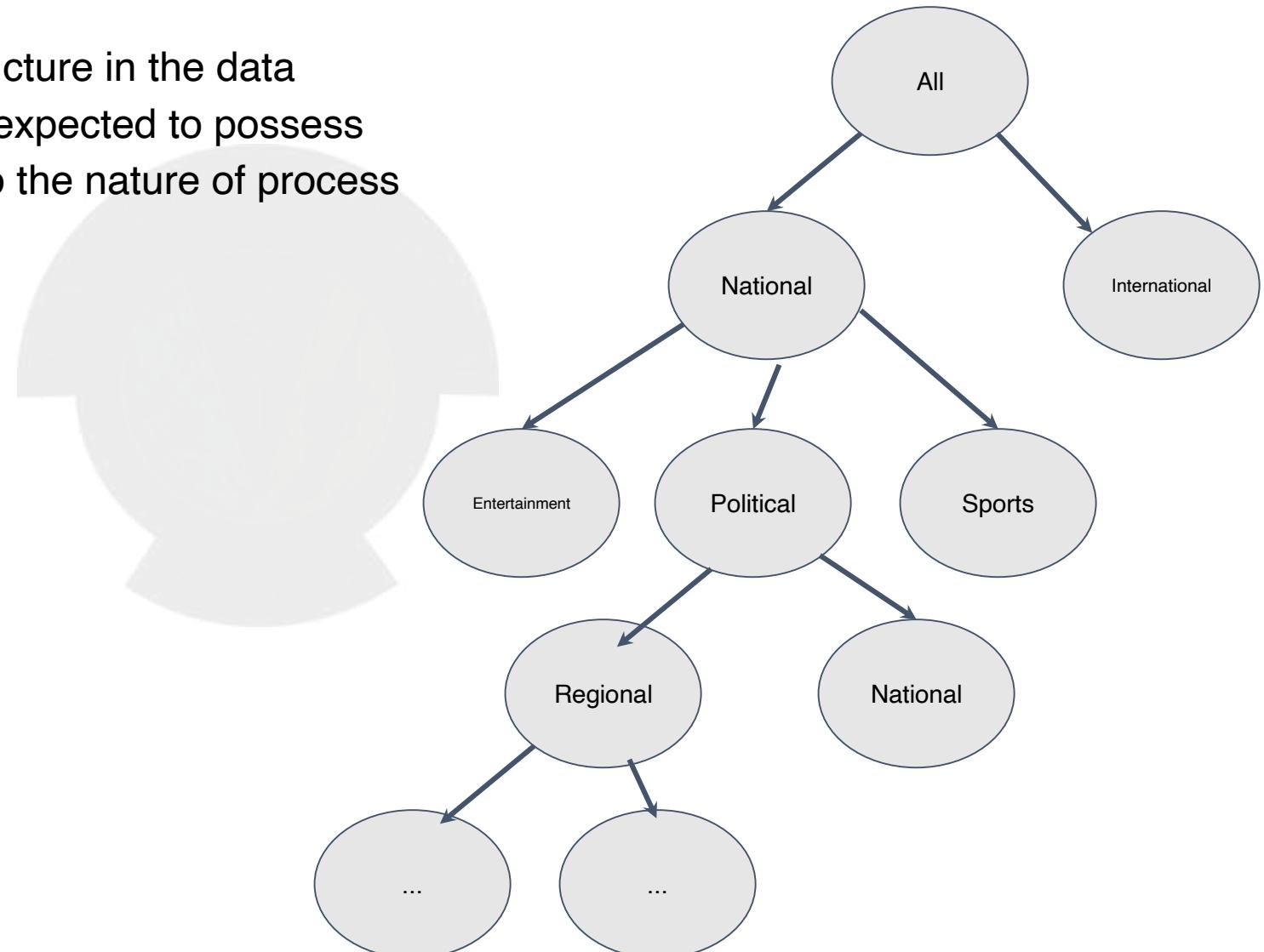
Hierarchical Clustering

- Helps discovering hierarchical structure in the data
 - Data from certain domains expected to possess hierarchical structure due to the nature of process generating the data
- Social Networks
- Travel Photographs



Hierarchical Clustering

- Helps discovering hierarchical structure in the data
 - Data from certain domains expected to possess hierarchical structure due to the nature of process generating the data
- Social Networks
- Travel Photographs
- Documents / News articles / ...



Hierarchical Clustering

Some Limitations of Partition based clustering

- K - Value
- Initialization
- Thresholds

Hierarchical Clustering

Hierarchical Approaches to clustering

- Greedy
- Deterministic
- Visually Interpretable (Mostly)

In this module

- Approaches to Hierarchical Clustering
- Basic hierarchical clustering algorithms – Single, Complete, Average Linkage

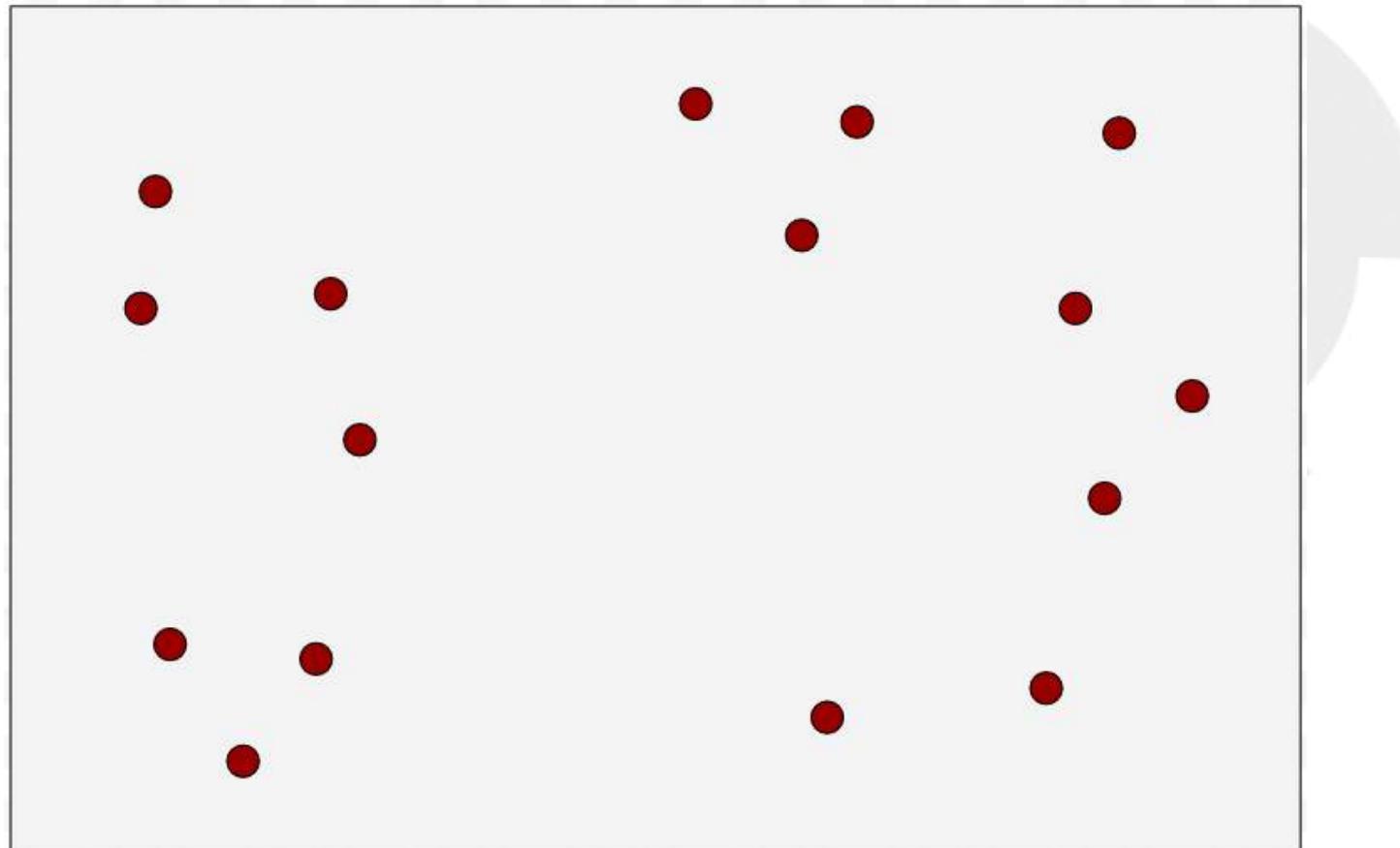
Agenda

Agglomerative & Divisive Hierarchical Clustering Algorithms



Agglomerative Algorithms

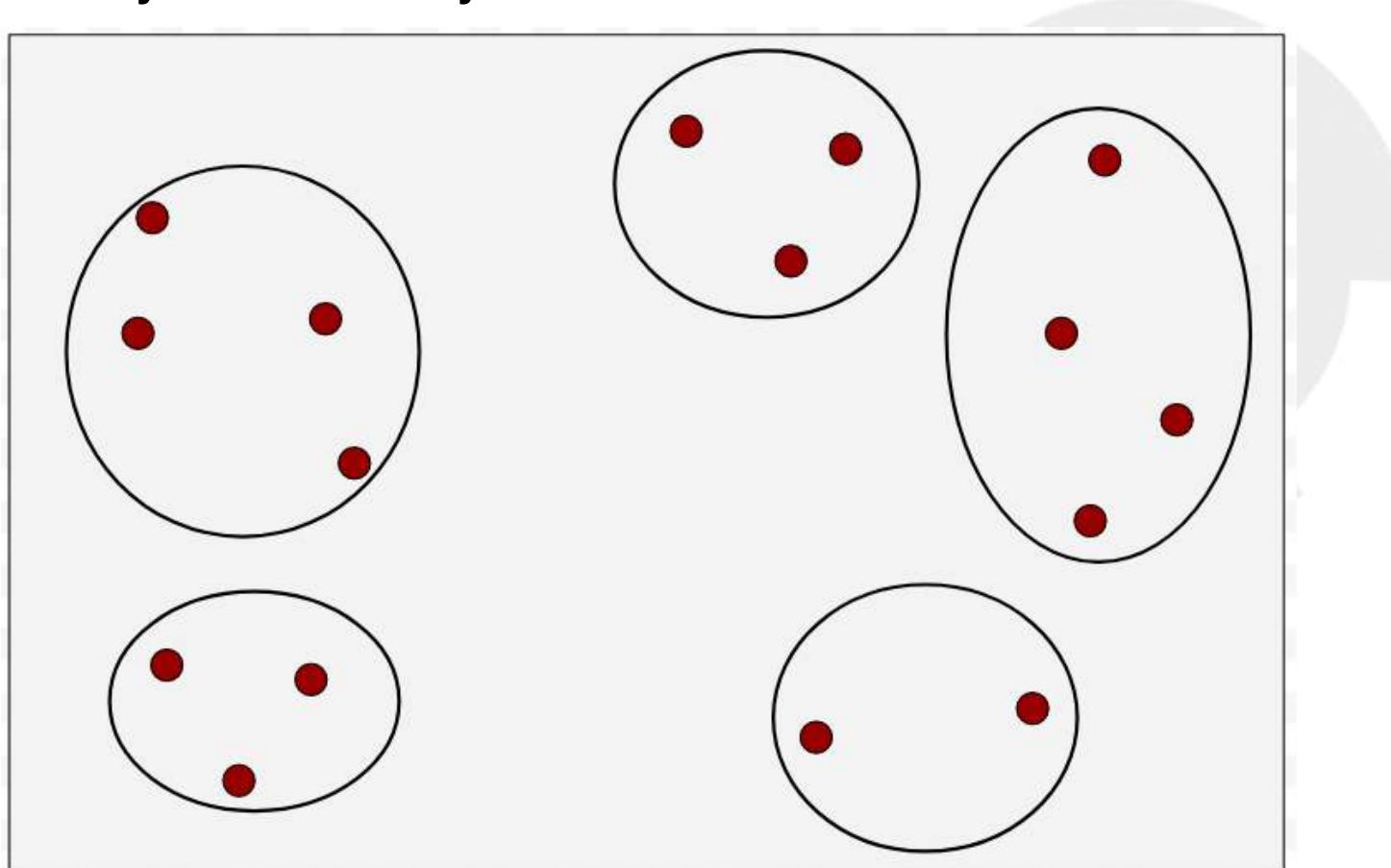
- Begins placing each objects in individual clusters and works to merge objects systematically



- Works bottom - up all objects are placed in single cluster or a termination is reached

Agglomerative Algorithms

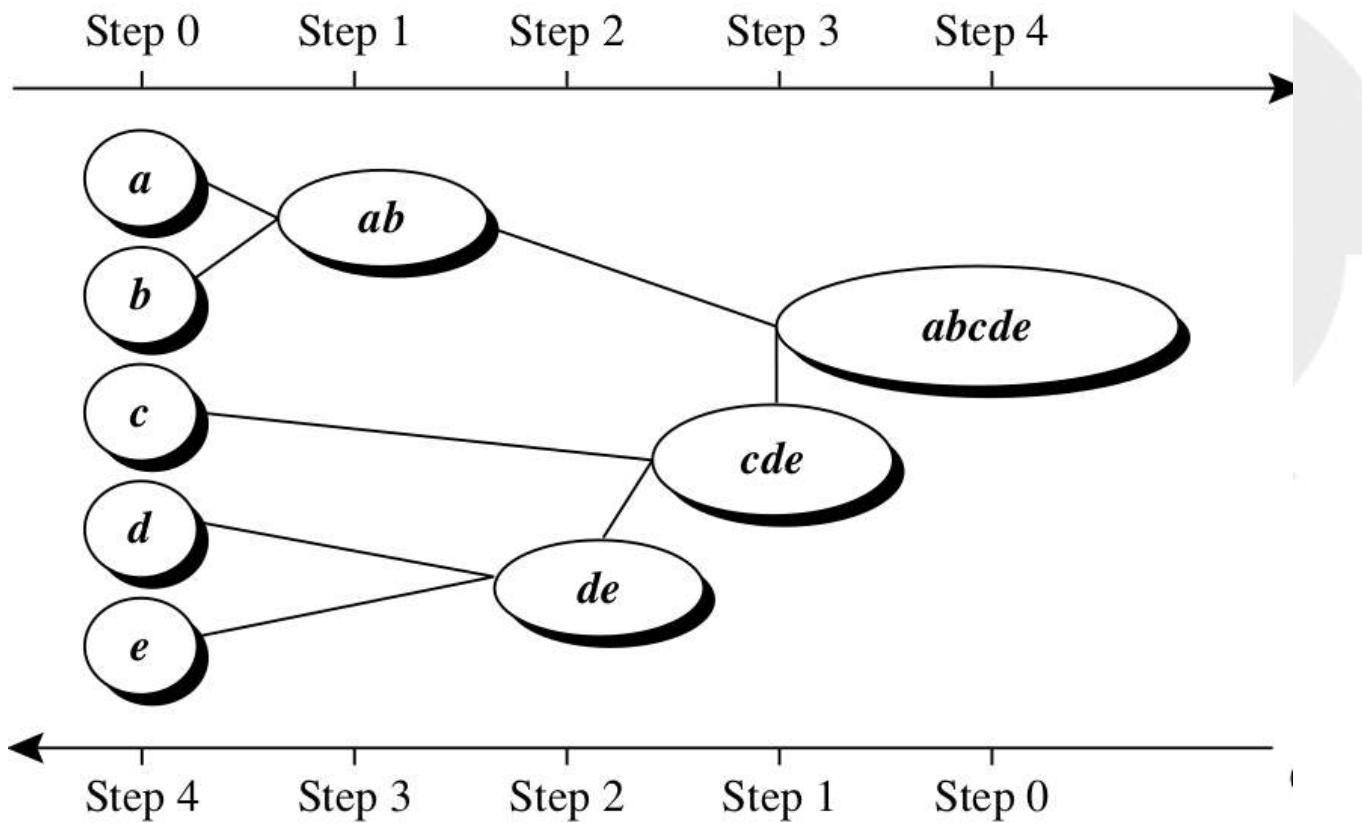
- Begins placing each objects in individual clusters and works to merge objects systematically



- Works bottom - up until objects are placed in single cluster or a termination is reached
 - What is the termination condition?
- Measures of inter-cluster distances plays role in merging majorly
 - What is the measure?
- Noisy points and Outliers can change the merging decisions - Algorithm is sensitive to noise

Agglomerative Algorithms

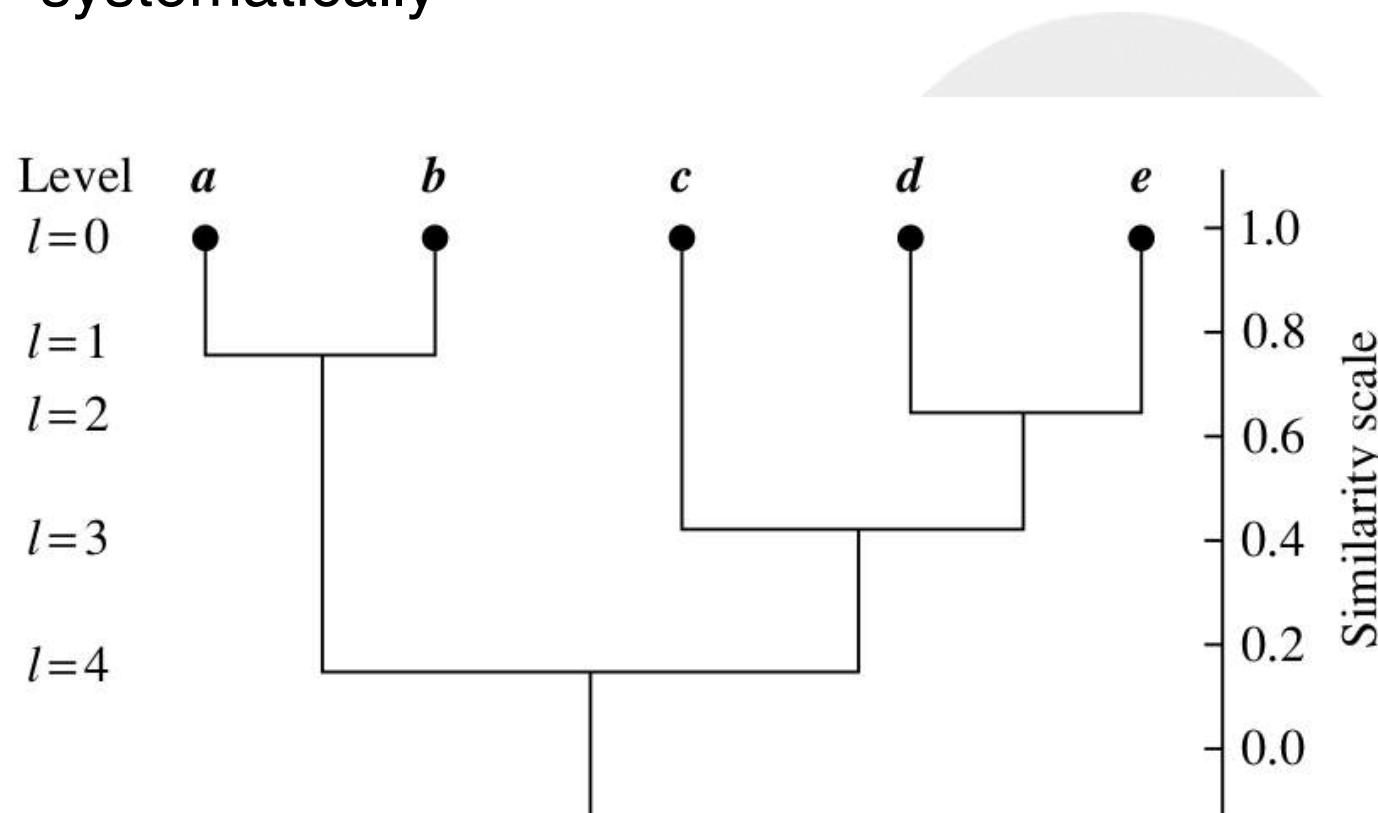
- Begins placing each objects in individual clusters and works to merge objects systematically



- Works bottom - up until objects are placed in single cluster or a termination is reached
 - What is the termination condition?
- Measures of inter-cluster distances plays role in merging majorly
 - What is the measure?
- Noisy points and Outliers can change the merging decisions - Algorithm is sensitive to noise

Agglomerative Algorithms

- Begins placing each objects in individual clusters and works to merge objects systematically



Dendrogram:

- Graphical way to represent the process of hierarchical clustering
- Used for both top-down and bottom-up clustering

Outline of General Agglomerative Algorithm

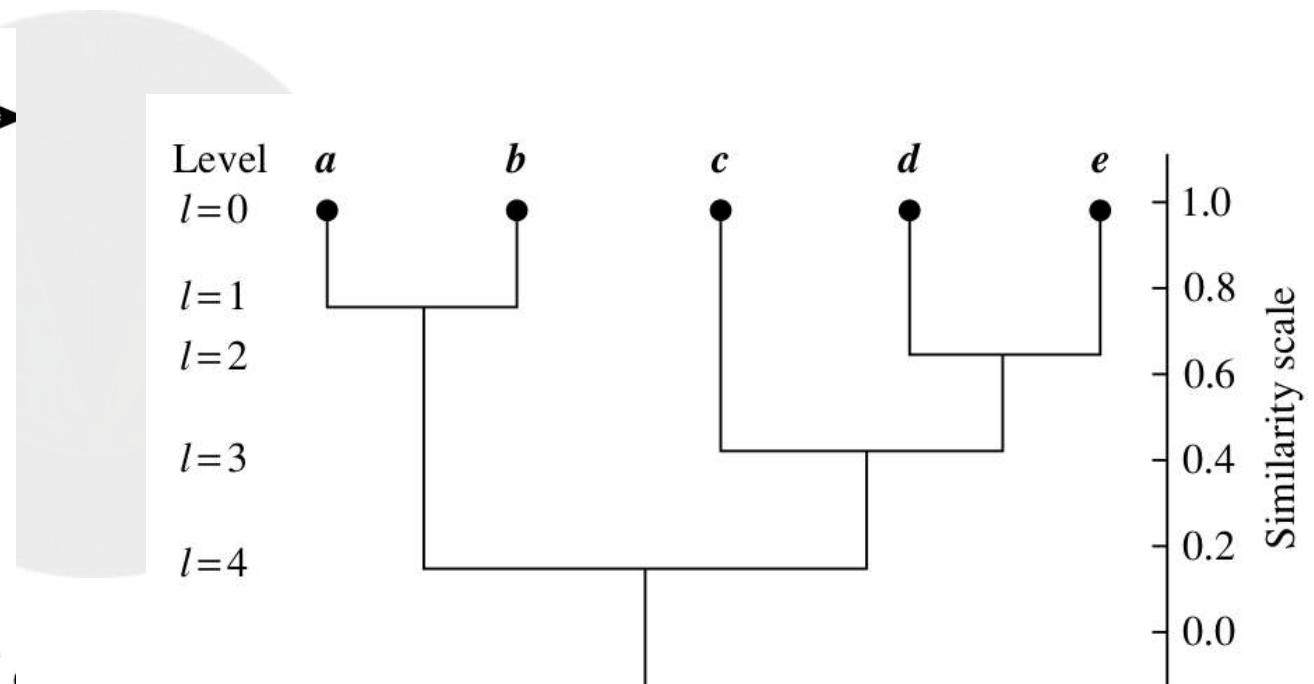
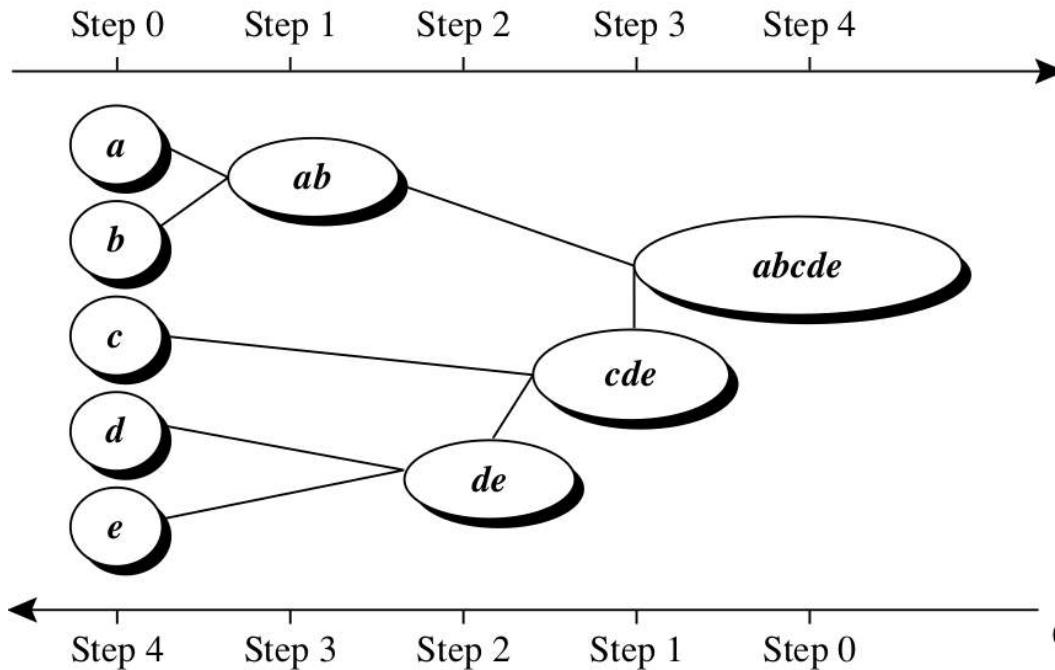
```
1. for i = 1 ... n :  
    1-1: Let  $C_i = \{x_i\}$   
2. while there is more than one cluster left do  
    2-1: find i and j such that  $D(C_i, C_j)$  is the lowest  
    2-2: Merge  $C_i, C_j$  as  $C_i$   
    2-3: Remove  $C_j$   
2. end
```

Agglomerative Algorithm

- Use of dissimilarity function guiding merges
 - This dissimilarity will keep increasing with consecutive merges
- Works well when the data is inherently hierarchical
- Does not scale well
- Do not undo merges was done previously

Divisive Algorithms

- Employs a top down strategy



Divisive Algorithms

Challenges in Divisive Clustering:

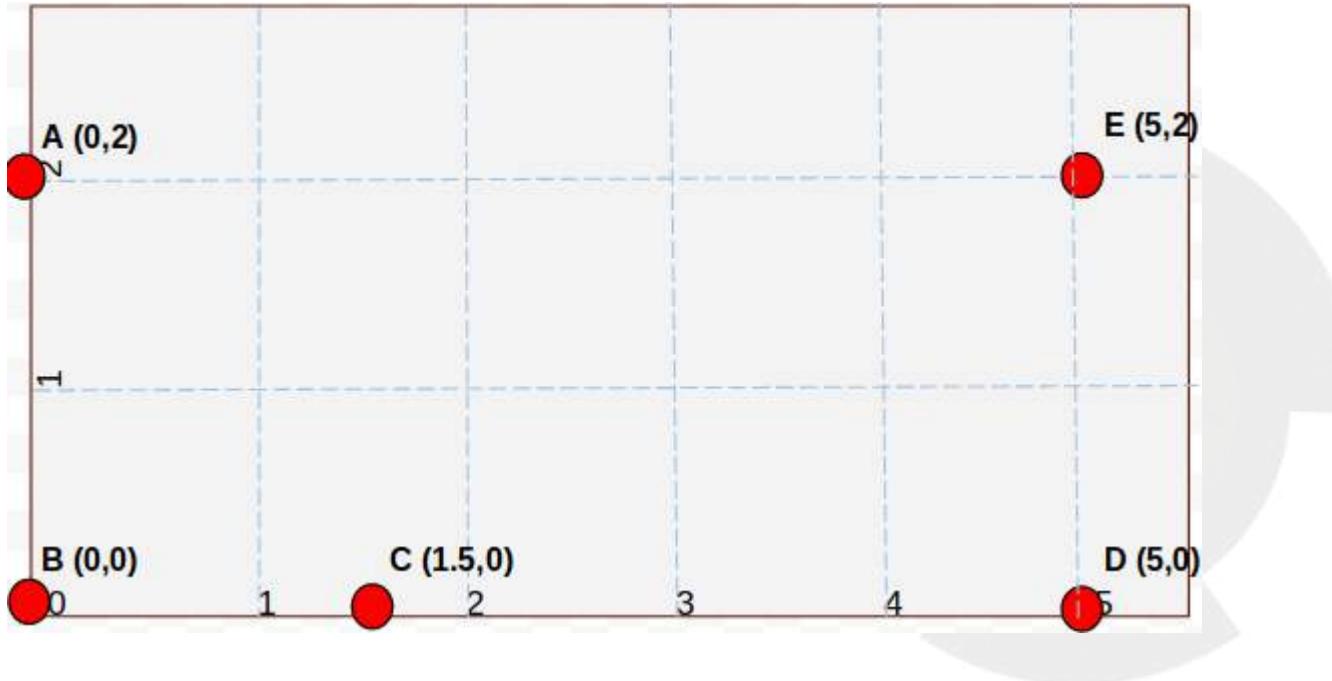
- (1) there are $2^{n-1}-1$ ways to partition a set of n objects into two exclusive subsets, where n is the number of objects
 - (a) Uses heuristics in partitioning, which leads to biases in the clustering process

Agenda

- Single Linkage Clustering
- Discussions

Single Linkage Clustering

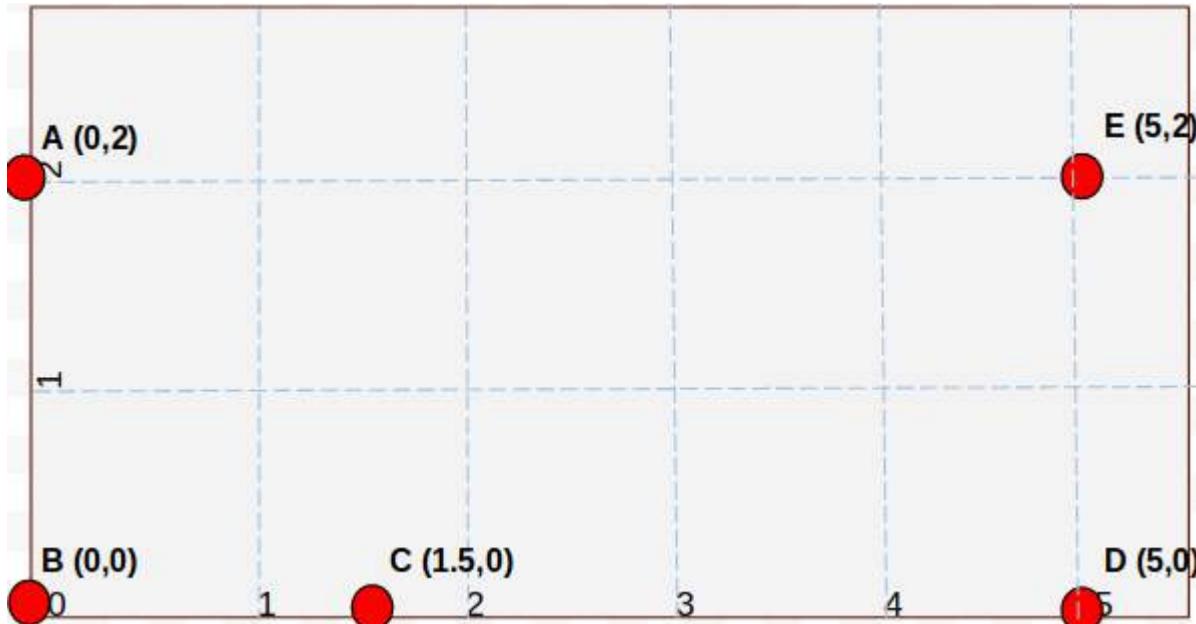
- An Agglomerative Hierarchical Clustering Approach



$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|$$

Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



Distance between B & C is 1.5

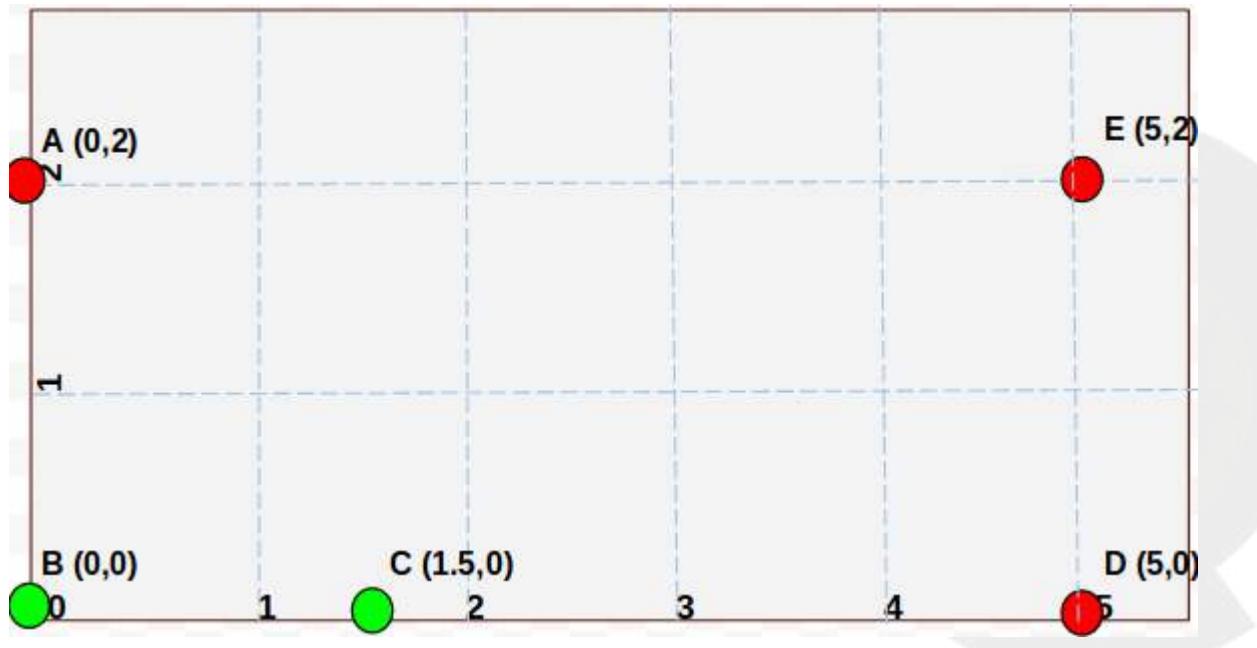
Merge B and C now.



Dendrogram : It is a tree which shows how clusters are merged/ split hierarchically

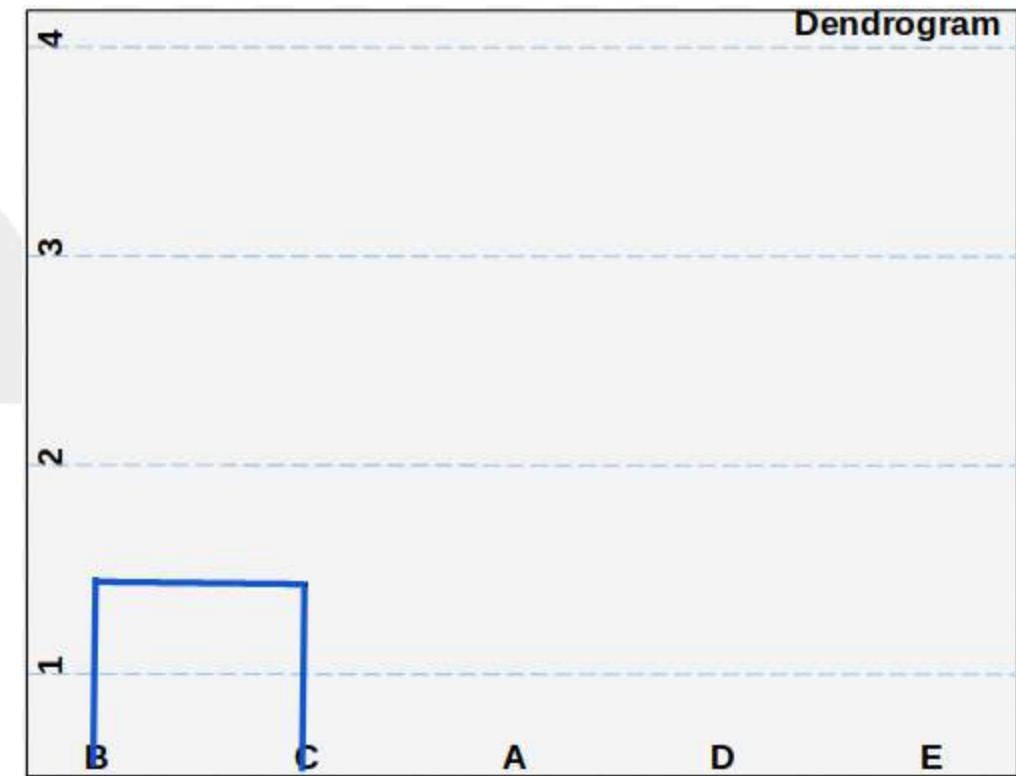
Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'| \}$$

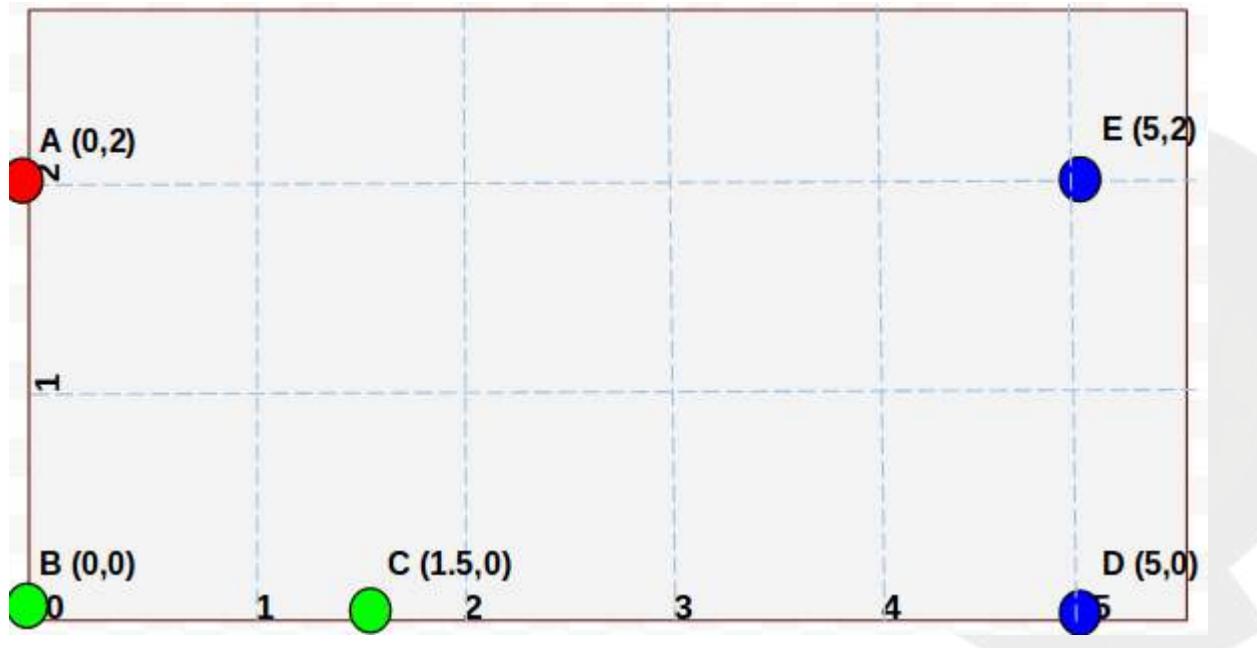
Distance between D & E are the next closest with the distance between them is 2
Merge B and C now.



Dendrogram : It is a tree which shows how clusters are merged/split hierarchically

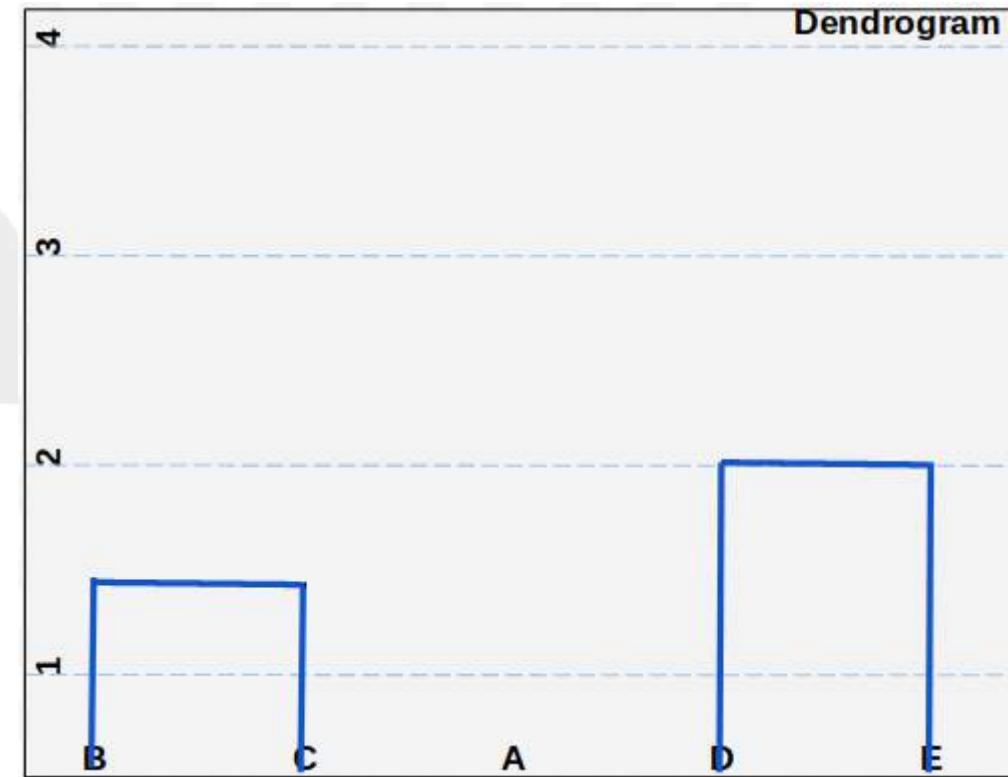
Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{ |p - p'| \}$$

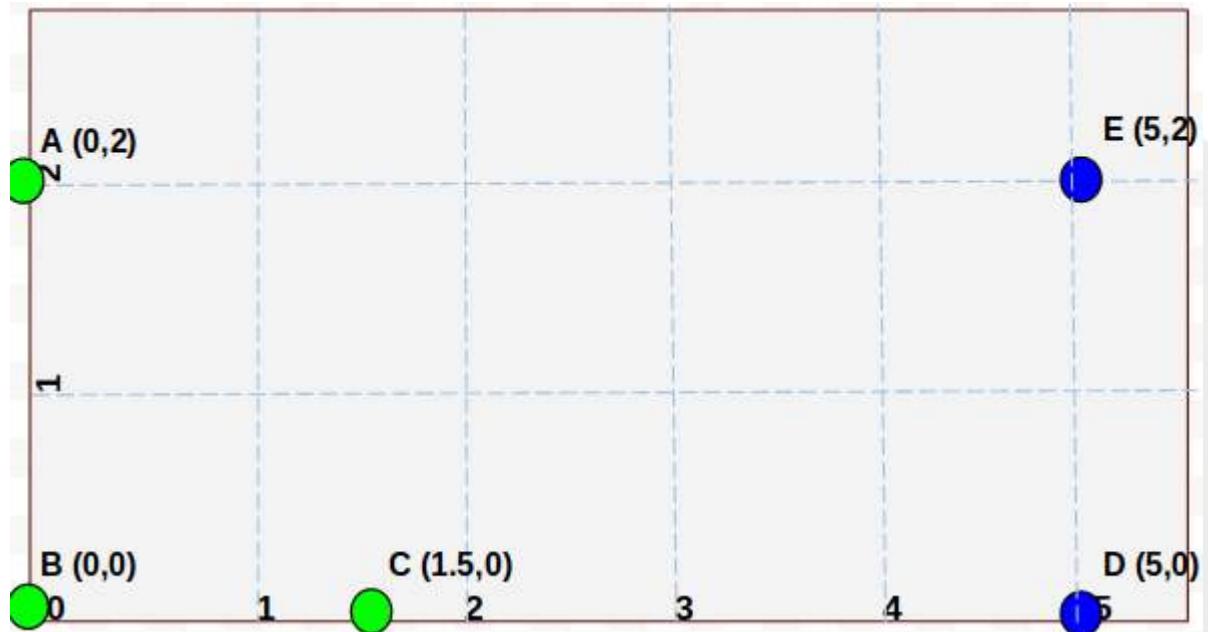
Distance between D & E are the next closest with the distance between them is 2
Merge B and C now.



Dendrogram : It is a tree which shows how clusters are merged/split hierarchically

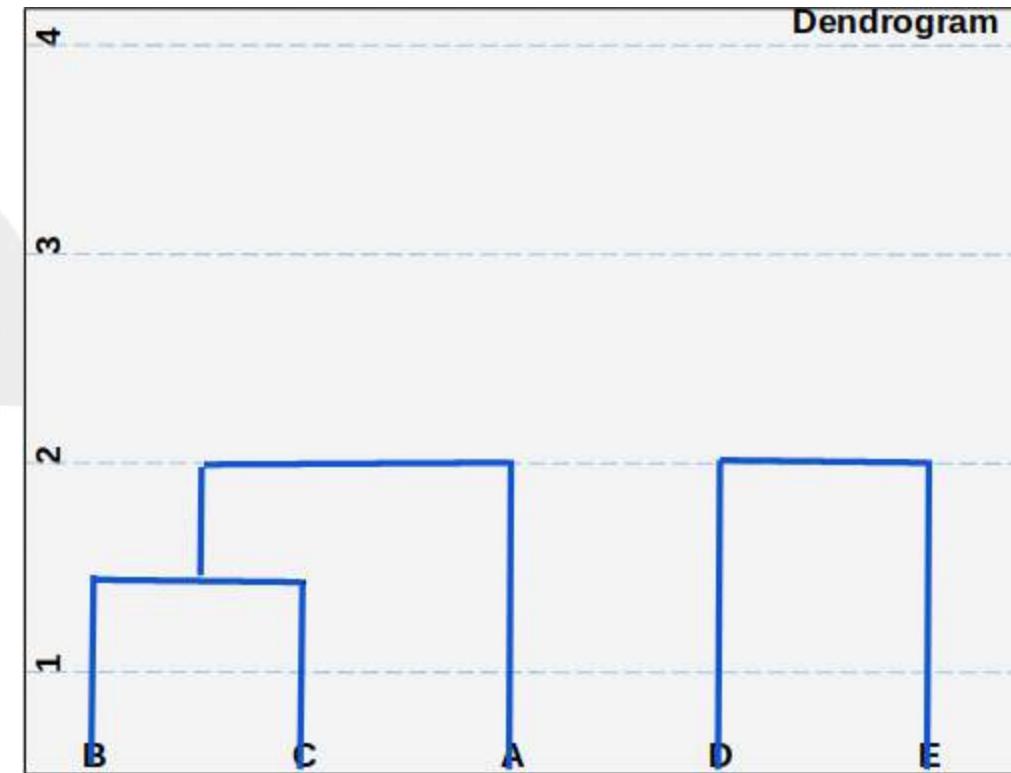
Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'| \}$$

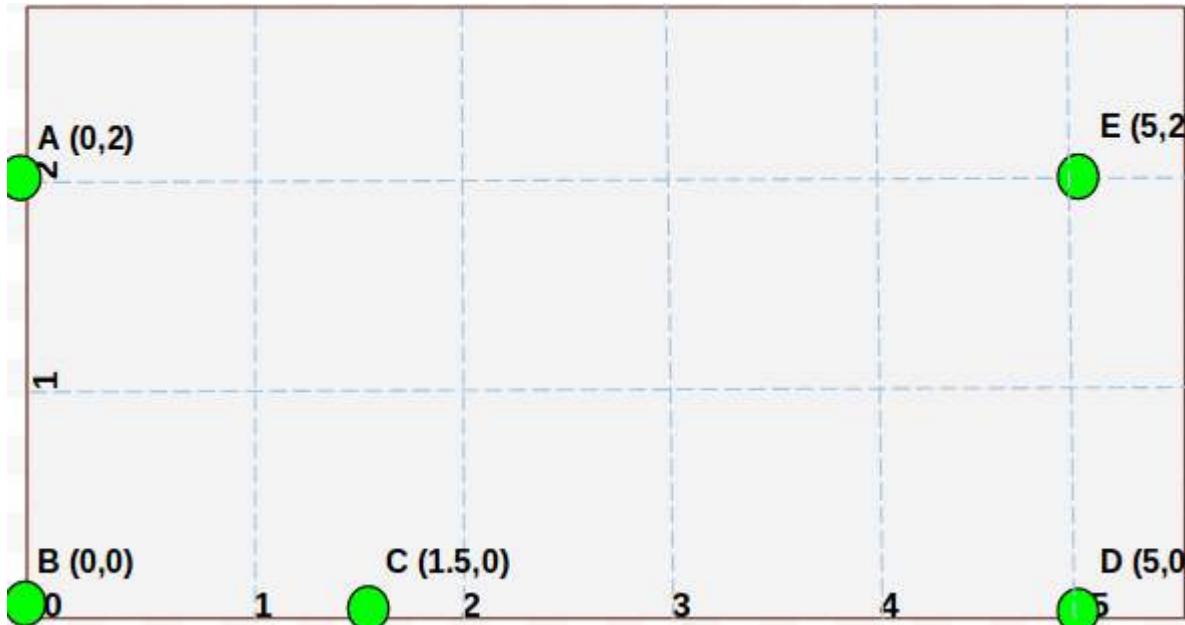
Distance between {B,C} & A : 2
Merged



Dendrogram : It is a tree which shows how clusters are merged/split hierarchically

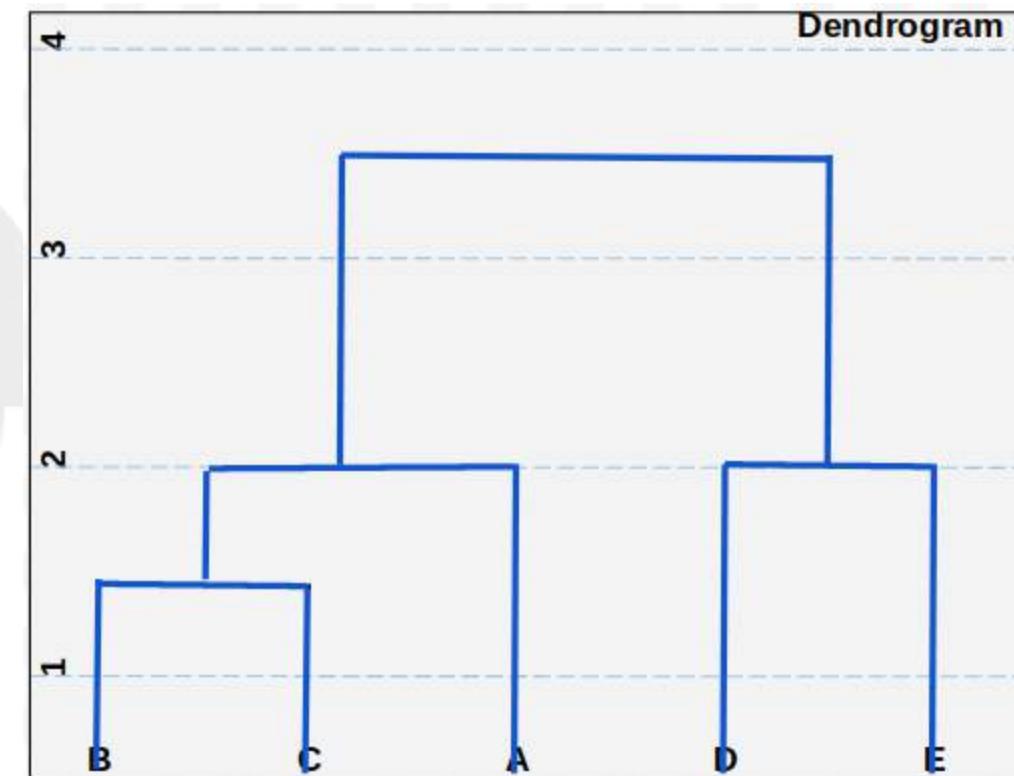
Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'| \}$$

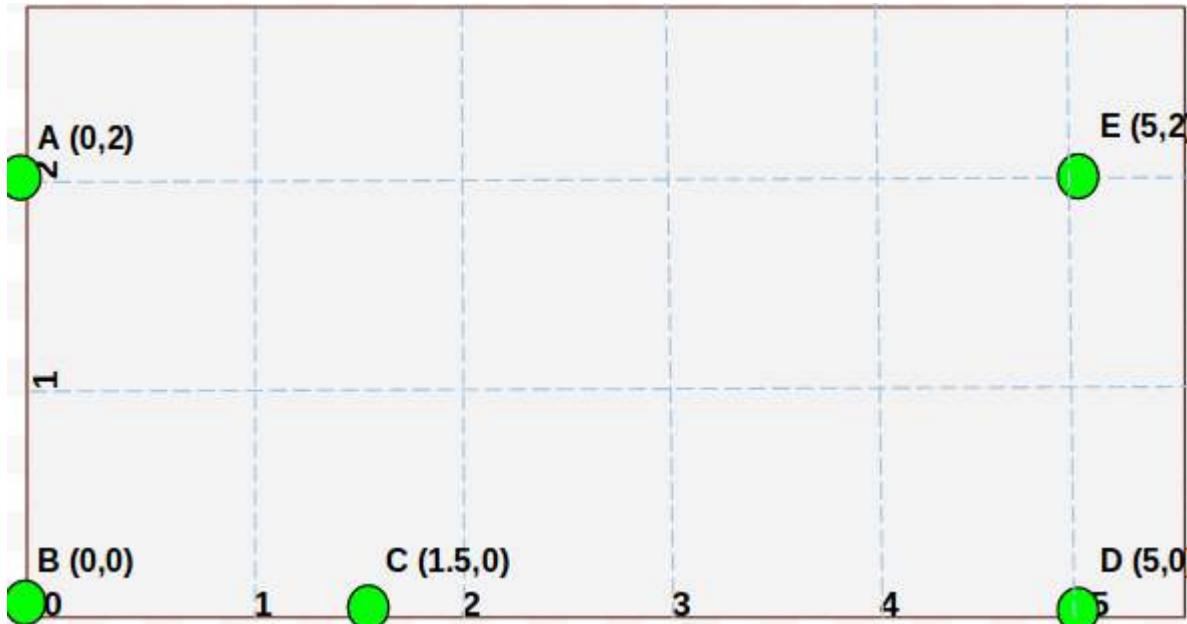
Distance between {A,B,C} and {D,E} is 3.5
Merged



Dendrogram : It is a tree which shows how clusters are merged/split hierarchically

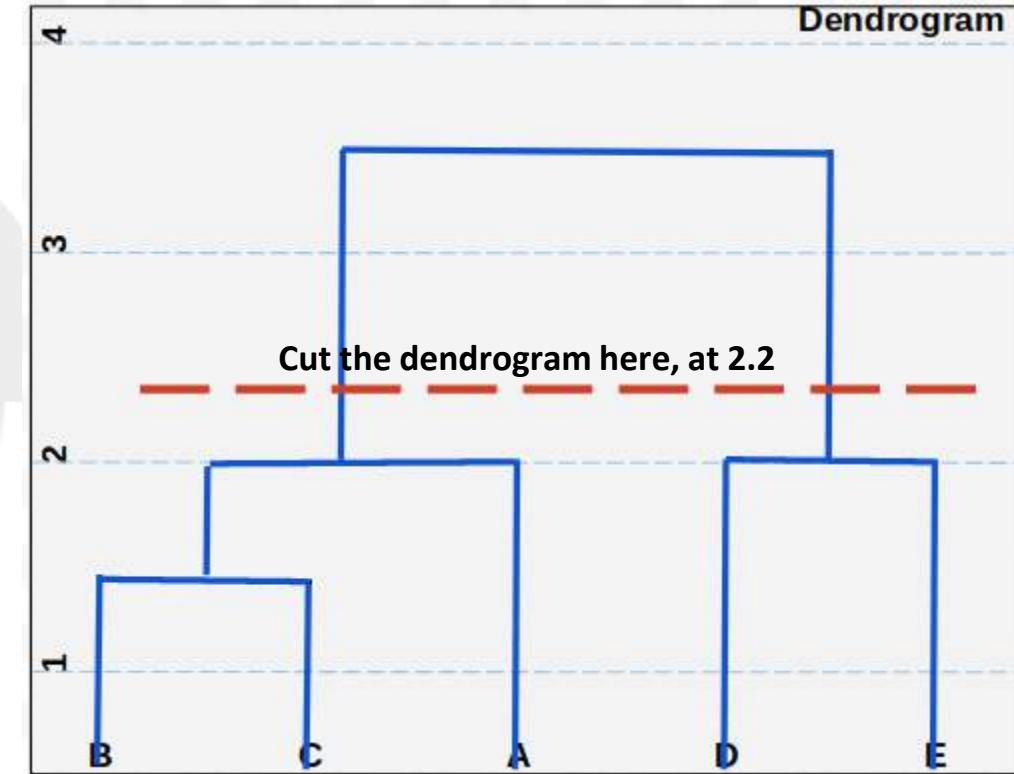
Single Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



How to terminate the process?

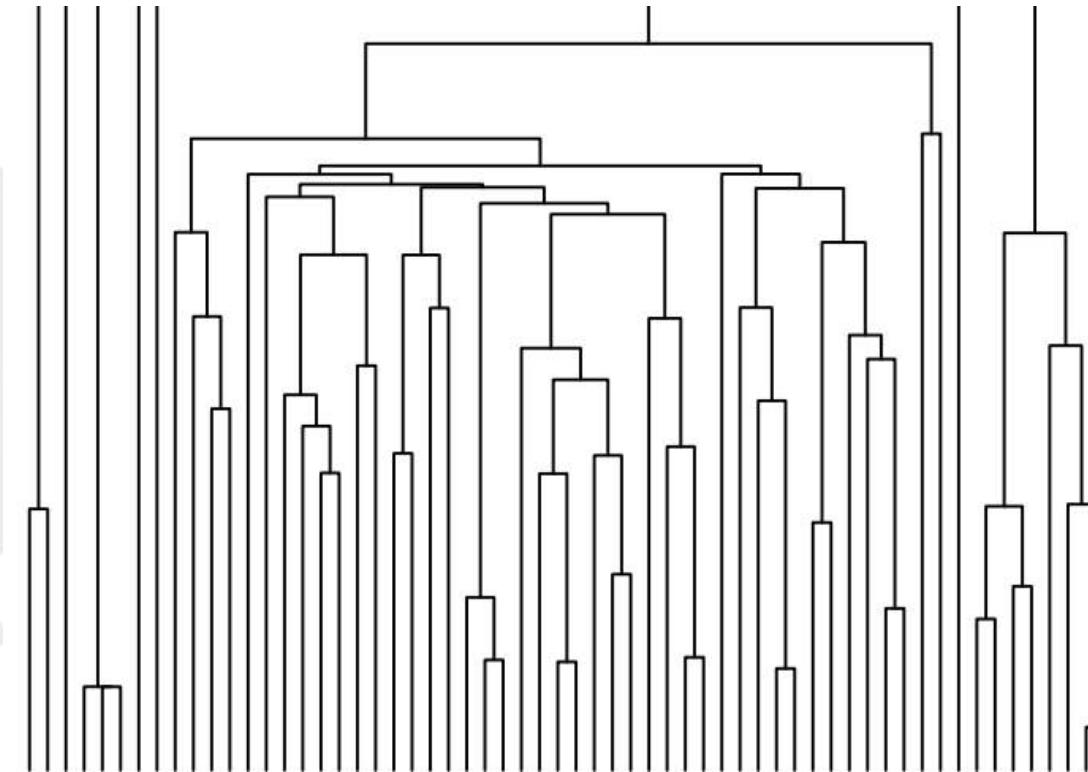
- (1) Use a predetermined K
- (2) Use a threshold on the closest distance between clusters
- (3) Use threshold on measures of compactness of individual clusters



Dendrogram : It is a tree which shows how clusters are merged/split hierarchically

Comments on the algorithm

- As the size of data grows, the dendrogram gets huge
[general issue]
 - Interpreting the dendrogram is a difficult task



Comments on the algorithm

- As the size of data grows, the dendrogram gets huge [general issue]
 - Interpreting the dendrogram is a difficult task

Chaining Issue:

- Merging criterion is local (nearest neighbour)
- It is likely that for large iterations, only adjacent points are added to cluster and no larger clustering pattern emerges



Comments on the algorithm

Time Complexity:

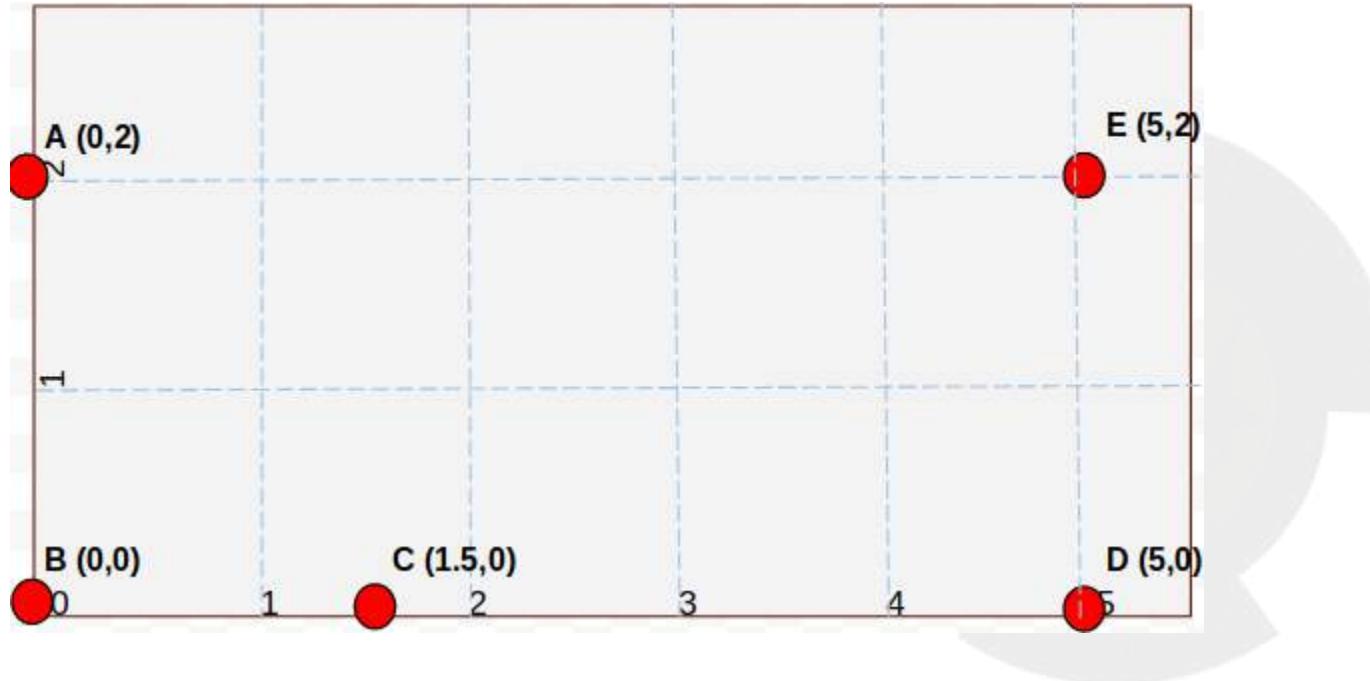
- Naive Implementation of the algorithm takes $O(n^3)$ time
 - Pairwise distance between points $O(n^2)$ time
 - Number of iterations $O(n)$
- Use of priority queue helps bringing the complexity to $O(n^2 \log n)$

Agenda

- Complete Linkage Clustering
- Discussions

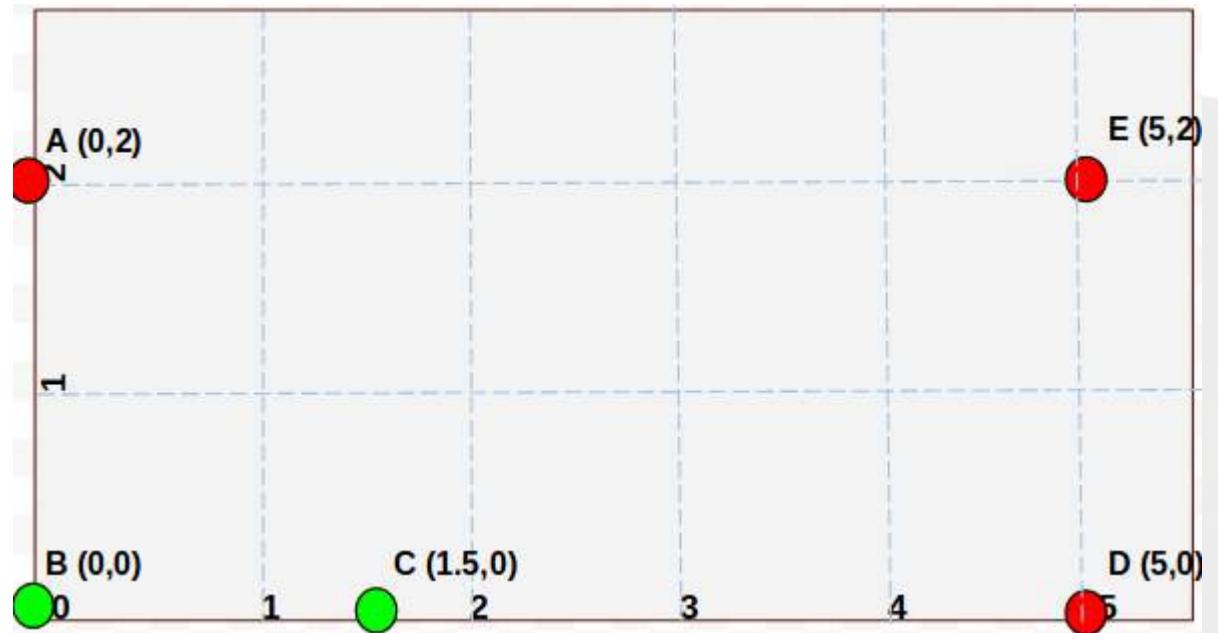
Complete Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach



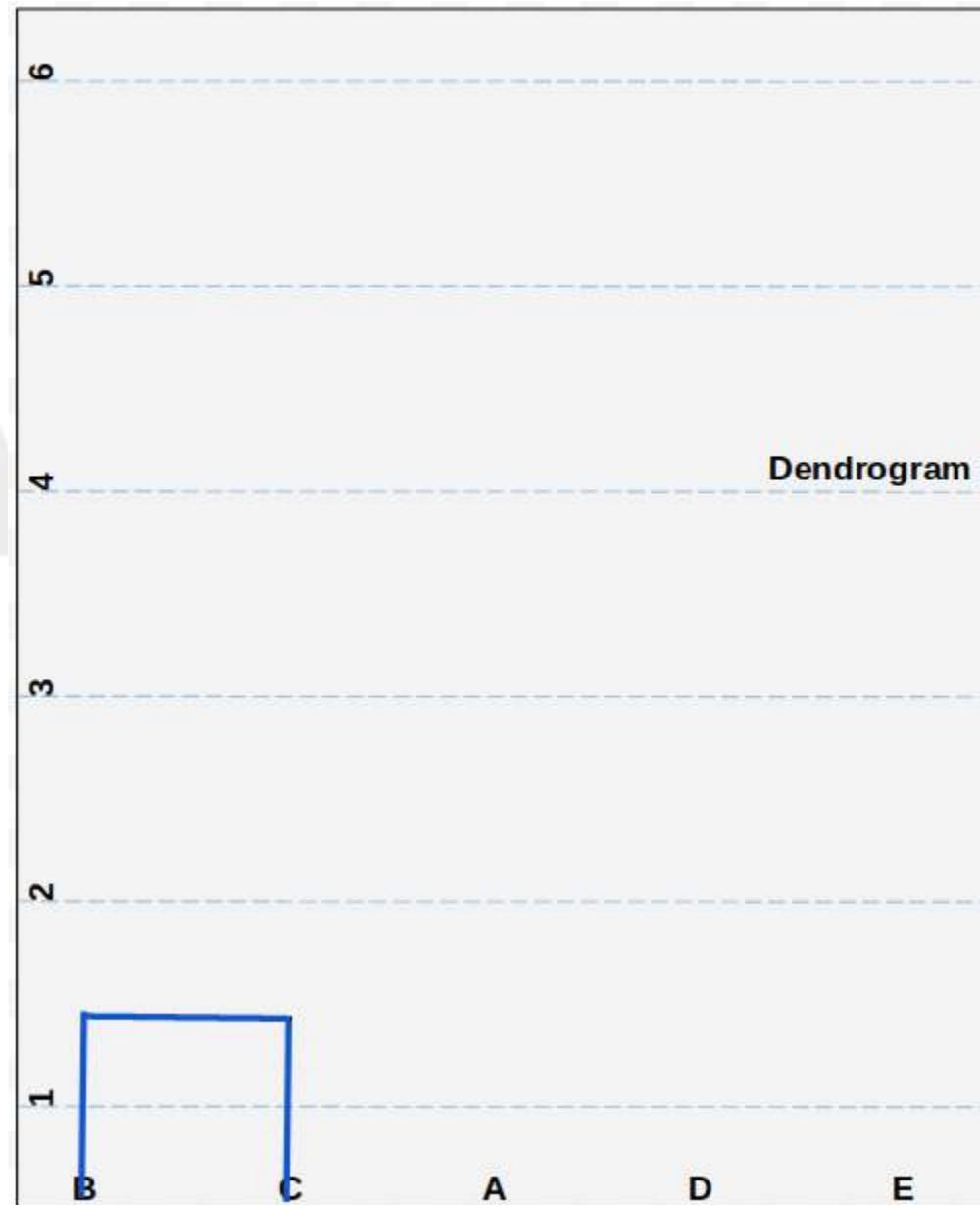
$$dist_{max}(C_i, C_j) = \max_{\mathbf{p} \in C_i, \mathbf{p}' \in C_j} \{ |\mathbf{p} - \mathbf{p}'| \}$$

Complete Linkage Clustering

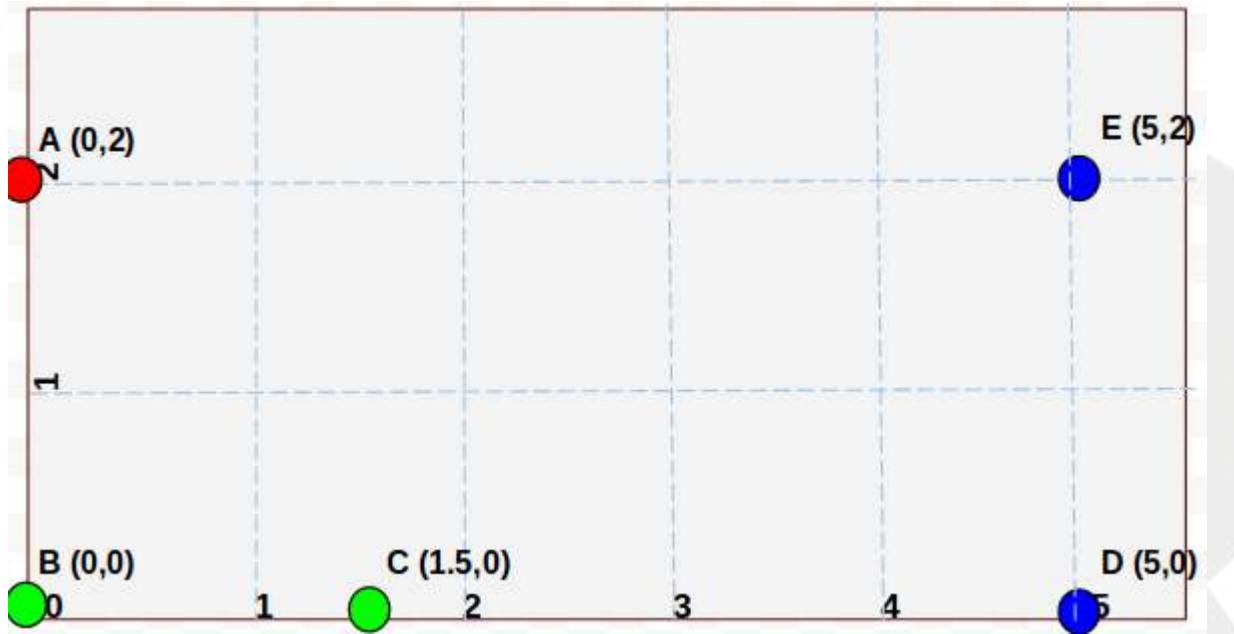


$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'| \}$$

Distance between B & C : 1.5 (Merged)

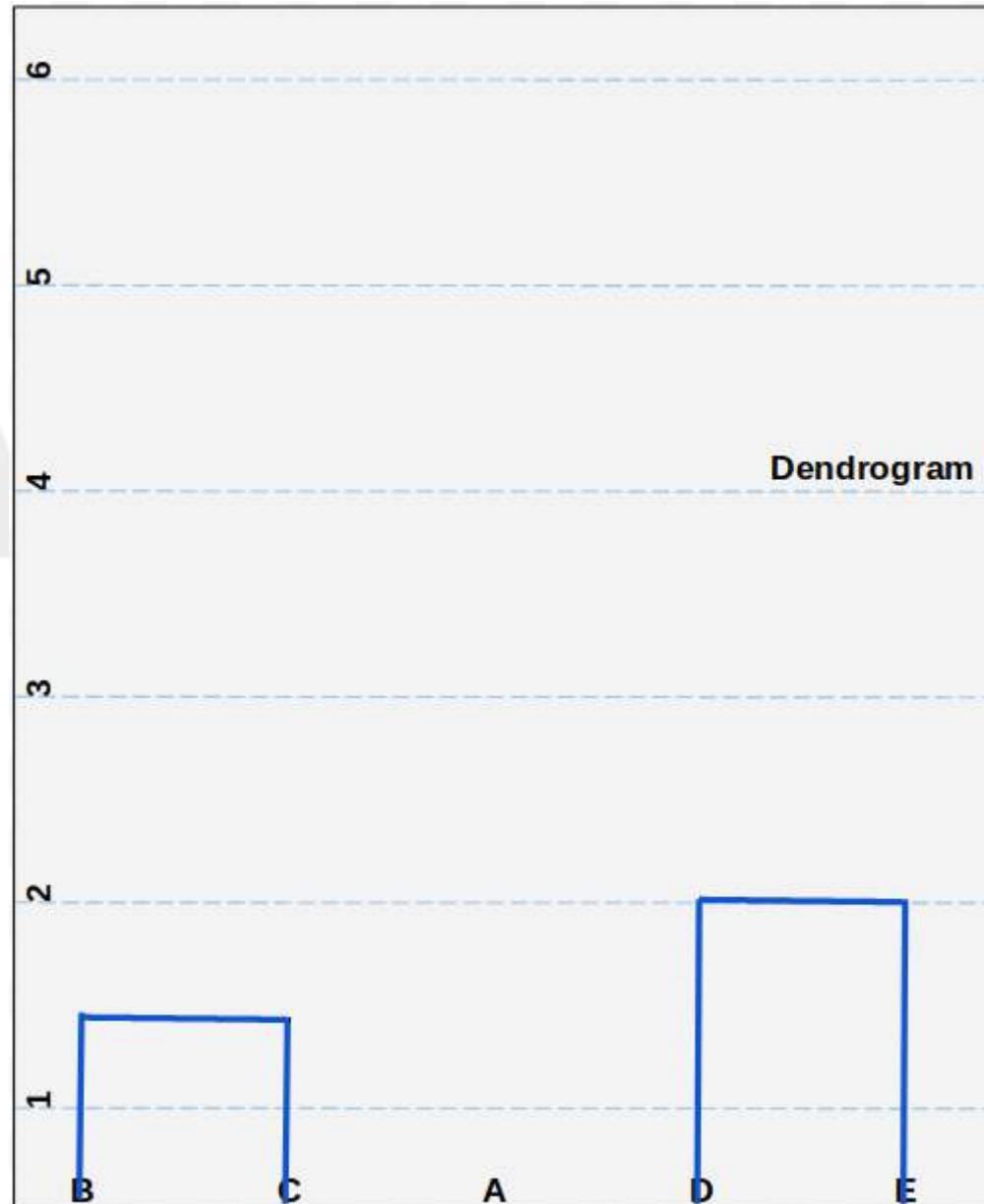


Complete Linkage Clustering

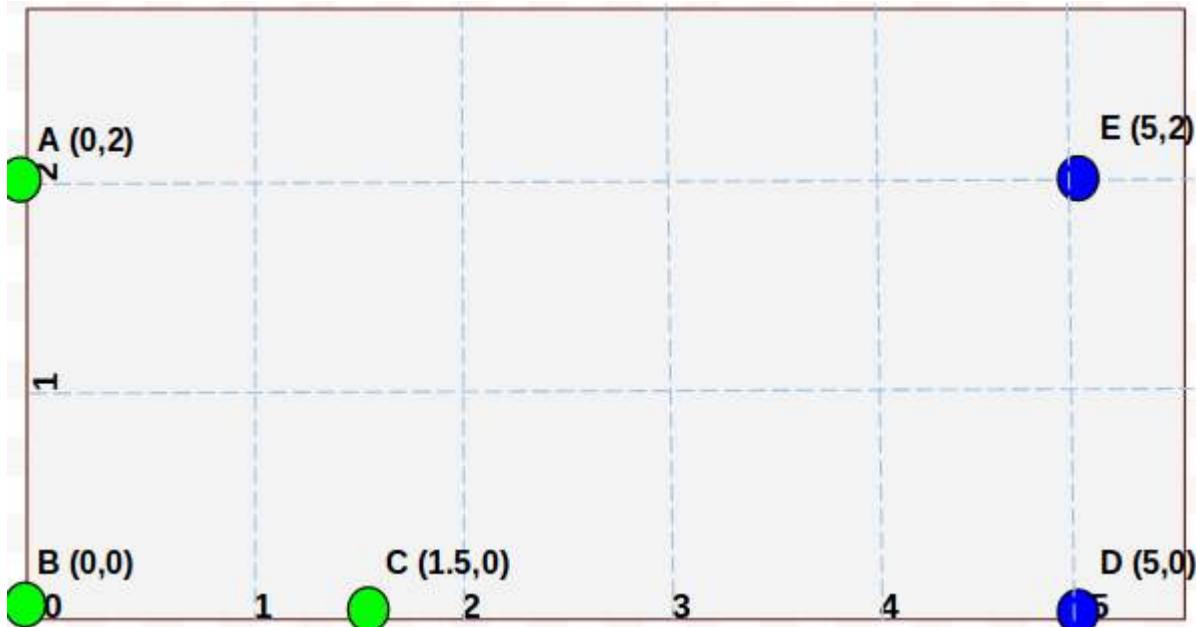


$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\}$$

Distance between D & E : 2.0 (Merged)

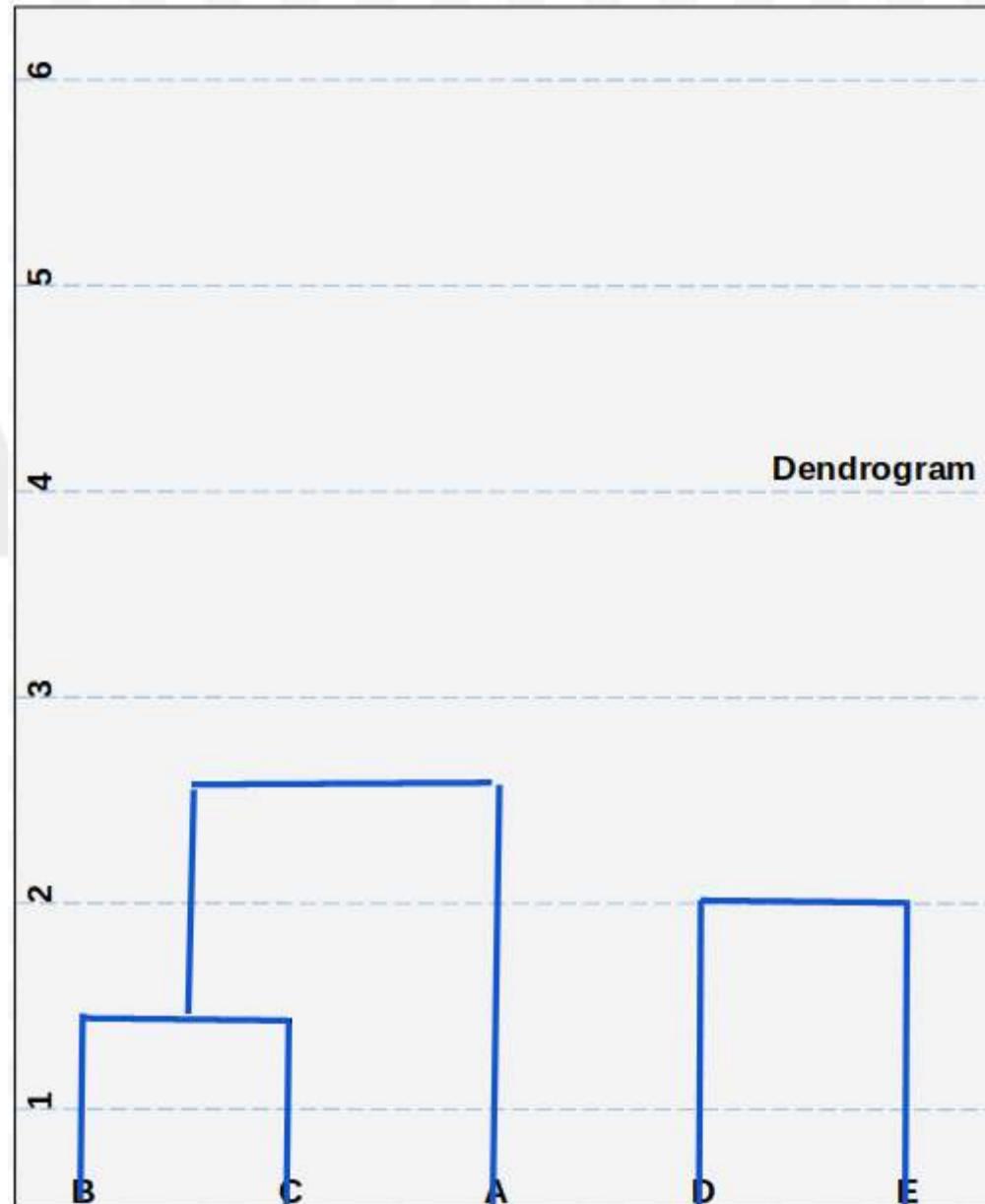


Complete Linkage Clustering



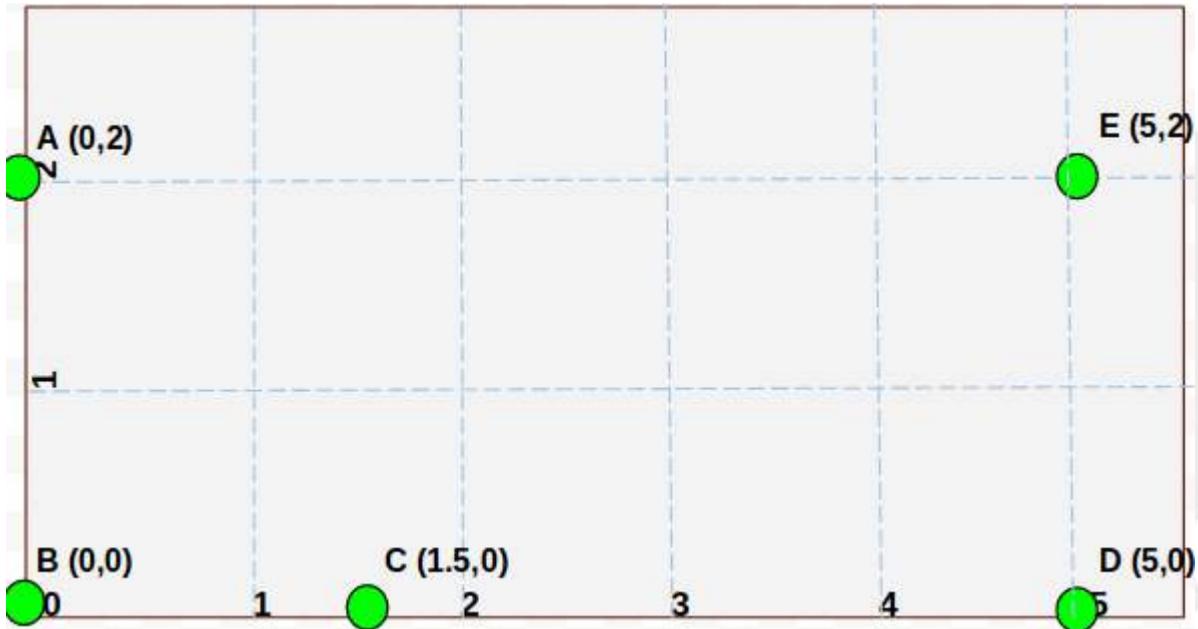
$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{ |p - p'| \}$$

Distance between {B,C} & A : 2.5 (Merged)



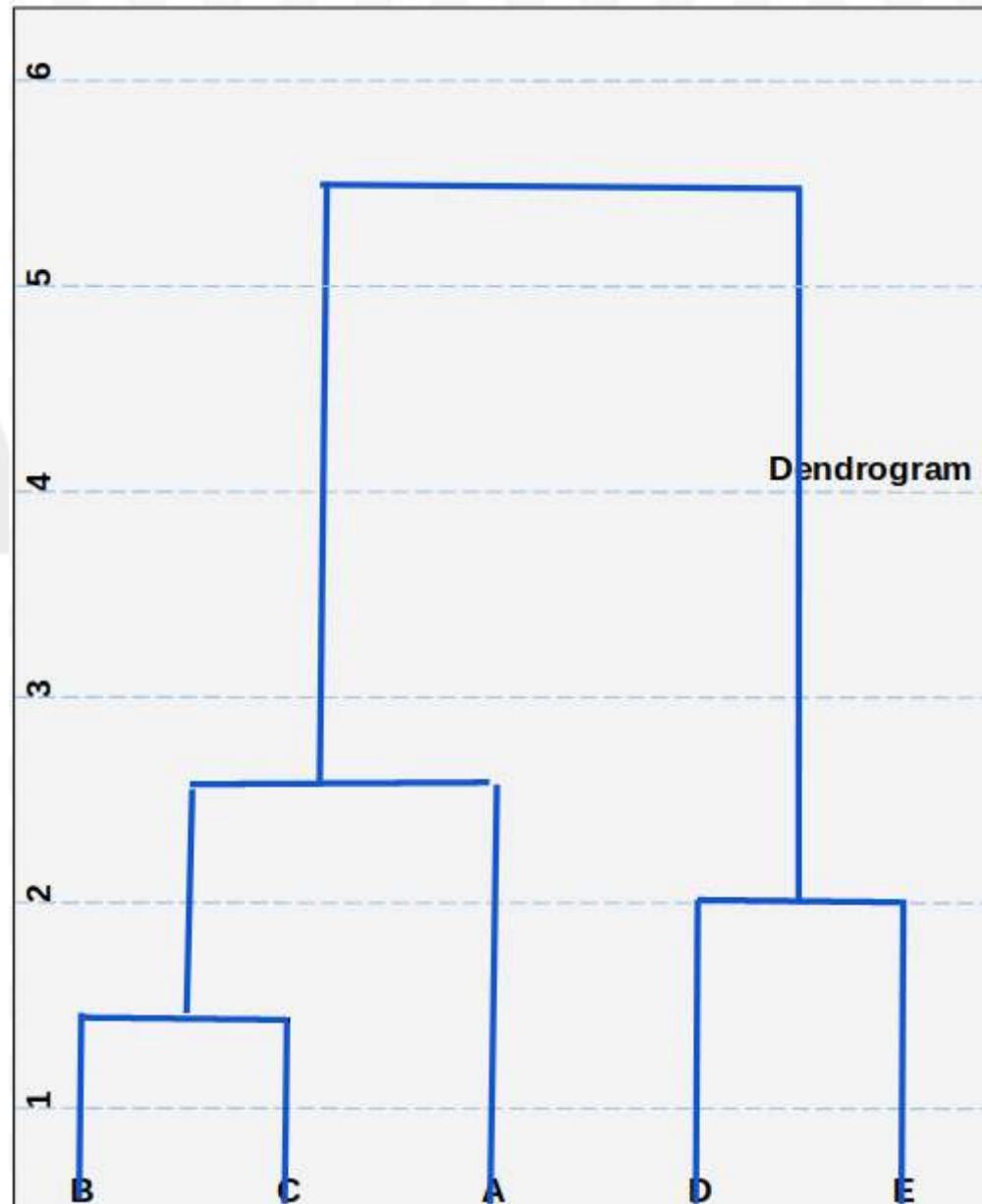
Complete Linkage Clustering

- An Agglomerative Hierarchical Clustering Approach

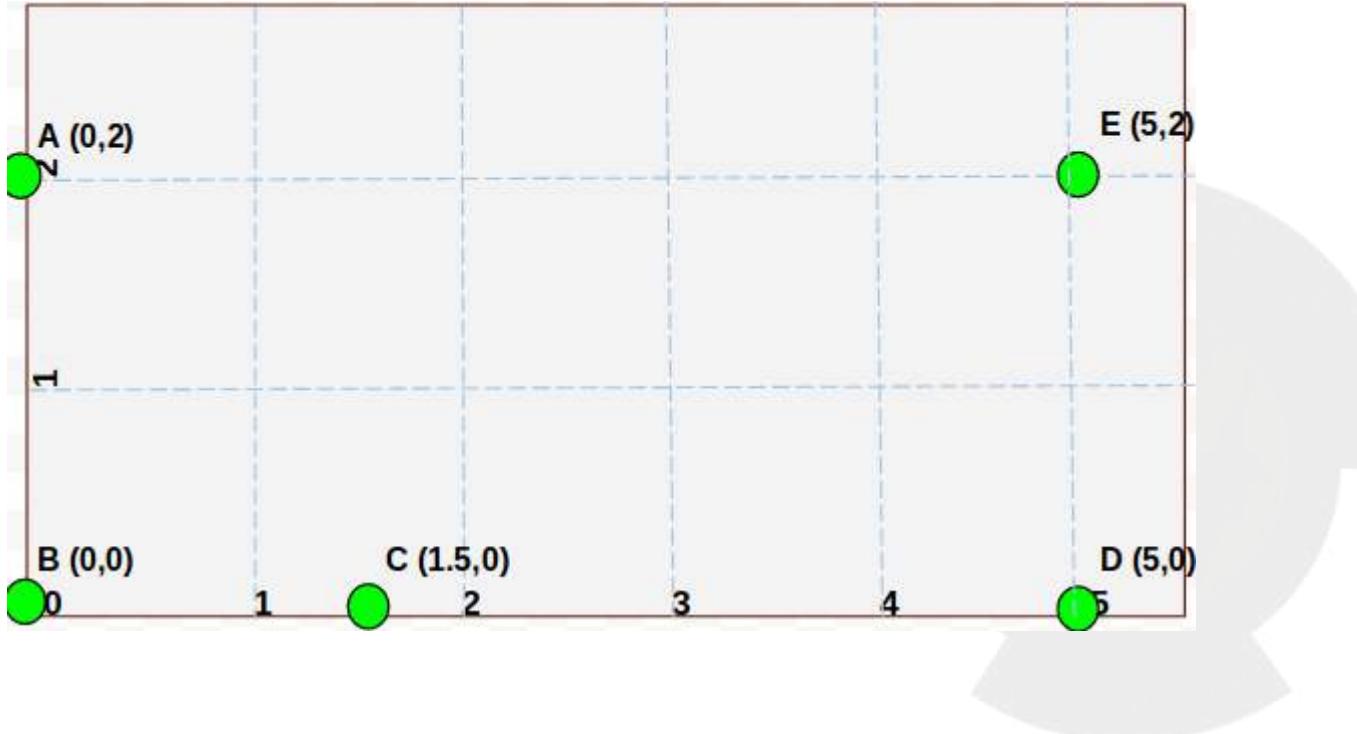


$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\}$$

Distance between {A,B,C} and {D,E} : 5.4 (Merged)

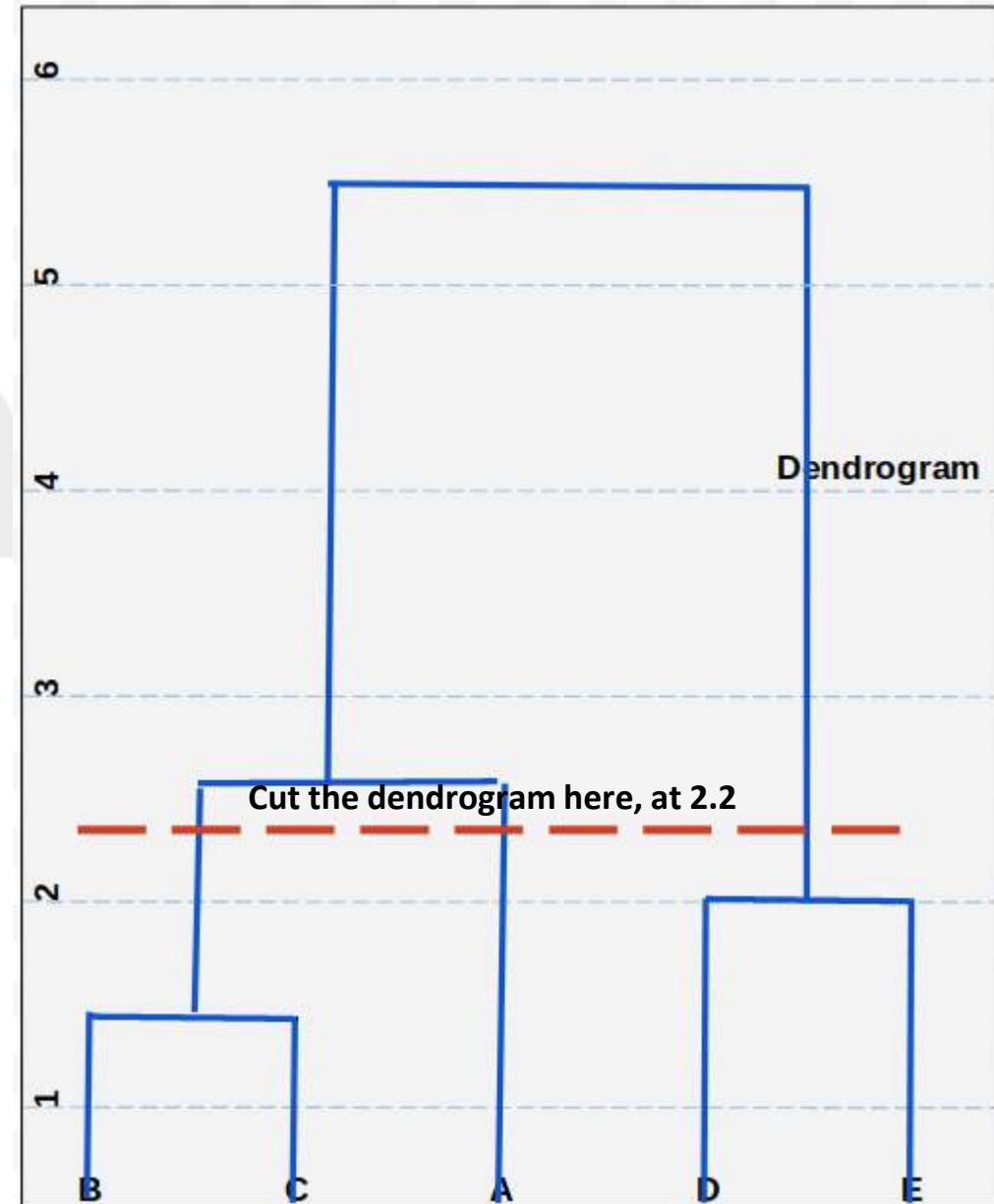


Complete Linkage Clustering

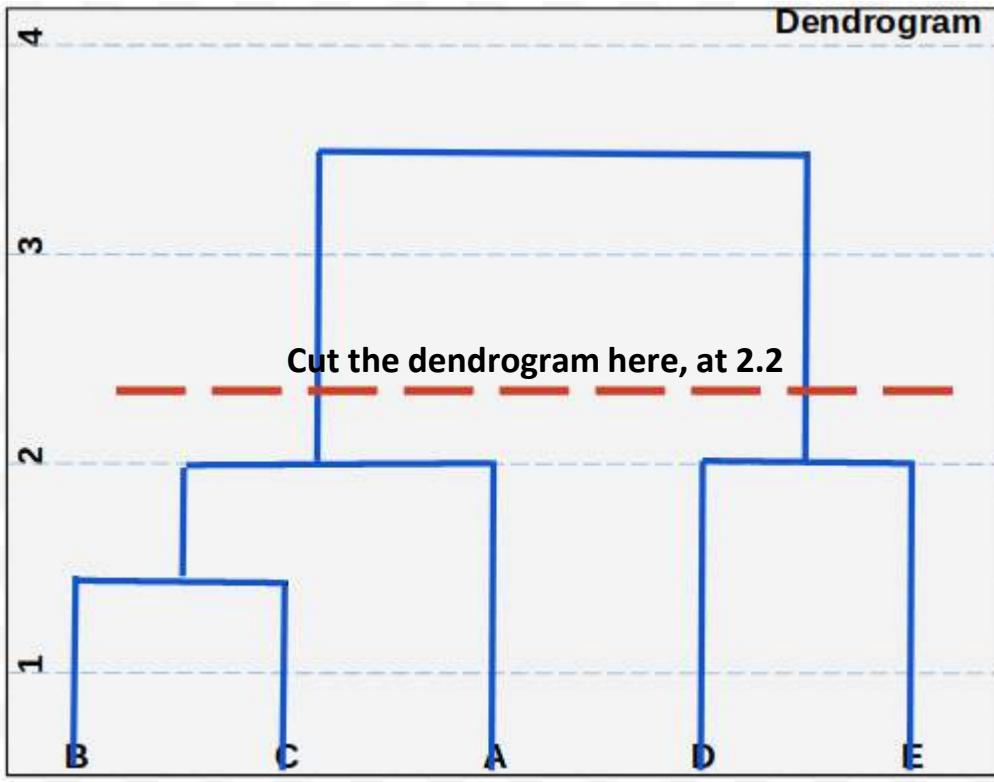


How to terminate the process?

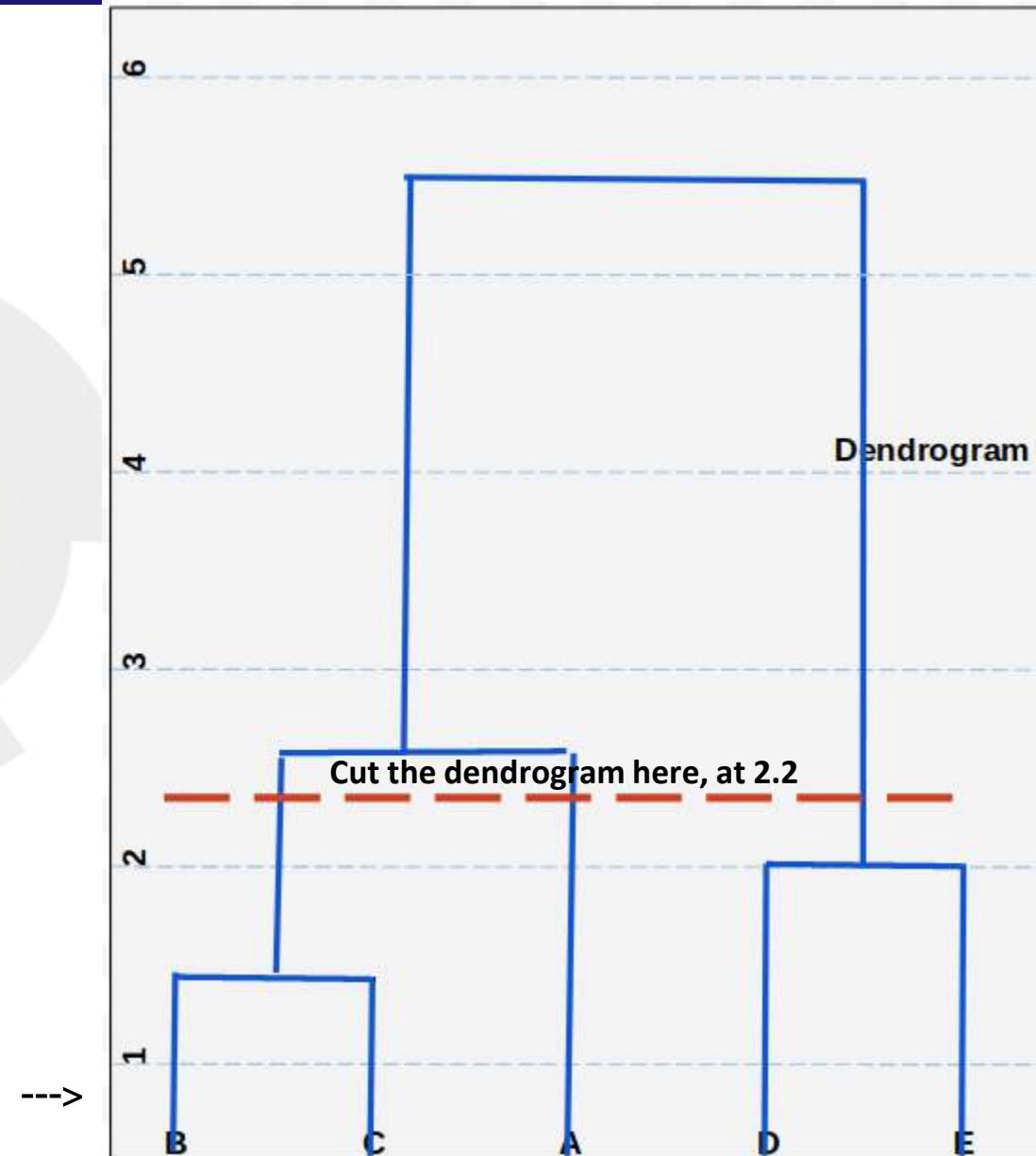
- (1) Use a predetermined K
- (2) Use a threshold on the closest distance between clusters
- (3) Use threshold on measures of compactness of individual clusters



Complete Linkage Clustering



Complete Linkage Clustering



Comments on the algorithm

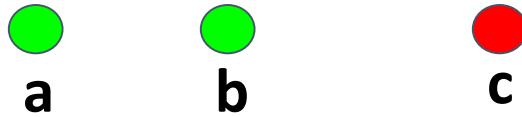
- Merge Criterion is non-local
 - We decide to merge clusters based on the distance between farthest points in clusters
- Complete Link is not merge-persistent



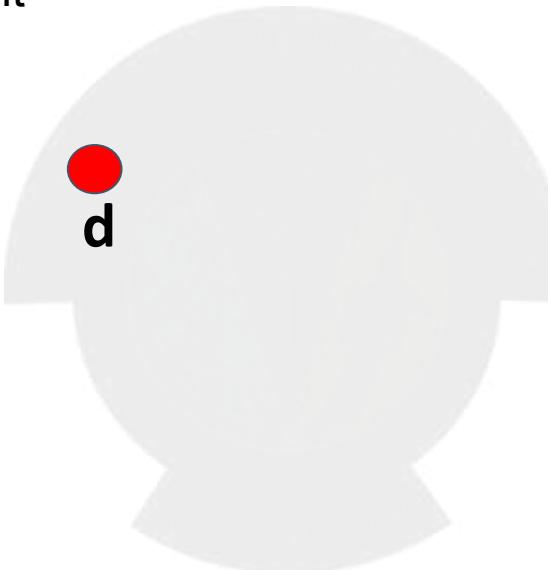
It appears from the distance matrix, c is closer to b than d

Comments on the algorithm

- Merge Criterion is non-local
 - We decide to merge clusters based on the distance between farthest points in clusters
- Complete Link is not merge-persistent



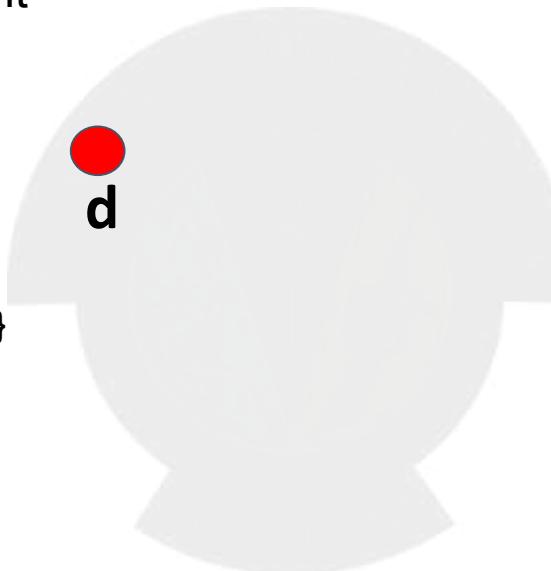
Now merge a and b



Comments on the algorithm

- Merge Criterion is non-local
 - We decide to merge clusters based on the distance between farthest points in clusters
- Complete Link is not merge-persistent

a b c



Now, c merges with d, but not with the cluster {a,b}

Comments on the algorithm

- Merge Criterion is non-local
 - We decide to merge clusters based on the distance between farthest points in clusters
- Complete Link is not merge-persistent
- Tends to find compact clustering with approximately same width
- Does not suffer from chaining

Time Complexity:

- Naive Implementation of the algorithm takes $O(n^3)$ time
 - Pairwise distance between points $O(n^2)$ time
 - Number of iterations $O(n)$
- Use of priority queue helps bringing the complexity to $O(n^2 \log n)$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mean and Average Linkage Clustering

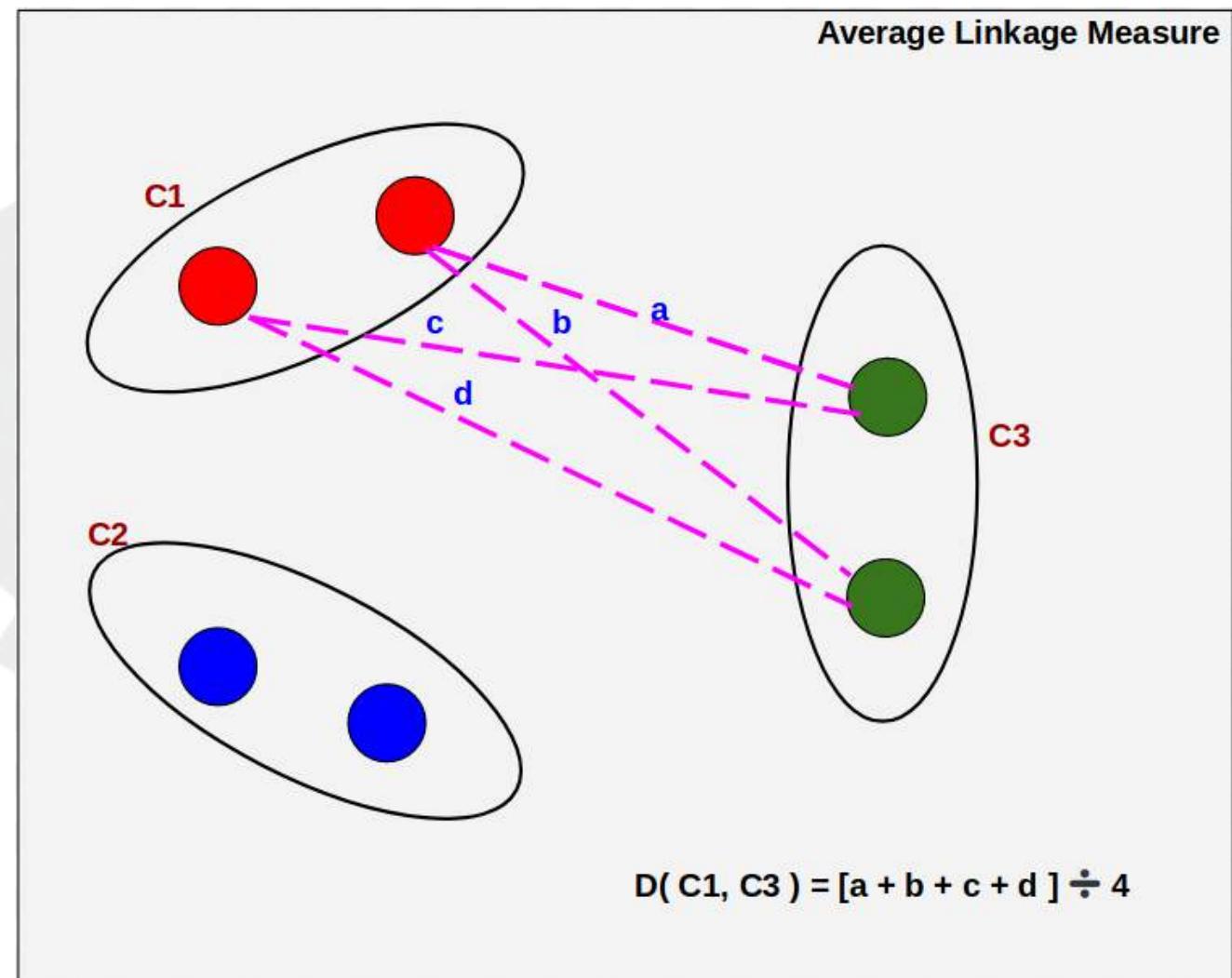
S.P.Vimal

Agenda

- Mean and Average Linkage Algorithms

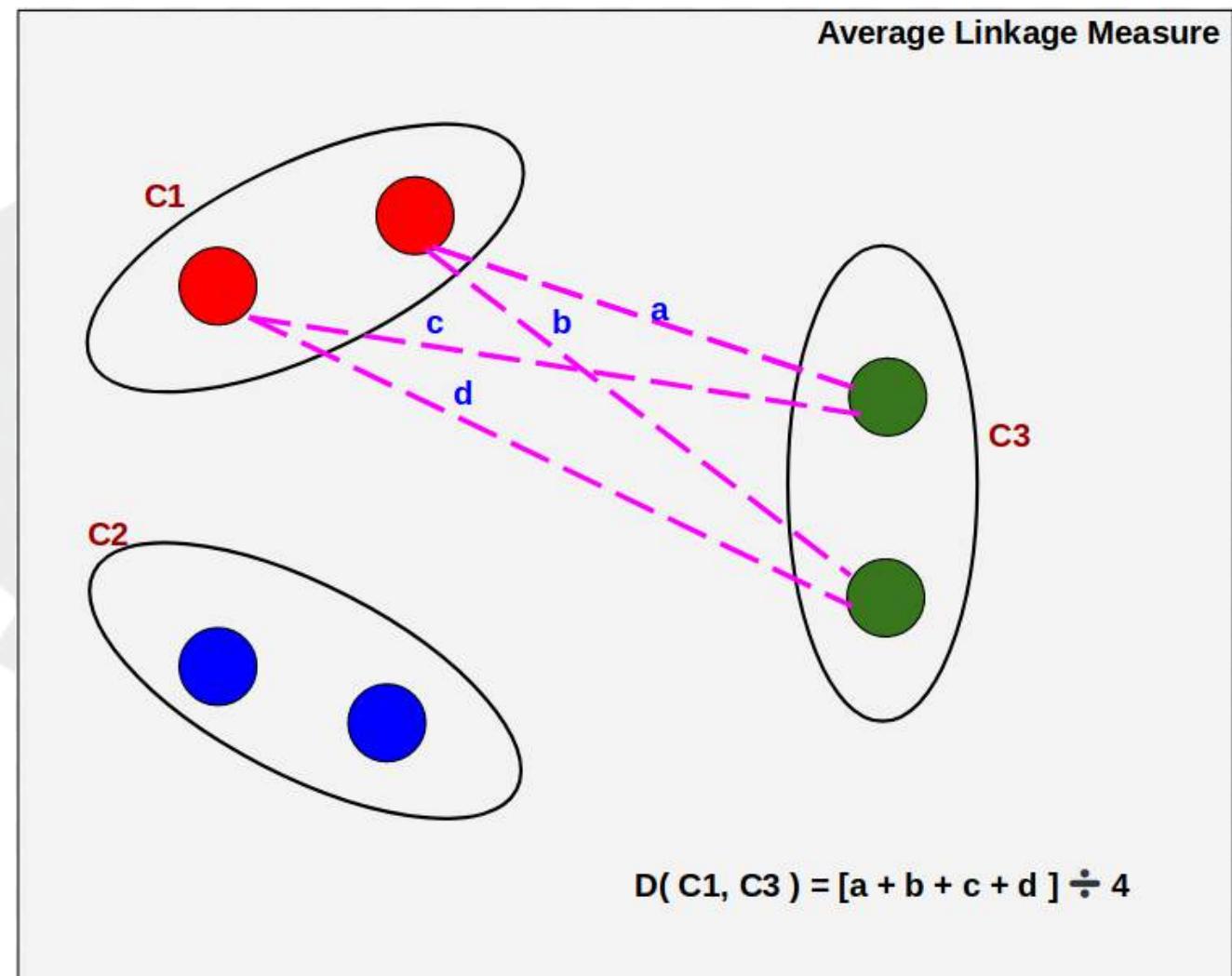
Average Linkage Clustering

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'|$$



Average Linkage Clustering

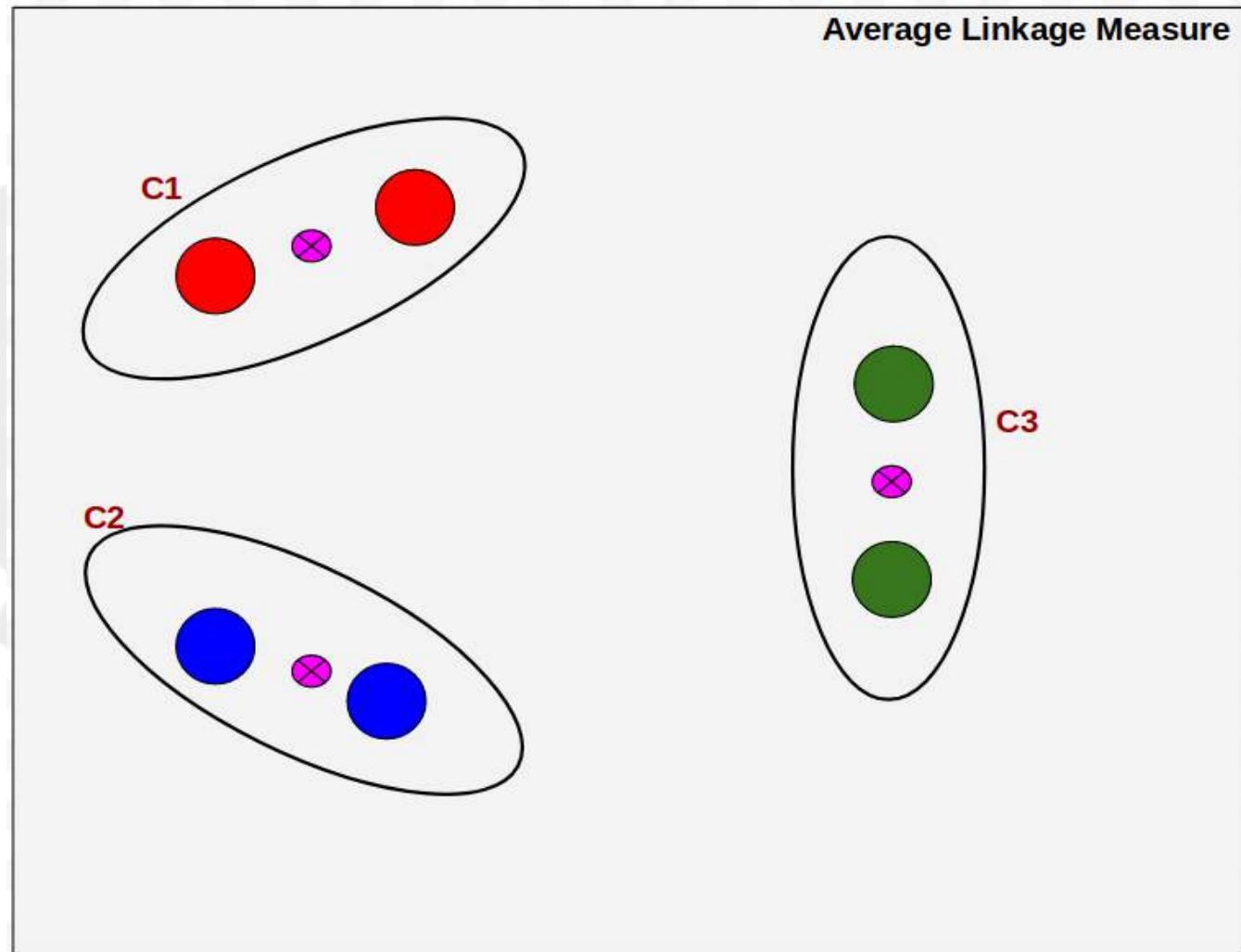
- Avoids issues with Single & Complete Linkage clustering
- Less sensitive to noise & outliers
- Better merge persistent
- Non elliptical shape
 - Not quite well understood on the nature of resultant clusters on an arbitrary cut in the dendrogram



Average Linkage Clustering

$$dist_{mean}(C_i, C_j) = |\mathbf{m}_i - \mathbf{m}_j|$$

Simpler computation of D(.) is the key benefit !!!





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Density Based Clustering

Raja Vadhana

Asst. Professor
WILP Division, BITS-Pilani



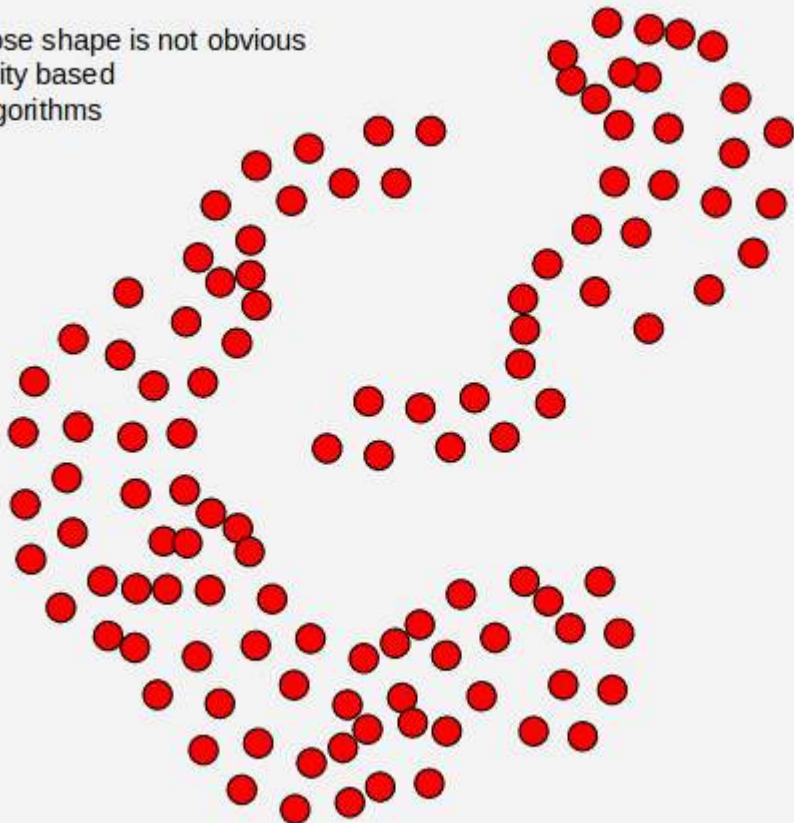
Agenda

- What is Density Based Clustering & Why?

Density Based Clustering Algorithms

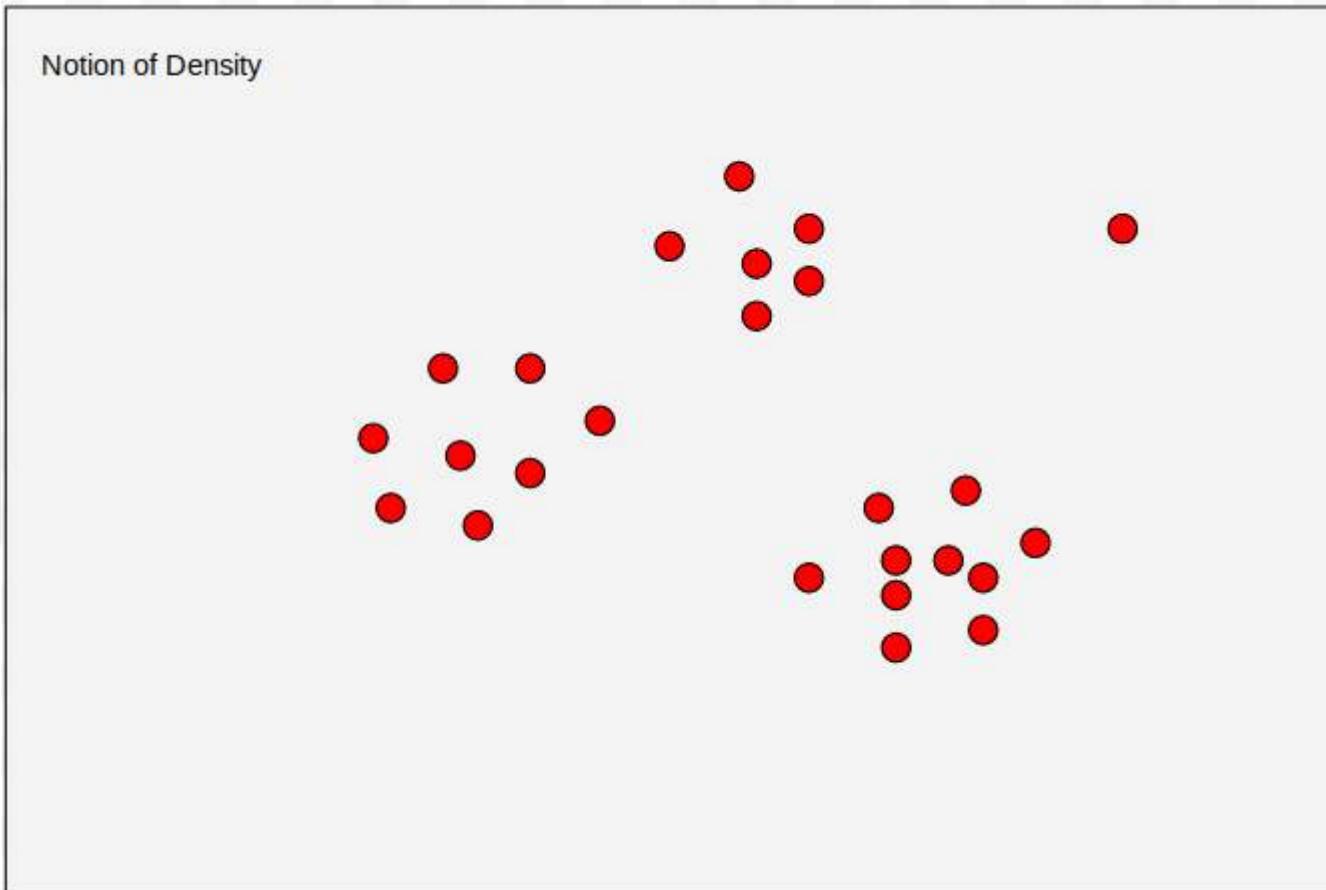
- Partition based & Hierarchical algorithms are biased towards finding spherical shaped clusters
- How do you describe the shape of each of the clusters you see in the picture?
 - There are just two dense regions
 - No particular shape, indeed.

Clusters whose shape is not obvious
for dissimilarity based
clustering algorithms



Density Based Clustering Algorithms

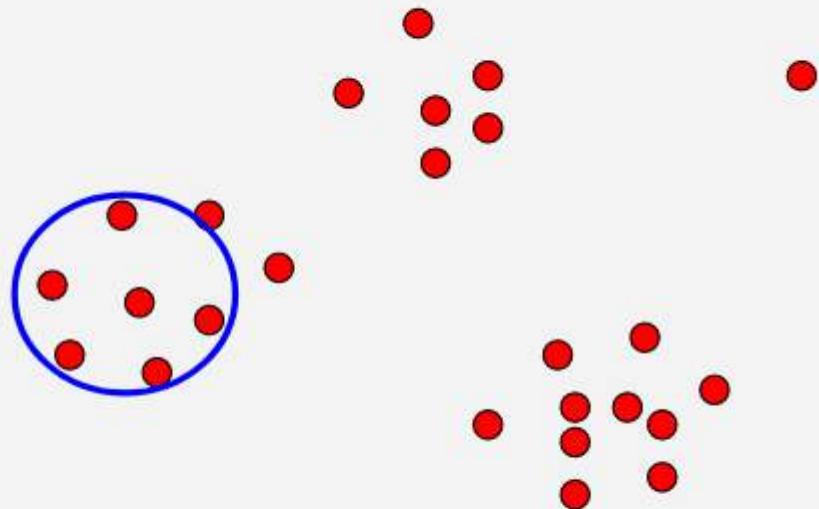
- Density (over a neighbourhood)
 - Neighbourhood defined by ϵ
 - Region with radius ϵ is dense if the region has at least MinPts points
- ϵ & MinPoints together defines Density



Density Based Clustering Algorithms

- Views Clusters as regions of high density objects - separated by regions of low-density regions
 - A cluster is a maximal set of density connected points

Notion of Density

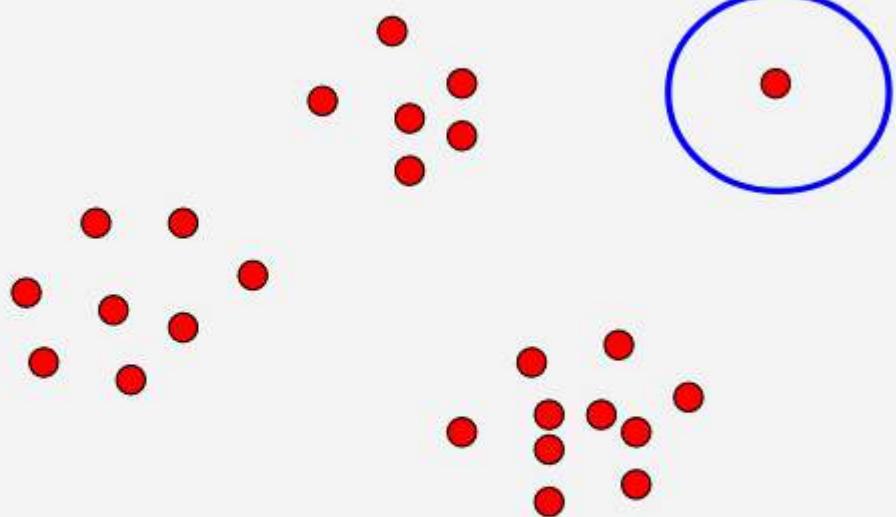


Let the radius of the circle is $\epsilon > 0$ & MinPts
number of points inside the number of in the
 ϵ -neighbourhood

Density Based Clustering Algorithms

- Not sensitive to noise/ outliers
 - These points does not typically form a dense cluster

Notion of Density

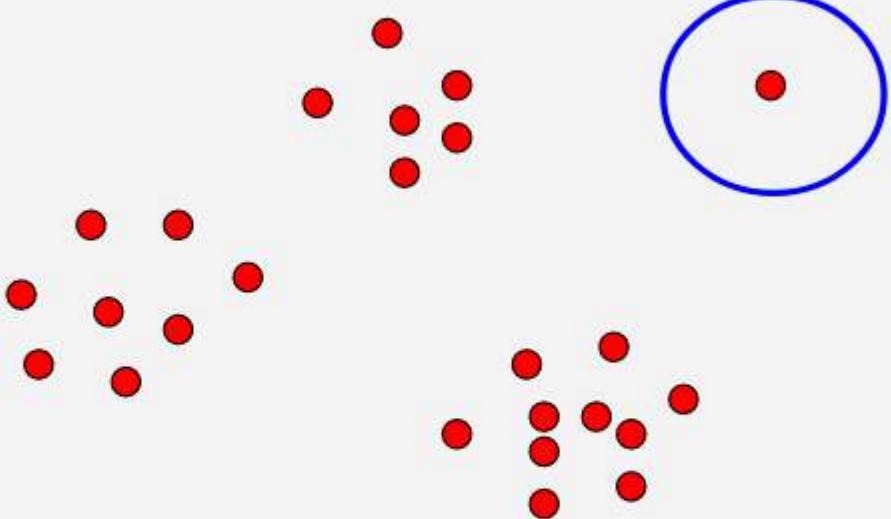


Let the radius of the circle is $\epsilon > 0$ & MinPts
number of points inside the number of in the
 ϵ -neighbourhood

Density Based Clustering Algorithms

- Let us learn a specific Density based algorithm (DSSCAN) to understand how density based algorithm work.

Notion of Density



Let the radius of the circle is $\epsilon > 0$ & MinPts
number of points inside the number of in the
 ϵ -neighbourhood



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

DBSCAN - Understanding Terminologies (1 / 2)

S.P.Vimal

Agenda

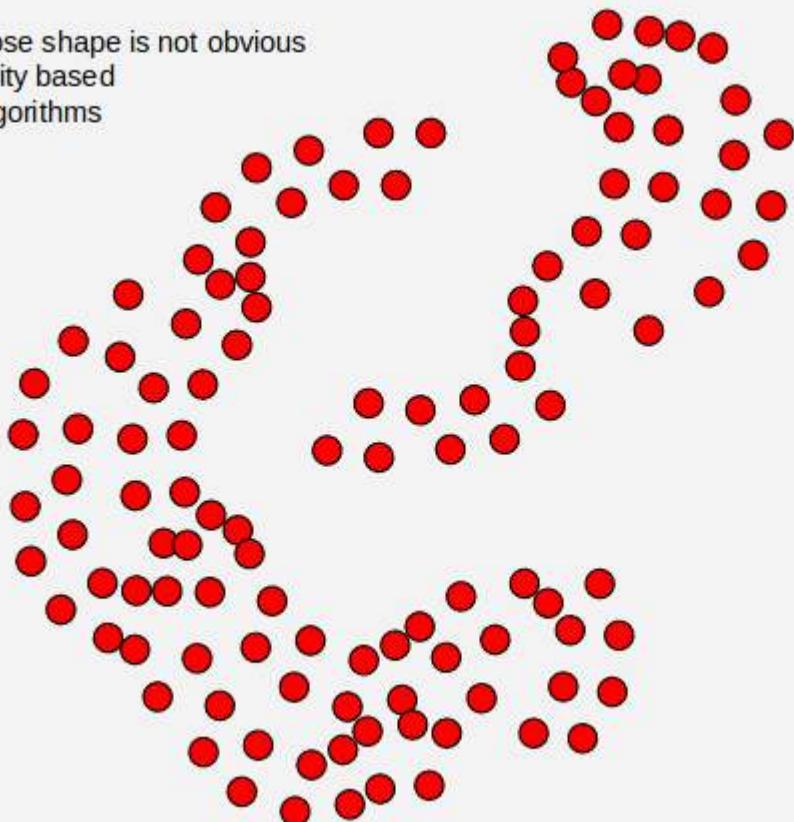
- DBSCAN - Notion of Density
- Eps-Neighbourhood ()
- Core points and Border points



DBSCAN

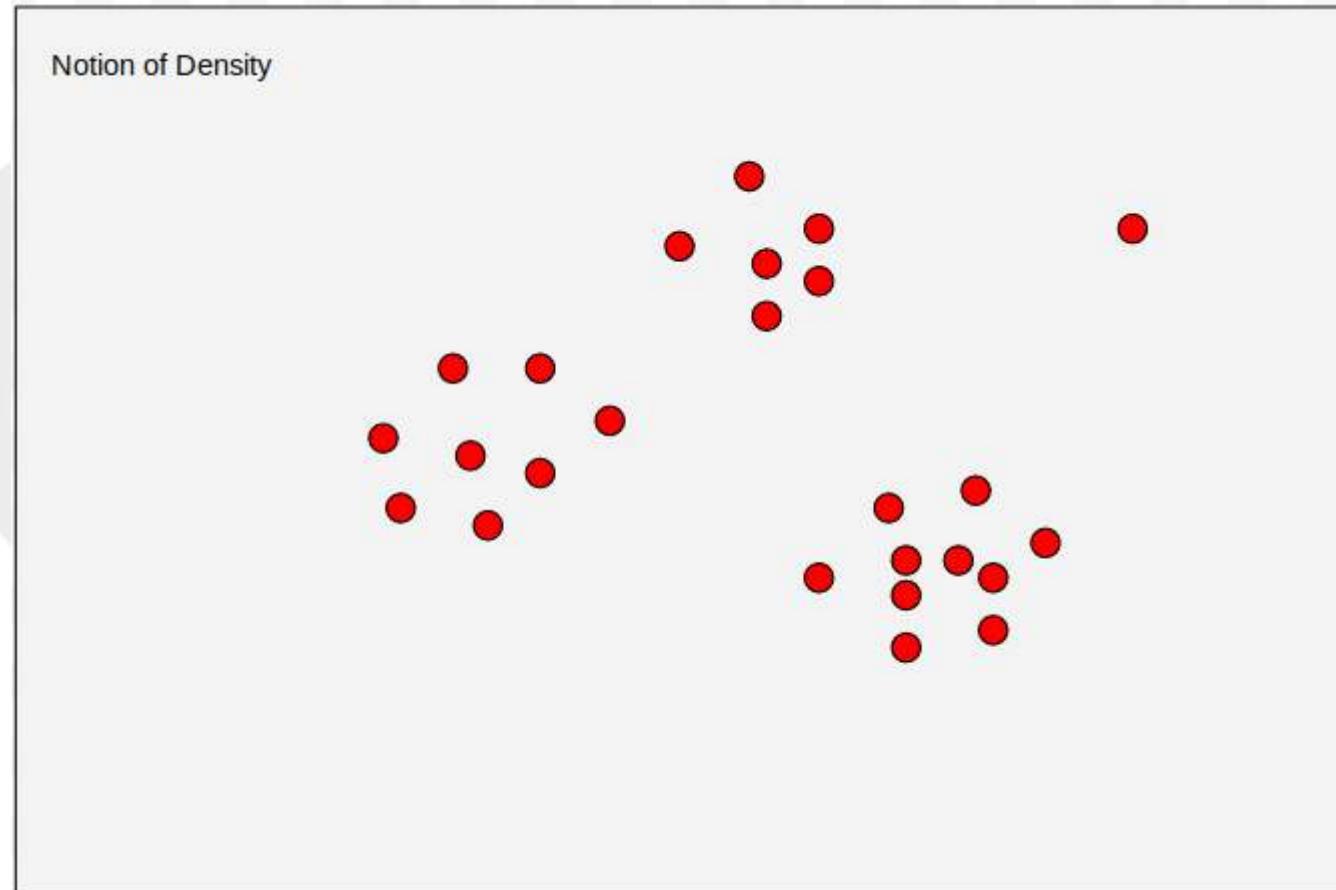
- DBSCAN - Density-Based Spatial Clustering of Applications with Noise [KDD'96 - Martin Ester et. al]

Clusters whose shape is not obvious
for dissimilarity based
clustering algorithms



How DBSCAN defines density?

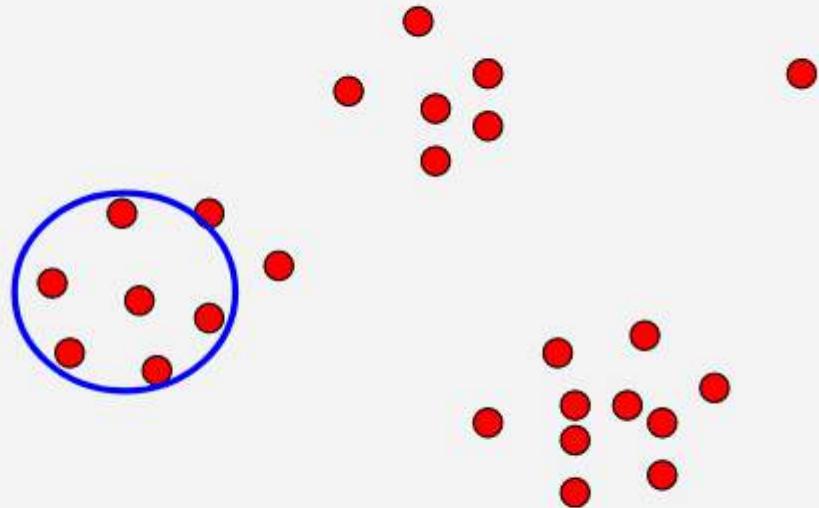
- Density (over a neighbourhood)
 - Neighbourhood defined by ϵ
 - Region with radius ϵ is dense if the region has at least MinPts points
- ϵ & MinPoints together defines Density



How DBSCAN defines density?

- Density (over a neighbourhood)
 - Neighbourhood defined by ϵ
 - Region with radius ϵ is dense if the region has at least MinPts points
- ϵ (written as eps mostly) & MinPoints together defines Density

Notion of Density



Let the radius of the circle is $\epsilon > 0$ & MinPts number of points inside the number of in the ϵ -neighbourhood

Terms used in DBSCAN

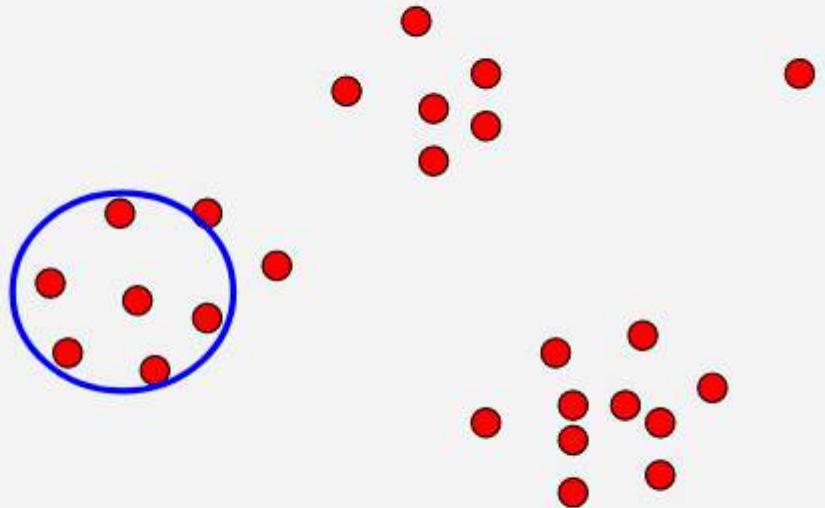
Definition

Eps-Neighbourhood of a point:

The Eps-neighborhood of a point p, denoted by $N_{Eps}(P)$, is defined

$$N_{Eps}(P) = \{q \in D \mid \text{dist}(p, q) \leq Eps\}$$

Notion of Density



Let the radius of the circle is $\epsilon > 0$ & MinPts
number of points inside the number of in the
 ϵ -neighbourhood

Terms used in DBSCAN

Definition

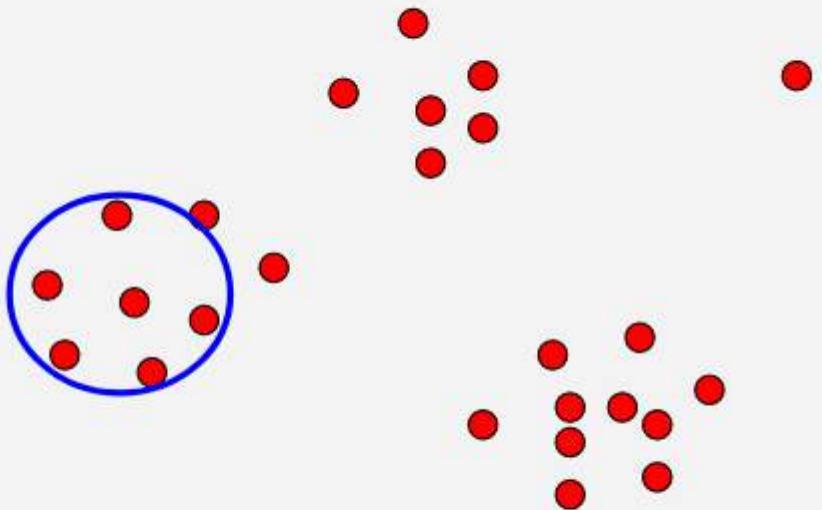
Eps-Neighbourhood of a point:

The Eps-neighborhood of a point p, denoted by $N_{Eps}(P)$, is defined

$$N_{Eps}(P) = \{q \in D \mid \text{dist}(p, q) \leq Eps\}$$

- Do we require all points in a cluster to have MinPts number of points?
 - Points in the border need not have this !

Notion of Density



Let the radius of the circle is $\epsilon > 0$ & MinPts number of points inside the number of in the ϵ -neighbourhood

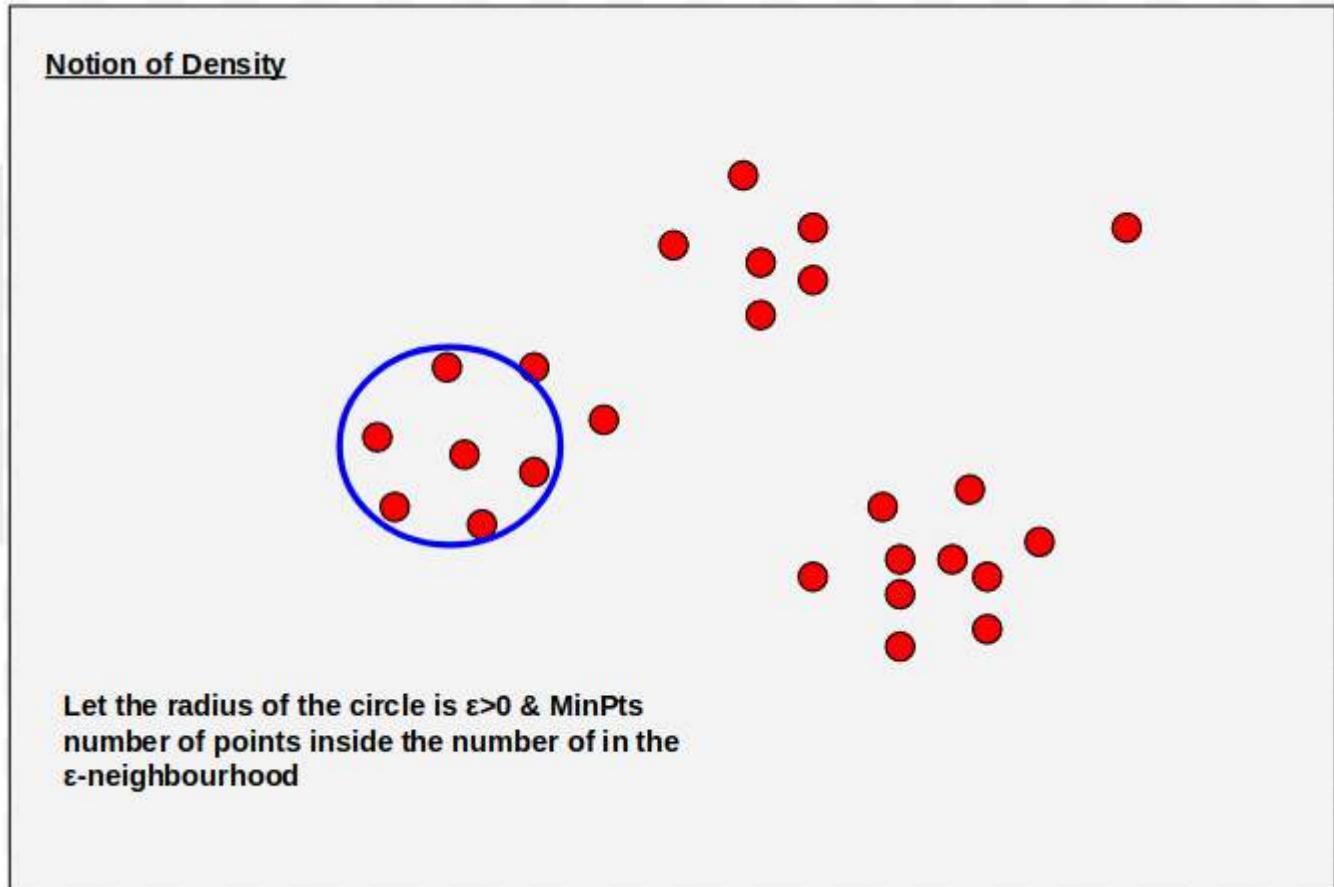
Terms used in DBSCAN

Two kinds of points in the cluster:

(1) Core Points

- Points inside the cluster
- These points have in MinPts in their Eps-neighborhood

(2) Border Points



Terms used in DBSCAN

Two kinds of points in the cluster:

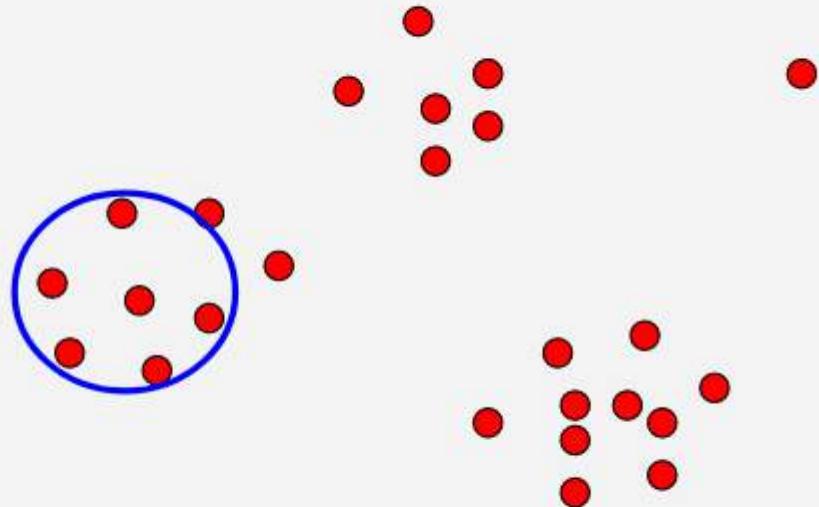
(1) Core Points

- Points inside the cluster
- These points have in MinPts in their Eps-neighborhood

(2) Border Points

- Points which does not have MinPts in their Eps-neighborhood
- If border points are a part of a cluster, then it is in the Eps-neighborhood of some core point

Notion of Density



Let the radius of the circle is $\epsilon > 0$ & MinPts
number of points inside the number of in the
 ϵ -neighbourhood



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

DBSCAN - Understanding Terminologies (2 / 2)

S.P.Vimal

Agenda

- Terms
 - Directly density reachable
 - Density reachable
 - Density connected

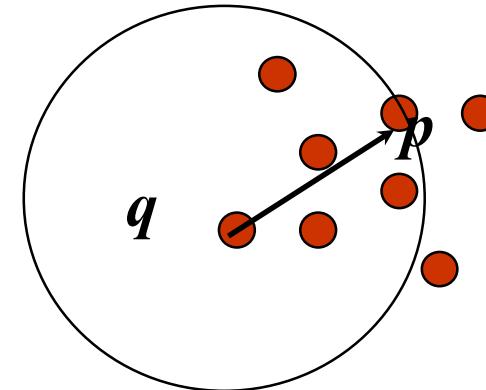
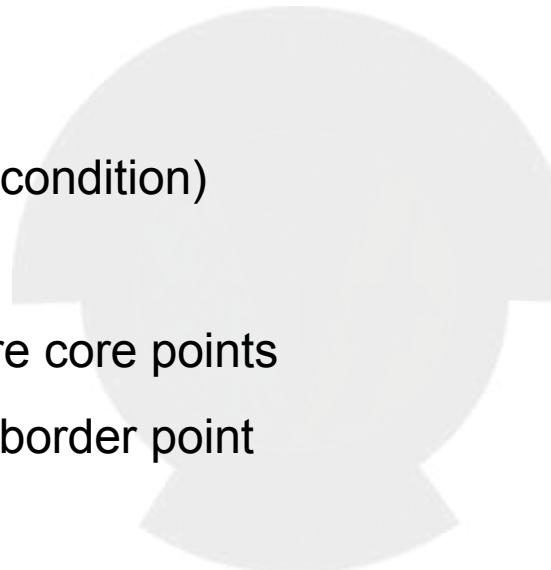
DBSCAN Terminologies

Directly Density Reachable

A point p is *directly density-reachable* from a point q wrt. Eps and MinPts if

- 1) $p \in N_{\text{Eps}}(q)$
- 2) $|N_{\text{Eps}}(q)| > \text{MinPts}$ (core point condition)

- A symmetric relation if p and q are core points
- Not a symmetric relation if q is a border point



p is density reachable from q

DBSCAN Terminologies

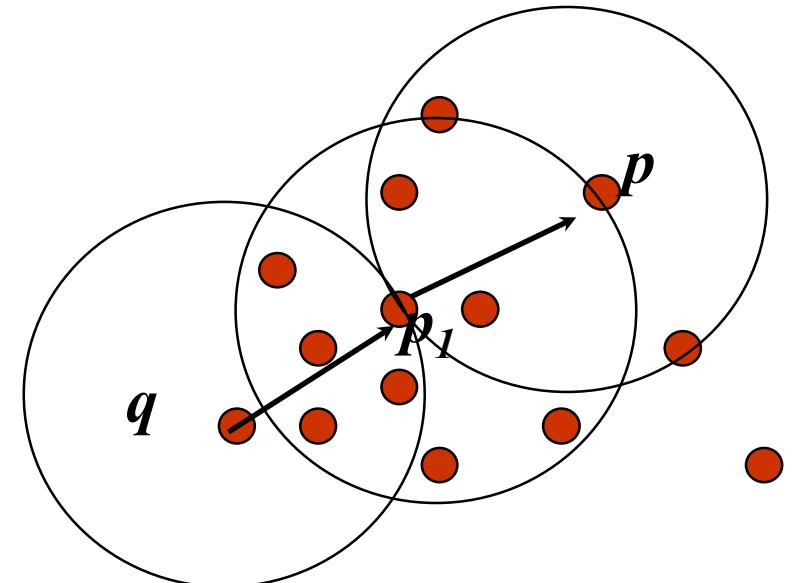
Density Reachable

A point p is **density-reachable** from a point q w.r.t. Eps , $MinPts$ if there is a chain of points

$$p_1, \dots, p_n, p_1 = q, p_n = p$$

such that p_{i+1} is directly density-reachable from p_i

- Note: The starting point q must be a core point

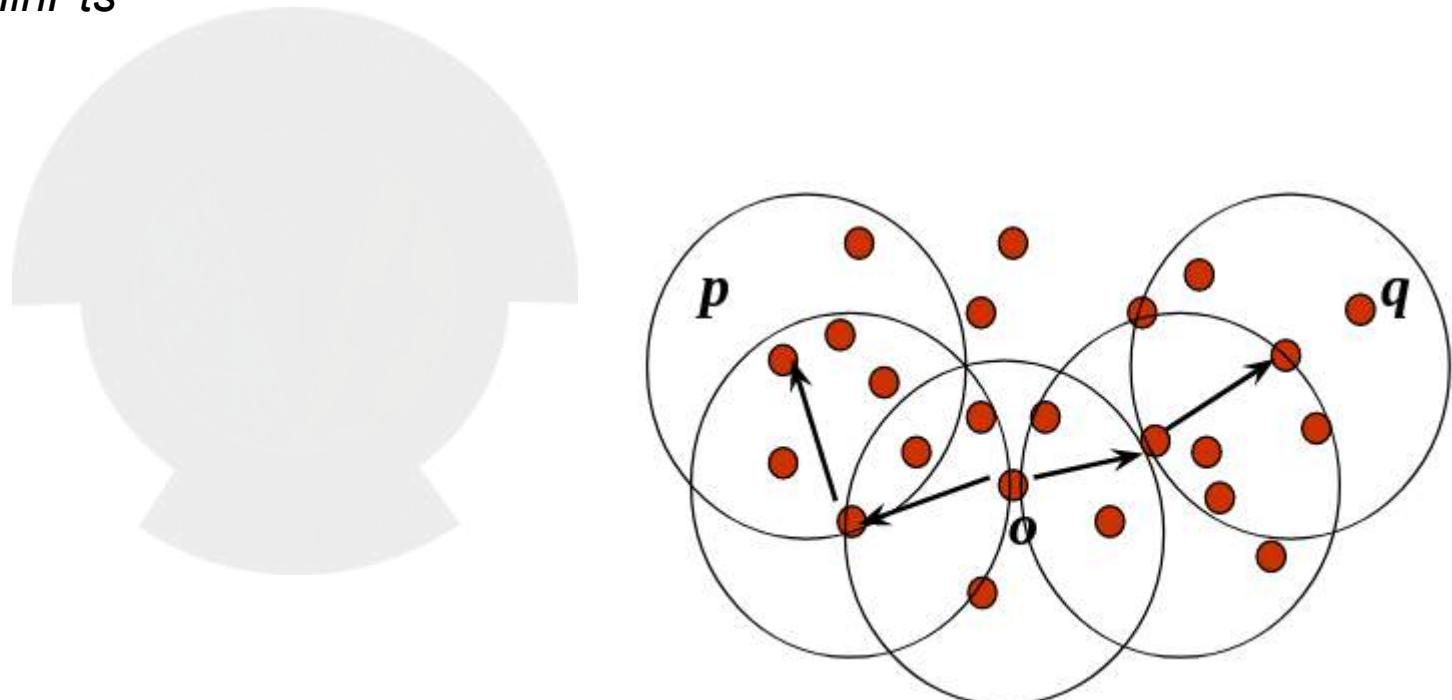


DBSCAN Terminologies

Density Connected

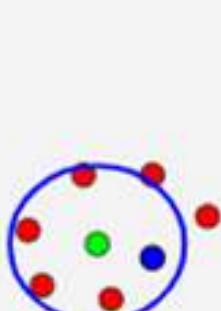
A point p is **density-connected** to a point q w.r.t. Eps , $MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$

- Symmetric relation

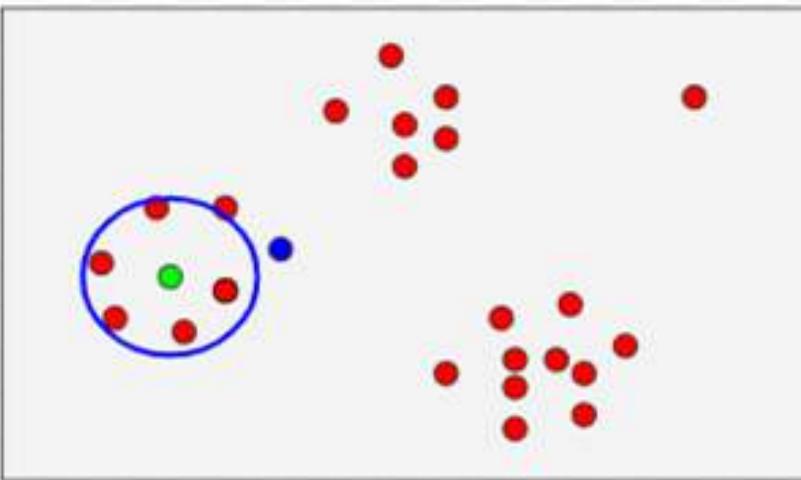


DBSCAN Terminologies

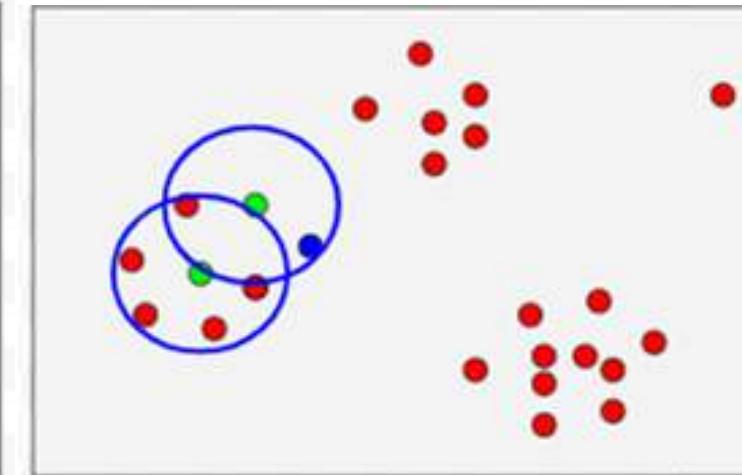
Density Connected



Directly Density Reachable

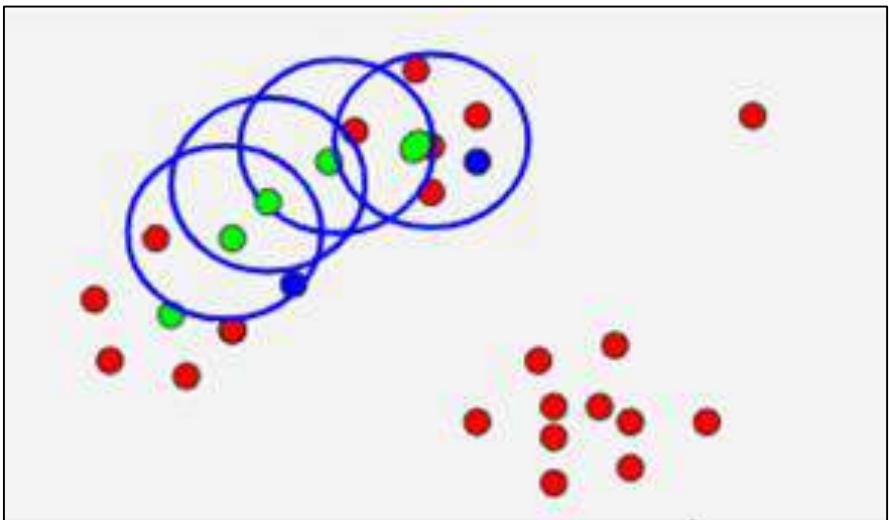


Density Reachable



DBSCAN Terminologies

Density Connected



Directly Connected



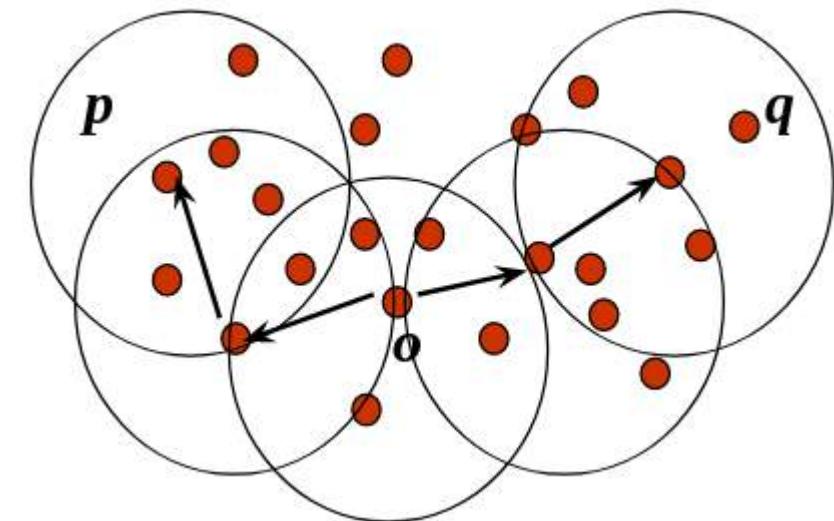
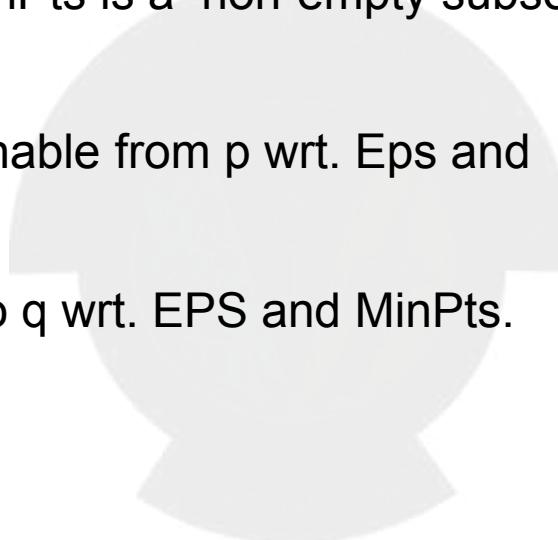
DBSCAN Terminologies

Cluster - Definition

Let D be a database of points.

A cluster C with respect to Eps and MinPts is a non-empty subset of D satisfying the following conditions:

- 1) $\forall p, q: \text{if } p \in C \text{ and } q \text{ is density-reachable from } p \text{ wrt. Eps and MinPts, then } q \in C.$ (Maximality)
- 2) $\forall p, q \in C: p \text{ is density-connected to } q \text{ wrt. EPS and MinPts.}$ (Connectivity)



Agenda

- Cluster in DBSCAN
- Example

DBSCAN

External online demo

From Naftali Harris's site <https://www.naftaliharris.com/>

Available from the link - <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>

DBSCAN Algorithm

1. Mark all objects as **unvisited**
2. Do
 - (2-1) Randomly select an unvisited object p & mark it as **visited**
 - (2-2) if the eps -neighborhood of p has at least MinPts objects
 - (2-2-1) Create a new cluster C, and add p to C
 - (2-2-2) Let N be the set of objects in the eps -neighborhood of p
 - (2-2-3) For each point p' in N
 - if p' is **unvisited**
 - Mark p' as **visited**
 - if the eps -neighborhood of p' has at least MinPts points,
 - Add those points to N ;
 - if p' is not yet a member of any cluster, add p' to C;
 - (2-2-4) End For
 - (2-2-5) Output C
 - (2-3) Else mark p as noise
 3. Until no object is unvisited

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



	X	Y
A	2	2
B	5	8
C	2	4
D	4	3
E	3	5

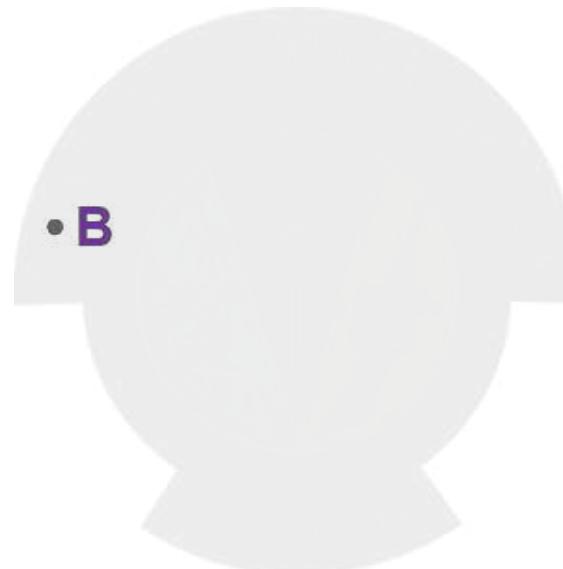
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



0 1 2 3 4 5 6

Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

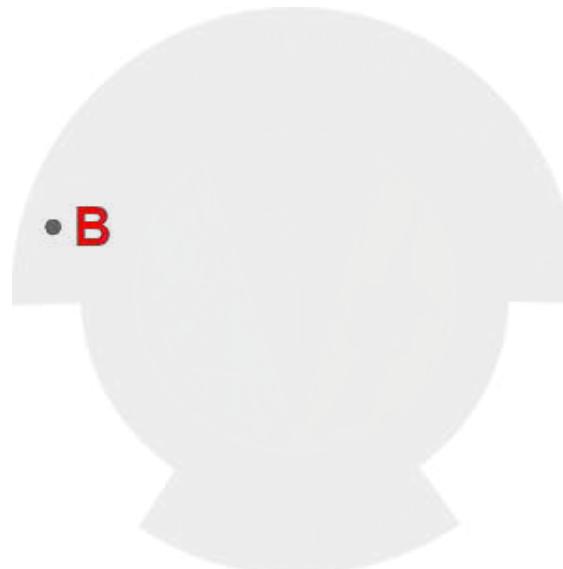
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

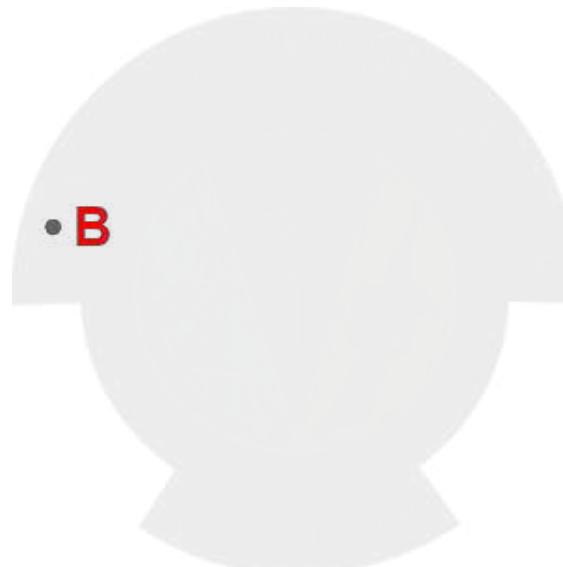
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

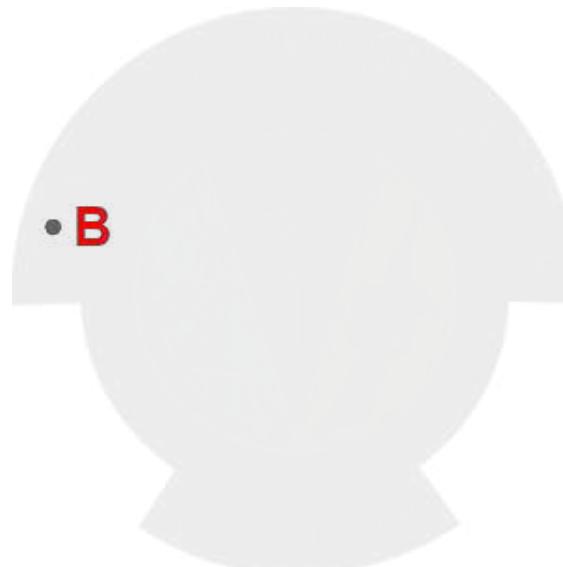
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



0 1 2 3 4 5 6

Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

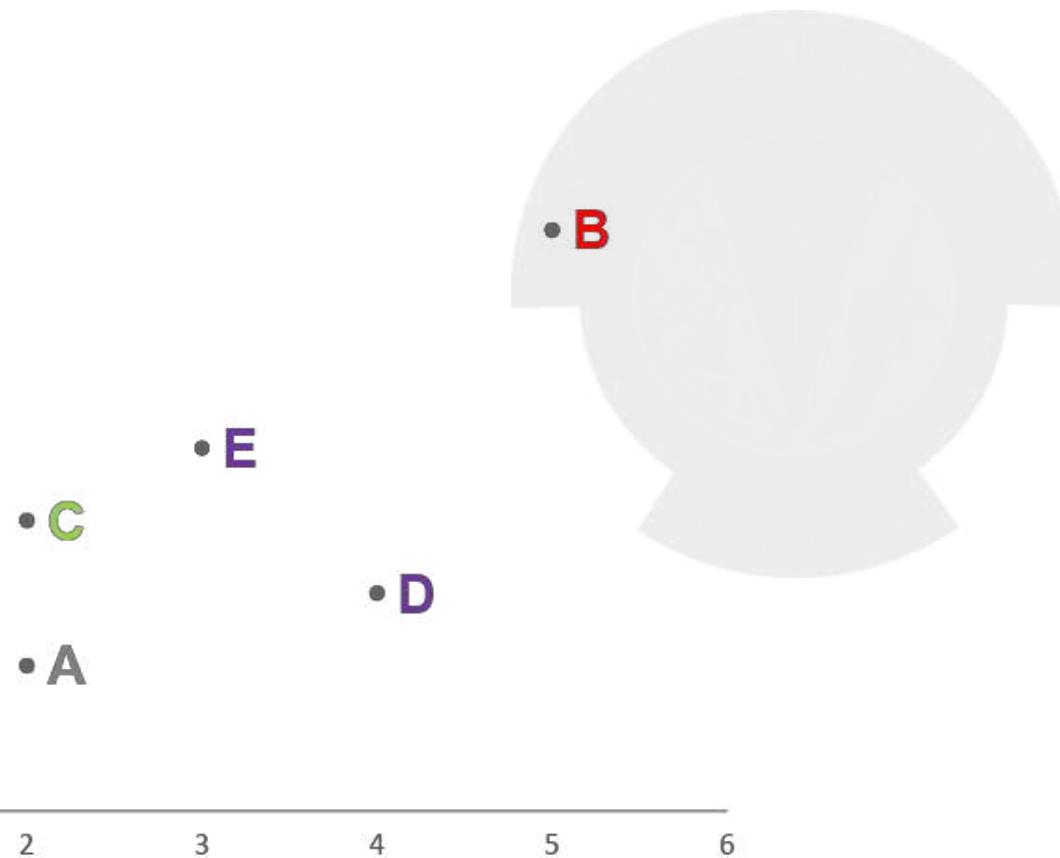
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

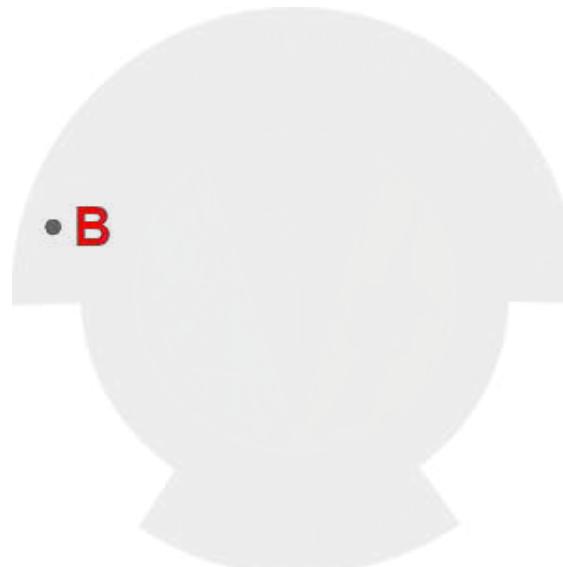
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

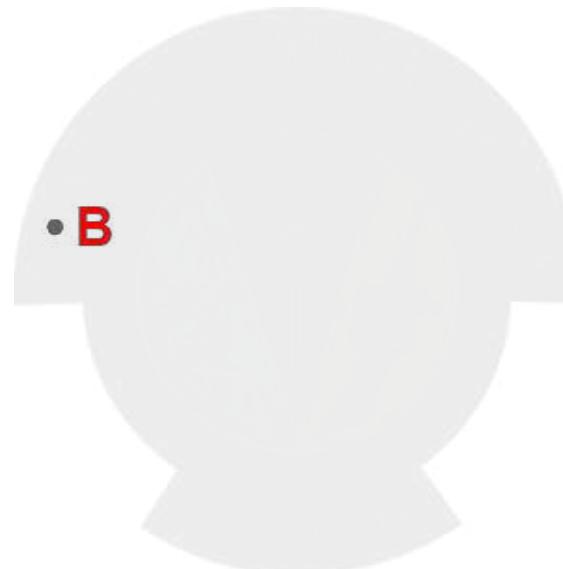
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

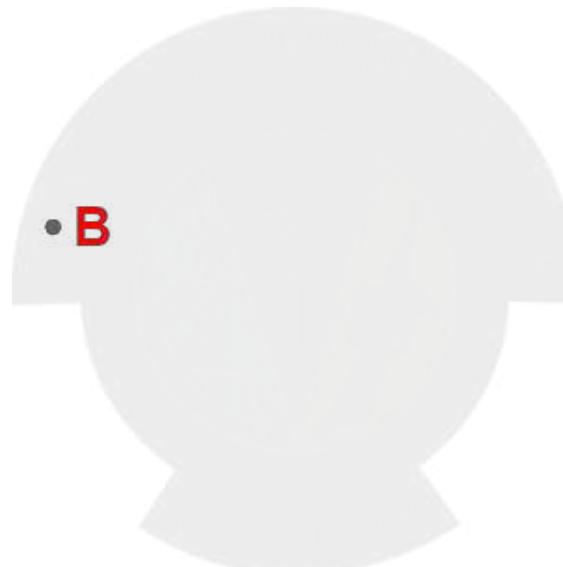
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

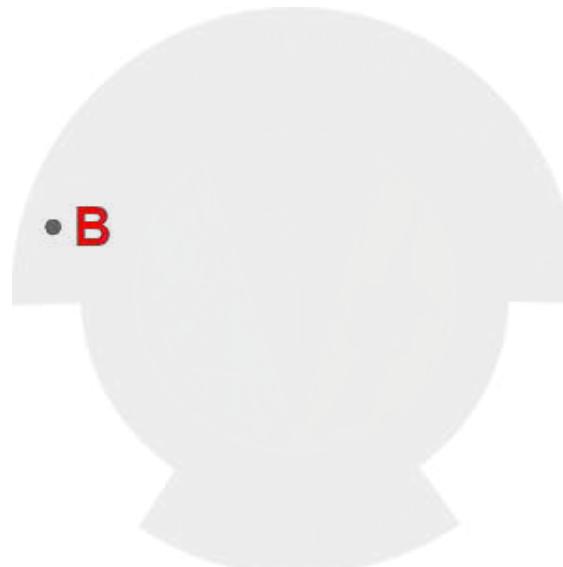
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

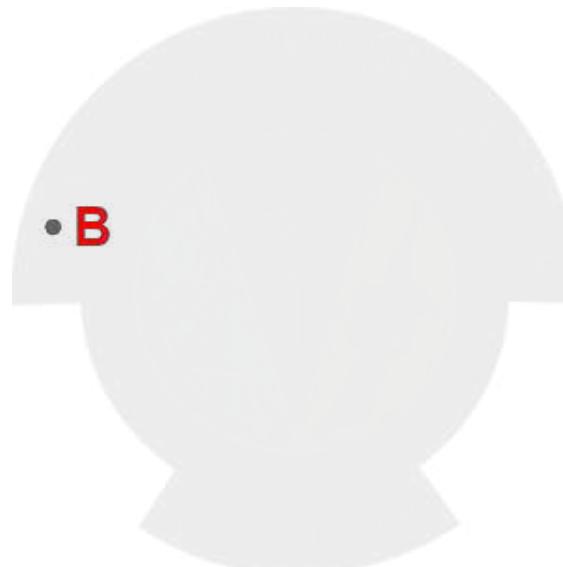
	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0

DBSCAN

Example

Eps / ϵ = 2.5

MinPts = 4



0 1 2 3 4 5 6

Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

	A	B	C	D	E
A	0	6.7	2.0	2.2	3.2
B		0	5.0	5.1	3.6
C			0	2.2	1.4
D				0	2.2
E					0



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

DBSCAN Algorithm - Some Points

S.P.Vimal

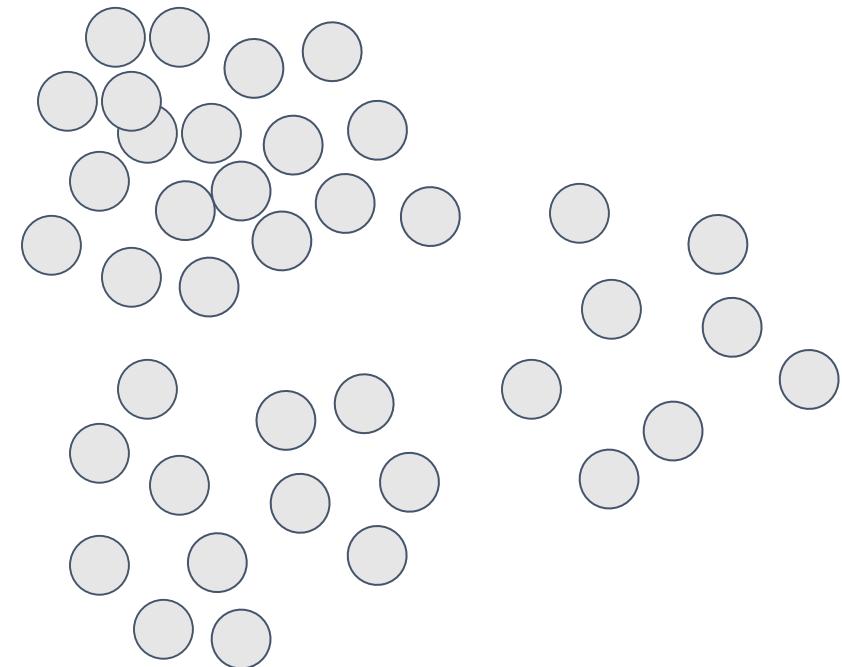
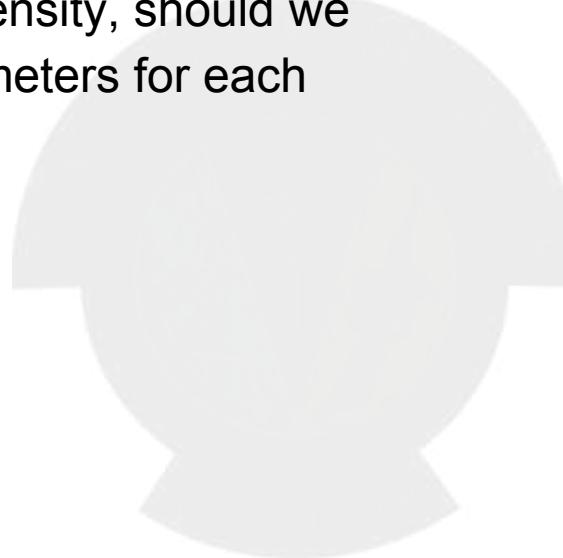
Agenda

- On the choice of parameters eps, MinPts
- Performance

DBSCAN

On the parameters Eps and MinPts

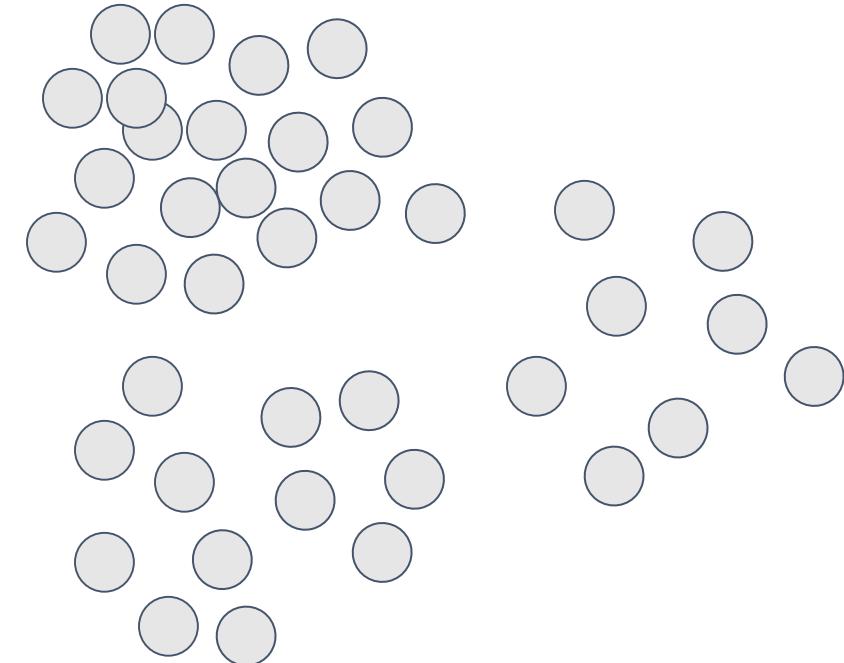
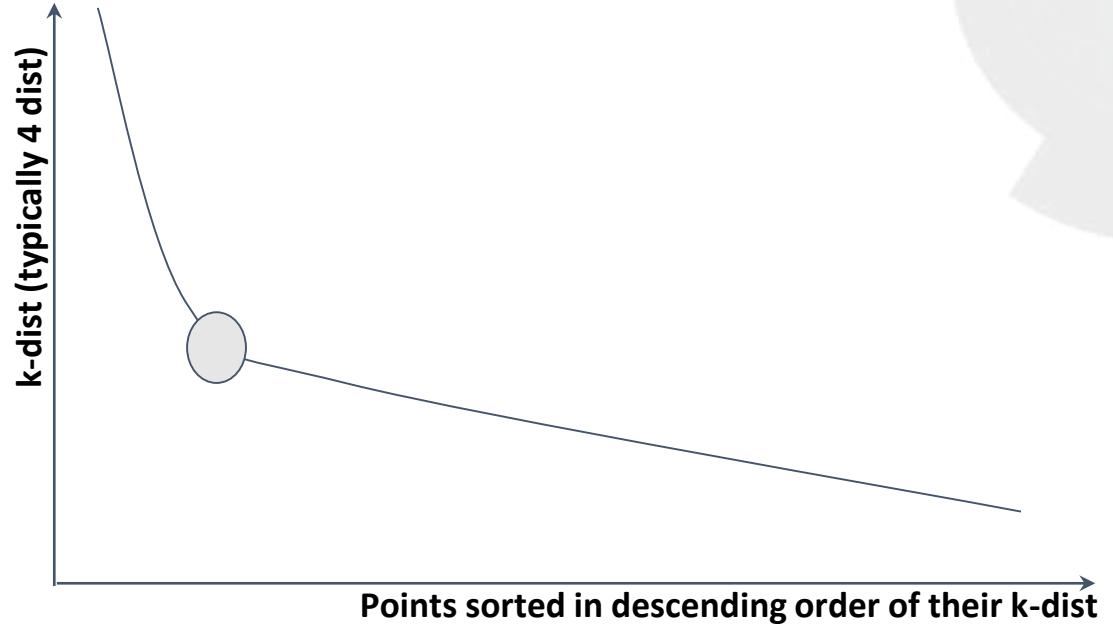
- Quality of clustering depends on the parameters
 - Assuming clusters of varying density, should we consider different density parameters for each clusters separately?



DBSCAN

On the parameters Eps and MinPts

- Quality of clustering depends on the parameters
 - Assuming clusters of varying density, should we consider different density parameters for each clusters separately?



DBSCAN

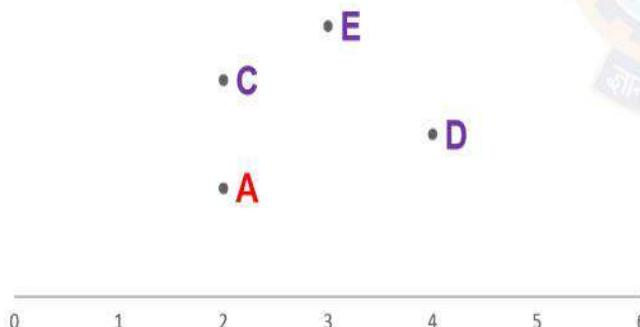
Performance

- Average time complexity reported by Ester et.al (KDD, '96) is $O(n \log n)$
 - Uses R^* -tree to for all the neighbourhood queries
 - Each query is $O(\log n)$
 - Popular Algorithm !!!

Example

Eps / ϵ = 2.5

MinPts = 4



Neighbors		X	Y
C, D	A	2	2
-	B	5	8
A, D, E	C	2	4
A, C, E	D	4	3
C, D	E	3	5

	A	B	C	D	E
A	06.7	2.0	2.2	3.2	
B		05.0	5.1	3.6	
C			02.2	1.4	
D				02.2	
E					0



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Evaluation of Clustering - Introduction

S.P.Vimal

In this segment

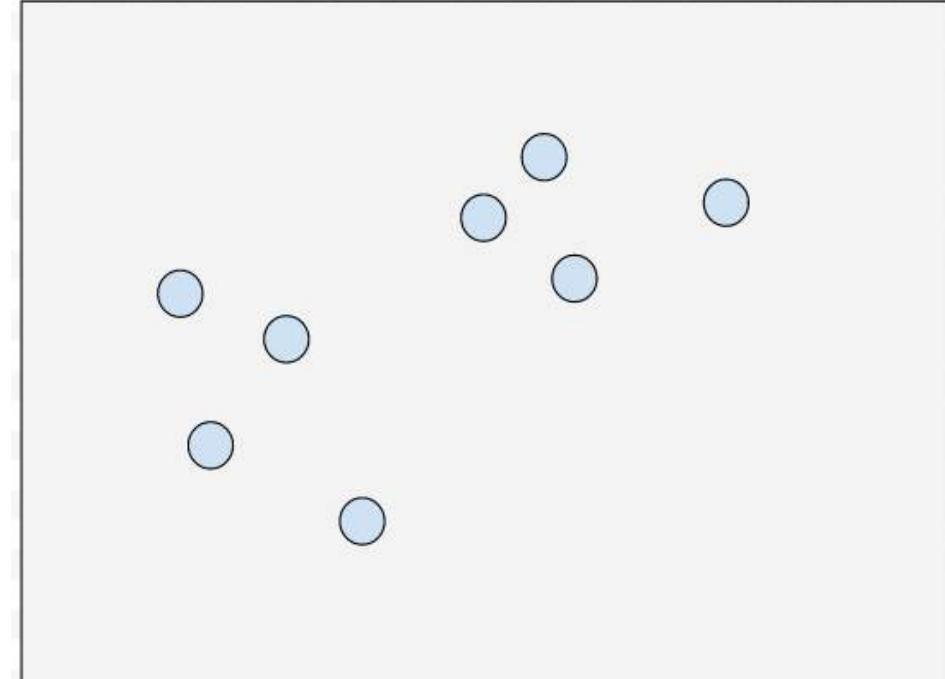
- Evaluation of Clustering - Introduction



Evaluation of Clustering - Introduction

How good are the clusters obtained?

- Does my clusters correspond to reality ?
- How do I measure if one set of clusters is better than others?
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
 - Ex. Silhouette coefficient
 - Use ground truth / how well the clusters fit the data



In this segment

- Silhouette Value

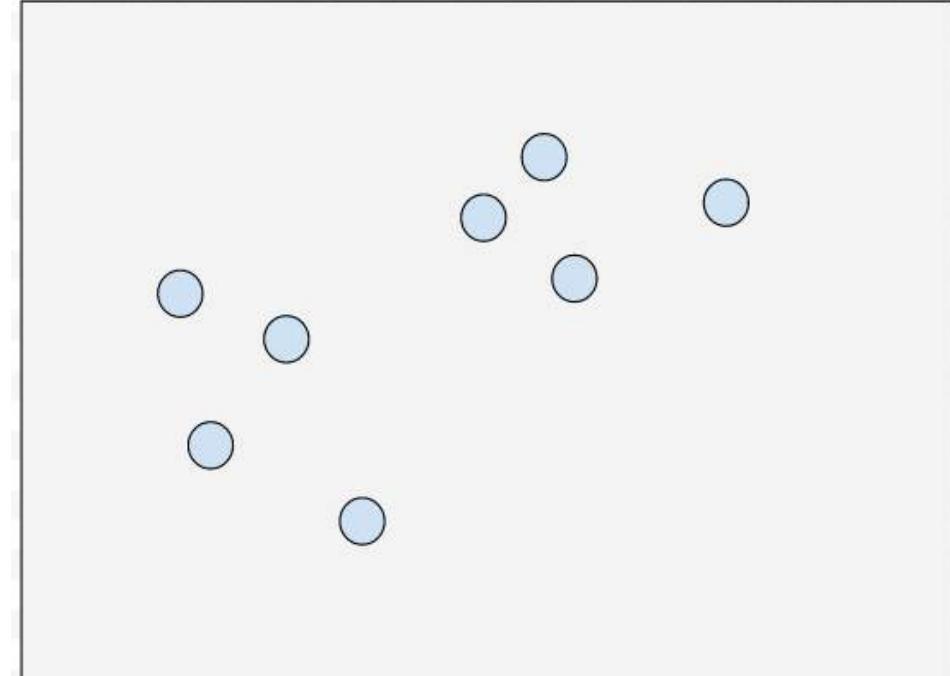
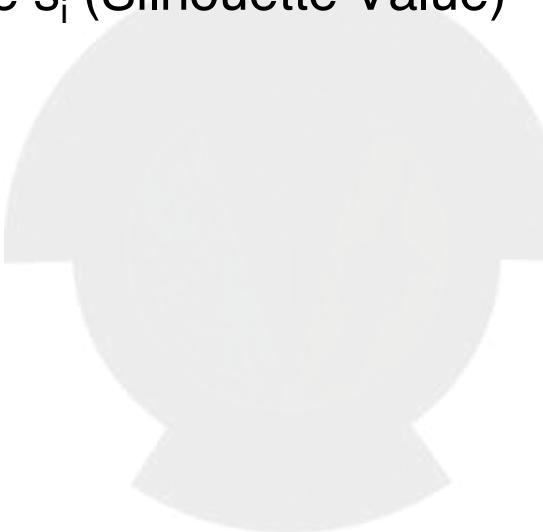


Silhouette Method

Measure

- For each data point x_i , determine s_i (Silhouette Value) as

$$s_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$



Where

a_i = distance between a data value and its center

b_i = distance between a data value and its next closest center

Silhouette Method

Measure

- For each data point x_i , determine s_i (Silhouette Value) as

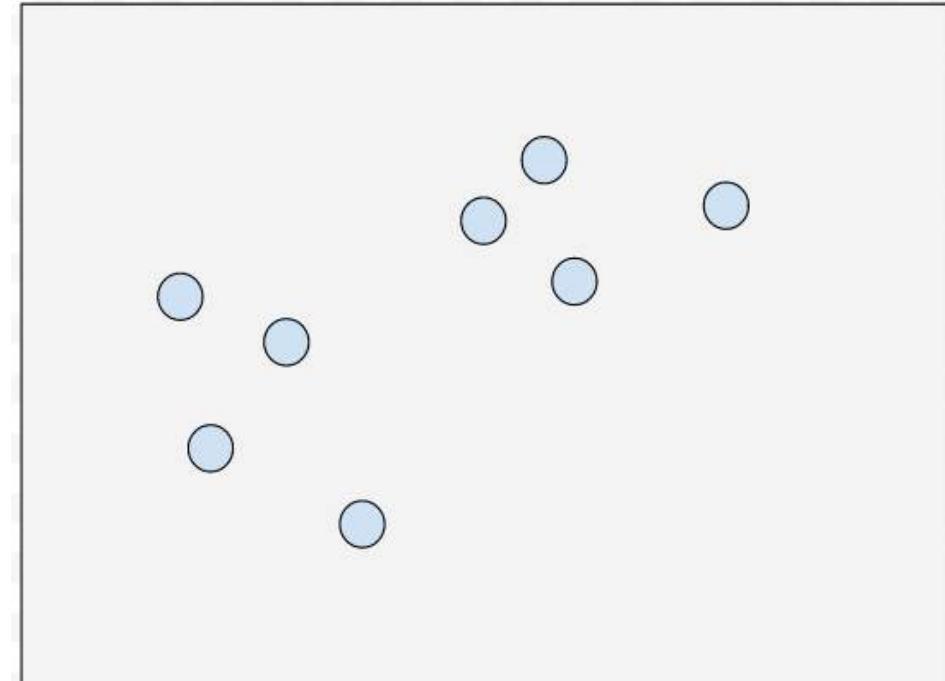
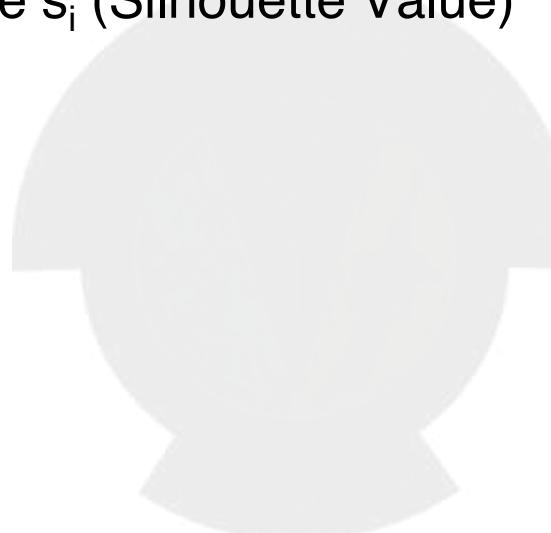
$$s_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$

s_i is

+ve - Assignments are good

-ve - Assignments must be improved

higher s_i are indicate good clustering



Silhouette Method

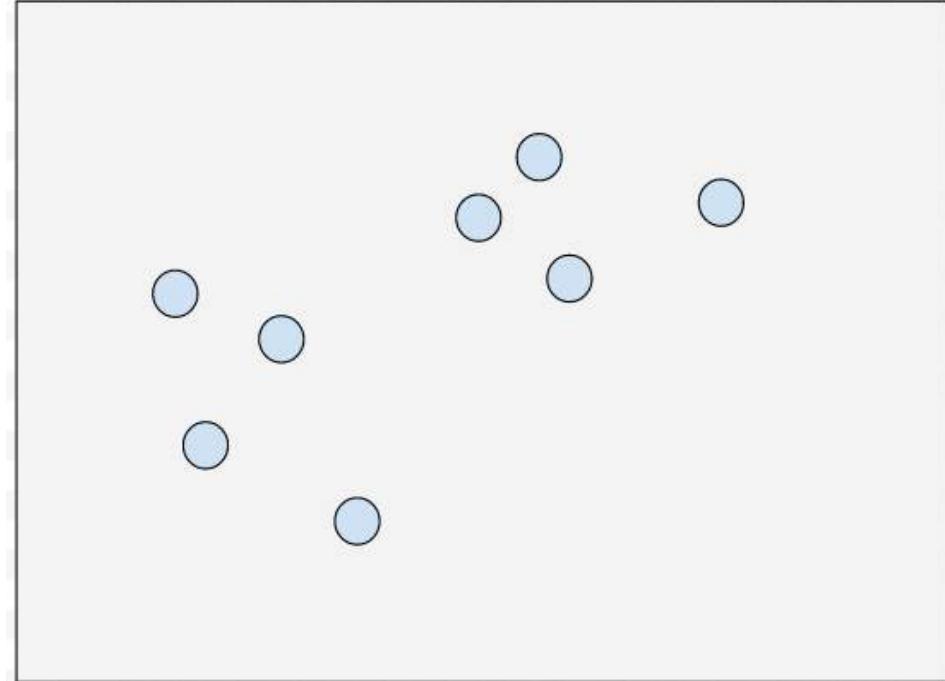
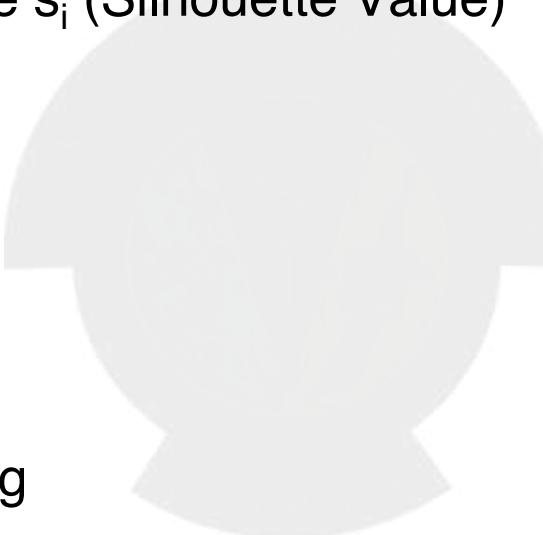
Measure

- For each data point x_i , determine s_i (Silhouette Value) as

$$s_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$

higher s_i are indicate good clustering

s_i -closer to 0, weak assignment



Silhouette Method

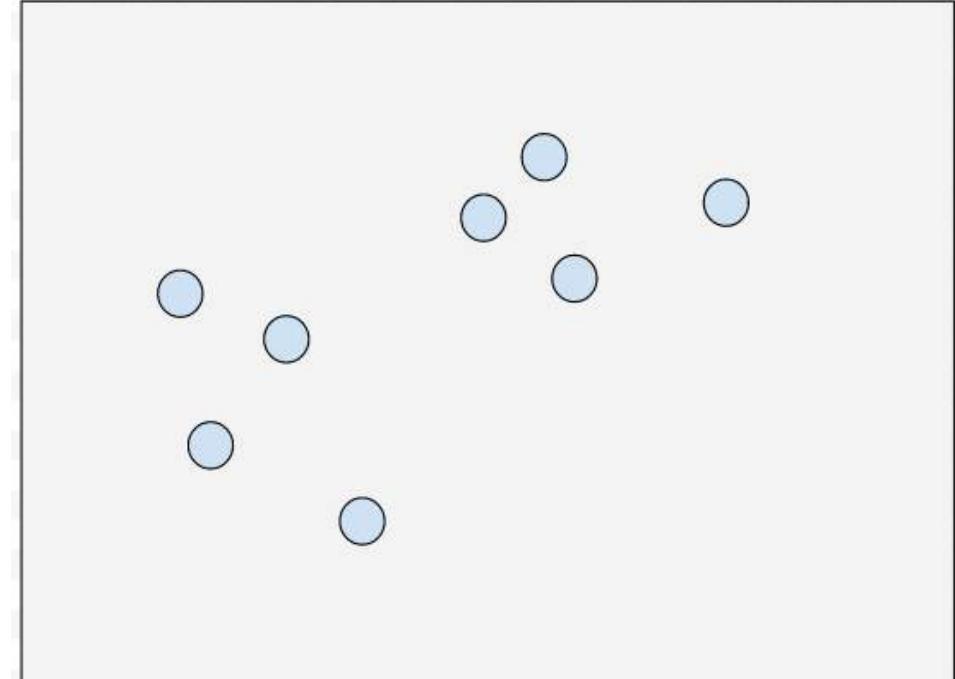
Measure

- For each data point x_i , determine s_i (Silhouette Value) as

$$s_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$

Usage:

Take the average s_i values across all the points and use this as a measure of goodness of clustering



Silhouette Method

Measure

- For each data point x_i , determine s_i (Silhouette Value) as

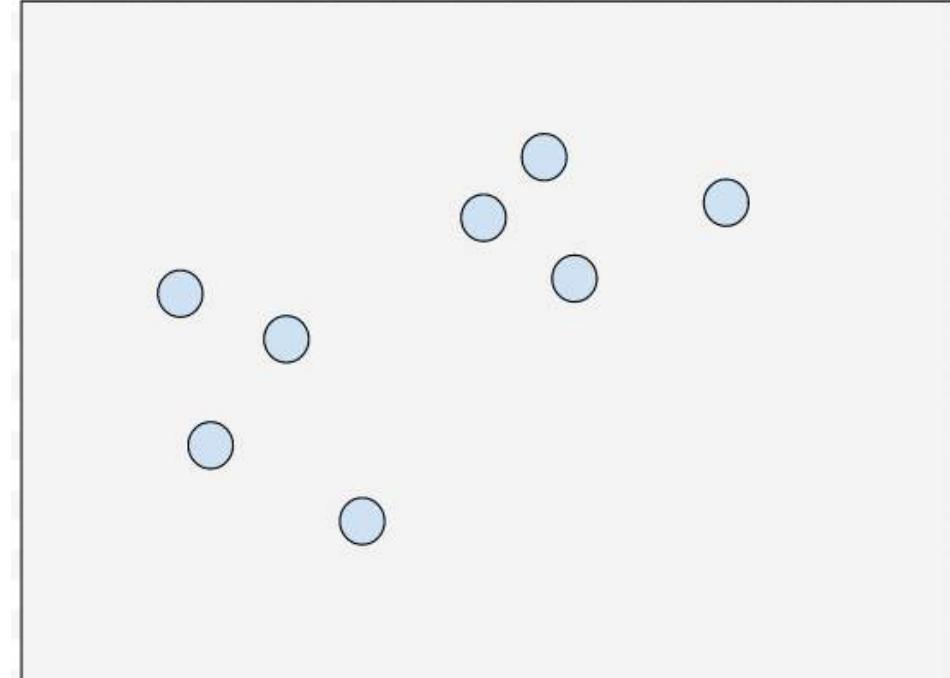
$$s_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$

Interpretation of Average Silhouette Value:

≥ 0.5 : Evidence of good clustering

between 0.25 and 0.5: Some evidence of reality, check with the domain expert

<0.25 : not adequate evidence of cluster reality



In this segment

- Rand Index
- Purity

Rand Index

- Most popular external measures
 - Evaluation in a supervised manner
- Assume ground truth data is available
 - Compare the clustering results with a priori labels
 - Uses pairwise agreements between the set of discovered clusters K and labels C

Rand Index

Rand Index

Let

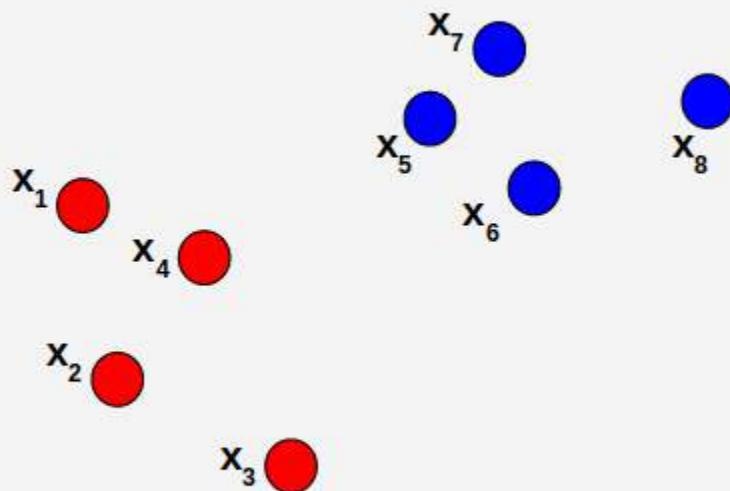
- a - Number of pairs of data points with the same cluster in C and assigned to the same cluster in K
- b - Number of pairs with the same cluster in C, but in different clusters K
- c - Number of pairs in the same cluster in K, but with different clusters in C
- d - Number of pairs with a different cluster in C and assigned to a different cluster in K

$$R = \frac{(a + d)}{(a + b + c + d)}$$

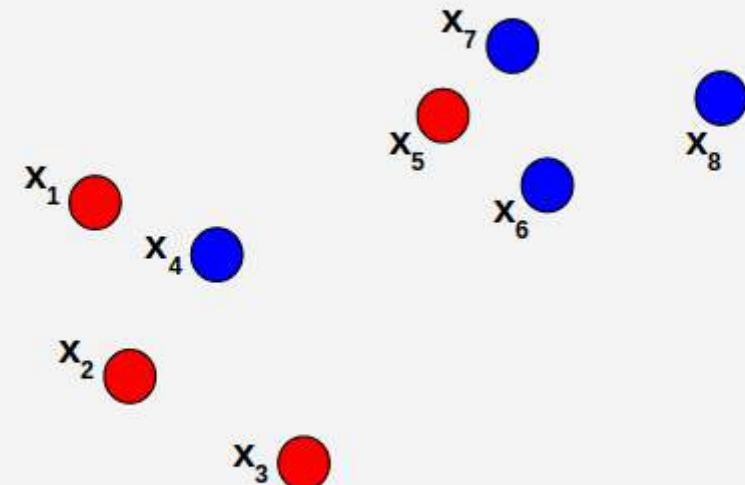
0 \leq R \leq 1, where a value of 1 indicates that C and K are identical
Higher R values are preferred.

Rand Index

Ground Truth



Cluster Result



Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

We consider only unordered pairs.

Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

a - Number of pairs of data points with the same cluster in C and assigned to the same cluster in K = 6

Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

b - Number of pairs with the same cluster in C, but in different clusters K = 6

Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

c - Number of pairs with the different cluster in C, but in same clusters K = 6

Rand Index

How to get a,b,c & d

Kij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	1	0	0	0	0
x ₂	1	1	1	1	0	0	0	0
x ₃	1	1	1	1	0	0	0	0
x ₄	1	1	1	1	0	0	0	0
x ₅	0	0	0	0	1	1	1	1
x ₆	0	0	0	0	1	1	1	1
x ₇	0	0	0	0	1	1	1	1
x ₈	0	0	0	0	1	1	1	1

From Ground Truth

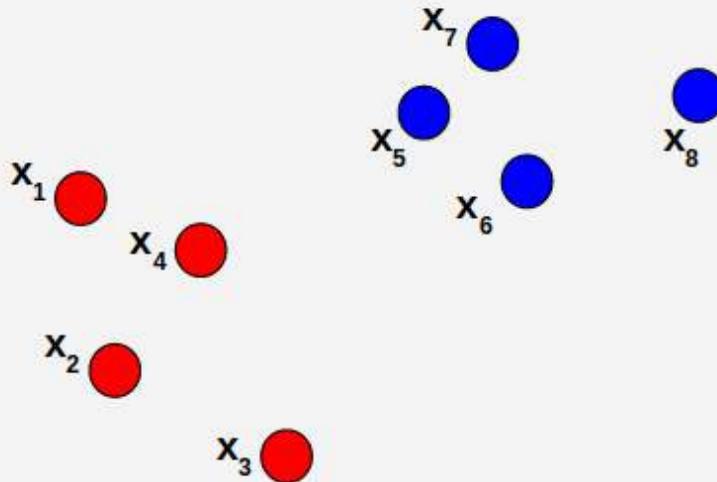
Cij	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	1	1	1	0	1	0	0	0
x ₂	1	1	1	0	1	0	0	0
x ₃	1	1	1	0	1	0	0	0
x ₄	0	0	0	1	0	1	1	1
x ₅	1	1	1	0	1	0	0	0
x ₆	0	0	0	1	0	1	1	1
x ₇	0	0	0	1	0	1	1	1
x ₈	0	0	0	1	0	1	1	1

From Clusters Obtained

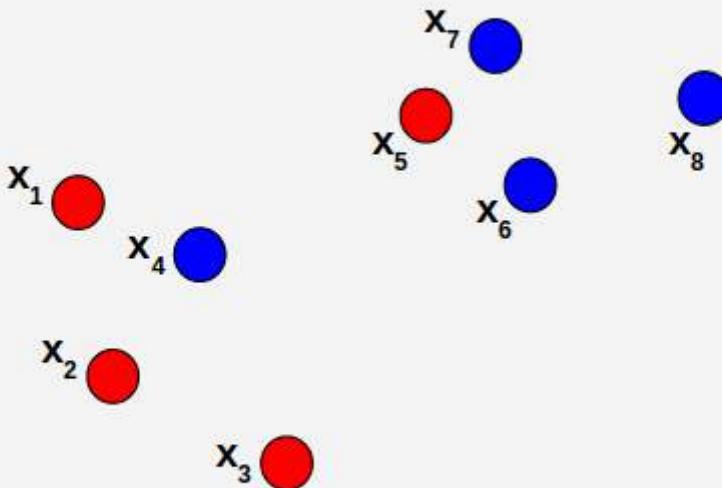
d - Number of pairs with a different cluster in C and assigned to a different cluster in K = **10**

Rand Index

Ground Truth



Cluster Result



$$a = 6$$

$$b = 6$$

$$c = 6$$

$$d = 10$$

$$\text{Rand Index} = (a+d) / (a+b+c+d) = 16 / 28 = 0.79$$

Purity

Measure

- An external measure of evaluation

$$P = \text{for all } K \left[p_i / M \right]$$

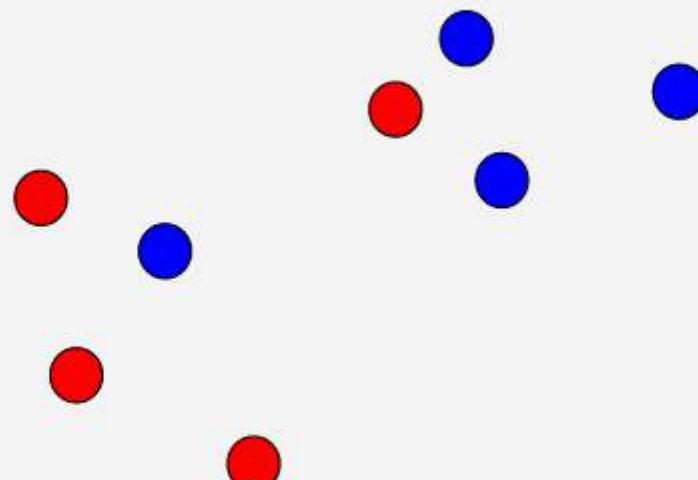
Where

p_i are purity values for each cluster [#of majority label]

K is the number of clusters, and

M is the total number of samples in data set

Cluster Result



Purity

Measure

- An external measure of evaluation

$$P = \text{for all } K \left[\frac{\pi_i}{M} \right]$$

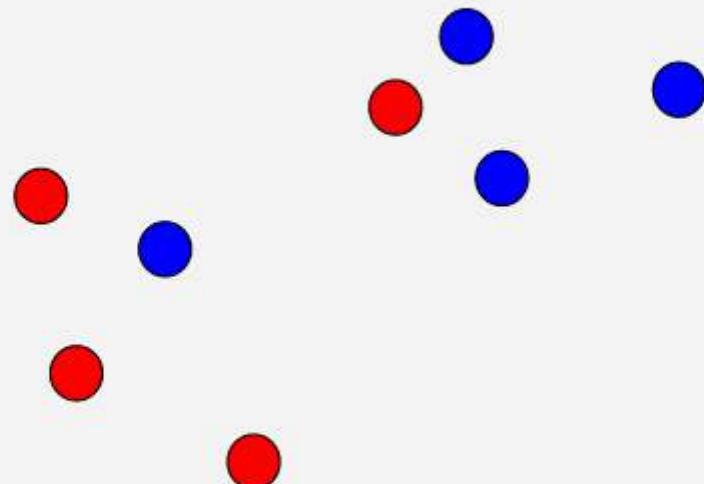
Purity of the clustering is the sum of local purity.

Local purity is π_i/M

$$\text{Purity} = [3 + 3] / 8$$



Cluster Result



In this Segment

Association Rule Mining

- Motivation
- Market Basket Analysis

Motivation

Applications

- Rakesh Agrawal, Tomasz Imieli ski and Arun Swami - 1993
- Discover **frequent patterns**: Patterns that appear frequently in a data set.

- Frequent Item Set
- Frequent Sequential Pattern
- Frequent Structured Pattern

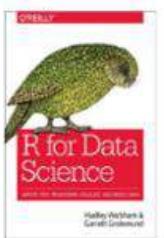
Motivation

Applications

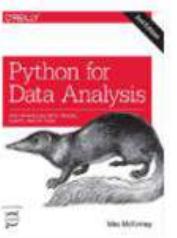
- **Frequent Item Set** : A set of items that often appear together in a transactional data set.
- Frequent Sequential Pattern
- Frequent Structured Pattern



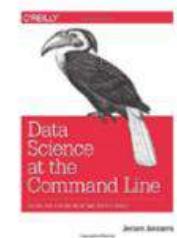
Customers who viewed this item also viewed



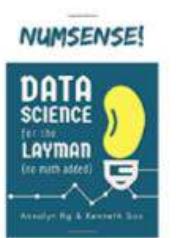
R for Data Science: Import, Tidy, Transform, Visualize, and Model Data
Hadley Wickham
★★★★★ 121
#1 Best Seller in Mathematical & Statistical...
Paperback \$18.17 ✓prime



Python for Data Analysis: Data Wrangling with Pandas, NumPy, and...
Wes McKinney
★★★★★ 74
Paperback \$35.65 ✓prime



Data Science at the Command Line: Facing the Future with Time-Tested...
Jeroen Janssens
★★★★★ 11
Paperback \$27.33 ✓prime



Numsense! Data Science for the Layman: No Math Added
Annalyn Ng
★★★★★ 87
Paperback \$27.54 ✓prime

amazon.com

Amazon.com has new recommendations for you based on items you purchased or told us you own.



The Little Big Things: 163 Ways to Pursue EXCELLENCE



Fascinate: Your 7 Triggers to Persuasion and Captivation



Sherlock Holmes [Blu-ray]



Alice in Wonderland [Blu-ray]



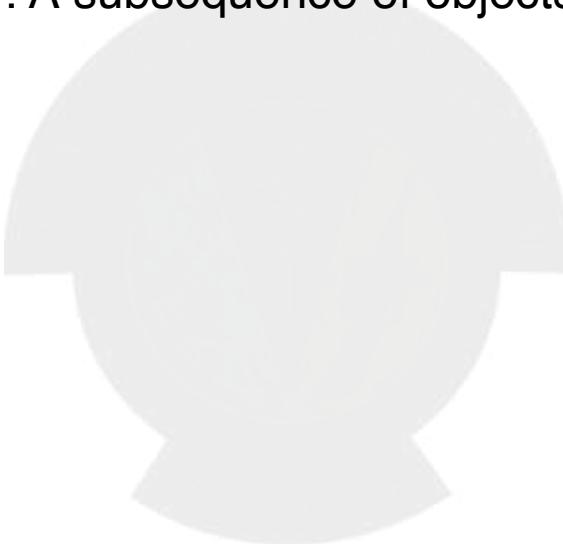
Motivation

Applications

- Frequent Item Set
- **Frequent Sequential Pattern** : A subsequence of objects/events that often appear together in specific order in a data set
- Frequent Structured Pattern



Image Credit: <http://news.mit.edu/>



Motivation

Applications

- Frequent Item Set
- Frequent Sequential Pattern
- **Frequent Structured Pattern:** Sub structures are Structural forms such as subgraphs, subtrees or sub-lattices which may be combined with item sets or subsequences. The frequent occurrence of it forms frequent structured pattern

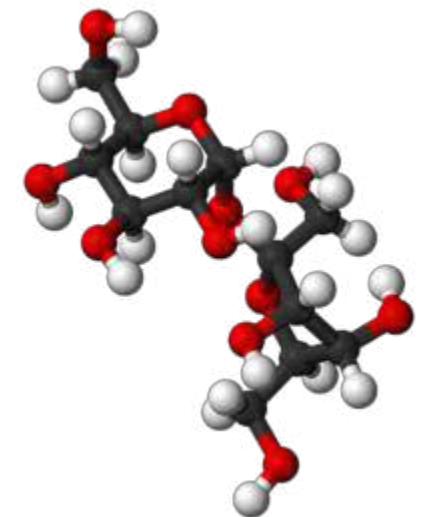
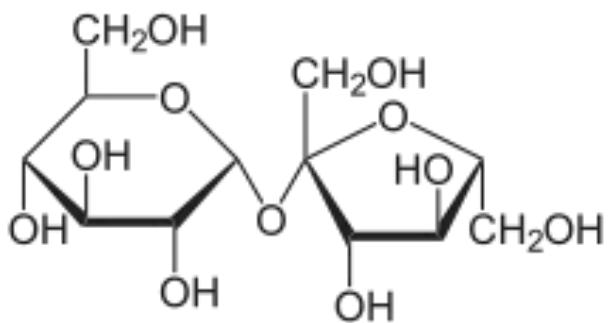


Image Credit: <https://en.wikipedia.org/>



Market Basket Analysis

Market Basket Analysis

- Market Basket Analysis process analyzes the customer buying habits by finding associations between the different items that customer place in their shopping basket.

	Purchase 1	Purchase 2
Jack	Paper, Pen, Medicine, Milk	Milk, Bread, Egg, Milk	
Jill	Rice, Medicine, Vegetable, Milk	Rice, Egg, Vegetable, Milk	
John	Bread, Jam, Butter , Jam	Milk, Bread, Pasta, Medicine	

Market Basket Analysis

- ✓ “Which groups or sets of items are customers likely to purchase on a given trip to the store?”
- ✓ “Which items could be placed near popular item sets to increase the customer satisfaction?”
- ✓ “Which items could be placed down the aisle to draw attention to other products?”
- ✓ “What promotional strategies will help increase the sale?”
- ✓ “How to strategize the product pricing?”

	Purchase 1	Purchase 2
Jack	Paper, Pen, Medicine, Milk	Milk, Bread, Egg, Milk	
Jill	Rice, Medicine, Vegetable, Milk	Rice, Egg, Vegetable, Milk	
John	Bread, Jam, Butter , Jam	Milk, Bread, Pasta, Medicine	

	Items Bought
Transaction T001	Paper, Pen, Medicine, Milk
Transaction T002	Rice, Medicine, Vegetable, Milk
Transaction T003	Milk, Bread, Egg, Milk
Transaction T004	Bread, Jam, Butter , Jam
Transaction T005	Milk, Bread, Pasta, Medicine
Transaction T006	Rice, Egg, Vegetable, Milk

Market Basket Analysis

- ✓ Catalog Design
- ✓ Demand Supply Management
- ✓ Product Placement
- ✓ Shelf Space Management
- ✓ Product Sale Strategy
- ✓ Product Promotion
- ✓ Targeted Marketing
- ✓ Bundle Pricing
- ✓ Cross Selling

	Purchase 1	Purchase 2
Jack	Paper, Pen, Medicine, Milk	Milk, Bread, Egg, Milk	
Jill	Rice, Medicine, Vegetable, Milk	Rice, Egg, Vegetable, Milk	
John	Bread, Jam, Butter , Jam	Milk, Bread, Pasta, Medicine	

	Items Bought
Transaction T001	Paper, Pen, Medicine, Milk
Transaction T002	Rice, Medicine, Vegetable, Milk
Transaction T003	Milk, Bread, Egg, Milk
Transaction T004	Bread, Jam, Butter , Jam
Transaction T005	Milk, Bread, Pasta, Medicine
Transaction T006	Rice, Egg, Vegetable, Milk

Market Basket Analysis

- Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

	Items Bought
Transaction T001	Paper, Pen, Medicine, Milk
Transaction T002	Rice, Medicine, Vegetable, Milk
Transaction T003	Milk, Bread, Egg, Milk
Transaction T004	Bread, Jam, Butter , Jam
Transaction T005	Milk, Bread, Pasta, Medicine
Transaction T006	Rice, Egg, Vegetable, Milk

	Bread	Butter	Egg	Jam	Medicine	Milk	Paper	Pasta	Pen	Rice	Vegetable
T001	0	0	0	0	1	1	1	0	1	0	0
T002	0	0	0	0	1	1	0	0	0	1	1
T003	1	0	1	0	0	1	0	0	0	0	0
T004	1	1	0	1	0	0	0	0	0	0	0
T005	1	0	0	0	1	1	0	1	0	0	0
T006	0	0	1	0	0	1	0	0	0	1	1

In this Segment

Association Rule Mining

- Terminologies
- Measures

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

K-Itemset = $\{I_1, I_2, I_3, \dots, I_k\}$

	Items Bought
Transaction T001	Paper, Pen, Medicine, Milk
Transaction T002	Rice, Medicine, Vegetable, Milk
Transaction T003	Milk, Bread, Egg, Milk
Transaction T004	Bread, Jam, Butter , Jam
Transaction T005	Milk, Bread, Pasta, Medicine
Transaction T006	Rice, Egg, Vegetable, Milk

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

$K\text{-Itemset} = \{I_1, I_2, I_3, \dots, I_k\}$

	Bread	Butter	Egg	Jam	Medicine	Milk	Paper	Pasta	Pen	Rice	Vegetable
T001	0	0	0	0	1	1	1	0	1	0	0
T002	0	0	0	0	1	1	0	0	0	1	1
T003	1	0	1	0	0	1	0	0	0	0	0
T004	1	1	0	1	0	0	0	0	0	0	0
T005	1	0	0	0	1	1	0	1	0	0	0
T006	0	0	1	0	0	1	0	0	0	1	1

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

$K\text{-Itemset} = \{I_1, I_2, I_3, \dots, I_k\}$



	Trend	Data	Story	Mining	Cloth	Learning
Document 1	5	10	4	8	0	6
Document 2	5	5	8	0	7	10
Document 3	2	8	2	4	0	0

	Trend	Data	Story	Mining	Cloth	Learning
Document 1	1	1	0	1	0	1
Document 2	1	1	1	0	1	1
Document 3	0	1	0	0	0	0

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

$K\text{-Itemset} = \{I_1, I_2, I_3, \dots, I_k\}$

Order ID	Customer Age	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase Amount LEVEL
101	Adult	1	1	1	0	0	0	Low
104	Adult	1	0	1	1	1	1	Medium
106	Senior	1	0	1	1	1	0	High

	Order ID	Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low
101	1	0	0	1	1	1	1	0	0	0	0	0	1
104	1	0	0	1	0	1	1	1	1	1	0	1	0
106	0	1	0	1	0	0	1	1	1	0	1	0	0

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

$K\text{-Itemset} = \{I_1, I_2, I_3, \dots, I_k\}$

Order ID	Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low
101	1	0	0	1	1	1	0	0	0	0	0	1
104	1	0	0	1	0	1	1	1	1	0	1	0
106	0	1	0	1	0	1	1	1	0	1	0	0
Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low	
	101	106		101	101	101	104	104	104	106	104	101
	104			104		104	106	106				
				106		106						

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

K-Itemset = $\{I_1, I_2, I_3, \dots, I_k\}$

Concept Hierarchy:

Items:

$\{Paper, Pen\} \in Stationary$

$\{Milk, Egg, Butter\} \in Non-Vegan$

$\{Vegetable, Jam, Bread, Rice\} \in Vegan$

$\{Medicine\} \in Medicinal$

Purchase-Level:

$\{<=200\} \in Low$

$\{200 < Rs <=1000\} \in Medium$

$\{>1000\} \in High$

	Items Bought	Purchase amount
Transaction T001	Paper, Pen, Medicine, Milk	Rs. 100
Transaction T002	Rice, Medicine, Vegetable, Milk	Rs. 2500
Transaction T003	Milk, Bread, Egg, Milk	Rs. 200
Transaction T004	Bread, Jam, Butter , Jam	Rs. 150
Transaction T005	Milk, Bread, Pasta, Medicine	Rs. 600
Transaction T006	Rice, Egg, Vegetable, Milk	Rs. 1800

Terminologies

K-Item set

Each item in $I = \{I_1, I_2, I_3, \dots, I_N\}$ is represented as boolean variable

$K\text{-Itemset} = \{I_1, I_2, I_3, \dots, I_k\}$

Order ID	Stationary	Non-Vegan	Vegan	Medicinal	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low
T001	1	1	0	1	0	0	1
T002	0	1	1	1	1	0	0
T003	0	1	1	0	0	0	1
T004	0	1	1	0	0	0	1
T005	0	1	1	1	0	1	0
T006	0	1	1	0	1	0	0

	Items Bought	Purchase amount
Transaction T001	Paper, Pen, Medicine, Milk	Rs. 100
Transaction T002	Rice, Medicine, Vegetable, Milk	Rs. 2500
Transaction T003	Milk, Bread, Egg, Milk	Rs. 200
Transaction T004	Bread, Jam, Butter , Jam	Rs. 150
Transaction T005	Milk, Bread, Pasta, Medicine	Rs. 600
Transaction T006	Rice, Egg, Vegetable, Milk	Rs. 1800

Terminologies

Association Rules

An association rule is an implication of the form $X \rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, $X \neq \emptyset$, $Y \neq \emptyset$, $X \cap Y = \emptyset$.
where:

$$I = \{i_1, i_2, i_3, \dots, i_d\}$$

$$T = \{t_1, t_2, t_3, \dots, t_N\}$$

Example:

$\{\text{Bread, Butter}\} \rightarrow \{\text{Milk}\}$

$\{\text{Data, Mining}\} \rightarrow \{\text{Features, Pattern}\}$

$\{\text{Gene-T1, Gene-T15}\} \rightarrow \{\text{Gene-T3}\}$

$\{\text{Shoes}\} \rightarrow \{\text{Watches, Costume}\}$

$\{\text{VideoY}\} \rightarrow \{\text{AdversimentX}\}$

Terminologies

Association Rules

An association rule is an implication of the form $X \rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, $X \neq \emptyset$, $Y \neq \emptyset$, $X \cap Y = \emptyset$.
where:

$$I = \{i_1, i_2, i_3, \dots, i_d\}$$

$$T = \{t_1, t_2, t_3, \dots, t_N\}$$

Choice of Association Rule:

Association between values

$$\{\text{Milk}\} \rightarrow \{\text{Bread}\}$$

Association between categories of values

$$\{\text{Vegan}\} \rightarrow \{\text{Medicinal}\}$$

Association between values of attributes

$$\{\text{Items Purchased : Milk}\} \rightarrow \{\text{Payment Mode : Cash}\}$$

Association over a time period

$$\{2018 : \text{CustomerID}\} \rightarrow \{2019 : \text{CustomerID}\}$$



Measures

Measures

Support

The occurrence frequency of an item set is the number of transactions that contain the itemset and is known as frequency , support count or count of the itemset.

X → Y

{Bread} → {Juice}

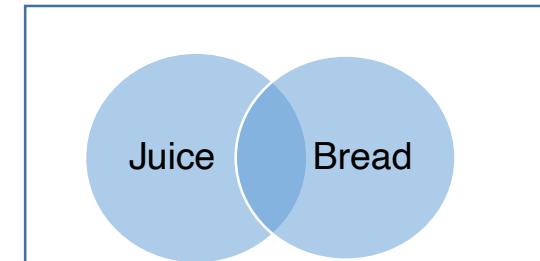
Support Threshold = 60%

Support Count (σ)

Frequency of occurrence of an itemset

$(X \rightarrow Y)$

Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread



Support (s)

Fraction of transactions that contain an itemset

$$s(X \rightarrow Y) = \frac{(X \rightarrow Y)}{|T|} = \frac{(X \rightarrow Y)}{N}$$

$$s(X \rightarrow Y) = P(X \text{ and } Y) = P(\overline{X} \cup Y) = P(X \mid Y)$$

Measures

Support

The occurrence frequency of an item set is the number of transactions that contain the itemset and is known as frequency , support count or count of the itemset.

X → Y

{Bread} → {Juice}

Support Threshold = 60% (*minsup*)

Frequent Item set

An itemset whose support is greater than or equal to a *minsup* threshold

Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread

Measures

Confidence

Confidence is a measure of certainty associated with each discovered rule. It determines how frequently items in Y appear in transactions that contain X.

$X \rightarrow Y$

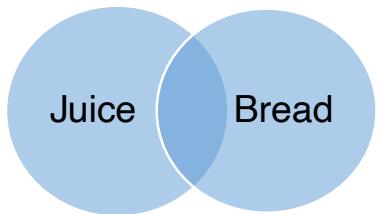
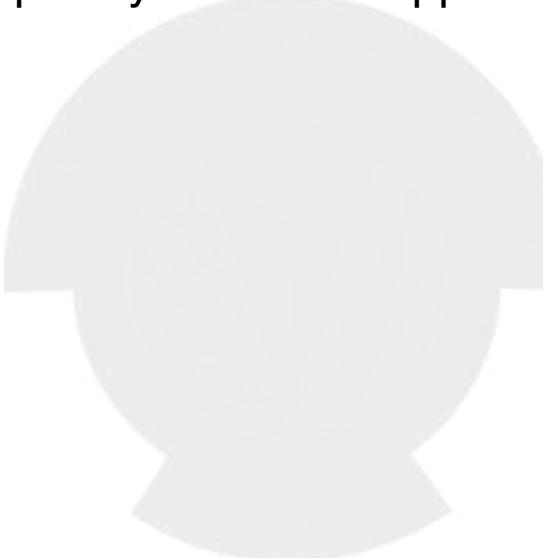
$\{Bread\} \rightarrow \{Juice\}$

Confidence Threshold = 60%

$$c(X \rightarrow Y) = \frac{(X \rightarrow Y)}{(X)} = \frac{(X \cap Y)}{(X)}$$
$$= P(Y | X)$$

Range: [0,1]

Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Association Rule Mining

Computational Complexity in Itemset Generation

Raja Vadhana P

In this Segment

Association Rule Mining

- Computational Complexity in Itemset Generation
- Maximal & Closed Frequent Itemsets

Computational Complexity in Itemset Generation

Computational Complexity

Goal of Association Rule mining

- Find frequent itemsets
- Find set of all rules that satisfies below properties:

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Computational Complexity

Goal of Association Rule mining

- **Find frequent itemsets**

Item set Generation is the process of listing all possible candidates / k-itemsets to derive association rules.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

1-itemset : {{Bread}, {Milk}, {Diaper}, {Butter}, {Egg}.....}

2-itemset : {{Bread, Milk}, {Bread, Diaper}, {Bread, Butter}, {Bread, Egg}, {Milk, diaper}, {Milk, Butter}.....}

3-itemset : {{Bread, Milk, Butter}, {Bread, Milk, Diaper}, {Milk, Butter, Diaper},.....}

- Find set of all rules that satisfies below properties:

Support($X \rightarrow Y$) \geq minsup

Confidence ($X \rightarrow Y$) \geq minconf

Computational Complexity

Goal of Association Rule mining

Item set Generation is the process of listing all possible candidates / k-itemsets to derive association rules.

1-itemset : {{Bread}, {Milk}, {Diaper}, {Butter}, {Egg}.....}

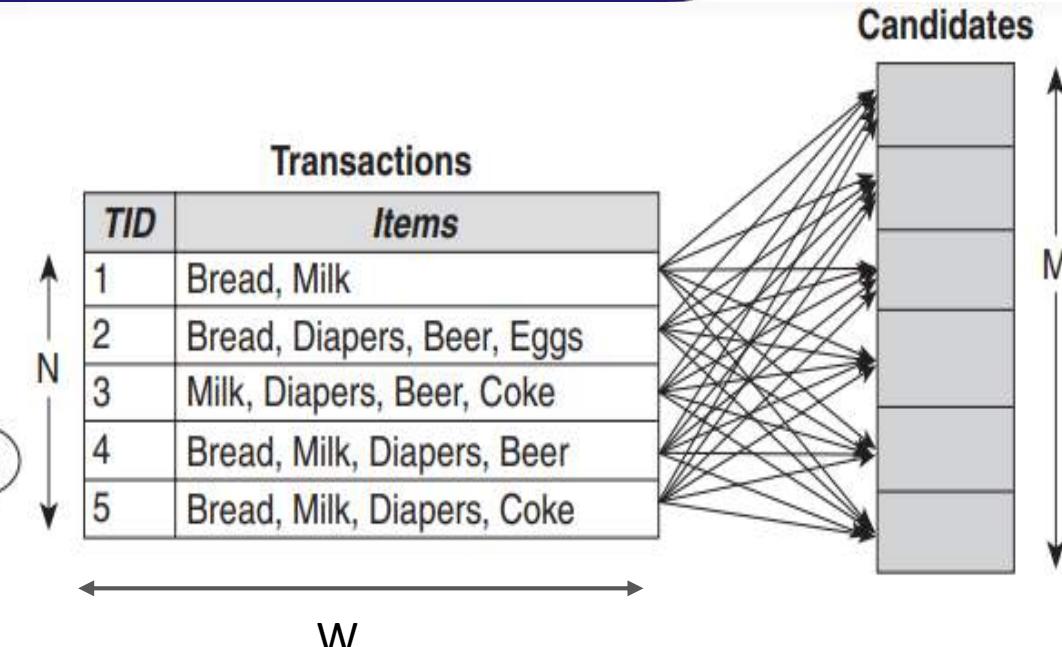
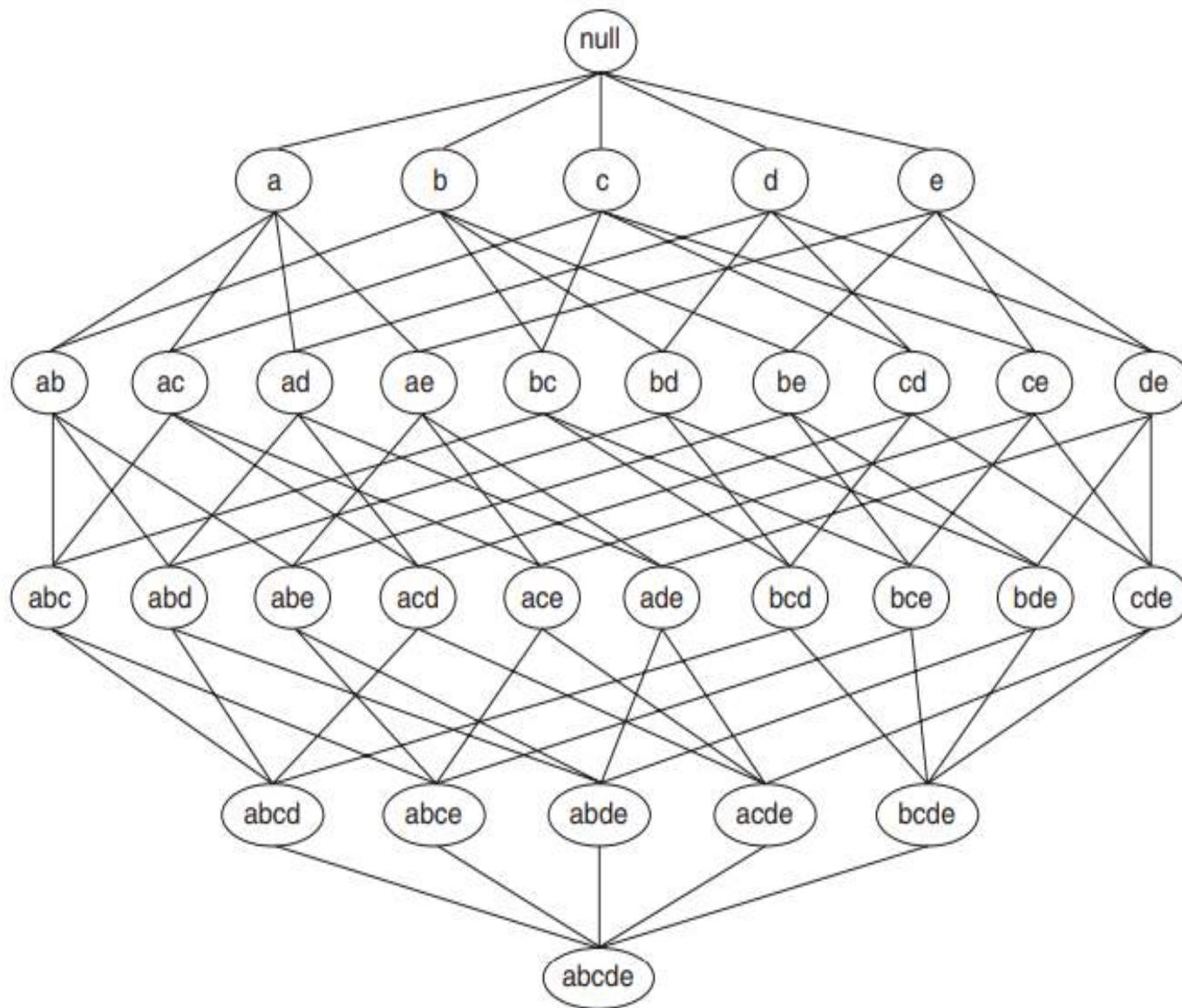
2-itemset : {{Bread, Milk}, {Bread, Diaper}, {Bread, Butter}, {Bread, Egg}, {Milk, diaper}, {Milk, Butter}.....}

3-itemset : {{Bread, Milk, Butter}, {Bread, Milk, Diaper}, {Milk, Butter, Diaper},.....}

Given d items, there are $2^d - 1$ possible candidate item sets

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Computational Complexity



N – Total number of Transactions

M – Total number of itemsets / candidates

W – Width of the Transactions

$$M = 2^d - 1$$

Computational Complexity

Strategies to reduce “M”

Idea: Use the support count to reduce the candidates generation

Apriori Principle:

If an item set is frequent then all of its subsets must also be frequent.

If an item set is infrequent then all of its supersets must also be infrequent.

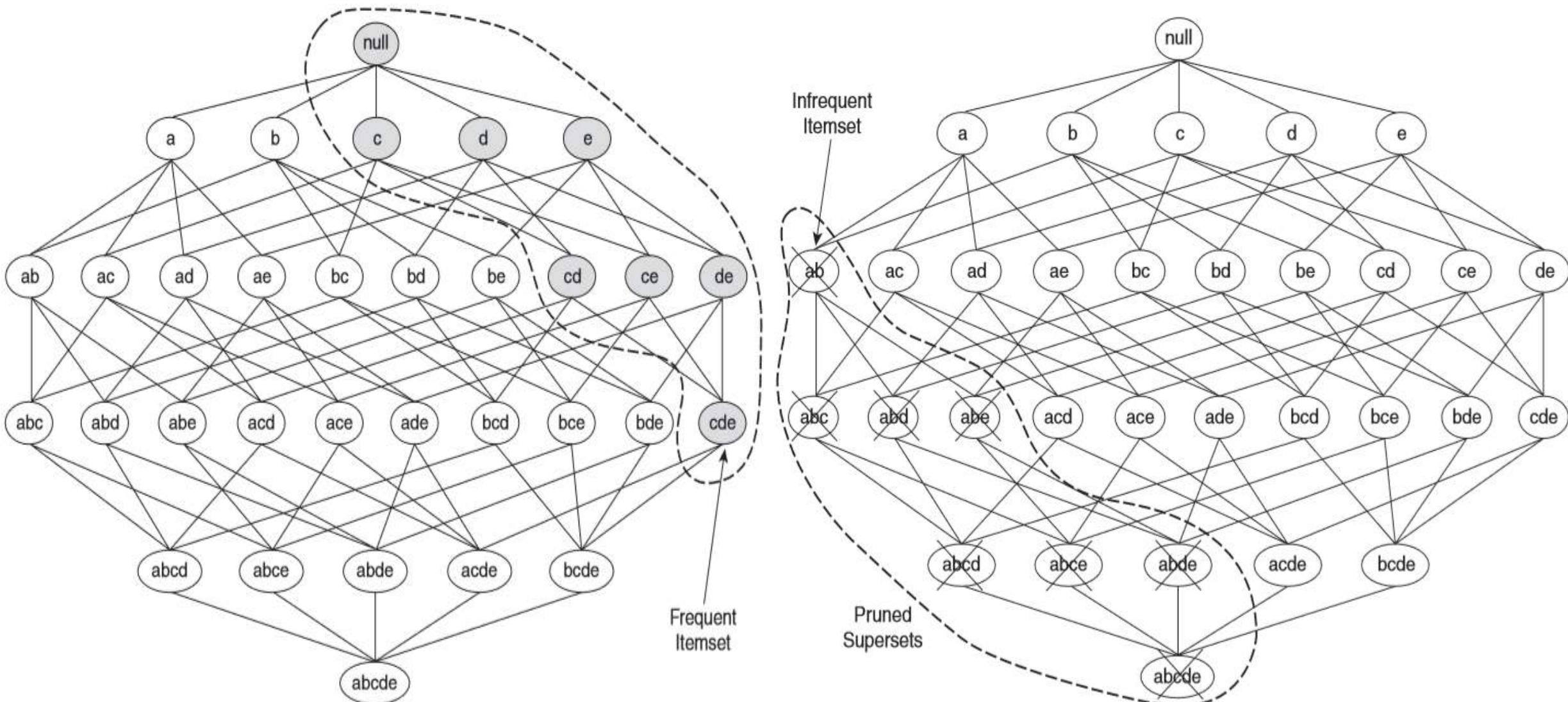
This intuition is known as the anti-monotone property of the support measure and this strategy of pruning the item set based on support is called **Support Based Pruning**

Monotone: $\forall X, Y \in J : (X \subseteq Y) \rightarrow f(X) \leq f(Y)$

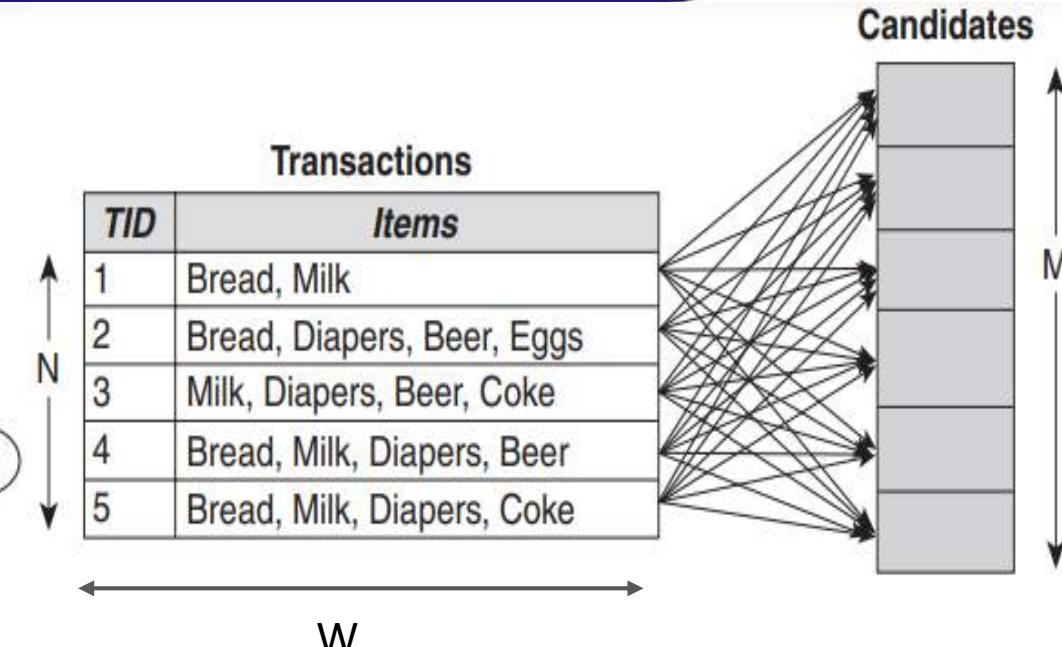
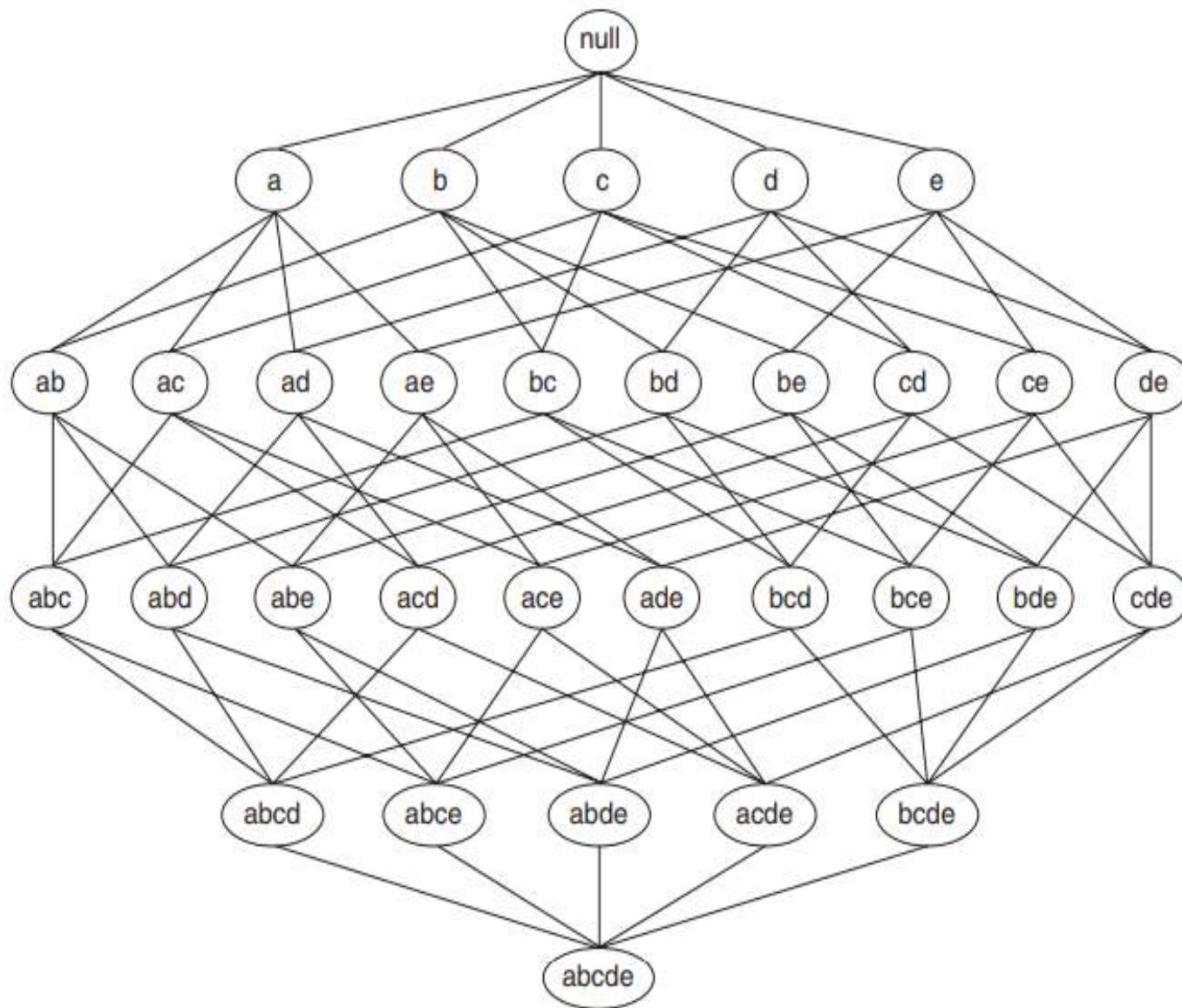
Anti-monotone : $\forall X, Y \in J : (X \subseteq Y) \rightarrow f(Y) \leq f(X)$
where : $J = 2^I$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Computational Complexity



Computational Complexity



N – Total number of Transactions

M – Total number of itemsets / candidates

W – Width of the Transactions

$$M = 2^d - 1$$

Computational Complexity

Strategies to reduce “N” & “W”

#1: Use appropriate data organization to reduce the I/O cost during candidate's support count computation.

Horizontal Data Layout

TID	Items
1	a,b,e
2	b,c,d
3	c,e
4	a,c,d
5	a,b,c,d
6	a,e
7	a,b
8	a,b,c
9	a,c,d
10	b

Vertical Data Layout

a	b	c	d	e
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

Order ID	Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low
101	1	0	0	1	1	1	0	0	0	0	0	1
104	1	0	0	1	0	1	1	1	1	0	1	0
106	0	1	0	1	0	1	1	1	0	1	0	0

Computational Complexity

Strategies to reduce “N” & “W”

#1: Use appropriate data organization to reduce the I/O cost during candidate's support count computation.

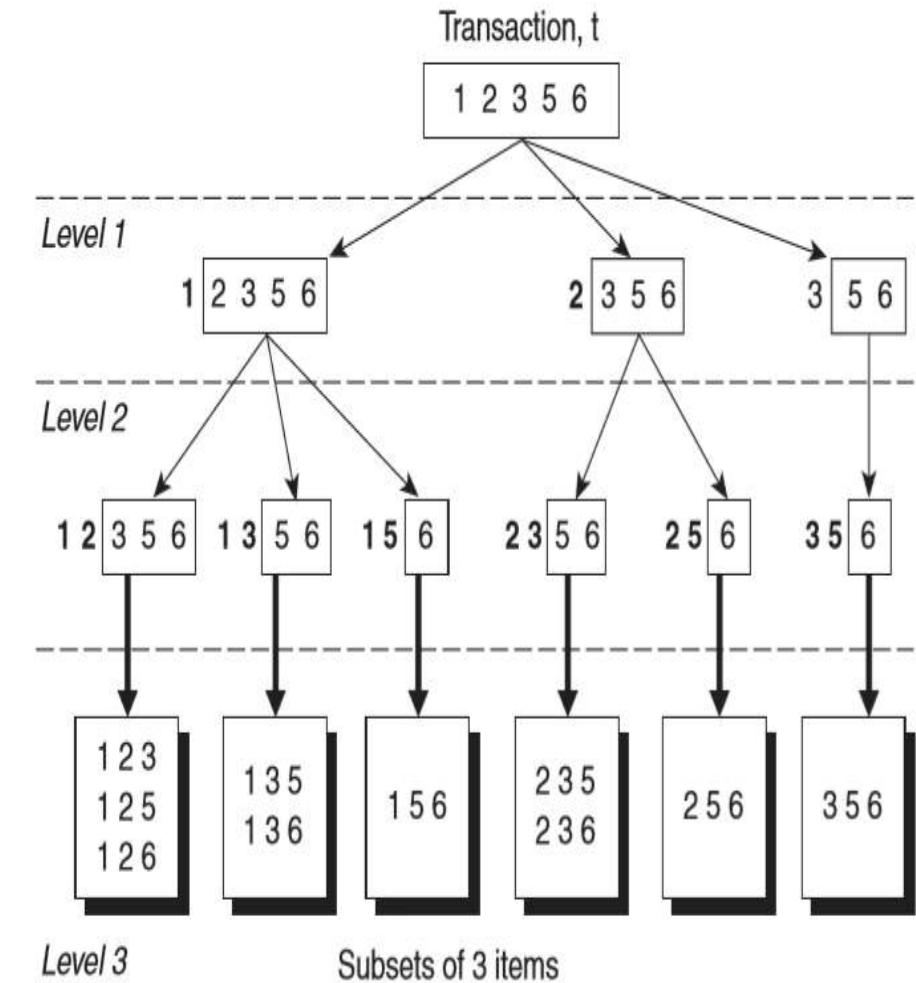
Order ID	Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low
101	1	0	0	1	1	1	0	0	0	0	0	1
104	1	0	0	1	0	1	1	1	1	0	1	0
106	0	1	0	1	0	1	1	1	0	1	0	0
Customer Age = Adult	Customer Age = Senior	Customer Age = Young	Milk	Bread	Vegetable	Cleaner	Medicine	KitchenStaples	Purchase LEVEL = High	Purchase LEVEL = Medium	Purchase LEVEL = Low	
	101	106		101	101	101	104	104	104	106	104	101
	104			104		104	106	106				
				106		106						

Computational Complexity

Strategies to reduce “N” & “W”

#1: Use appropriate data organization to reduce the I/O cost during candidates support count computation.

#2: Use systematic enumeration of itemsets to reduce repeated iteration of same transaction to generate k-itemset



Computational Complexity

Strategies to reduce “N” & “W”

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Code	Items
1	Bread
2	Butter
3	Coke
4	Diaper
5	Eggs
6	Milk

Computational Complexity

Strategies to reduce “N” & “W”

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Code	Items
1	Bread
2	Butter
3	Coke
4	Diaper
5	Eggs
6	Milk

↓

TID	Items
1	1,6
2	1,4,2,5
3	6,4,2,3
4	1,6,4,2
5	1,6,4,3

→

TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6

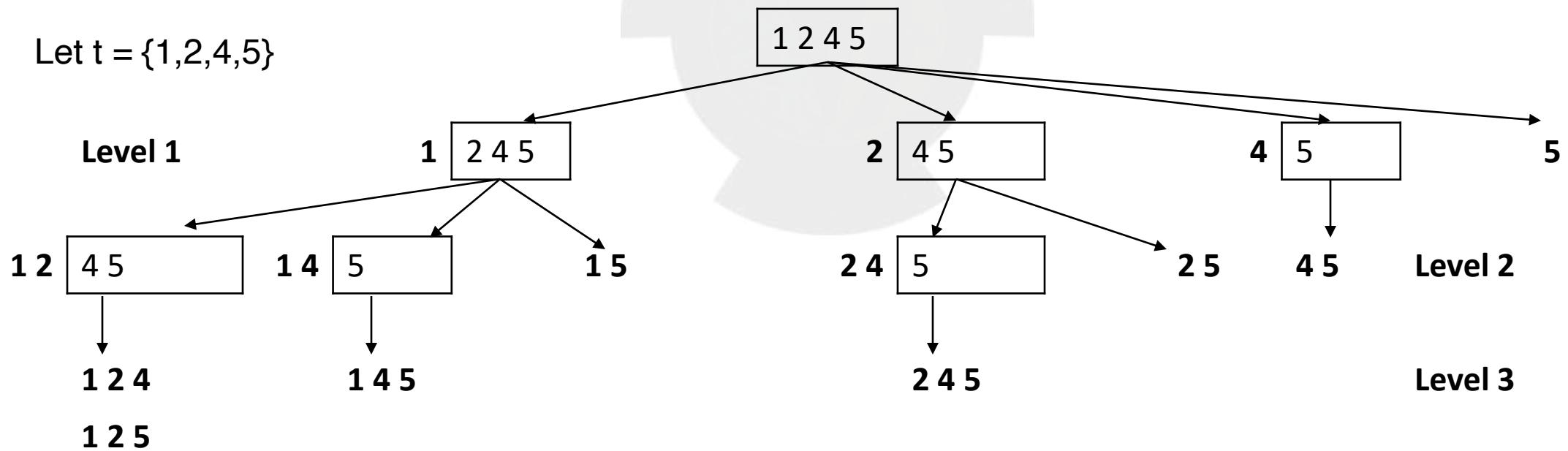
Computational Complexity

Strategies to reduce “N” & “W”

TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Let $t = \{1,2,4,5\}$



Computational Complexity

Strategies to reduce “N*M” Comparisons

Idea: Use the Hash tree structure to efficiently store candidate itemsets M. This reduces the comparisons between every itemset per N transaction with M candidate itemsets.

- Candidate itemsets (with ordered items) are stored in a hash-tree
- Leaf node of hash-tree contains a list of itemsets and counts
- Interior node contains a hash table
- All the candidates contained in a transaction are generated by subset function
- Every subset is mapped against the hash tree and support count is updated or new node is added

Computational Complexity

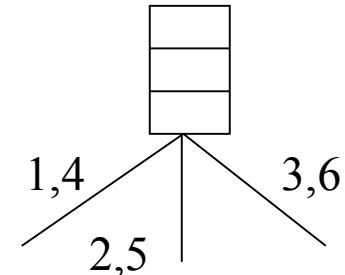
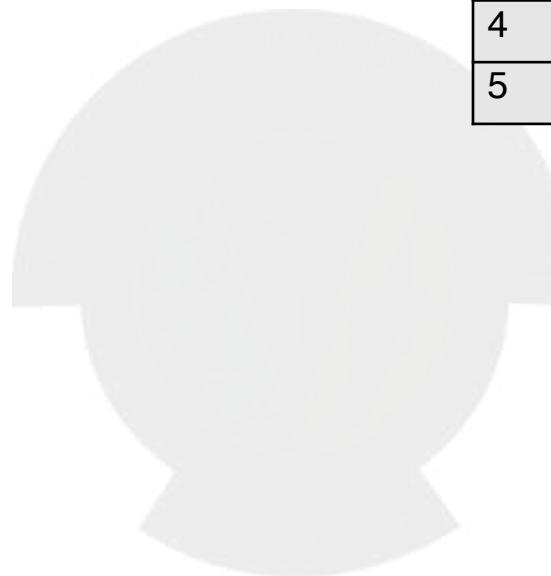
Strategies to reduce “N*M” Comparisons

- Candidate itemsets (with ordered items) are stored in a hash-tree

Example: $h = k \bmod 3$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

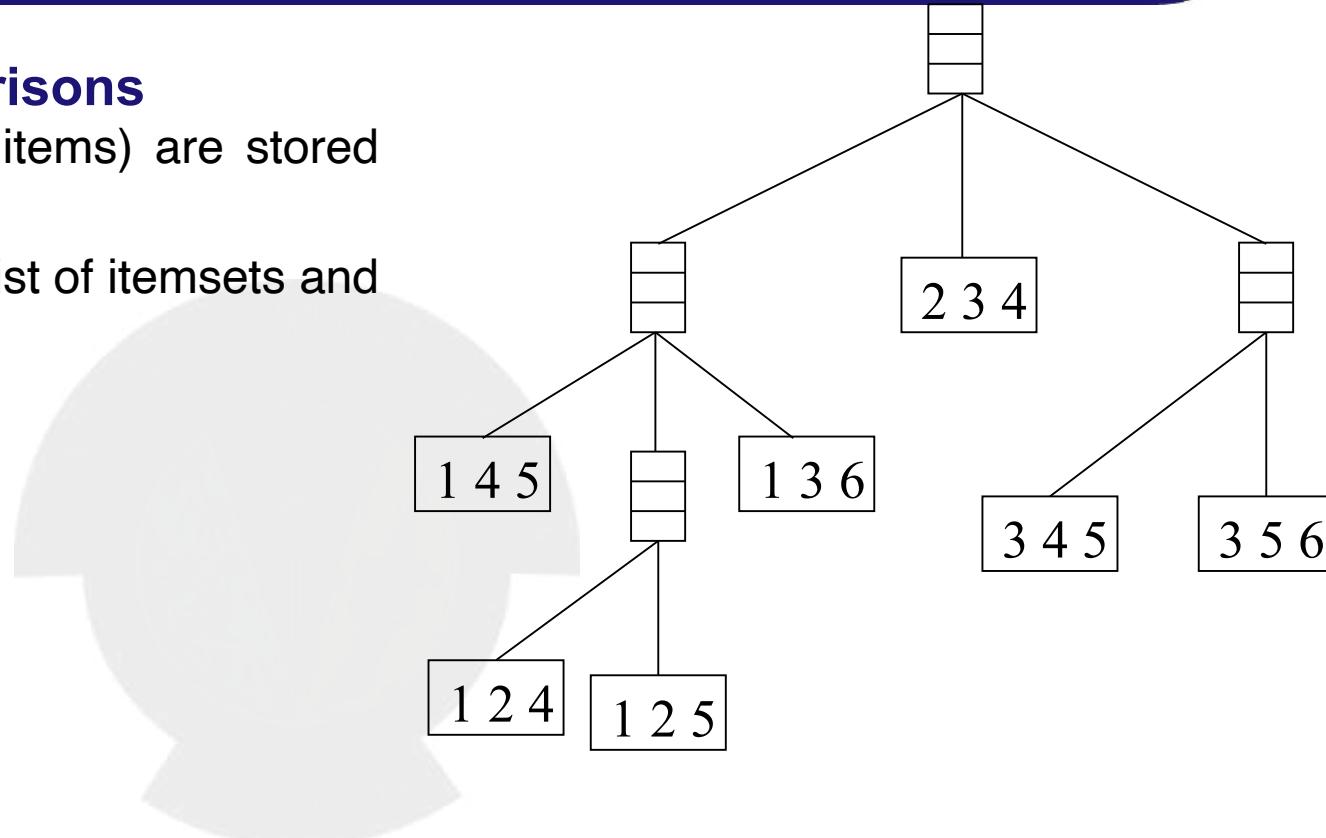
Code	Items
1	Bread
2	Butter
3	Coke
4	Diaper
5	Eggs
6	Milk



Computational Complexity

Strategies to reduce “N*M” Comparisons

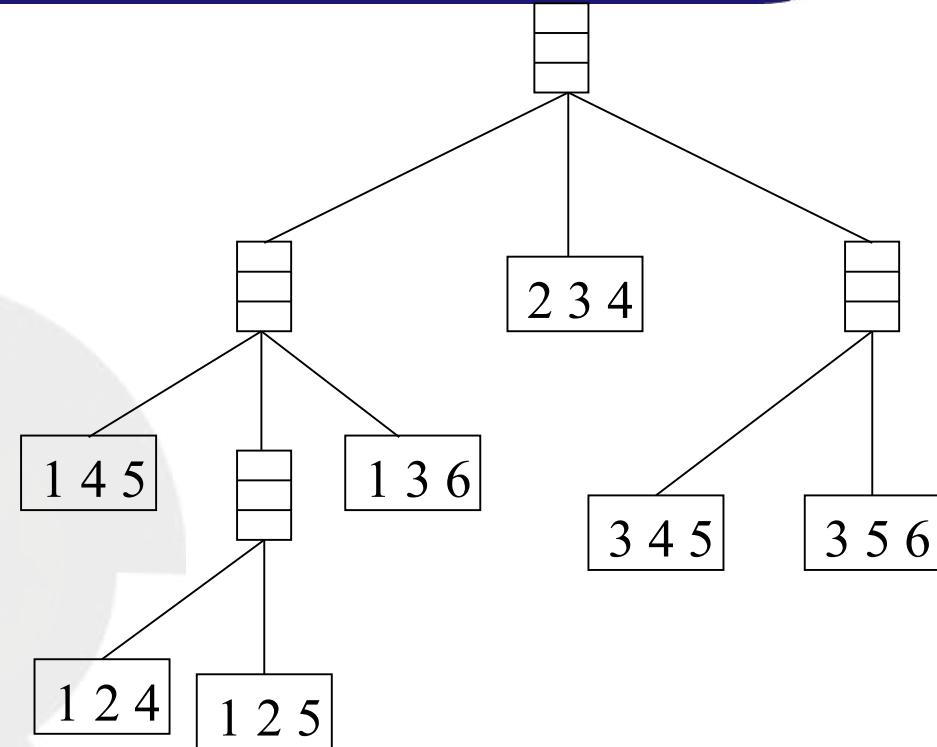
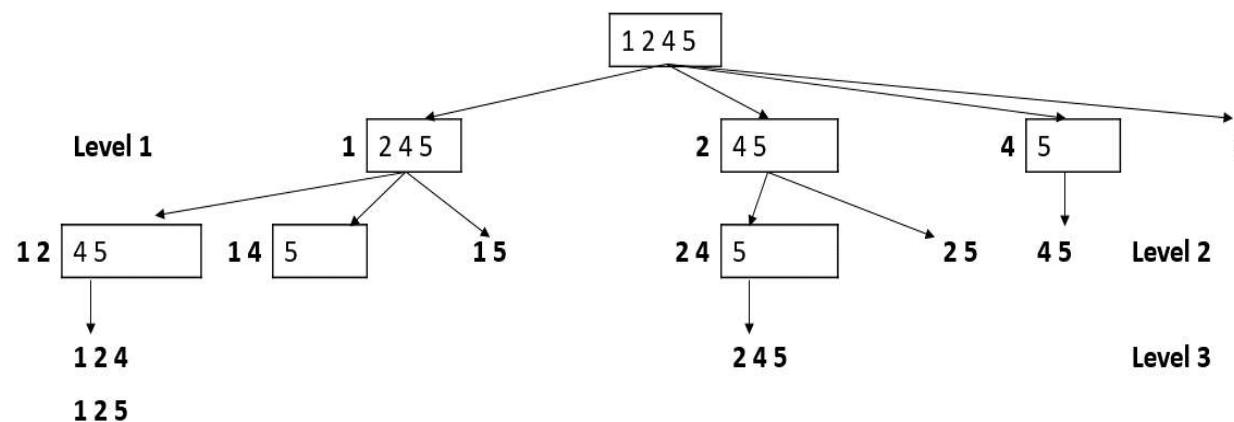
- Candidate itemsets (with ordered items) are stored in a hash-tree
- Leaf node of hash-tree contains a list of itemsets and counts
- Interior node contains a hash table



Computational Complexity

Strategies to reduce “N*M” Comparisons

- Candidate itemsets (with ordered items) are stored in a hash-tree
- Leaf node of hash-tree contains a list of itemsets and counts
- Interior node contains a hash table
- All the candidates contained in a transaction are generated by subset function
- Every subset is mapped against the hash tree and support count is updated or new node is added

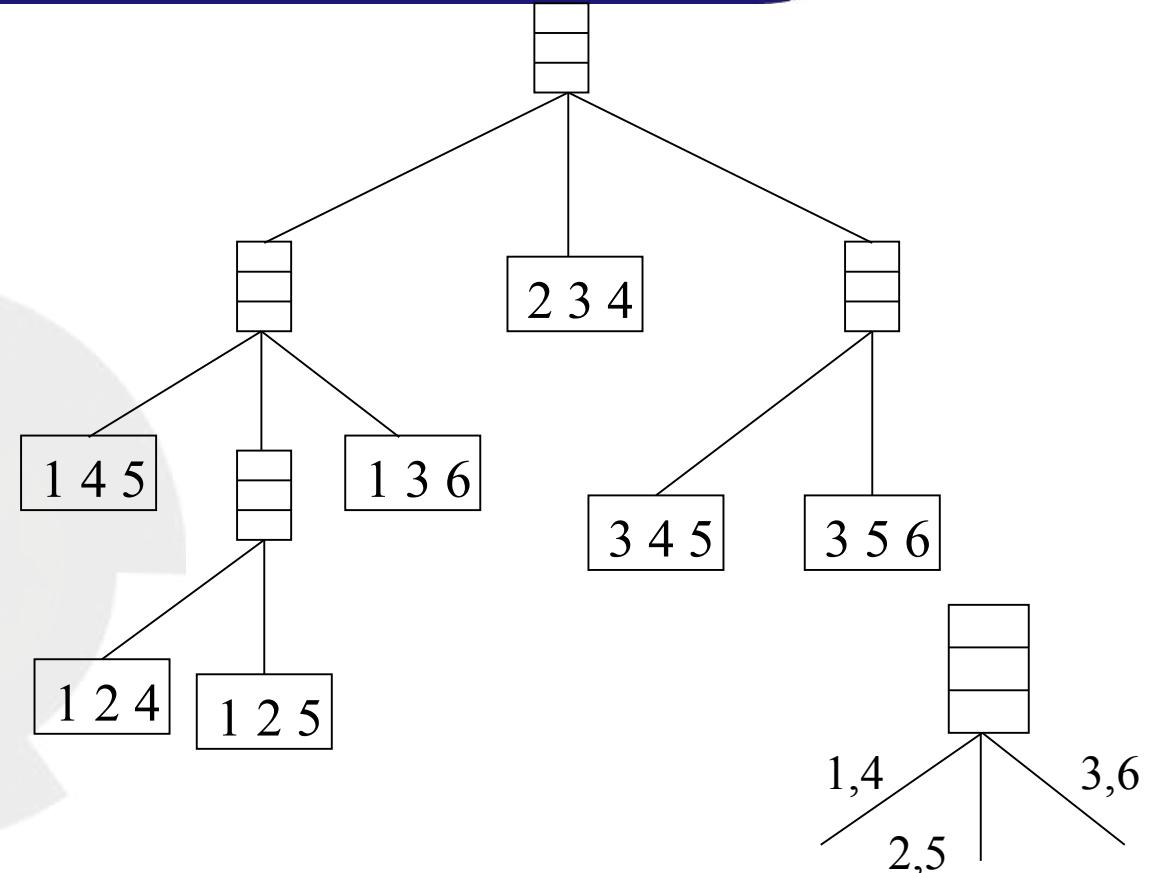
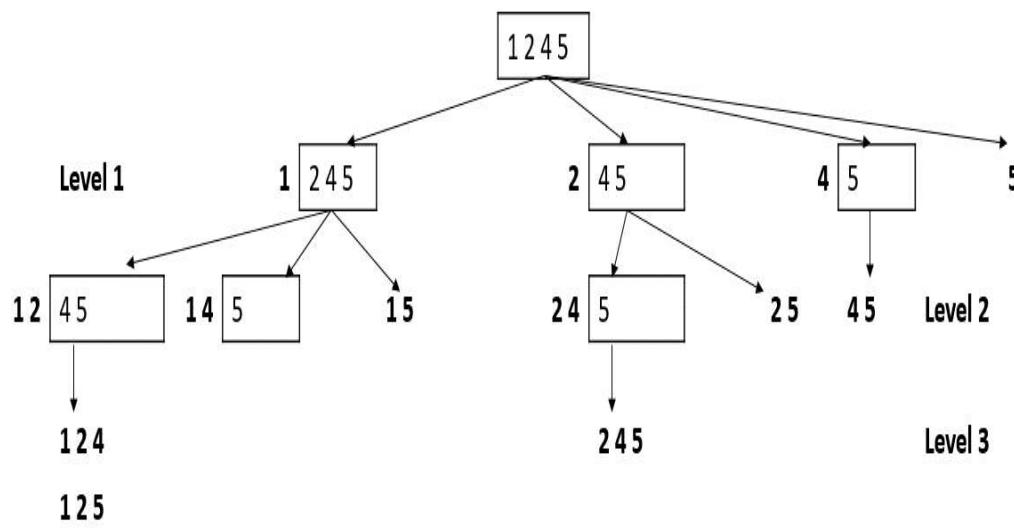


TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6

Computational Complexity

Strategies to reduce “N*M” Comparisons

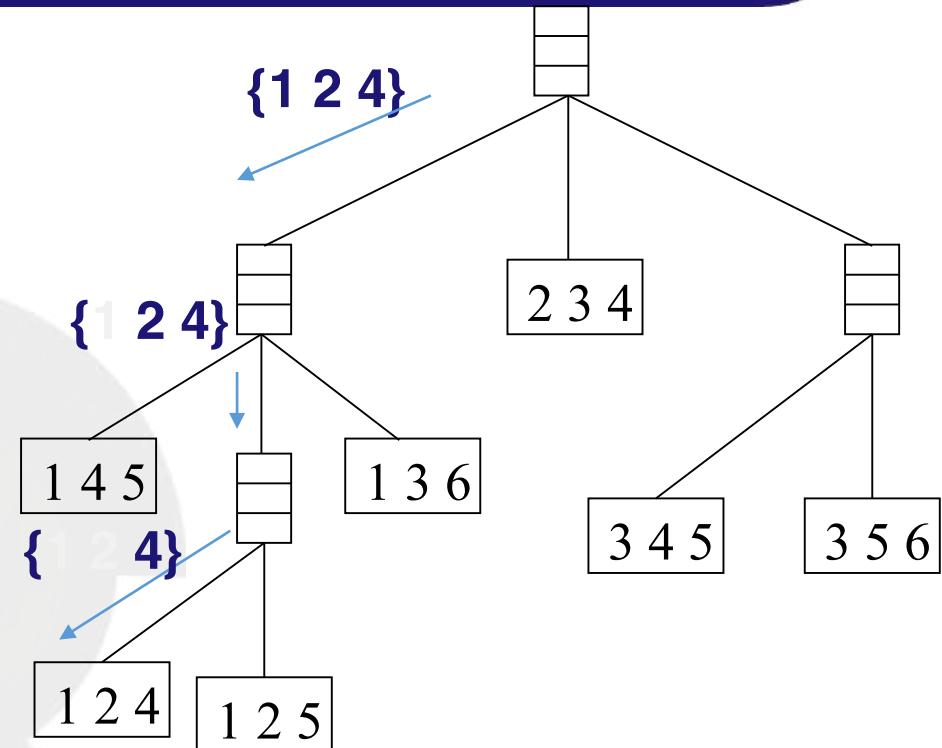
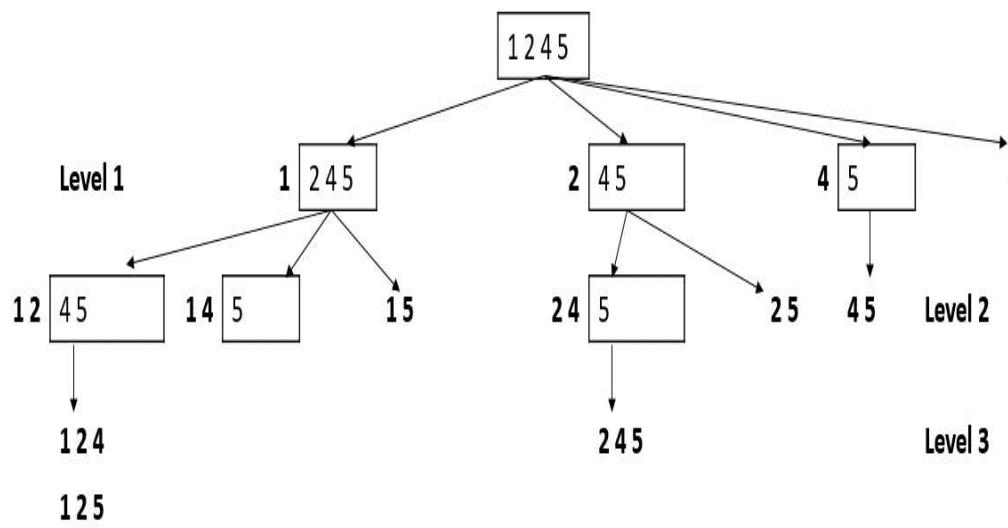
TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6



Computational Complexity

Strategies to reduce “N*M” Comparisons

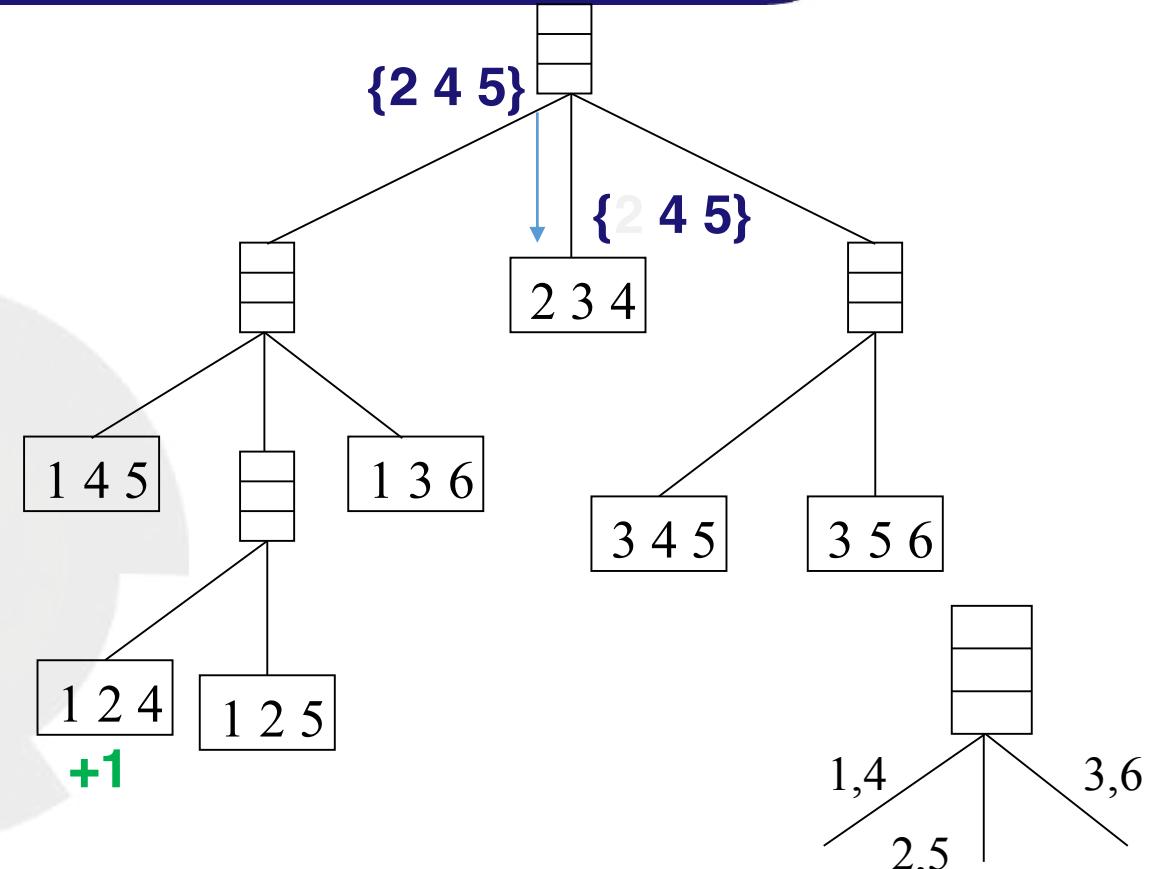
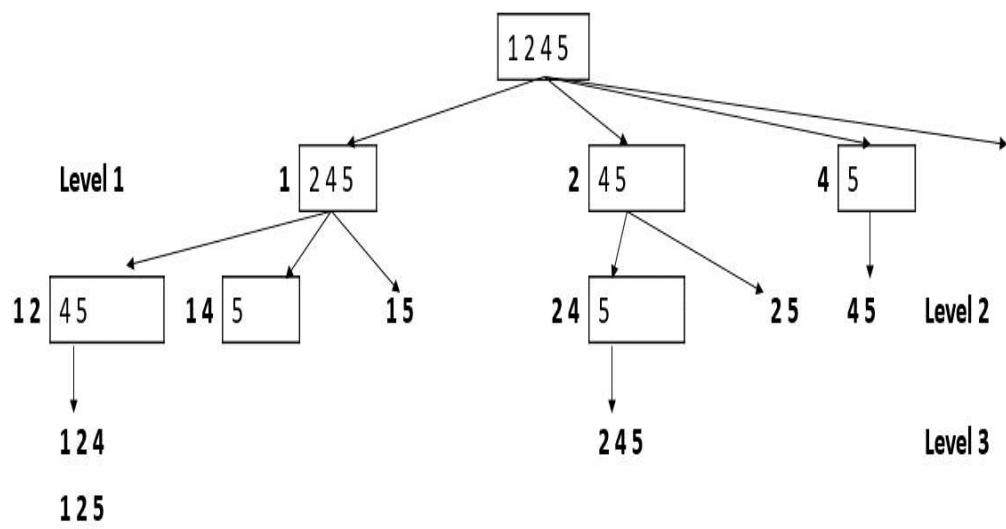
TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6



Computational Complexity

Strategies to reduce “N*M” Comparisons

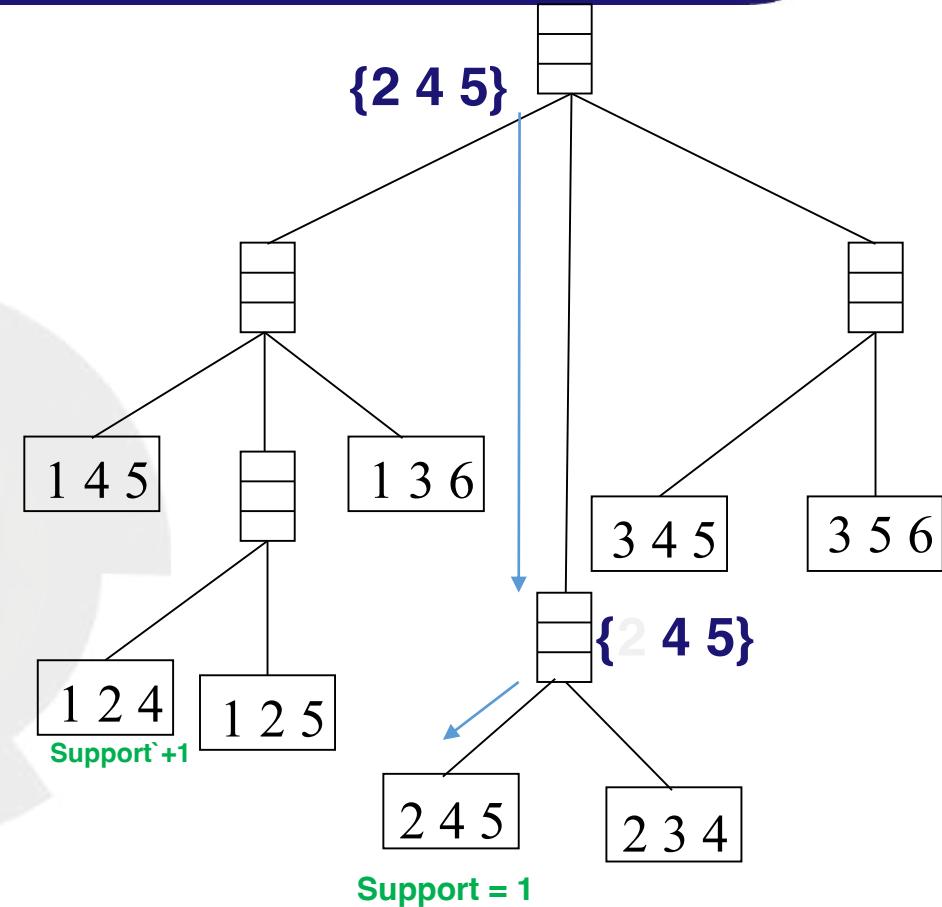
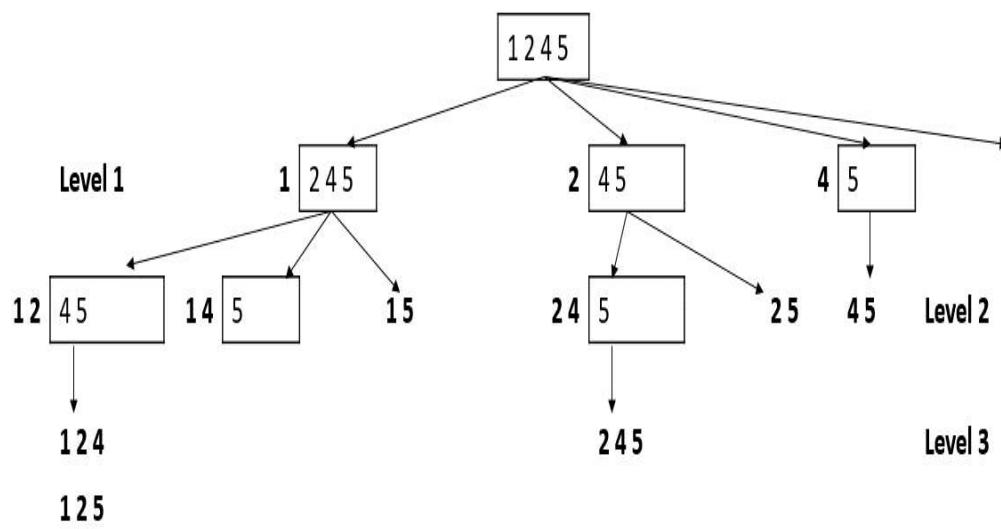
TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6



Computational Complexity

Strategies to reduce “N*M” Comparisons

TID	Items
1	1,6
2	1,2,4,5
3	2,3,4,6
4	1,2,4,6
5	1,3,4,6





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Association Rule Mining Algorithm

Raja Vadhana P

In this Segment

Association Rule Mining

- Apriori Algorithm
- FP Tree Growth Algorithm



Association Mining Algorithm

Association Rule Mining

Steps

1. Frequent Item set Generation
2. Strong Rules Generation

$$T = \{t_1, t_2, t_3, \dots, t_N\}$$

$$I = \{i_1, i_2, i_3, \dots, i_d\}$$

$X \rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, $X \neq \emptyset$, $Y \neq \emptyset$, $X \cap Y = \emptyset$.

$\text{Support}(X \rightarrow Y) \geq \text{minsup}$

$\text{Confidence}(X \rightarrow Y) \geq \text{minconf}$



TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

minsup:

Support threshold= 60%

minconf:

Confidence threshold= 75%



Apriori Algorithm

Frequent Itemset Generation

Apriori algorithm

Generate Candidate Itemset:

```
K=1  
 $F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$   
repeat  
     $k=k+1$   
     $C_k = \text{apriori-gen } (F_{k-1})$   
    for each transaction  $t \in T$  do  
         $C_t = \text{subset}(C_k, t)$   
        for each candidate itemset  $c \in C_t$  do  
             $\sigma(c) = \sigma(c) + 1$   
        end for  
    end for
```

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

1- Itemset

Item	Count
Bread	4
Coke	2
Milk	4
Butter	3
Diaper	4
Eggs	1

Frequent Itemset = {Bread, Milk, Butter, Diaper}

Assume Minimum Support count = 3

or support = 0.6

C_k -Set of frequent k-itemsets

F_k -Set of frequent k-itemsets

➤ **Candidate Generation : Self Join the itemsets of F_{k-1}**

➤ **Support Based Pruning :**

Apriori Pruning Principle:Supersets of infrequent itemsets are infrequent

Apriori algorithm

Extract Frequent K-Itemset

```

 $K=1$ 
 $F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$ 
repeat
   $k=k+1$ 
   $C_k = \text{apriori-gen } (F_{k-1})$ 
  for each transaction  $t \in T$  do
     $C_t = \text{subset}(C_k, t)$ 
    for each candidate itemset  $c \in C_t$  do
       $\sigma(c) = \sigma(c) + 1$ 
    end for
  end for
   $F_k = \{c | c \in C_k \wedge \sigma(\{c\}) \geq N * \text{minsup}\}$ 
Until  $F_k = \emptyset$ 
Result = Union all  $F_k$ 

```

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Item	Count
Bread	4
Coke	2
Milk	4
Butter	3
Diaper	4
Eggs	1

1- Itemset

Frequent Itemset = {Bread, Milk, Butter, Diaper}

Assume Minimum Support count = 3
or support = 0.6

Itemset	Count

Itemset	Count
{Bread,Milk}	3
{Bread,Butter}	2
{Bread,Diaper}	3
{Milk,Butter}	2
{Milk,Diaper}	3
{Butter,Diaper}	3

2 - Itemset

Frequent Itemset = {
 {Bread, Milk} ,
 {Bread, Diaper} ,
 {Milk, Diaper} ,
 {Butter, Diaper}}

3- Itemset

Apriori algorithm

Frequent Item set generation

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

repeat

$k=k+1$

$C_k = \text{apriori-gen } (F_{k-1})$

for each transaction $t \in T$ do

$C_t = \text{subset}(C_k, t)$

for each candidate itemset $c \in C_t$

do

$\sigma(c) = \sigma(c) + 1$

end for

end for

$F_k = \{c | c \in C_k \wedge \sigma(\{i\}) \geq N * \text{minsup}$

}

Until $F_k = \emptyset$

Result = Union all F_k

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

1- Itemset

Item	Count
Bread	4
Coke	2
Milk	4
Butter	3
Diaper	4
Eggs	1

Frequent Itemset = {Bread, Milk, Butter, Diaper}

Assume Minimum Support count = 3

or support = 0.6

Itemset	Count
{Bread, Milk, Diaper}	2

3- Itemset

Itemset	Count
{Bread,Milk}	3
{Bread,Butter}	2
{Bread,Diaper}	3
{Milk,Butter}	2
{Milk,Diaper}	3
{Butter,Diaper}	3

$F_3 = \emptyset$



FP Growth Algorithm

Frequent Itemset Generation

FP Tree Growth

FP Tree Construction

$K=1$

$$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$$

$$F_k = \text{sort_desc}(F_k)$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

ItemID	Count
Bread	4
Coke	2
Milk	4
Butter	3
Diaper	4
Eggs	1

Assume Minimum Support count = 3
or support = 0.6

Sorted Frequent List : **F-List:**

FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N^* \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Sorted Frequent List : **F-List:**
 $\{Bread, Diaper, Milk, Butter\}$

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

Assume Minimum Support count = 3
or support = 0.6

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree(null(item_name= null, support_count=0), T)

for each transaction $t \in T$ **do**

$C_t = [p|P]$

repeat

 insert_tree([p|P], T)

 update node link

Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

Sorted Frequent List : F-List:
 $\{Bread, Diaper, Milk, Butter\}$

{null}
Count =

{Bread}
Count =

{Milk}
Count =

FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree(null(item_name= null, support_count=0), T)

for each transaction $t \in T$ **do**

$C_t = [p|P]$

repeat

 insert_tree([p|P], T)

 update node link

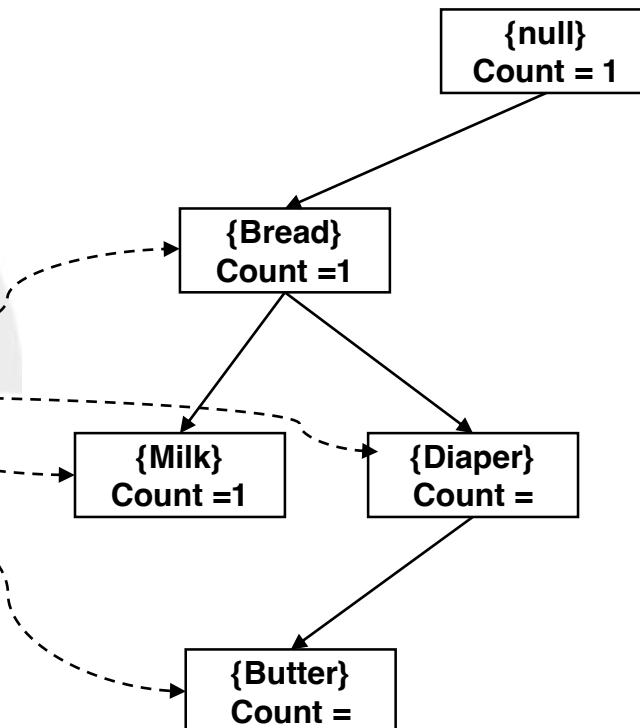
Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

Sorted Frequent List : F-List:
 $\{Bread, Diaper, Milk, Butter\}$



FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree(null(item_name= null, support_count=0), T)

for each transaction $t \in T$ **do**

$C_t = [p|P]$

repeat

 insert_tree([p|P], T)

 update node link

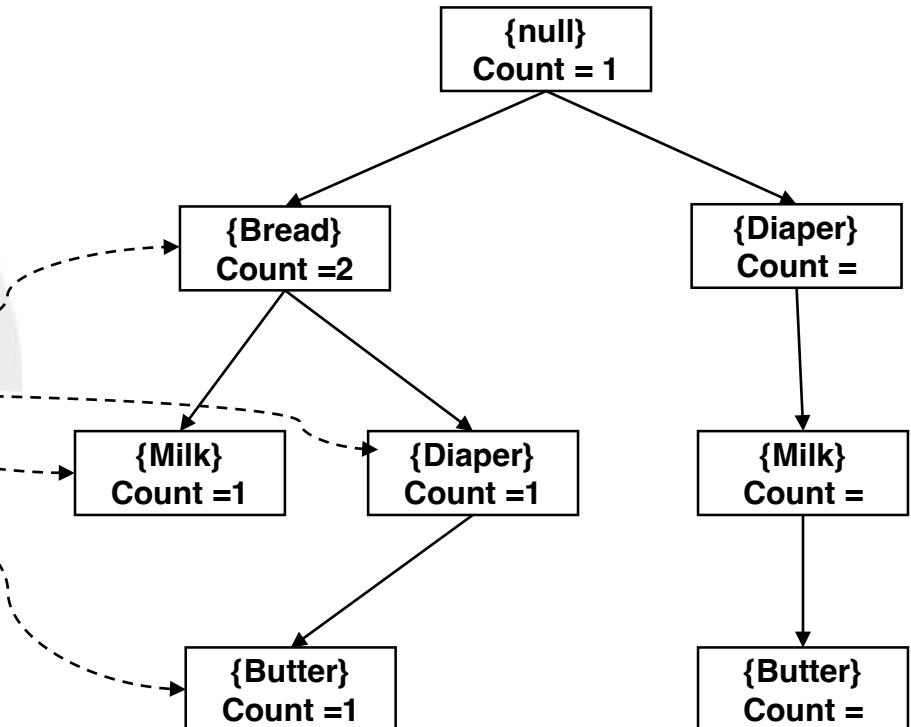
Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

Sorted Frequent List : F-List:
 $\{Bread, Diaper, Milk, Butter\}$



FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree(null(item_name= null, support_count=0), T)

for each transaction $t \in T$ **do**

$C_t = [p|P]$

repeat

 insert_tree([p|P], T)

 update node link

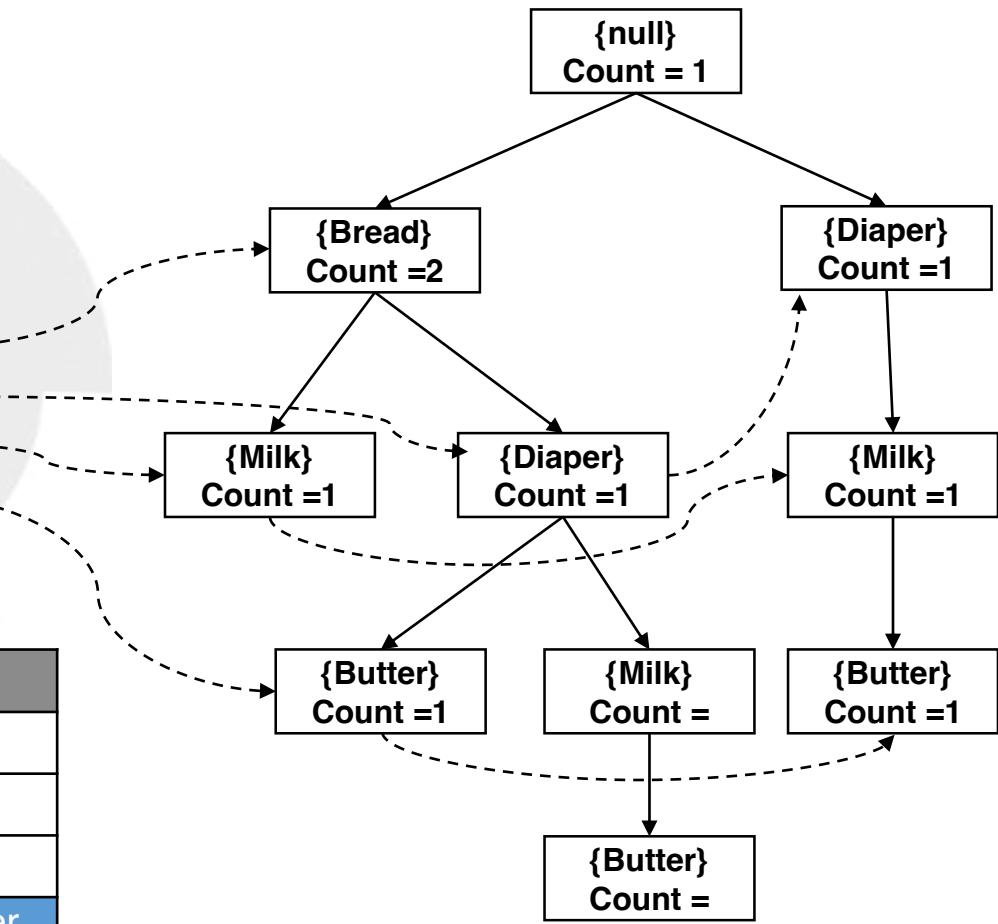
Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

*Sorted Frequent List : F-List:
 {Bread, Diaper, Milk, Butter}*



FP Tree Growth

FP Tree Construction

$K=1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$

for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree(null(item_name= null, support_count=0), T)

for each transaction $t \in T$ **do**

$C_t = [p|P]$

repeat

 insert_tree([p|P], T)

 update node link

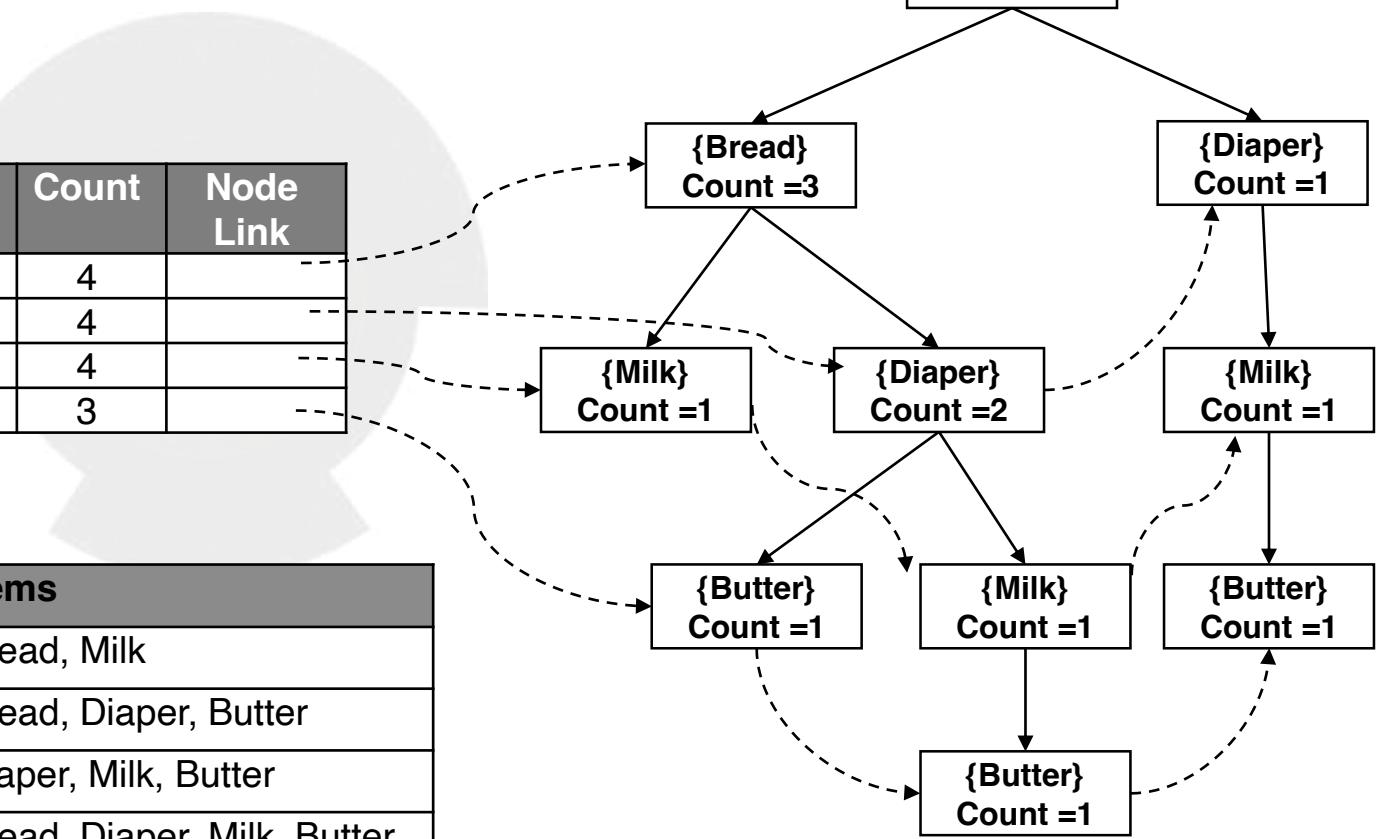
Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

*Sorted Frequent List : F-List:
 {Bread, Diaper, Milk, Butter}*



FP Tree Growth

FP Tree Construction

$K=1$
 $F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N * \text{minsup}\}$

$F_k = \text{sort_desc}(F_k)$
for each transaction $t \in T$ **do**

$C_t = \text{sort_desc}(C_t, F_k)$

end for

Create FP_tree($\text{null}(\text{item_name}= \text{null}, \text{support_count}=0)$, T)

for each transaction $t \in T$ **do**
 $C_t = [p|P]$
repeat

insert_tree($[p|P]$, T)

update node link

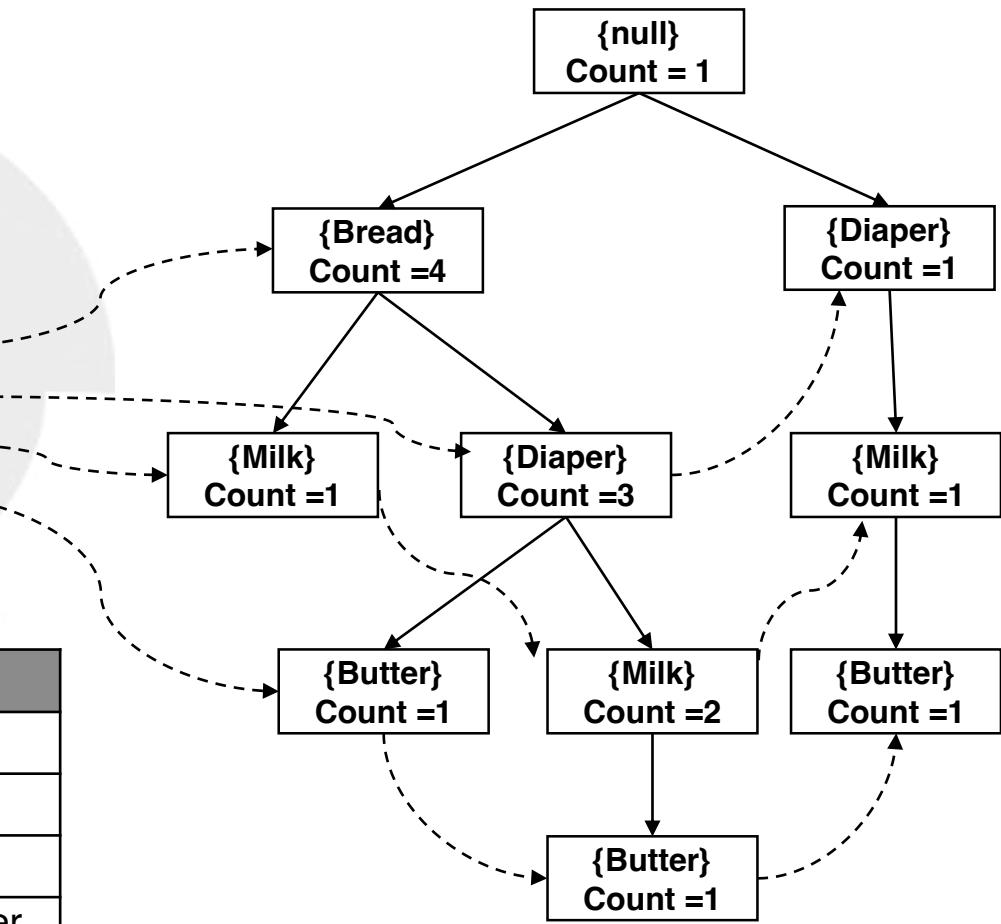
Until $P = \emptyset$

end for

ItemID	Count	Node Link
Bread	4	
Diaper	4	
Milk	4	
Butter	3	

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter
3	Diaper, Milk, Butter
4	Bread, Diaper, Milk, Butter
5	Bread, Diaper, Milk

*Sorted Frequent List : F-List:
 $\{Bread, Diaper, Milk, Butter\}$*



FP Tree Growth

FP Tree Growth – Pattern Mining

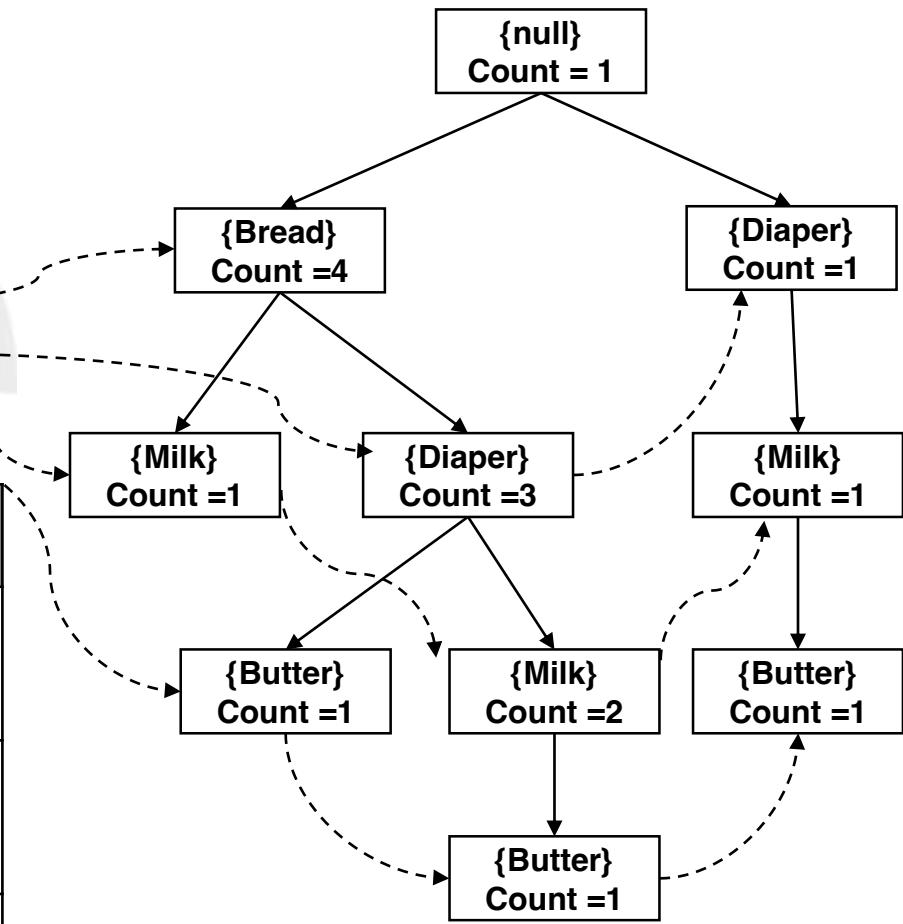
{I4, I3, I2, I1, {I2,I4}, {I1,I3}, {I2,I3}, {I1,I2}}

```
{
  {Butter}, {Milk}, {Diaper}, {Bread},
  {Diaper,Butter}, {Bread,Milk},
  {Diaper,Milk}, {Bread,Diaper}
}
```

Item	Conditional Pattern Base	Conditional FP Tree	Frequent Patterns Generated
I4	{I1, I2 : 1} {I1, I2, I3 : 1} {I2, I3 : 1}	{I2 : 3}	{I2, I4 : 3}
I3	{I1 : 1} {I1, I2 : 2} {I2 : 1}	{I1 : 3} {I2 : 3}	{I1, I3 : 3} {I2, I3 : 3}
I2	{I1 : 3}	{I1 : 3}	{I1, I2 : 3}
I1	-	-	-

ItemID	Count	Node Link
Bread : I1	4	-
Diaper : I2	4	-
Milk : I3	4	-
Butter: I4	3	-

Sorted Frequent List : F-List:
{Bread, Diaper, Milk, Butter}



Choice of Algorithm

Frequent Item set generation

Apriori

- Level wise approach similar to breadth first search technique
- Item set oriented with expected higher data ingestion rate
- Helps in the absence of item set ordering

Eg., Document similarity among k-item set matching

FP Growth

- Suffix based approach employing depth first technique
- Pattern Oriented with expected less dynamicity of the basket window
- Suitable in the presence of lexicographic item set ordering

Eg., DNA sequencing , Gene matching

In this Segment

Association Rule Mining

- Association Rule Generation

Association Rule Mining

Steps

1. Frequent Item set Generation
2. **Strong Rules Generation**

$$T = \{t_1, t_2, t_3, \dots, t_N\}$$

$$I = \{i_1, i_2, i_3, \dots, i_d\}$$

$X \rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, $X \neq \emptyset$, $Y \neq \emptyset$, $X \neq Y$, \emptyset .

$\text{Support}(X \rightarrow Y) \geq \text{minsup}$

$\text{Confidence}(X \rightarrow Y) \geq \text{minconf}$



TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

minsup:

Support threshold= 60%

minconf:

Confidence threshold= 75%



Apriori Algorithm

Association Rule Generation

Apriori algorithm

Rule Generation

Rule Generation is a process of generating high confidence rules from each frequent item set, where each rule is a binary partitioning of a frequent item set

Eg., If {A,B,C,D} is a frequent 4-itemset, candidate rules are given by:

$$\begin{array}{lll} ABC \rightarrow D, & AB \rightarrow CD, & A \rightarrow BCD, \\ ABD \rightarrow C, & AC \rightarrow BD, & B \rightarrow ACD, \\ ACD \rightarrow B, & AD \rightarrow BC, & C \rightarrow ABD, \\ BCD \rightarrow A, & BC \rightarrow AD, & D \rightarrow ABC, \\ & BD \rightarrow AC, & \\ & CD \rightarrow AB, & \end{array}$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke , Bread
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

If |ItemSets| = k, then there are $2^k - 2$ candidate association rules

Apriori algorithm

Rule Generation

- Assume Minimum Confidence = 75% and Support =0.4

Itemset generation by Apriori:

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{6}{4}$$

Without support based pruning

$$= 6+15+20+15 = 56$$

$$\binom{6}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4}$$

With support based pruning

$$= 6+10+10+5 = 31$$

	Itemset\	Count
1	{Bread}	5
2	{Coke}	2
3	{Milk}	4
4	{Butter}	3
5	{Diaper}	4
6	{Bread,Coke}	2
7	{Bread,Milk}	4
8	{Bread,Butter}	3
9	{Bread,Diaper}	4
10	{Coke,Milk}	2
11	{Coke,Diaper}	2
12	{Milk,Butter}	2
13	{Milk,Diaper}	3
14	{Butter,Diaper}	3

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke , Bread
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

	Itemset\	Count
15	{Bread,Coke,Milk}	2
16	{Bread,Coke,Diaper}	2
17	{Bread,Milk,Butter}	2
18	{Bread,Milk,Diaper}	3
19	{Bread,Butter,Diaper}	3
20	{Coke,Milk,Diaper}	2
21	{Milk,Butter,Diaper}	2
22	{Bread,Diaper,Butter,Milk}	3
23	{Bread,Milk,Diaper,Coke}	2

Apriori algorithm

Rule Generation

- Assume Minimum Confidence = 75% and Support = 0.4
1. Generate the rule by binary partition
 2. Compute Confidence
 3. Prune rules using idea of monotone property applied to the rule antecedent
 4. Eliminate weak rules

Itemset\	Count
{Bread}	5
{Coke}	2
{Milk}	4
{Butter}	3
{Diaper}	4
{Bread,Coke}	2
{Bread,Milk}	4
{Bread,Butter}	3
{Bread,Diaper}	4
{Coke,Milk}	2
{Coke,Diaper}	2
{Milk,Butter}	2
{Milk,Diaper}	3
{Butter,Diaper}	3
{Bread,Milk,Diaper}	3

Apriori algorithm

Rule Generation *Minimum Confidence = 75% and Support = 0.4*

Itemset	Count
{Bread, Milk, Diaper}	3

Candidate Rule	Conf
{Bread, Milk} -> {Diaper}	0.75
{Bread, Diaper} -> {Milk}	0.75
{Diaper, Milk} -> {Bread}	0.75

Candidate Rule	Conf
{Bread} -> {Milk, Diaper}	0.6
{Milk} -> {Bread, Diaper}	0.75

Candidate Rule	Conf
{Bread, Milk} -> {Diaper}	0.75
{Bread, Diaper} -> {Milk}	0.75

Candidate Rule	Conf
{Bread} -> {Milk, Diaper}	
{Diaper} -> {Bread, Milk}	0.75

Candidate Rule	Conf
{Bread, Milk} -> {Diaper}	0.75
{Bread, Diaper} -> {Milk}	0.75

Candidate Rule	Conf
{Diaper} -> {Milk, Bread}	
{Milk} -> {Bread, Diaper}	

Itemset\	Count
{Bread}	5
{Coke}	2
{Milk}	4
{Butter}	3
{Diaper}	4
{Bread, Coke}	2
{Bread, Milk}	4
{Bread, Butter}	3
{Bread, Diaper}	4
{Coke, Milk}	2
{Coke, Diaper}	2
{Milk, Butter}	2
{Milk, Diaper}	3
{Butter, Diaper}	3
{Bread, Milk, Diaper}	3

Apriori algorithm

Rule Generation

- Assume Minimum Confidence = 75% and Support =0.4

1. Generate the rule by binary partition
2. Compute Confidence
3. Prune rules using idea of monotone property applied to the rule antecedent
4. Eliminate weak rules

Confidence Based Pruning:

If a rule $X \rightarrow Y$ -X does not satisfy the confidence threshold , then any rule $X' \rightarrow Y$ - X' , where X' is a subset of X , must not satisfy the confidence threshold as well.

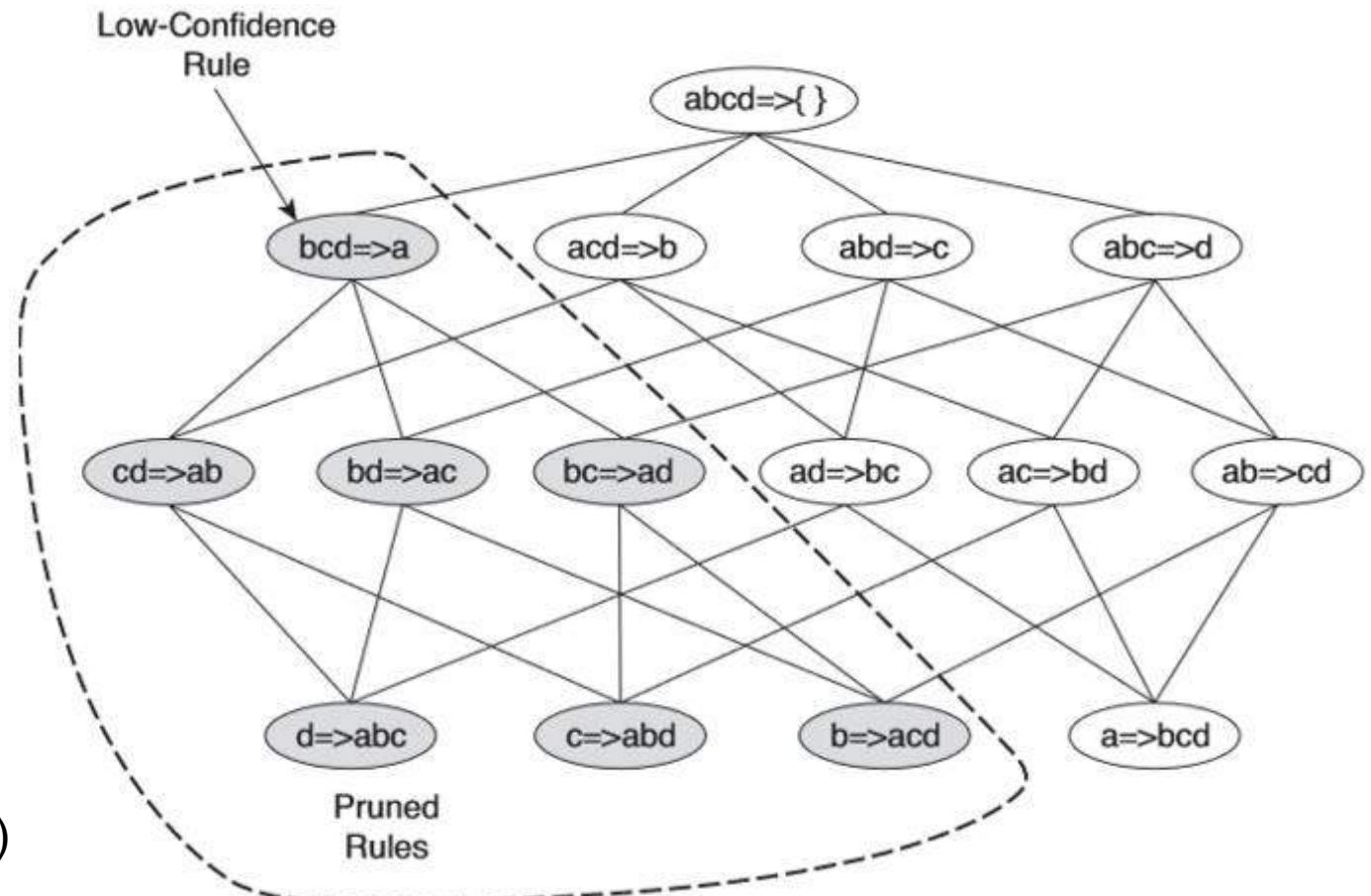
Itemset\	Count
{Bread}	5
{Coke}	2
{Milk}	4
{Butter}	3
{Diaper}	4
{Bread,Coke}	2
{Bread,Milk}	4
{Bread,Butter}	3
{Bread,Diaper}	4
{Coke,Milk}	2
{Coke,Diaper}	2
{Milk,Butter}	2
{Milk,Diaper}	3
{Butter,Diaper}	3
{Bread,Milk,Diaper}	3

Apriori algorithm

Rule Generation

Confidence Based Pruning:

If a rule $X \rightarrow Y - X$ does not satisfy the confidence threshold , then any rule $X' \rightarrow Y - X'$, where X' is a subset of X , must not satisfy the confidence threshold as well.



Frequent Itemsets = {A,B,C,D}:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

Apriori algorithm

Rule Generation

Minimum Confidence = 75% and Support = 0.4

Itemset	Count
{Coke,Milk,Diaper}	2

Candidate Rule	Conf
{Milk, Diaper} -> {Coke}	0.66
{Coke, Diaper} -> {Milk}	1
{Coke, Milk} -> {Diaper}	1

Candidate Rule	Conf
{Milk} -> {Coke, Diaper}	0.5
{Diaper} -> {Coke, Milk}	0.5

Confidence Based Pruning: If a rule $X \rightarrow Y$ does not satisfy the confidence threshold, then any rule $X' \rightarrow Y$ where X' is a subset of X , must not satisfy the confidence threshold as well.

Itemset\	Cnt
{Bread}	5
{Coke}	2
{Milk}	4
{Butter}	3
{Diaper}	4
{Bread,Coke}	2
{Bread,Milk}	4
{Bread,Butter}	3
{Bread,Diaper}	4
{Coke,Milk}	2
{Coke,Diaper}	2
{Milk,Butter}	2
{Milk,Diaper}	3
{Butter,Diaper}	3
{Bread,Coke,Milk}	2
{Bread,Coke,Diaper}	2
{Bread,Milk,Butter}	2
{Bread,Milk,Diaper}	3
{Bread,Butter,Diaper}	3
{Coke,Milk,Diaper}	2
{Milk,Butter,Diaper}	2
{Bread,Diaper,Butter,Milk}	3



Thank You!

In our next session:
Efficiency



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Efficiency of Association Mining Algorithm

Raja Vadhana P

In this Segment

Association Rule Mining

- Efficiency Issues
- Factors
- Maximal & Closed Item sets



Efficiency of Algorithm

Efficiency of algorithm

Influential Factors

- Configuration Parameters
- Scalability

Efficiency of algorithm

Influential Factors

- Configuration Parameters
- Scalability

Total no.of.Transactions considered = 15

	MinSup	Support
Milk	60%	75%
Oven	60%	1.6%

Scenario:

Food and Electronic products from same outlet.

Efficiency of algorithm

Influential Factors

- Configuration Parameters
- Scalability

Scenario:

Food and Electronic products from same outlet.

Workaround:

Item wise minimum support and confidence threshold.

Side effect:

Anti-monotone property is no longer useful.

Total no.of.Transactions considered = 15

	MinSup	Support
Milk	60%	50%
Oven	1%	1.6%
Milk&Oven	1%	5%

$$\{Milk\} + \{Oven\} \rightarrow \{Milk, Oven\}$$

$$\min(\text{MinSup}_{\text{Milk}}, \text{MinSup}_{\text{Oven}})$$

Efficiency of algorithm

Influential Factors

- Configuration Parameters
- **Scalability**

Scenario:

Computationally intensive application generating data streams

Workaround:

Sampling with relative support threshold

Efficiency at the cost of accuracy

Efficiency of algorithm

Influential Factors

- Configuration Parameters
- **Scalability**

Scenario:

Larger number (N) of transaction which are dense(W).

Workaround: *Divide & Conquer Strategy*

Partition into **n** non-overlapping transaction sets

Parallelize the local frequent item set search.

Local support count = size of n_i^* minsup

Filter item sets that are frequent in atleast one partition

Perform global search from local candidates

Total no.of.Transactions considered = **10**

T1,T2,.....T10

(*minSup = 0.5*)

T1,T4,T7,T10

$$0.5^*4 = 2$$

T2,T3,T5,T6,T8,T9

$$0.5^*6 = 3$$

Support count:

Sample candidates: {I4,I5, I8, I20-I25, I29}



Maximal & Closed Itemsets

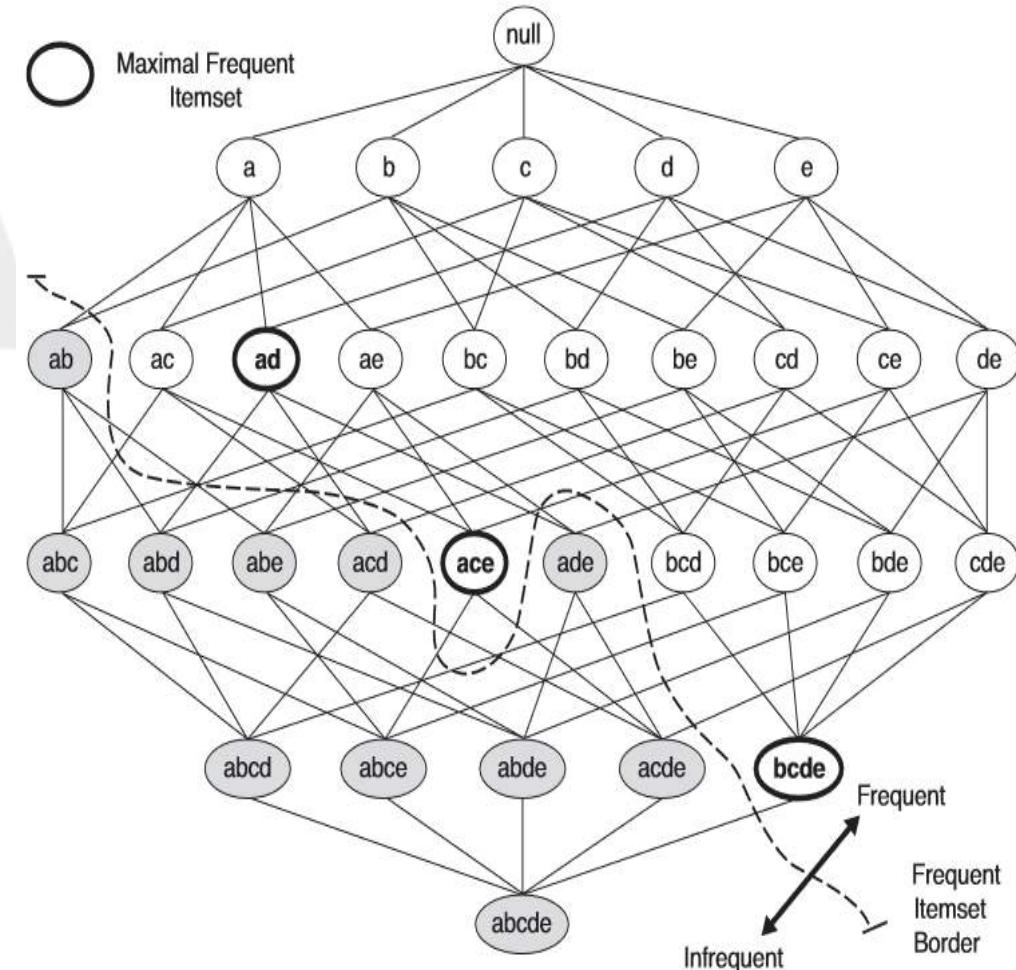
Computational Complexity

Maximal & Closed Frequent Item sets

Need: To identify a small representative set item set to form a compact representation.

Maximal Frequent Item sets:

An item set X is maximal frequent item set for which none of its immediate super sets are frequent



Computational Complexity

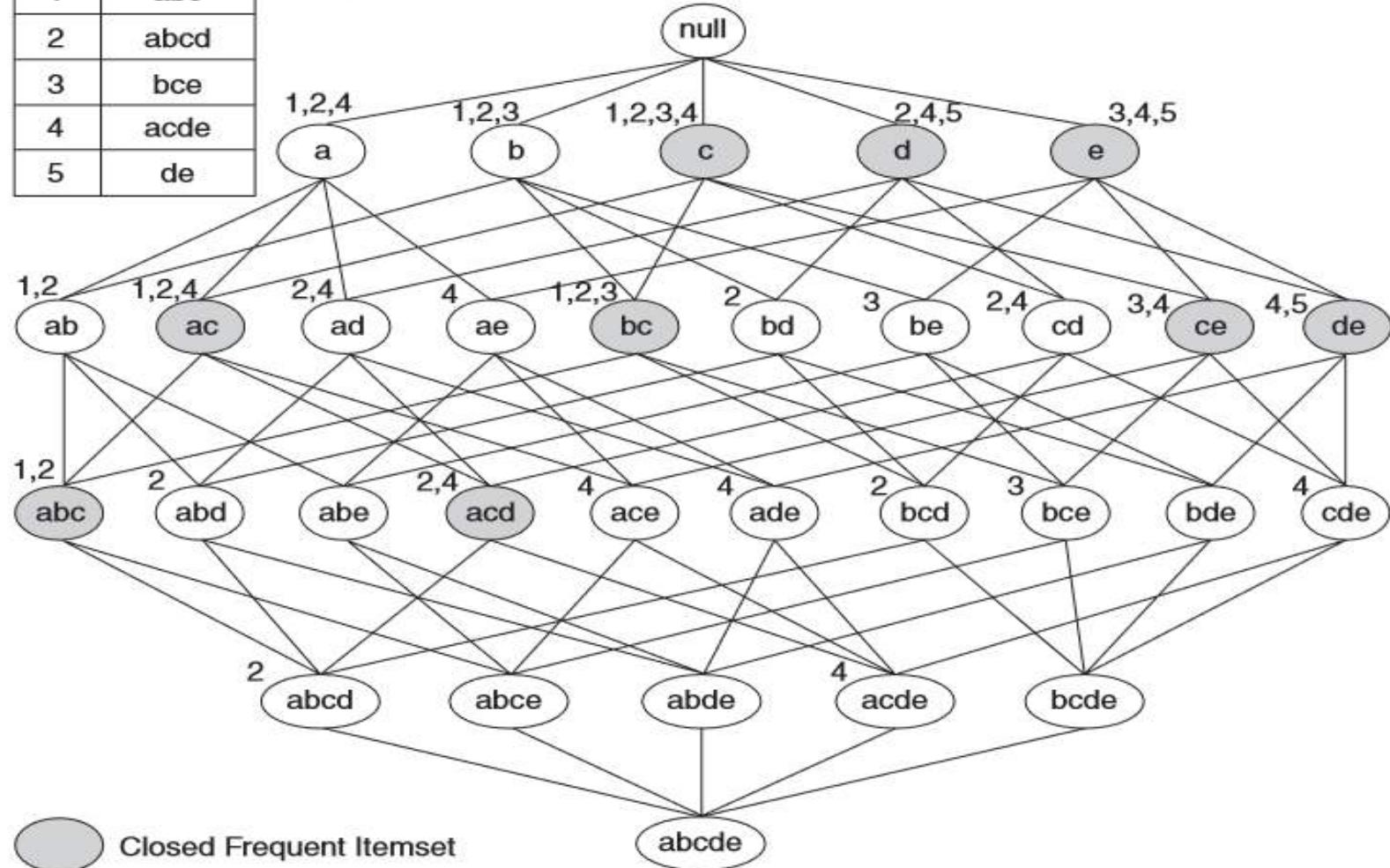
Maximal & Closed Frequent Item sets

Maximal Item sets =

$\{\{c,e\}, \{d,e\}, \{a,b,c\}, \{a,c,d\}\}$

TID	Items
1	abc
2	abcd
3	bce
4	acde
5	de

minsup = 40%



Computational Complexity

Maximal & Closed Frequent Item sets

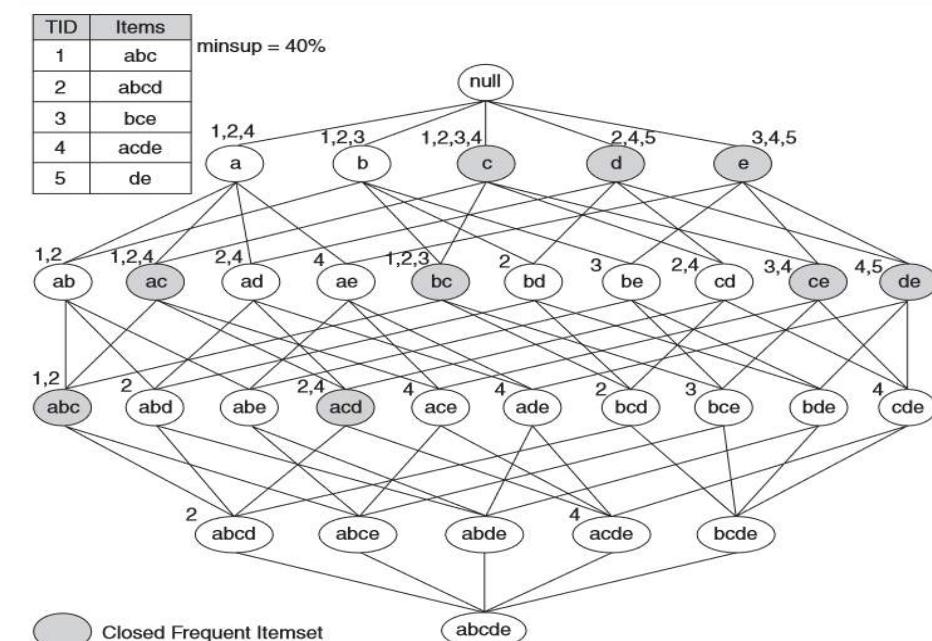
Closed Item sets:

An item set X is closed if none of its immediate supersets has exactly the same support count of X

Closed Frequent Item sets:

An item set X is closed frequent item set if its closed and its support is greater than or equal to *minsup*.

Significance of Closed Item set: Actual support information of {b,c} is lost in the absence of closed frequent itemset.





Thank You!

In our next session:
Evaluating interestingness of patterns



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Association Rule Mining

Evaluating interestingness of patterns

Raja Vadhana P

In this Segment

Association Rule Mining

- Evaluating interestingness of patterns



Pattern Evaluation

Pattern Evaluation

Confidence

Confidence determines how frequently items in Y appear in transactions that contain X

X → Y

{Bread} → {Milk}

Confidence Threshold = 60%

$$c(X \rightarrow Y) = \frac{(X \rightarrow Y)}{(X)} = \frac{(X \cap Y)}{(X)} = P(Y | X)$$

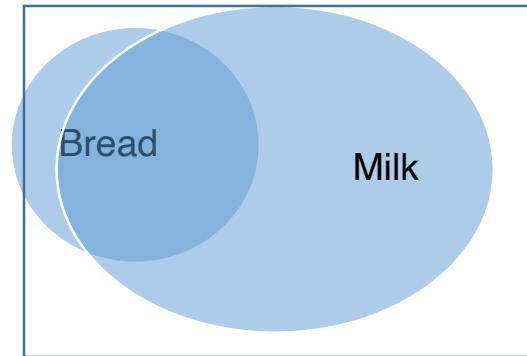
Range: [0,1]

Coverage

Measure of total of number transaction that can be analyzed.



Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread



Itemsets	Confidence
Bread → Jam	2/4 = 0.5
Bread → Milk	4/4 = 1
Bread → Juice	3/4 = 0.75



Interestingness Measure

Pattern Evaluation

Lift

Lift is a measures the relatedness of the inference made by the rules

The higher this value, the more likely that the existence of X and Y together in a transaction is not just a random occurrence, but because of some relationship between them.

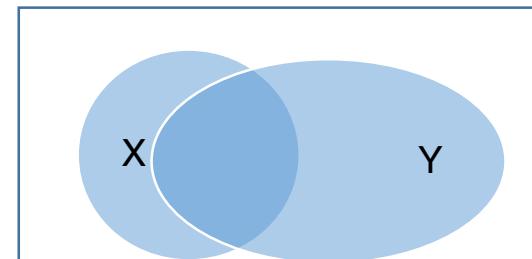
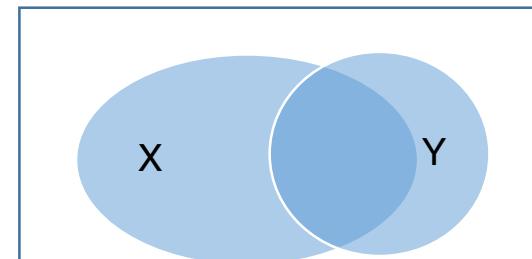
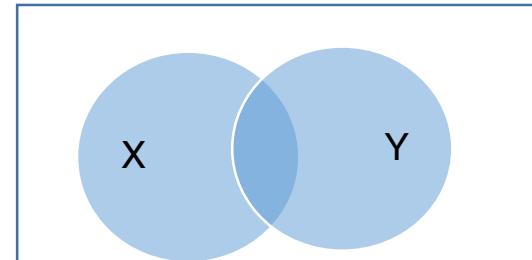
X → Y

{Bread} → {Milk}

$$\text{Lift}(X \rightarrow Y) = \frac{C(X \rightarrow Y)}{S(Y)} = \frac{s(X \cap Y)}{S(X) * S(Y)}$$
$$= \frac{P(Y | X)}{P(Y)}$$

Range: [0,]

Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread



Pattern Evaluation

Lift

Bread → Juice:

$$\text{Lift}(X \rightarrow Y) = \frac{C(X \rightarrow Y)}{S(Y)} = \frac{S(X \mid Y)}{S(X) * S(Y)}$$
$$= \frac{P(Y \mid X)}{P(Y)}$$

Bread → Jam:



Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread

{Milk} → {Rice} [confidence = 2/5 = 0.4]

Lift(X → Y) = $\frac{2/5}{2/5} = 1 \rightarrow X \text{ & } Y \text{ are independent}$

{Rice} → {Milk}

Lift(Y → X) = $\frac{2/2}{5/5} = 1$

Range: [0,]

Measures

Leverage

Leverage is a measure of the expected independent contribution of the components in the rules.

X → Y

{Bread} → {Juice}

$$\begin{aligned}\text{Lift}(X \rightarrow Y) &= P(X \cap Y) - P(X) * P(Y) \\ &= s(X \cap Y) - s(X) * s(Y)\end{aligned}$$

Range: [-1, +1]

Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread

Itemsets

Bread → Jam

Bread → Milk

Bread → Juice

Leverage

$$2/5 - (3/5 * 4/5) = -0.08$$

$$4/5 - (4/5 * 5/5) = 0$$

$$3/5 - (4/5 * 3/5) = 0.12$$

Pattern Evaluation

Conviction

Conviction is a measure of the direction of relatedness of the association rule.

X → Y

{Bread} → {Juice}

$$\text{conviction}(X \Rightarrow Y) = \frac{1 - s(Y)}{1 - C(X \Rightarrow Y)}$$
$$= \frac{P(X) * P(Y_c)}{P(X) P(Y)}$$

Range: [0, 1]



Transaction No.	Items
T1	Vegetables, Juice, Cleaner, Milk, Bread, Jam
T2	Medicine, Juice, Cleaner, Milk, Bread, Jam
T3	Vegetables, Butter, Milk, Bread
T4	Vegetables, Egg, Rice, Milk, Jam
T5	Rice, Juice, Milk, Bread

Itemsets

Bread → Jam

Bread → Juice

Jam → Bread

Juice → Bread

Conviction

$$\frac{(4/5)*(2/5)}{2/5} = 0.8$$

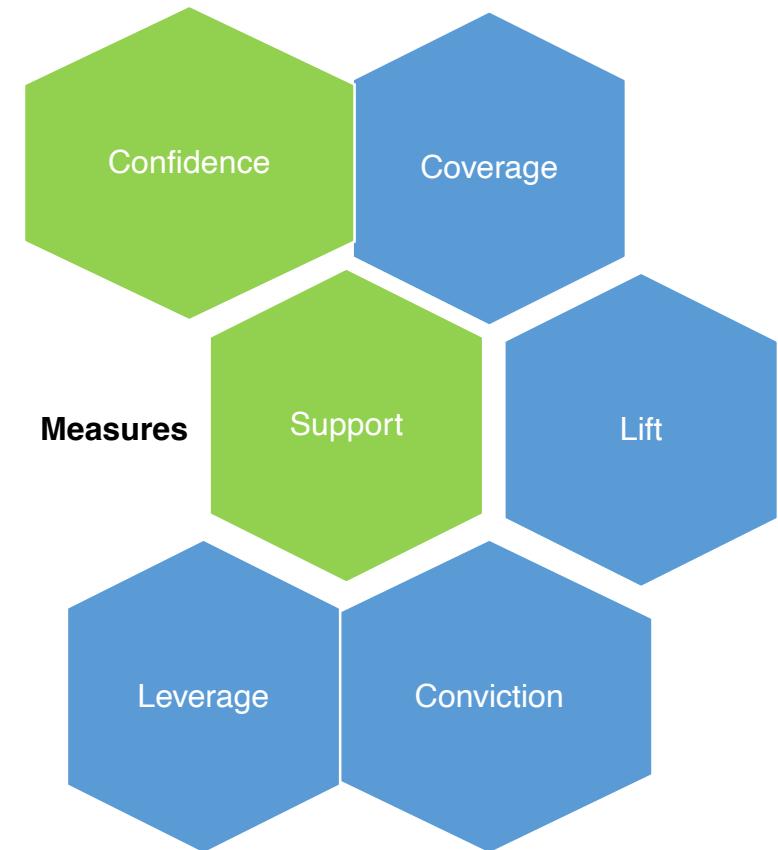
$$\frac{(4/5)*(2/5)}{1/5} = 1.6$$

$$\frac{(3/5)*(1/5)}{1/5} = 0.6$$

$$\frac{(3/5)*(1/5)}{0/5} =$$

Pattern Evaluation

Interestingness Measures



In this segment

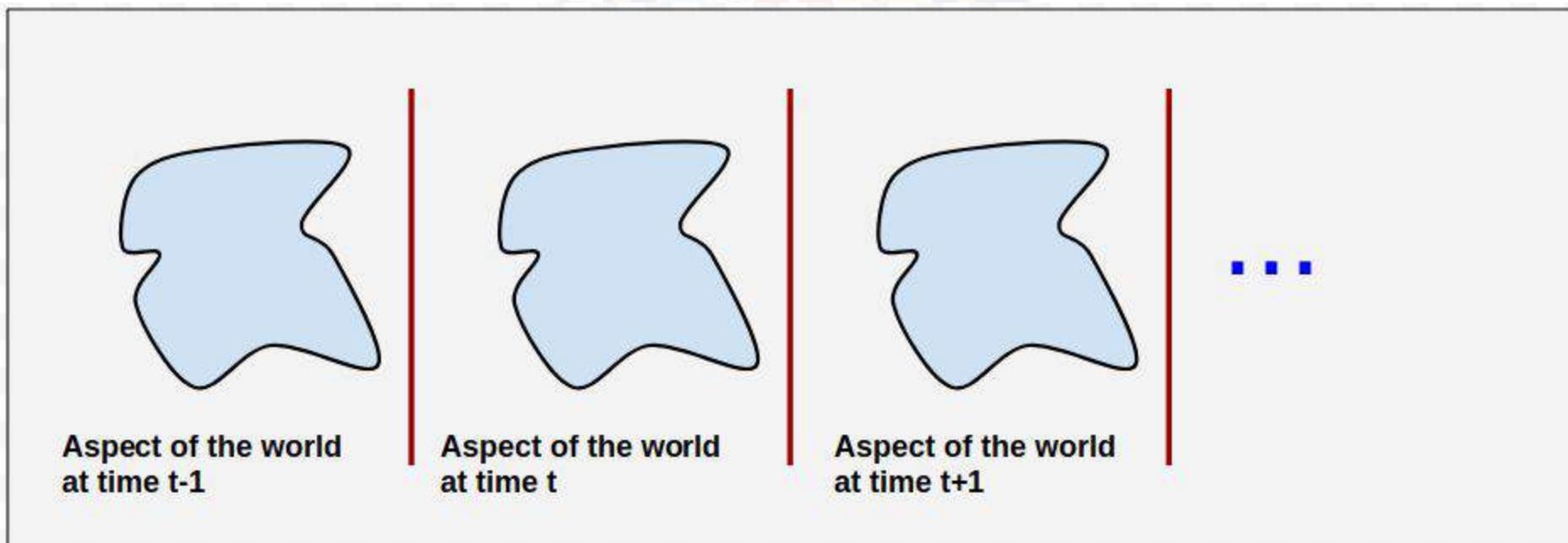
- States and Observations in a Sequential Data
- A case for the module (From Russell & Norvig)



Introduction

World as series of time slices

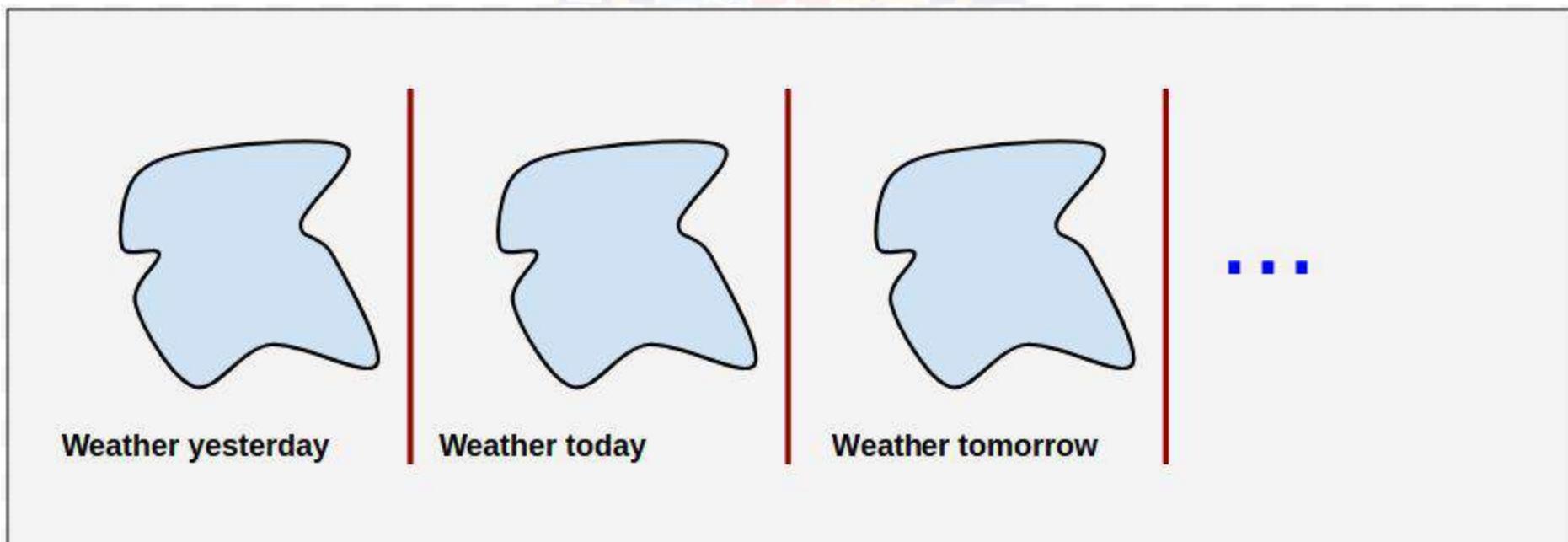
- World to as series of time slices / snap shots, with lengths appropriate for the problem



Introduction

World as series of time slices

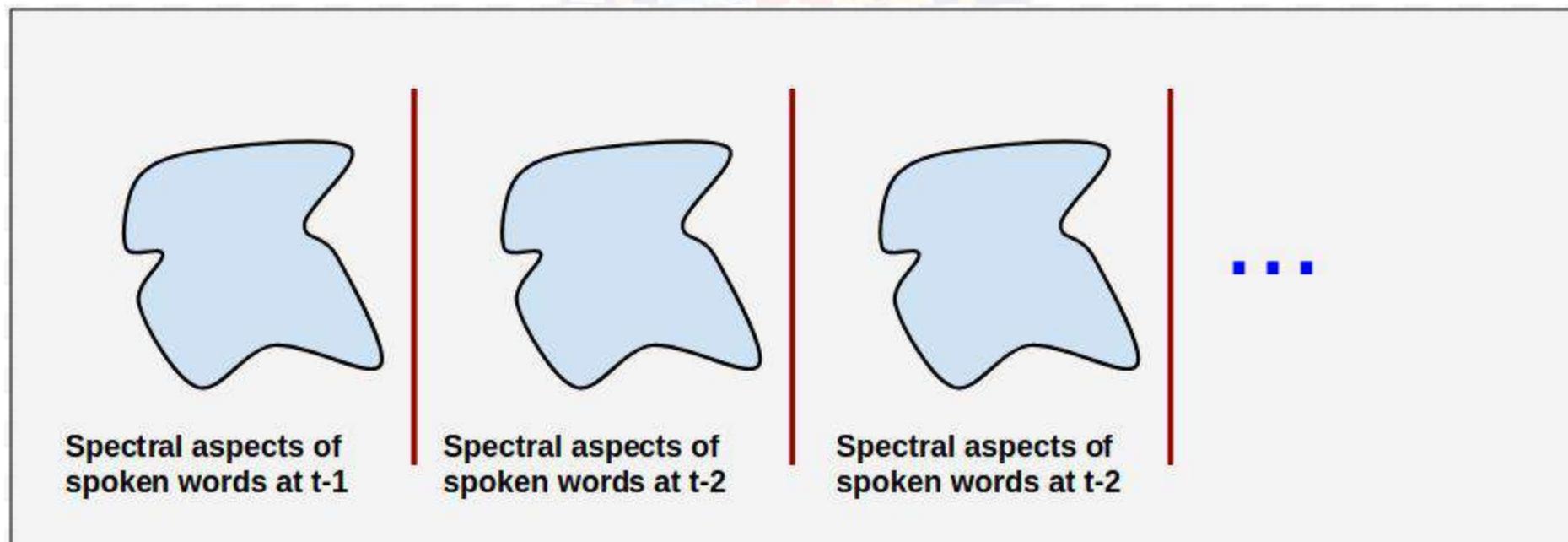
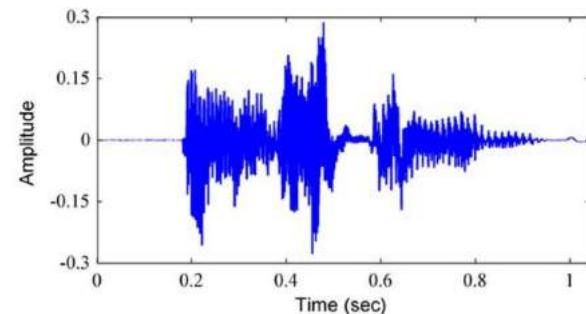
- World to as series of time slices / snap shots, with lengths appropriate for the problem



Introduction

World as series of time slices

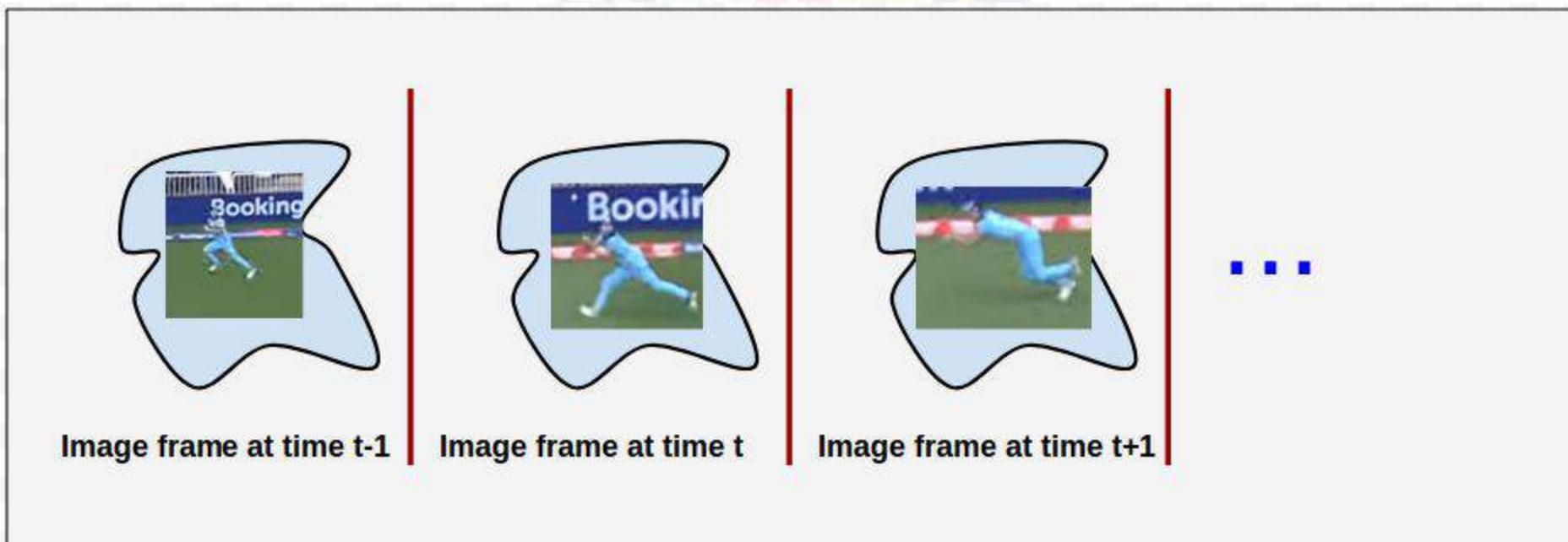
- World to as series of **time slices / snap shots**, with lengths appropriate for the problem



Introduction

World as series of time slices

- World to as series of time slices / snap shots, with lengths appropriate for the problem



Introduction

World as series of time slices

- World to as series of **time slices** / snap shots, with lengths appropriate for the problem
- Each time slice contain certain *observed* (for instance, measured) and *unobserved* aspects of the world
 - **Observed** : Temperature, Atmospheric pressure, Humidity, wind speed, wind direction etc
 - **Unobserved** : Season today , Snowfall category (category -1, 2), Rainfall

Introduction

World as series of time slices

- World to as series of **time slices** / snap shots, with lengths appropriate for the problem
- Each time slice contain certain *observed* (for instance, measured) and *unobserved* aspects of the world
 - **Observed** : Temperature, Atmospheric pressure, Humidity, wind speed, wind direction etc
 - **Unobserved** : Season today , Snowfall category (category -1, 2), Rainfall
- What are observed and unobserved aspects of recognizing spoken words problem?

A simple case from Russell & Norvig

Predicting rain by an agent employed in an underground installation

- A security guard employed at a underground secret installation, who has no way of seeing the world outside for longer time of his life
- Every morning, the director of the installation visits the station without fail
 - He appears with an umbrella, if it rains outside
 - He does not come with an umbrella, if it does not rain outside
- Security guard understands if it rains outside or not by watching this pattern of his director
 - Observed : Director carrying umbrella or not
 - Unobserved : Rains outside or not

A simple case from Russell & Norvig

Notations Used

- Observed (U_t) : Director carrying umbrella or not

t	0	1	2	3	4	...
U_t	-	True	True	False	False	...

- Unobserved (R_t) : Rains outside or not

t	0	1	2	3	4	...
R_t	True	True	True	False	False	...

A simple case from Russell & Norvig

Notations Used

- Observed (U_t) : Evidences / Measured (Evidence Variables)

t	0	1	2	3	4	...
U_t	-	True	True	False	False	...

- Unobserved (R_t) : State of the world (State variables)

t	0	1	2	3	4	...
R_t	True	True	True	False	False	...

A simple case from Russell & Norvig

Notations Used

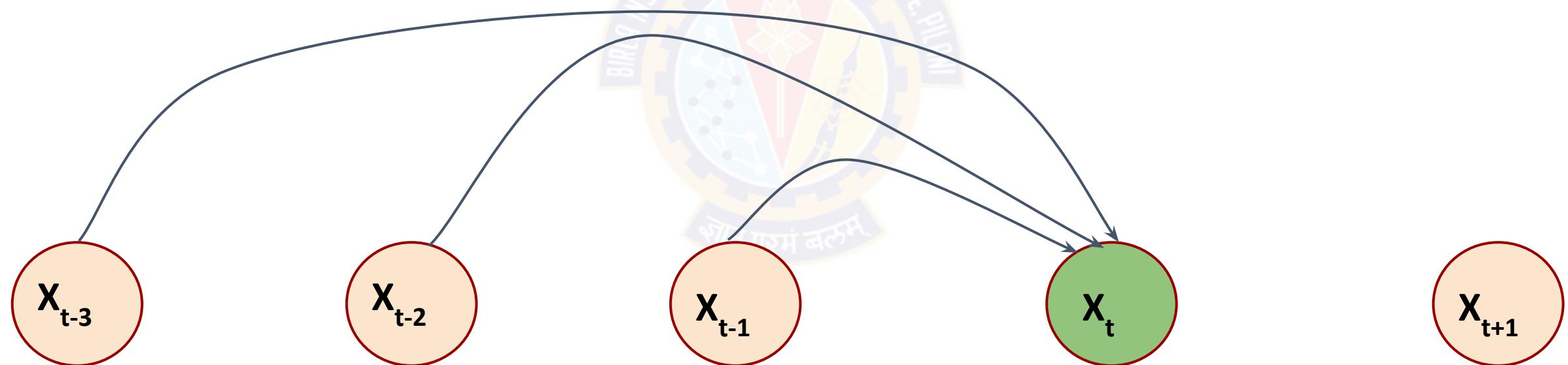
- Observed (U_t)
 - $U_{a:b}$ includes $U_a, U_{a+1}, U_{a+2}, U_{a+3}, \dots, U_b$
- Unobserved (R_t)
 - $R_{a:b}$ includes $R_a, R_{a+1}, R_{a+2}, R_{a+3}, \dots, R_b$
- In general, we will use freely shorthand notations of the form $X_{a:b}$ to represent $X_a, X_{a+1}, X_{a+2}, X_{a+3}, \dots, X_b$

Next : How do we model the connections between the variables of subsequent time slices?

Introduction

Transition Model

- Specifies how world (states) evolves?
 - A distribution over a state conditioned on previous world states.
 - $P(X_t | X_{0:t-1})$:

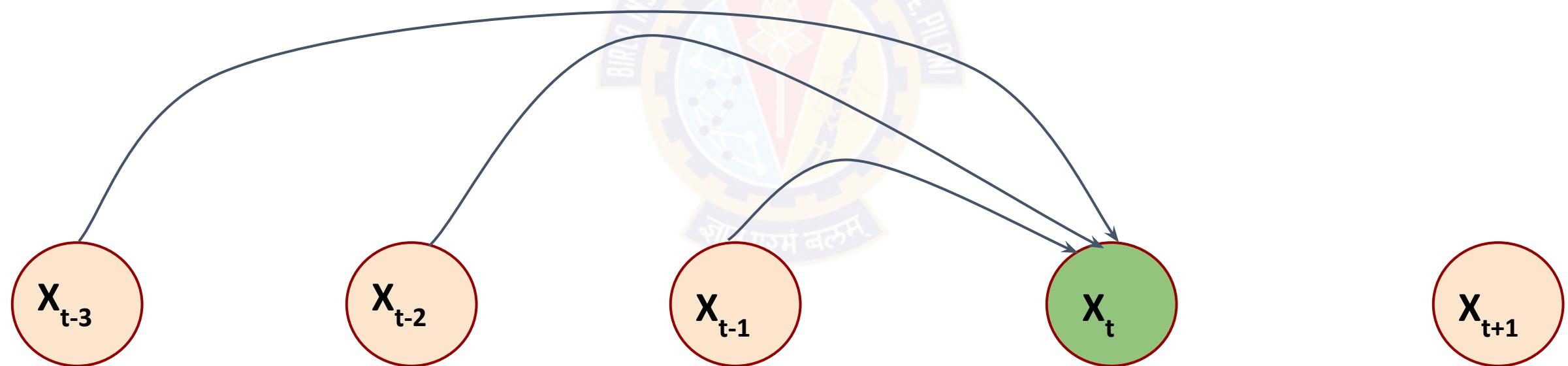


Introduction

Transition Model

$P(X_t | X_{0:t-1}) :$

- How do we specify this, as t increases?
- How many of my past states indeed carry information to help deciding on X_t

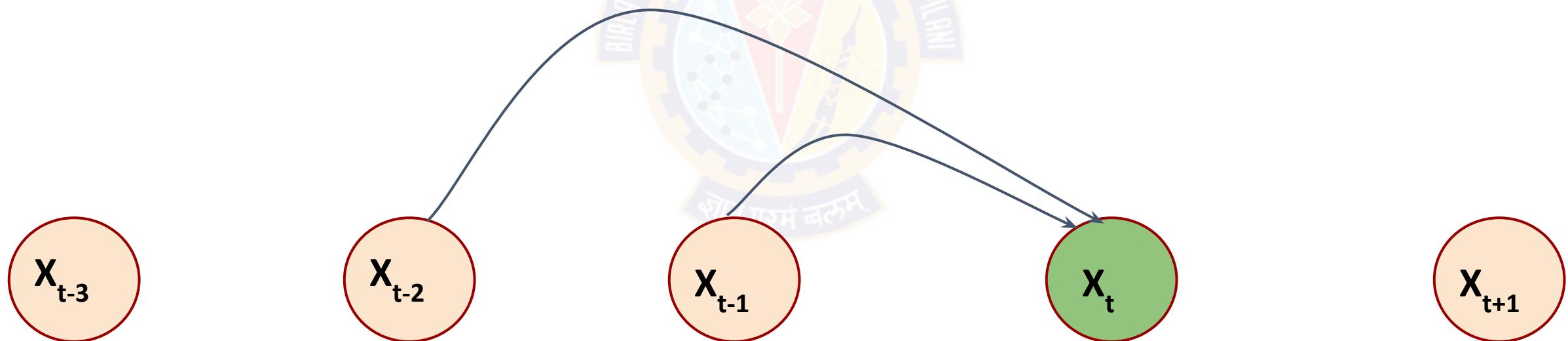


Markov Assumption

Andrei Markov (1856-1922)

$P(X_t | X_{0:t-1}) :$

- Current state depends on only a finite fixed number of previous states
 - Processes satisfying this property are called as **Markov Processes / Markov Chains**

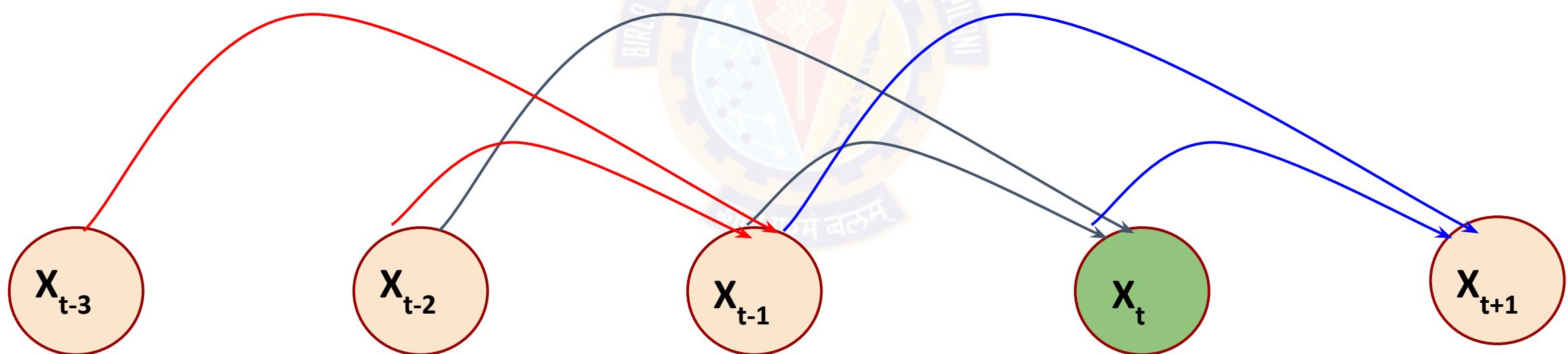


Markov Assumption

Andrei Markov (1856-1922)

$P(X_t | X_{0:t-1}) :$

- Current state depends on only a finite fixed number of previous states
 - Processes satisfying this property are called as **Markov Processes / Markov Chains**

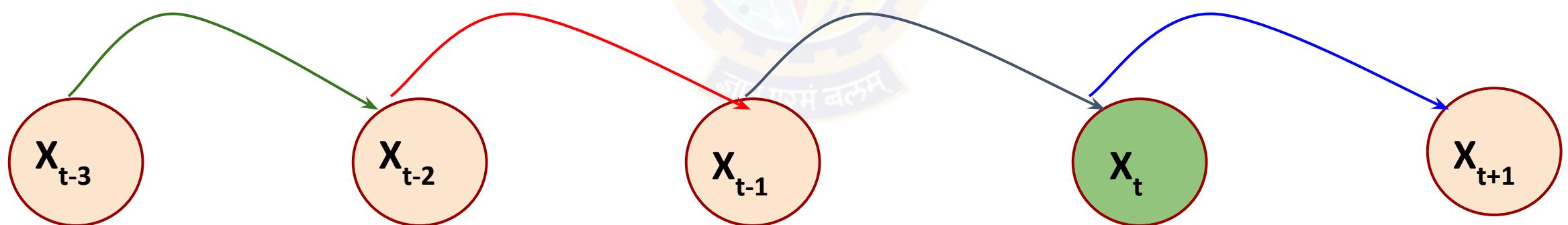


Markov Assumption

First Order Markov Process

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$$

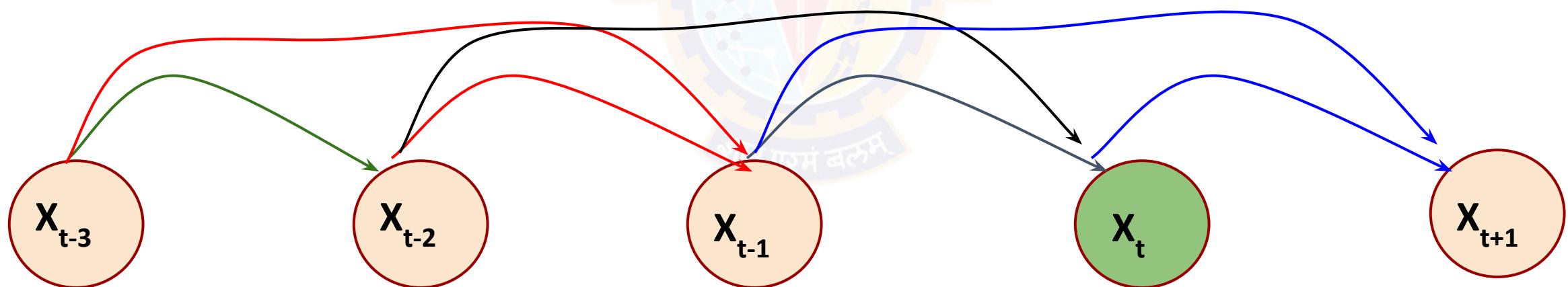
- Current state depends only on *previous state* and not on any other states
 - Processes satisfying this property are called as **Markov Processes / Markov Chains**



Markov Assumption

Stationary Markov Processes

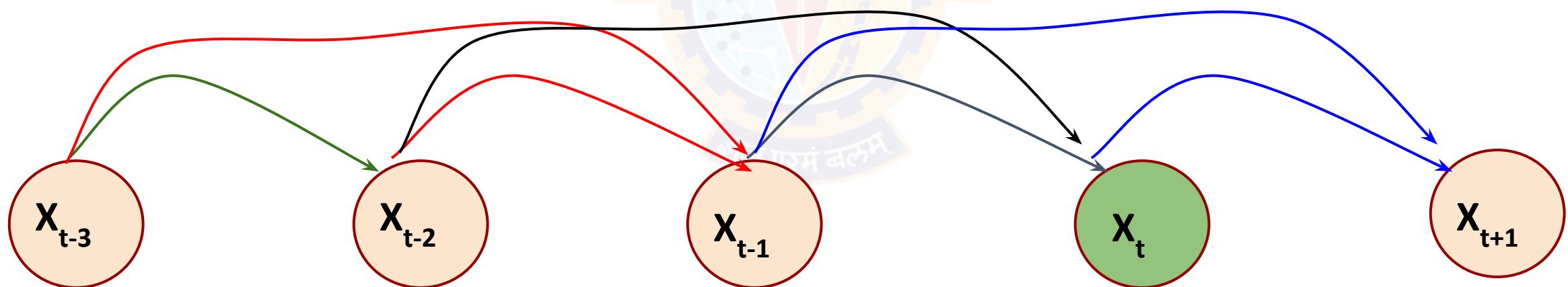
- Do we assume one $P(X_t | .)$ for each X_t ?
 - How do we handle this for infinite t then?
- Assume further that *the laws governing process of change does not change themselves*



Markov Assumption

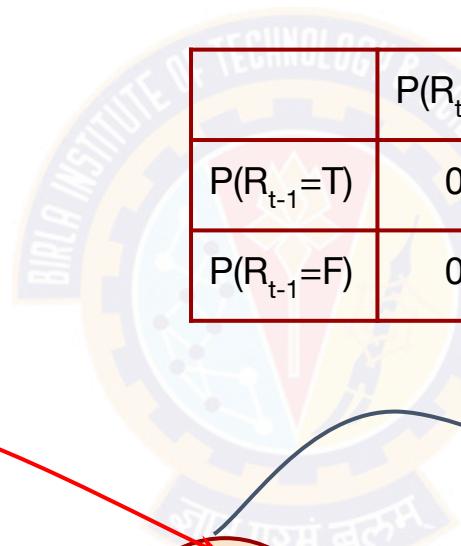
Stationary Markov Processes

- Do we assume one $P(X_t | .)$ for each X_t ?
 - How do we handle this for infinite t then?
- Assume further that the laws governing process of change does not change themselves

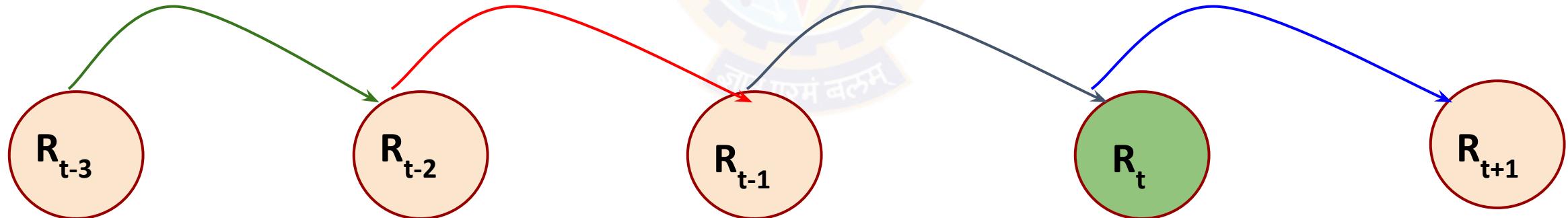


Markov Assumption

A first Order Stationary Markov Processes

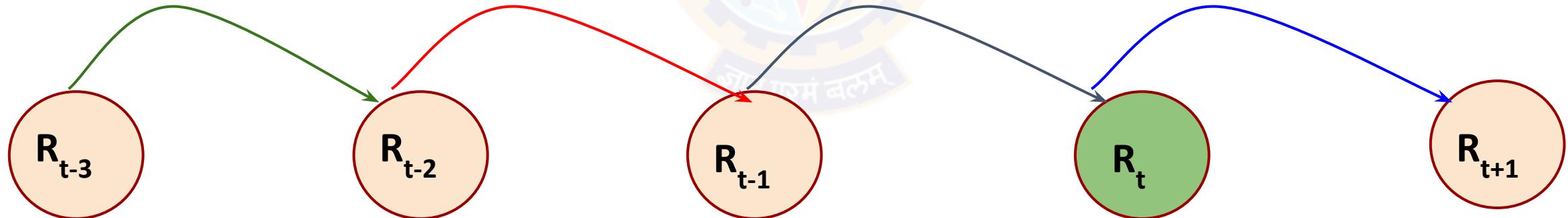
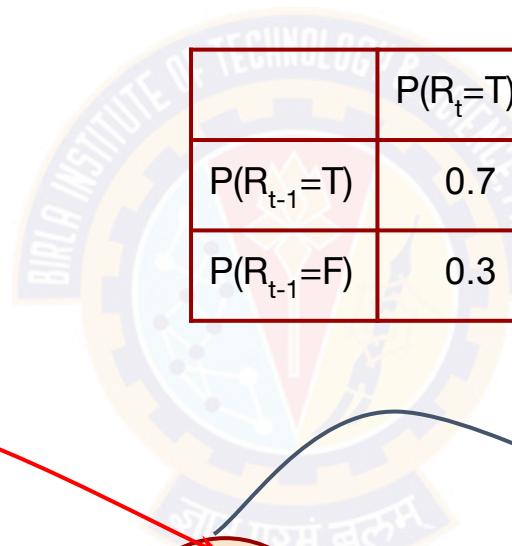


	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7



Markov Assumption

A First Order Stationary Markov Processes

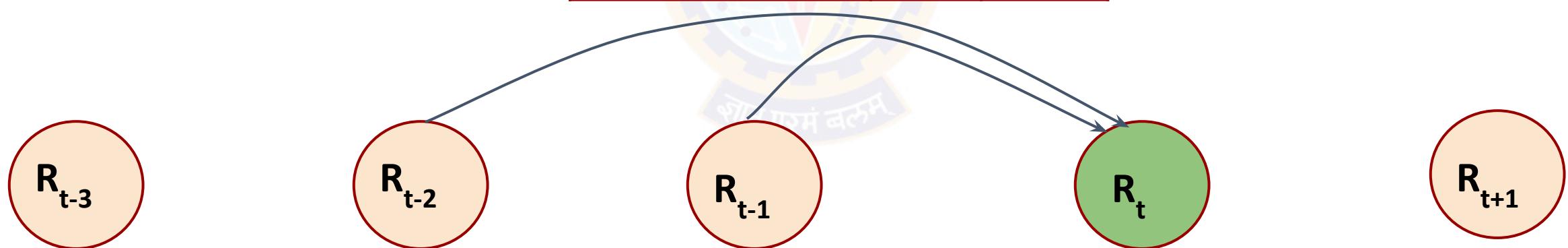


	$P(R_t=T)$
$P(R_{t-1}=T)$	0.7
$P(R_{t-1}=F)$	0.3

Markov Assumption

A Second Order Stationary Markov Processes

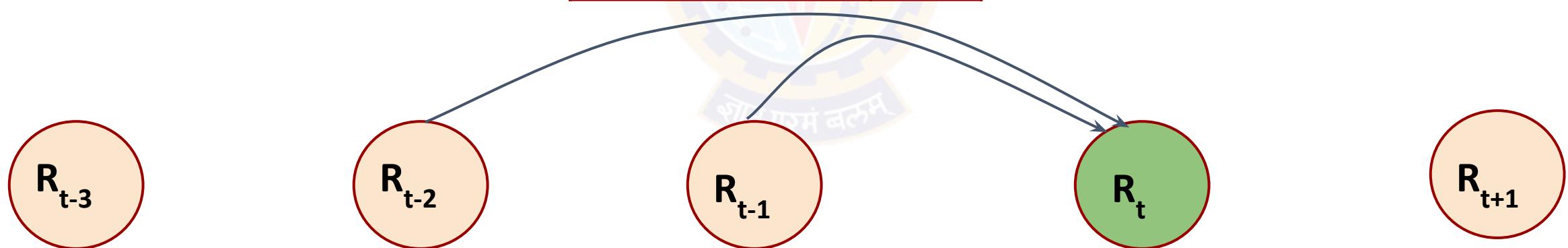
	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T \text{ and } R_{t-2}=T)$	0.7	0.3
$P(R_{t-1}=T \text{ and } R_{t-2}=F)$	0.3	0.7
$P(R_{t-1}=F \text{ and } R_{t-2}=T)$	0.6	0.4
$P(R_{t-1}=F \text{ and } R_{t-2}=F)$	0.1	0.9



Markov Assumption

A Second Order Stationary Markov Processes

	$P(R_t=T)$
$P(R_{t-1}=T \text{ and } R_{t-2}=T)$	0.7
$P(R_{t-1}=T \text{ and } R_{t-2}=F)$	0.3
$P(R_{t-1}=F \text{ and } R_{t-2}=T)$	0.6
$P(R_{t-1}=F \text{ and } R_{t-2}=F)$	0.1



Markov Assumption

A Second Order Stationary Markov Processes

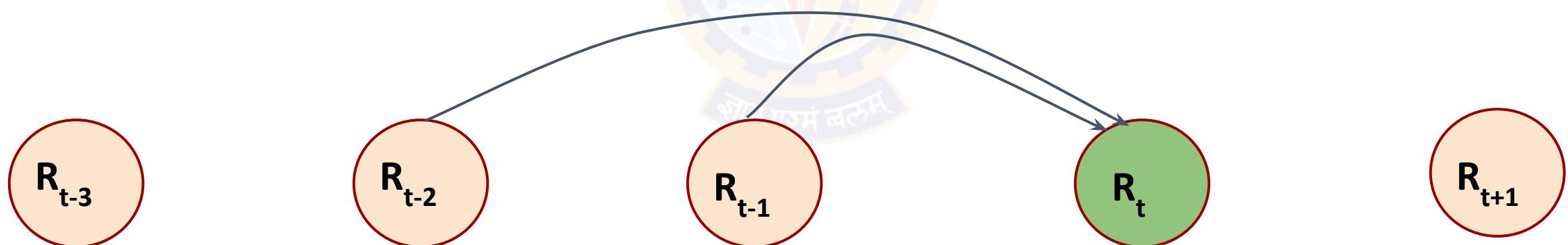
	$P(R_t=T)$
$P(R_{t-1}=T \text{ and } R_{t-2}=T)$	0.7
$P(R_{t-1}=T \text{ and } R_{t-2}=F)$	0.3
$P(R_{t-1}=F \text{ and } R_{t-2}=T)$	0.6
$P(R_{t-1}=F \text{ and } R_{t-2}=F)$	0.1

2 x 2 x (2-1) entries in the table for binary state variable.

3 x 3 x (3-1) entries in the table for variable which are ternary !!!

k x k x (k-1) entries in the table for variable which are ternary !!!

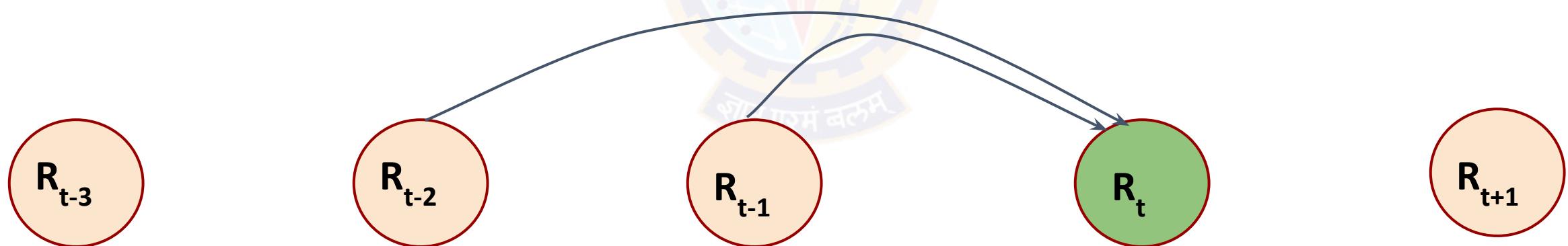
Larger the order, larger the transition model



Markov Assumption

A Second Order Stationary Markov Processes

	$P(R_t=T)$
$P(R_{t-1}=T \text{ and } R_{t-2}=T)$	0.7
$P(R_{t-1}=T \text{ and } R_{t-2}=F)$	0.3
$P(R_{t-1}=F \text{ and } R_{t-2}=T)$	0.6
$P(R_{t-1}=F \text{ and } R_{t-2}=F)$	0.1



But our model has evidence variables too, not just state variables !!!

How to model them now?



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.4: Sensor Model

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

- Sensor Markov Assumption for Sensor models
- Specifying Prior for initial state

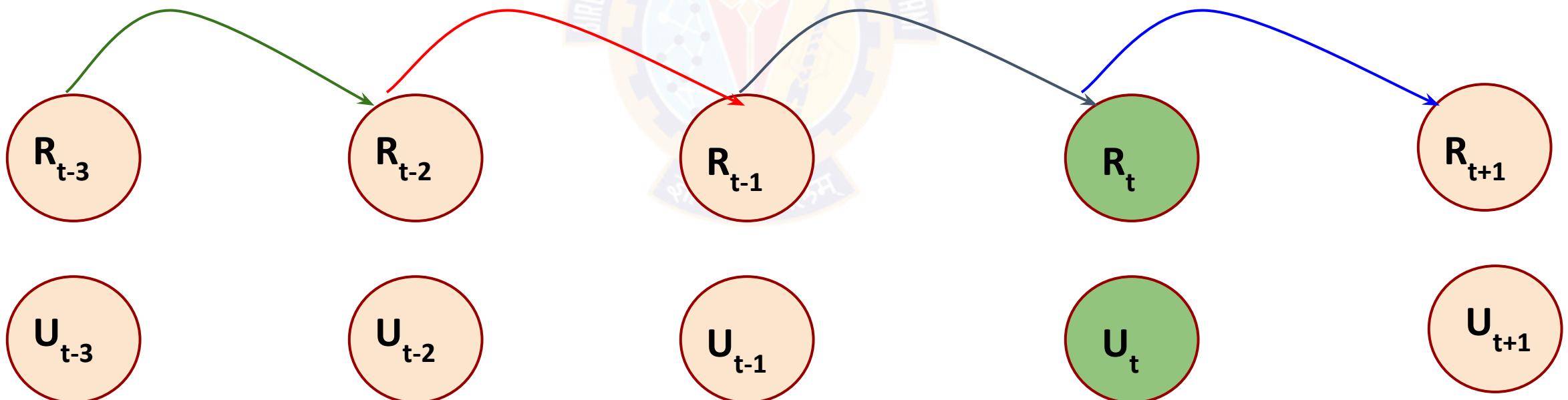


Sensor Model / Observation Model

Does my evidence variable depend on all the previous states?

- Possible !!!
- But if the present state has sufficient information then this is unlikely

	$P(R_t=T)$
$P(R_{t-1}=T)$	0.7
$P(R_{t-1}=F)$	0.3



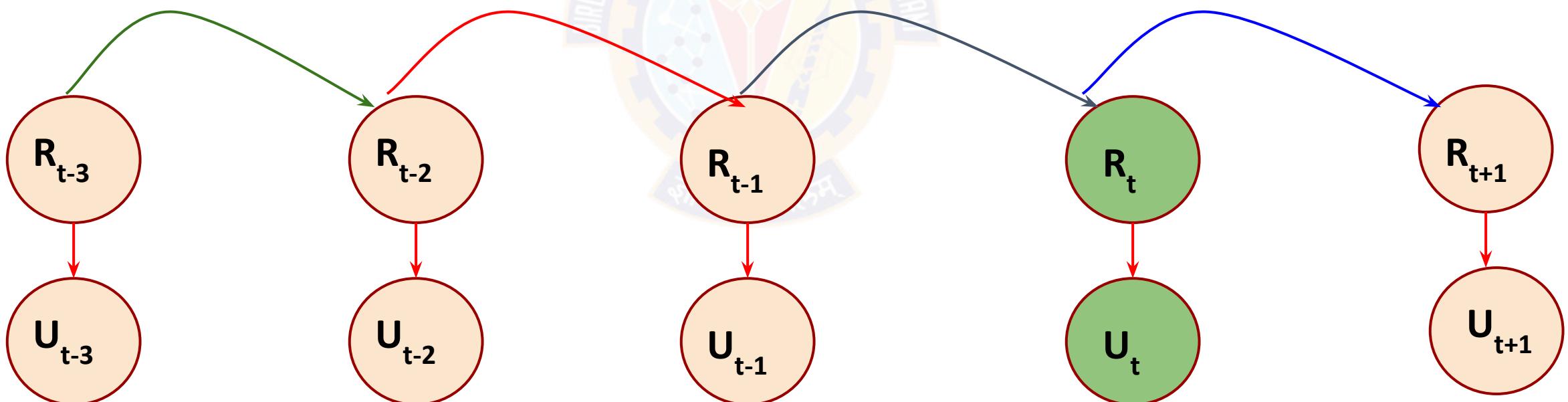
Sensor Model / Observation Model

Does my evidence variable depend on all the previous states?

Sensor Markov Assumption

$$P(E_t | X_{0:t-1}, E_{1:t-1}) = P(E_t | X_t)$$

	P(R _t =T)
P(R _{t-1} =T)	0.7
P(R _{t-1} =F)	0.3



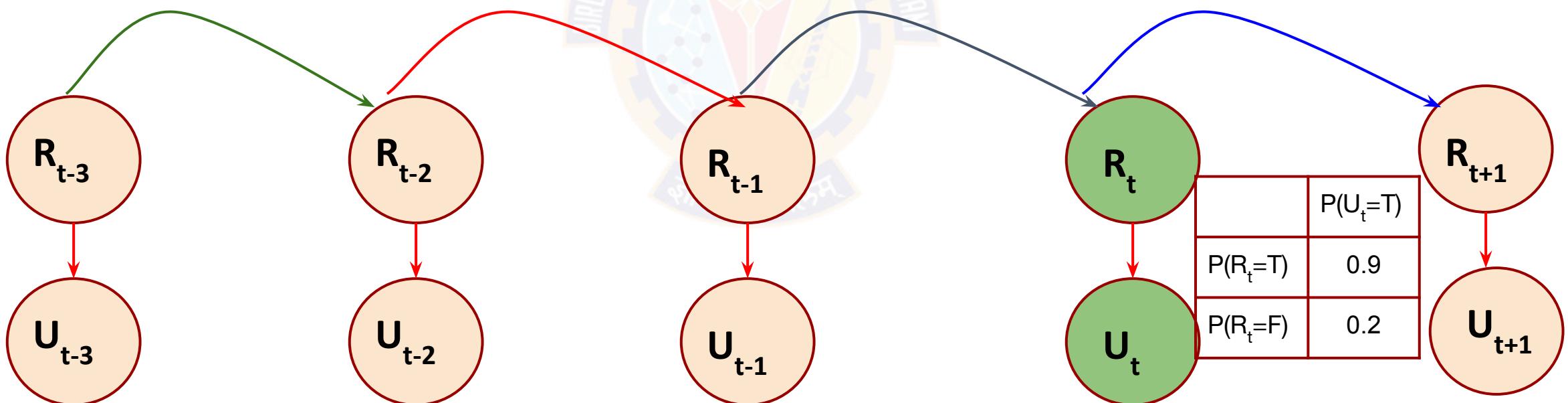
Sensor Model / Observation Model

Does my evidence variable depend on all the previous states?

Sensor Markov Assumption

$$P(E_t | X_{0:t-1}, E_{1:t-1}) = P(E_t | X_t)$$

	$P(R_t = T)$
$P(R_{t-1} = T)$	0.7
$P(R_{t-1} = F)$	0.3

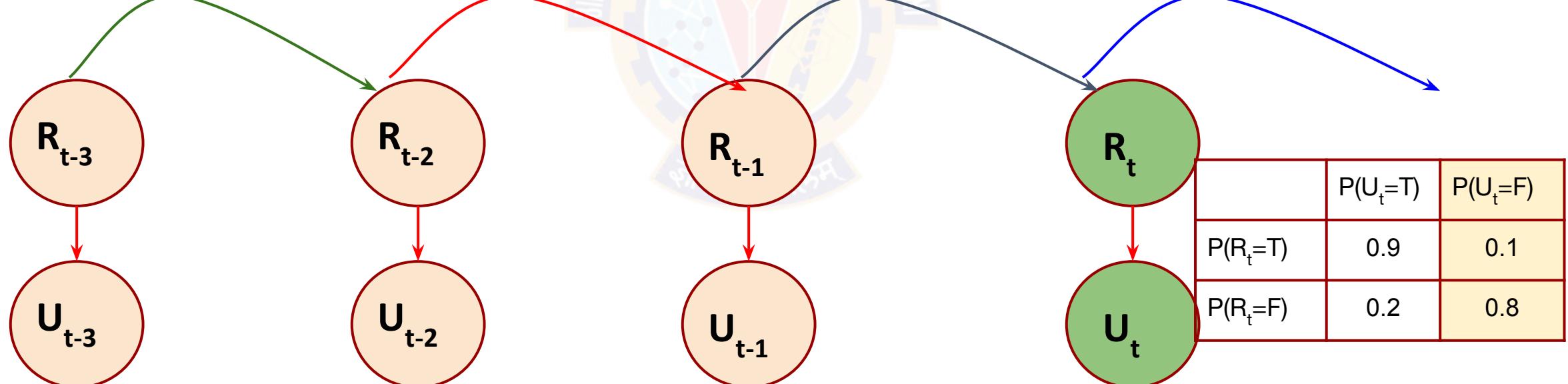


Sensor Model / Observation Model

Does my evidence variable depend on all the previous states?

Sensor Markov Assumption

$$P(E_t | X_{0:t-1}, E_{1:t-1}) = P(E_t | X_t)$$



Sensor Model / Observation Model

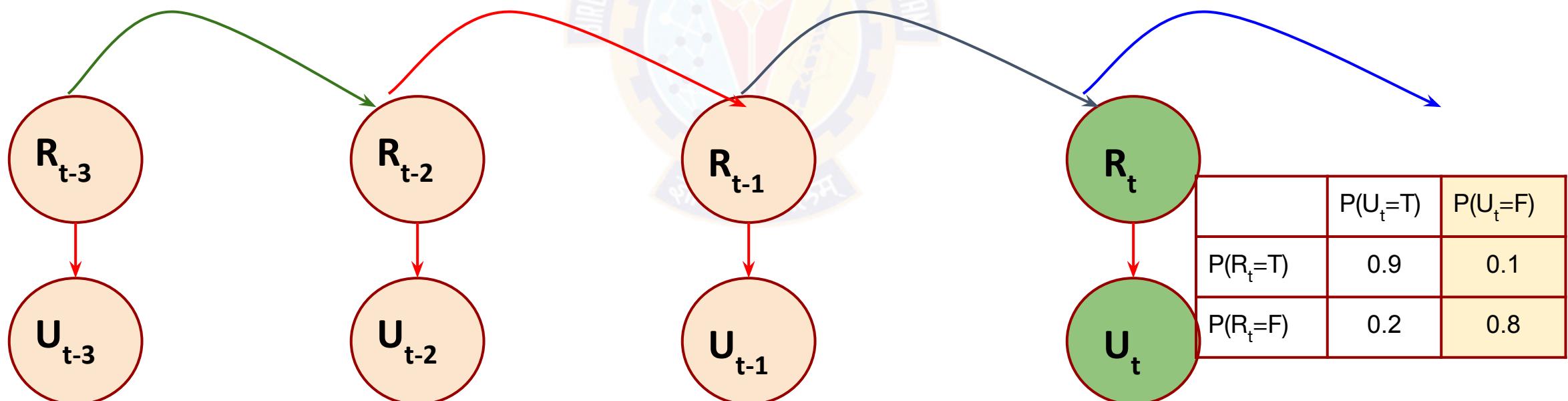
Does my evidence variable depend on all the previous states?

Observe the direction of edges

Rain causes umbrella to appear

i.e. Word state causes sensors to take particular values !!!

	$P(R_t=T)$
$P(R_{t-1}=T)$	0.7
$P(R_{t-1}=F)$	0.3



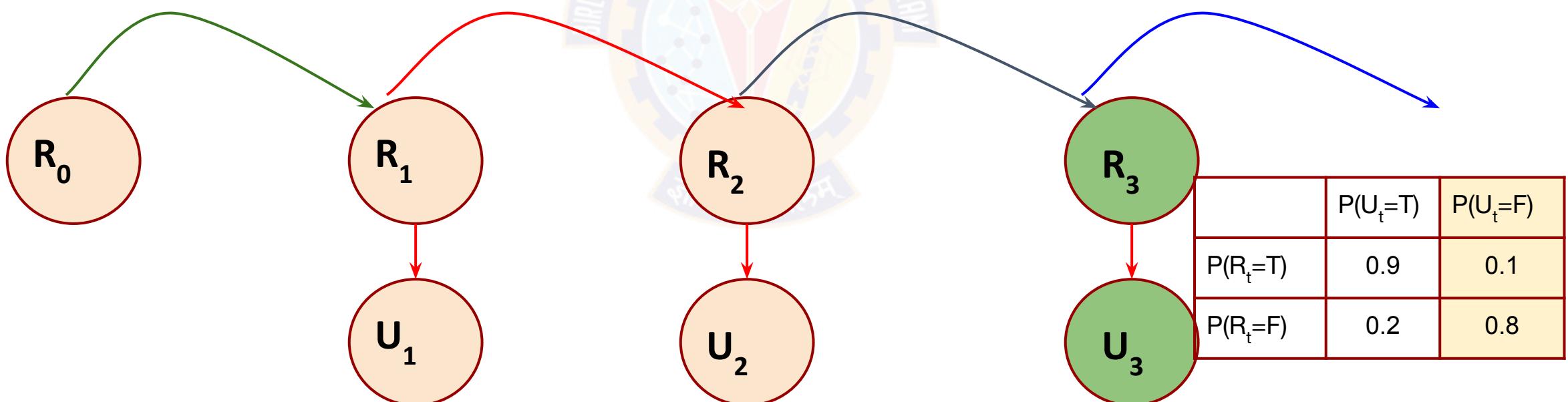
Sensor Model / Observation Model

How all this gets started?

Transition & Sensor model largely describe the network
and are not complete

Time $t=0$, which does not have a predecessor.

Specify $P(X_0)$, i.e. $P(R_0)$ here



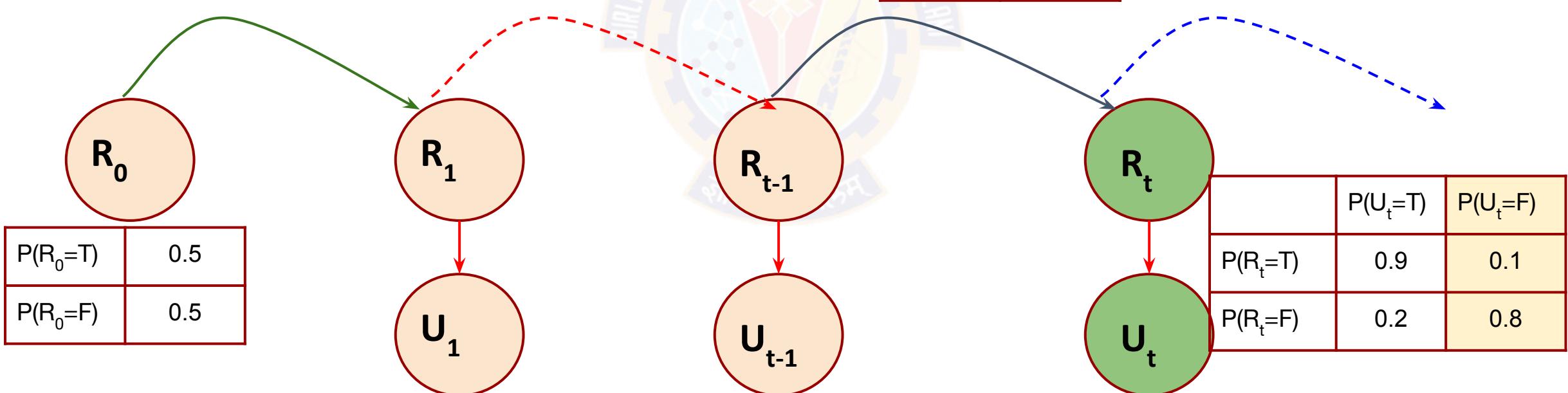
Sensor Model / Observation Model

How all this gets started?

Transition & Sensor model largely describe the network
and but are not complete

Time $t=0$, which does not have a predecessor.

Specify $P(X_0)$, i.e. $P(R_0)$ here





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.5: Hidden Markov Models

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

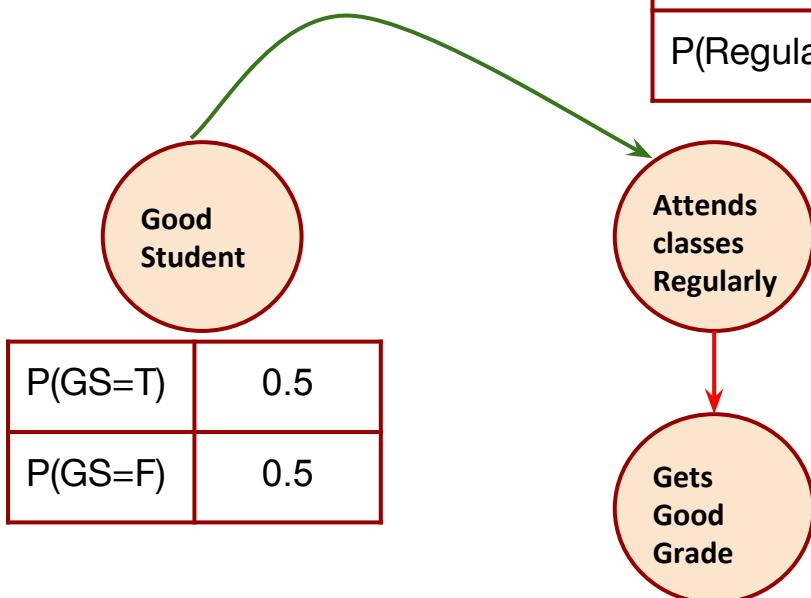
- Bayesian Networks (A gentle intro)
- HMM
- Outline of the rest of the module



Hidden Markov Models (HMM)

Bayesian Network

- Syntax: (a) a set of nodes, one per variable (b) a directed, acyclic graph (link \approx "directly influences") (c) conditional distribution for each node given its parents: $P(X_i | \text{Parents}(X_i))$

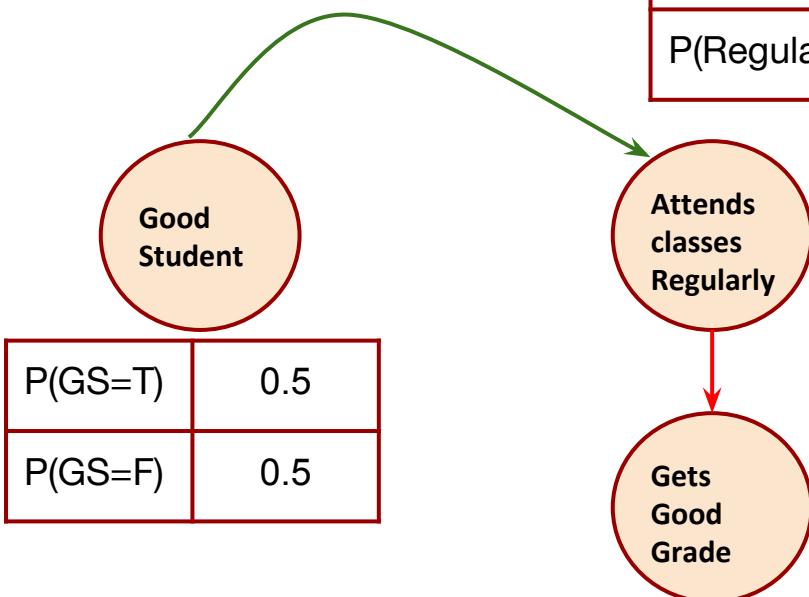


Hidden Markov Models (HMM)

Bayesian Network

- Full Joint Distribution:

$$P(X_1, \dots, X_n) = \pi_{i=1} P(X_i | \text{Parents}(X_i))$$



	P(GS=T)	P(GS=F)
P(Regular to Classes=T)	0.9	0.1
P(Regular to Classes=F)	0.3	0.7

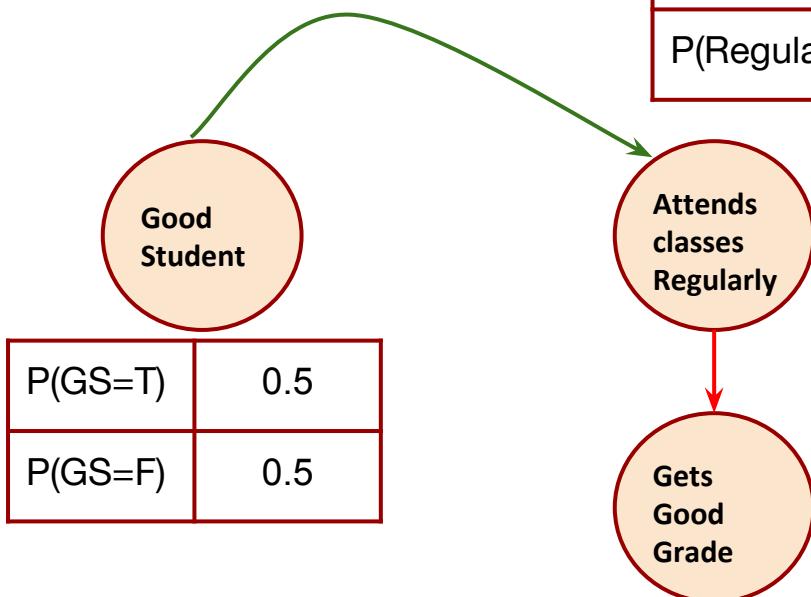
	P(Regular to Classes=T)	P(Regular to Classes=F)
P(GGG = T)	0.9	0.1
P(GGG = F)	0.2	0.8

Hidden Markov Models (HMM)

Bayesian Network

- Full Joint Distribution:

$$P(\text{GS, Regular, GGG}) = P(\text{GS}) P(\text{Regular} \mid \text{GS}) P(\text{GGG} \mid \text{Regular})$$



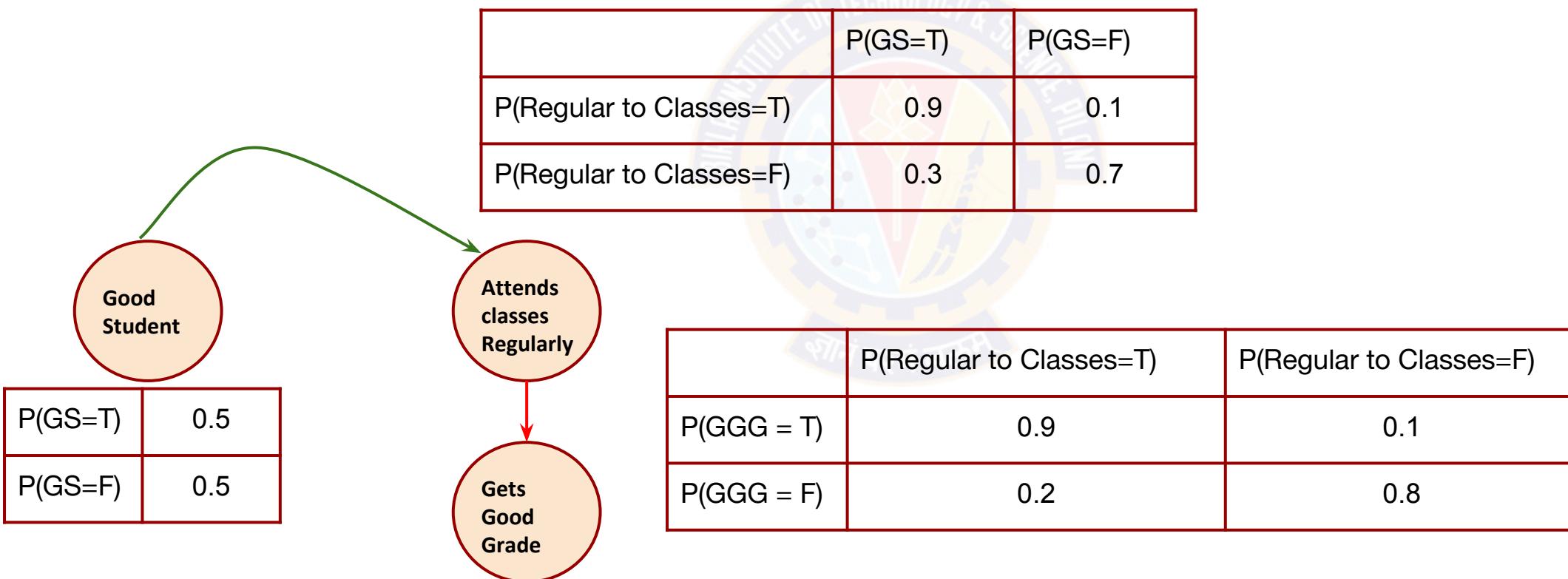
	P(GS=T)	P(GS=F)
P(Regular to Classes=T)	0.9	0.1
P(Regular to Classes=F)	0.3	0.7

	P(Regular to Classes=T)	P(Regular to Classes=F)
P(GGG = T)	0.9	0.1
P(GGG = F)	0.2	0.8

Hidden Markov Models (HMM)

Bayesian Network

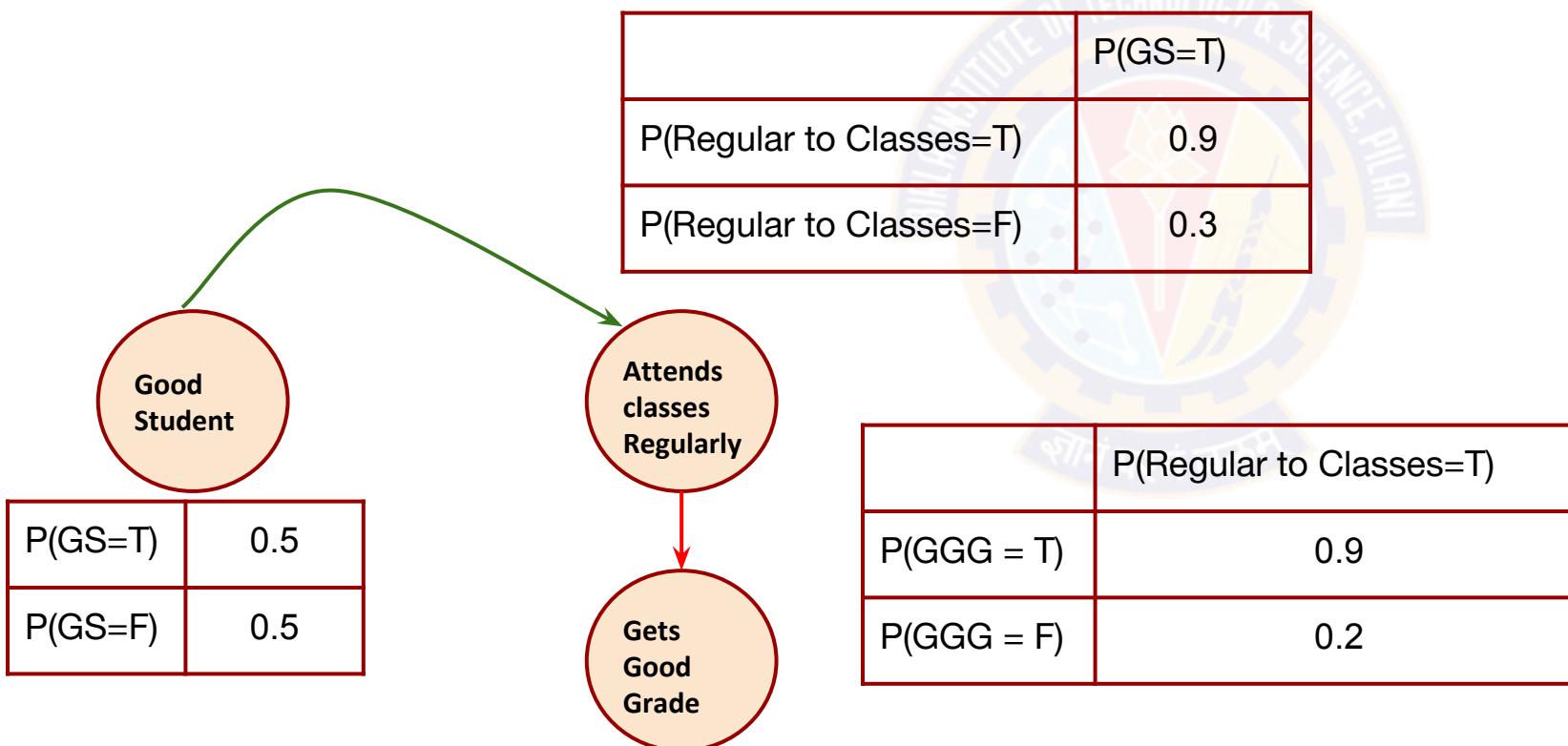
- A simple, graphical notation for conditional independence assertions and hence for compact specification of *full joint distributions*



Hidden Markov Models (HMM)

Bayesian Network

- A simple, graphical notation for conditional independence assertions and hence for compact specification of *full joint distributions*

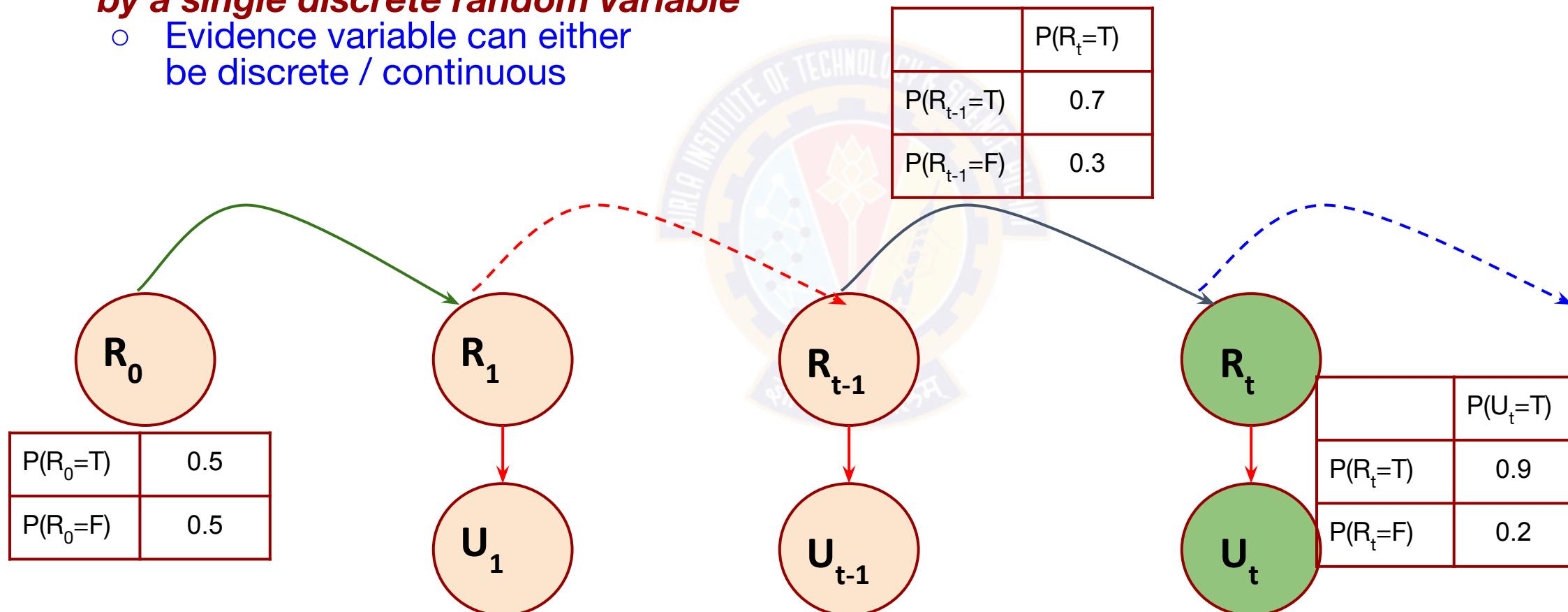


GS	Regular	GGG	P(GS, Regular, GGG)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Hidden Markov Models (HMM)

Bayesian Network

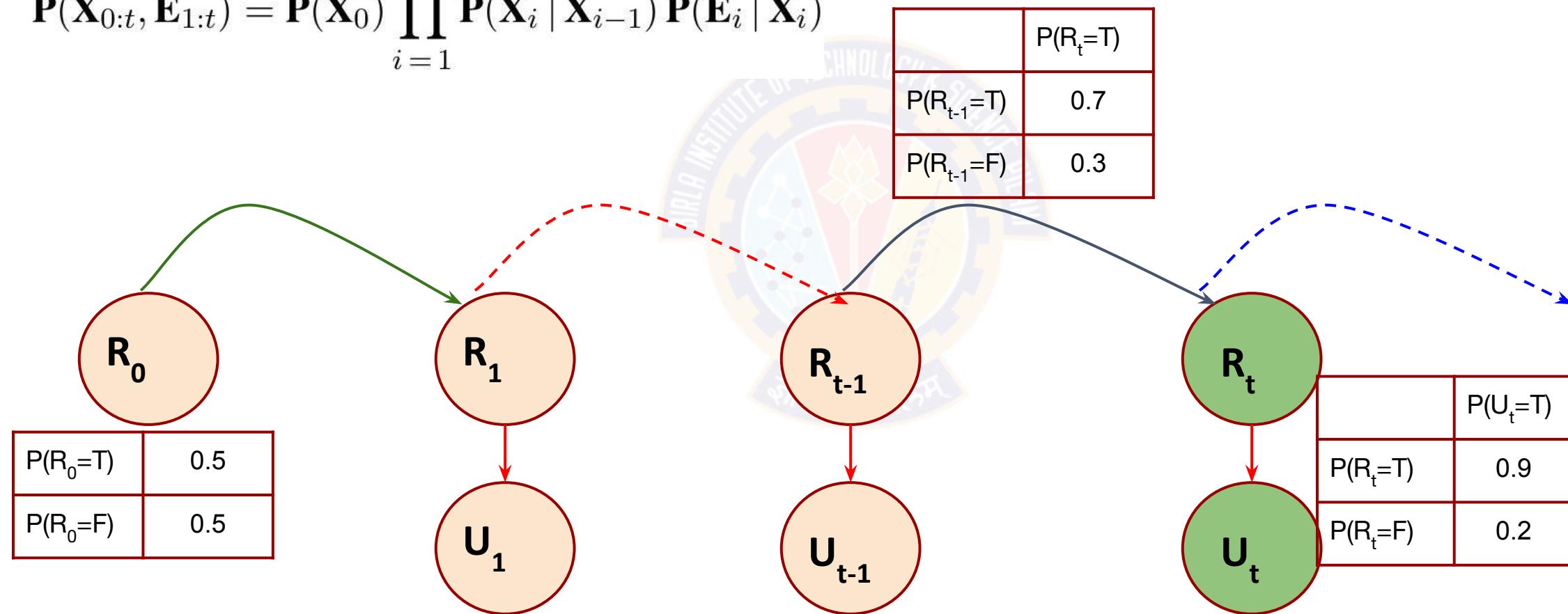
- An HMM is a temporal probabilistic model in which the ***state of the process is described by a single discrete random variable***
 - Evidence variable can either be discrete / continuous



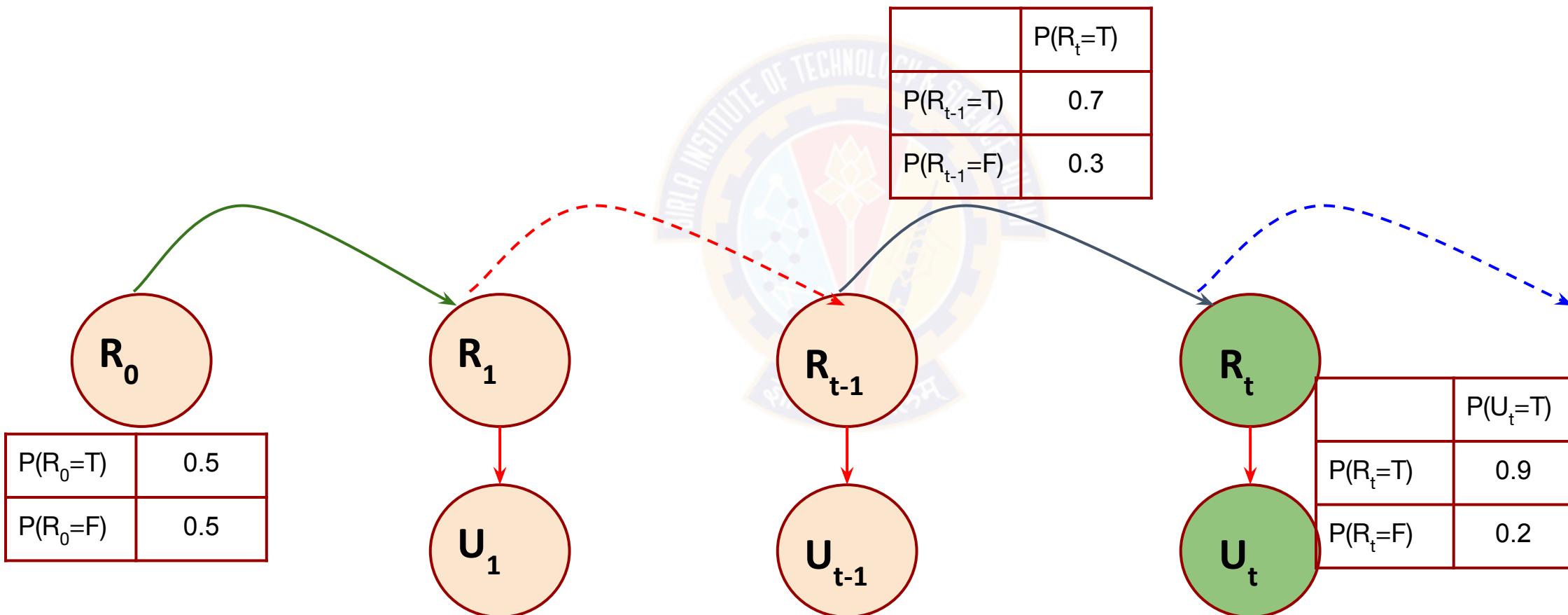
Hidden Markov Models (HMM)

Joint Distribution

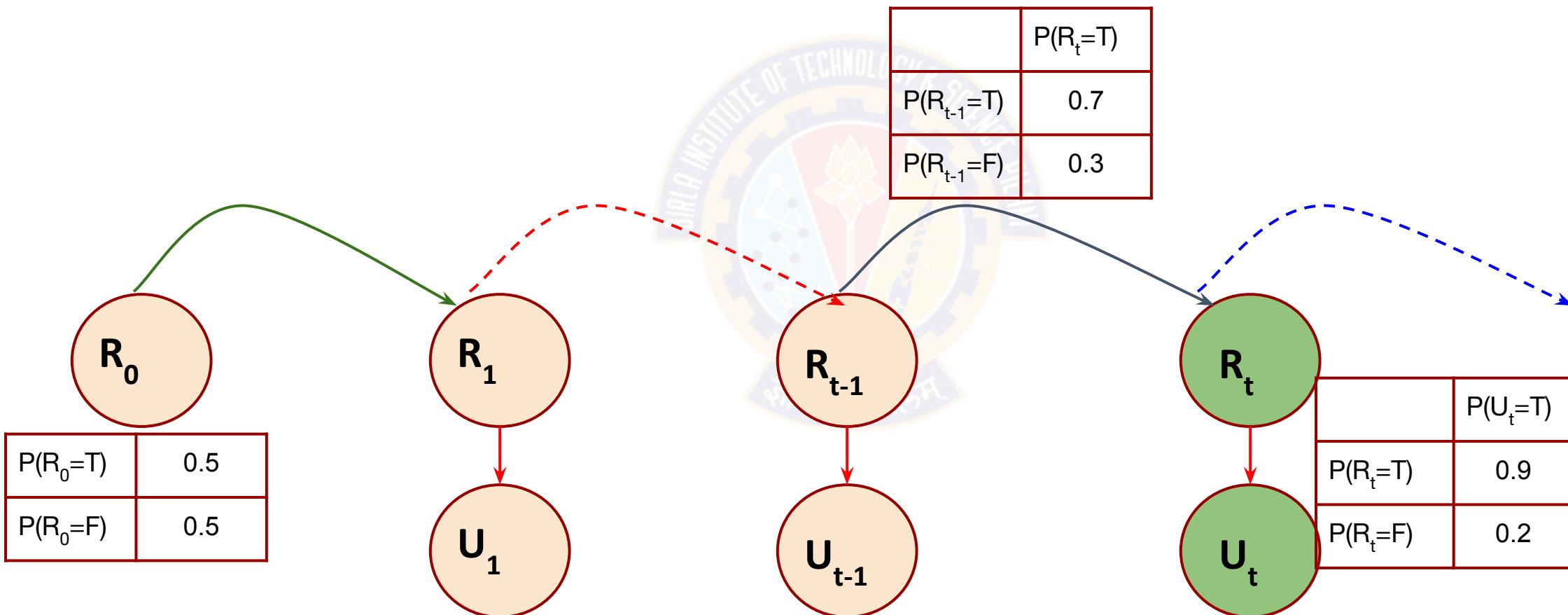
$$P(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = P(\mathbf{X}_0) \prod_{i=1}^t P(\mathbf{X}_i | \mathbf{X}_{i-1}) P(\mathbf{E}_i | \mathbf{X}_i)$$



Hidden Markov Models (HMM)



Hidden Markov Models (HMM)



Hidden Markov Models (HMM)

Problems [to be covered in the rest of the module]

- Filtering
 - State estimation : $P(X_t | e_{1:t})$
 - Ex: Given evidence $e_{1:4} = [U_1=T, U_2=F, U_3=F, U_4=T]$, estimate $P(R_4 | e_{1:4})$
- Prediction
 - Future State Estimation : $P(X_{t+k} | e_{1:t})$ where $k > 0$
 - Ex: Given evidence $e_{1:4} = [U_1=T, U_2=F, U_3=F, U_4=T]$, estimate $P(R_7 | e_{1:4})$
- Smoothing
 - Compute posterior over past state : $P(X_k | e_{1:t})$ where $0 \geq k > t$
 - Ex: Given evidence $e_{1:4} = [U_1=T, U_2=F, U_3=F, U_4=T]$, estimate $P(R_2 | e_{1:4})$
- Most Likely Explanation
 - Find the sequence of states that is most likely to have generated given observations:
 $\text{argmax } x_{1:t} P(x_{1:t} | e_{1:t})$
 - Ex: Given evidence $e_{1:4} = [U_1=T, U_2=F, U_3=F, U_4=T]$, estimate most likely $x_{1:t}$

Hidden Markov Models (HMM)

Problems [to be covered in the rest of the module]

- How to learn model parameters (transition, sensor models) from data using EM Algorithm



In this segment

- State estimation - Derivation



State Estimation Problem (Filtering)

Problem

State estimation : $P(R_t | u_{1:t})$

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R	[0.5, 0.5] [Prior]					[T=?, F=?]

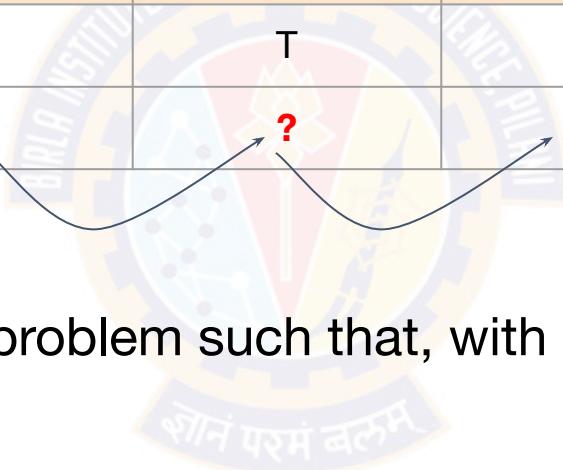
Pose it as a recursive estimation problem such that, with every new evidences we can estimate state incrementally

State Estimation Problem (Filtering)

Problem

State estimation : $P(R_t | u_{1:t})$

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R	[0.5, 0.5] [Prior]	?	?	?	?	[T=? , F=?]



Pose it as a recursive estimation problem such that, with every new evidences we can estimate state incrementally

$$P(X_{t+1} | e_{1:t+1}) = f(e_{t+1}, P(X_t | e_{1:t}))$$

State Estimation Problem (Filtering)

Problem

State estimation : $P(R_t | u_{1:t})$

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R	[0.5, 0.5] [Prior]	?	?	?	?	[T=? , F=?]

$$P(X_{t+1} | e_{1:t+1})$$

$$= P(X_{t+1} | e_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t}) \quad (\text{by the sensor Markov assumption})$$

State Estimation Problem (Filtering)

Problem

State estimation : $P(R_t | u_{1:t})$

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R	[0.5, 0.5] [Prior]	?	?	?	?	[T=? , F=?]

$$P(X_{t+1} | e_{1:t+1})$$

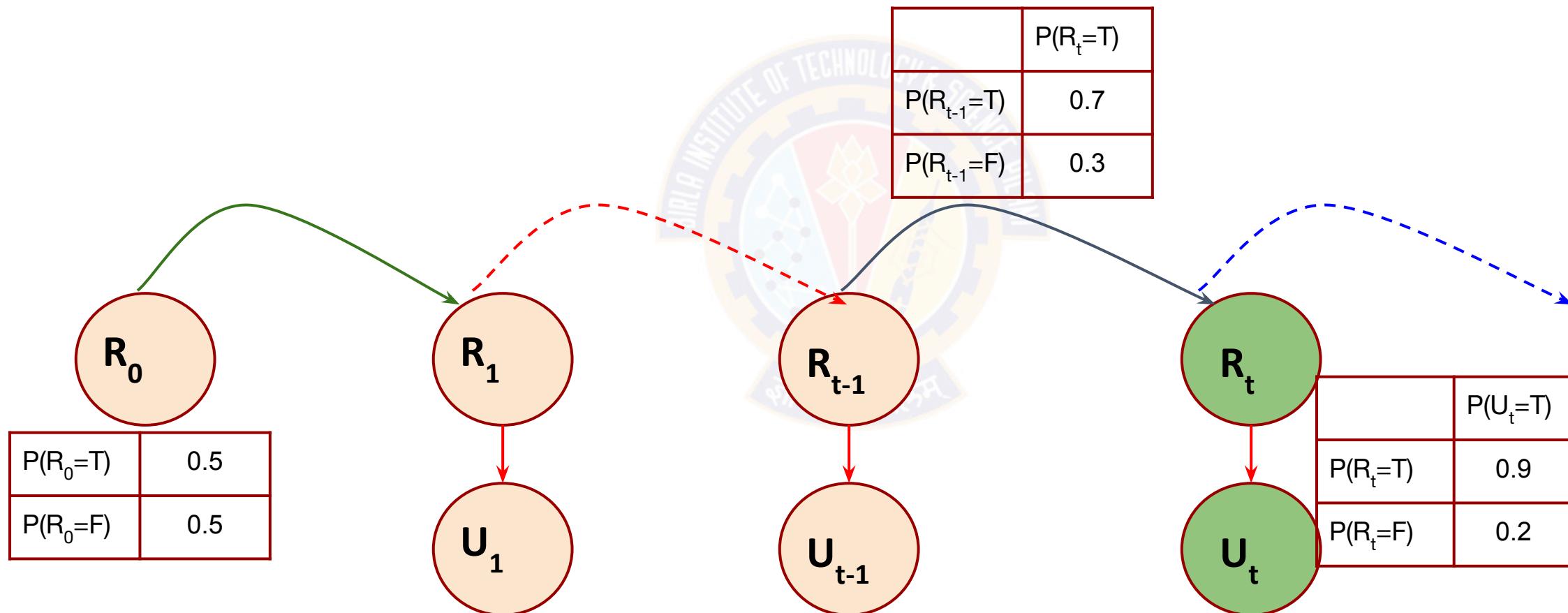
$$= P(X_{t+1} | e_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \underbrace{P(e_{t+1} | X_{t+1})}_{\substack{\uparrow \\ \text{Use Sensor model to} \\ \text{compute this}}} P(X_{t+1} | e_{1:t}) \quad (\text{by the sensor Markov assumption})$$

State Estimation Problem (Filtering)

Problem



State Estimation Problem (Filtering)

Problem

State estimation : $P(R_t | u_{1:t})$

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R	[0.5, 0.5] [Prior]	?	?	?	?	[T=? , F=?]

$$P(X_{t+1} | e_{1:t+1})$$

$$= P(X_{t+1} | e_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \underbrace{P(e_{t+1} | X_{t+1})}_{\substack{\text{Looks like an one step prediction, Let us expand} \\ \text{more to understand this.}}} \underbrace{P(X_{t+1} | e_{1:t})}_{\substack{}} \quad (\text{by the sensor Markov assumption})$$

State Estimation Problem (Filtering)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \underline{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})} \quad (\text{by the sensor Markov assumption})$$



Expanding this now. Since we have term $e_{1:t}$ let us try to introduce x_t by conditioning on it

State Estimation Problem (Filtering)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \underline{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})} \quad (\text{by the sensor Markov assumption})$$



$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{e}_{1:t}) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

State Estimation Problem (Filtering)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{by the sensor Markov assumption})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \underline{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{e}_{1:t})} P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

Distribution on \mathbf{X}_{t+1} solely depend on \mathbf{x}_t , by transition model.
That leads to a simplification.

State Estimation Problem (Filtering)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad (\text{dividing up the evidence})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{using Bayes' rule})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad (\text{by the sensor Markov assumption})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \underline{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{e}_{1:t})} P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \boxed{\sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)} P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

Let us try to understand this form now !!!

State Estimation Problem (Filtering)

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Comes from sensor model Comes from transition model



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Recursive Term, propagated forward !!!

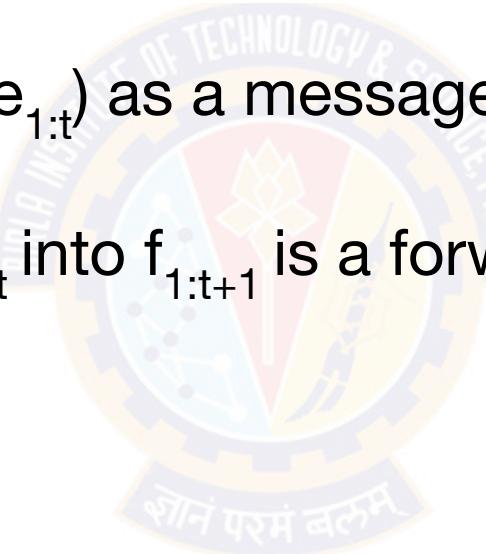


State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Let us call the quantity $P(X_t | e_{1:t})$ as a message $f_{1:t}$

The process that translates $f_{1:t}$ into $f_{1:t+1}$ is a forward propagation procedure.



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Let us call the quantity $P(\mathbf{X}_t | \mathbf{e}_{1:t})$ as a message $f_{1:t}$

The process that translates $f_{1:t}$ into $f_{1:t+1}$ is a forward propagation procedure.

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha \boxed{P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})}$$

Let us call the quantity $P(\mathbf{X}_t | \mathbf{e}_{1:t})$ as a message $f_{1:t}$

The process that translates $f_{1:t}$ into $f_{1:t+1}$ is a forward propagation procedure.

This can be written as:

$$\mathbf{f}_{1:t+1} = \alpha \text{ FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$$

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Let us call the quantity $P(\mathbf{X}_t | \mathbf{e}_{1:t})$ as a message $f_{1:t}$

The process that translates $f_{1:t}$ into $f_{1:t+1}$ is a forward propagation procedure.

This can be written as:

$$\mathbf{f}_{1:t+1} = \alpha \text{FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$$

Let us understand this better with a simple numerical example, next !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.7: Illustrating Filtering with an Example

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

- Illustrating Filtering



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

R_{t-1}

U_{t-1}

R_t

U_t

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

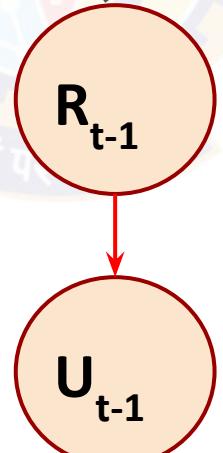
State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #0:

As per the setup, no evidences

$$P(R_0) = <0.5, 0.5>$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

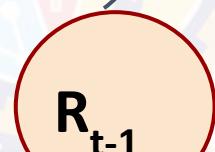
$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

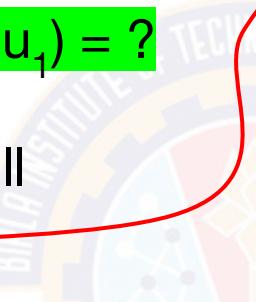
Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

Let us do this in two steps. First, we will
compute this part

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8



State Estimation Problem (Filtering)

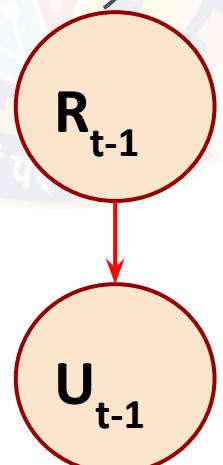
$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

$$\sum_{r_0} P(R_1 | r_0) P(r_0)$$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

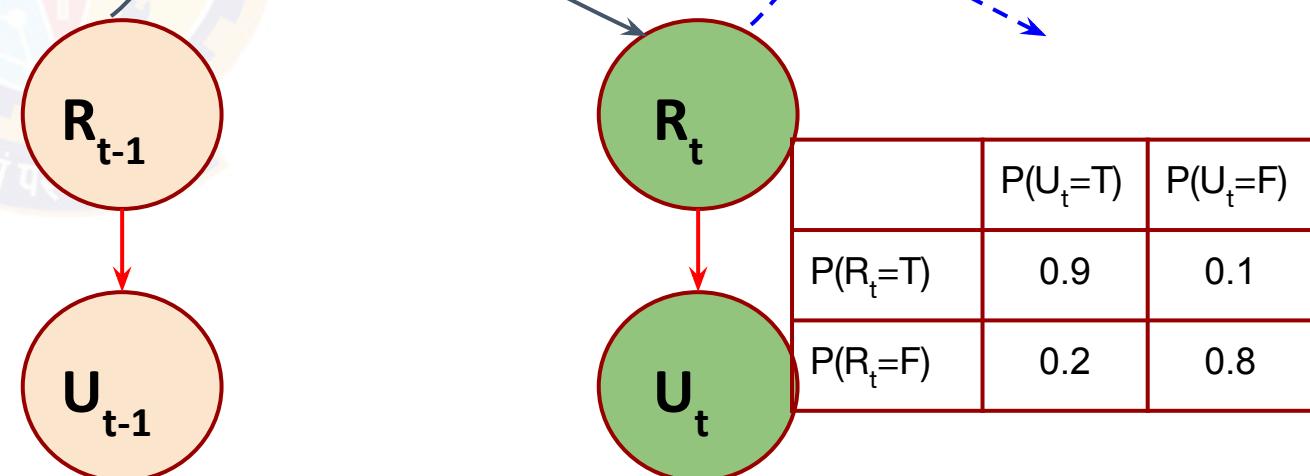
Day #1:

Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

$$\sum_{r_0} P(R_1 | r_0) P(r_0) = ?$$

We need to $P(R_1 | r_0=T)$ and $P(R_1 | r_0=F)$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

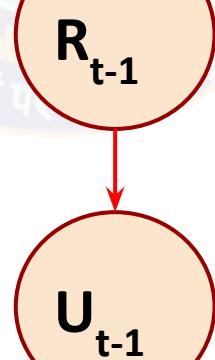
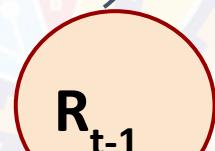
Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

$$\sum_{r_0} P(R_1 | r_0) P(r_0) = ?$$

We need to $P(R_1 | r_0=T)$ and $P(R_1 | r_0=F)$

$$P(R_1 | r_0=T) = <0.7, 0.3>$$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

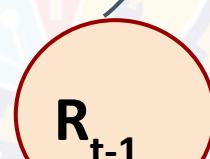
$$\sum_{r_0} P(R_1 | r_0) P(r_0) = ?$$

We need to $P(R_1 | r_0=T)$ and $P(R_1 | r_0=F)$

$$P(R_1 | r_0=T) = <0.7, 0.3>$$

$$P(R_1 | r_0=F) = <0.3, 0.7>$$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

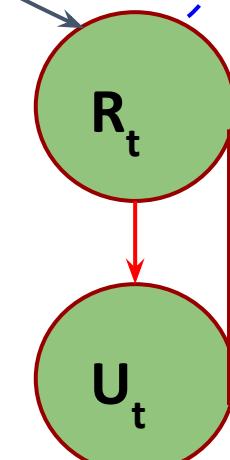
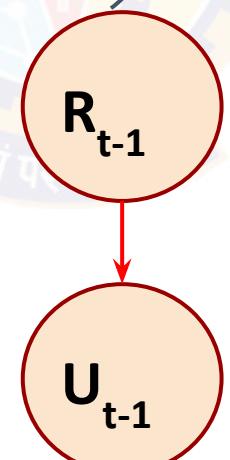
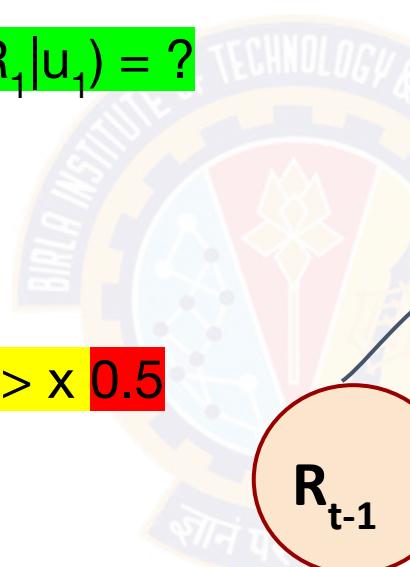
$$\sum_{r_0} P(R_1 | r_0) P(r_0)$$

$$= <0.7, 0.3> \times 0.5 + <0.3, 0.7> \times 0.5$$

$$P(R_1 | r_0=T) = <0.7, 0.3>$$

$$P(R_1 | r_0=F) = <0.3, 0.7>$$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

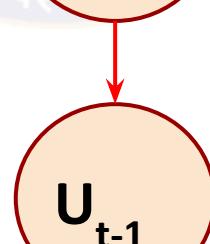
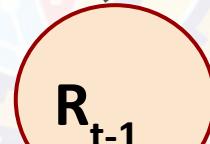
Umbrella appears. That is, $u_1=T$. $P(R_1|u_1) = ?$

$$\sum_{r_0} P(R_1 | r_0) P(r_0)$$

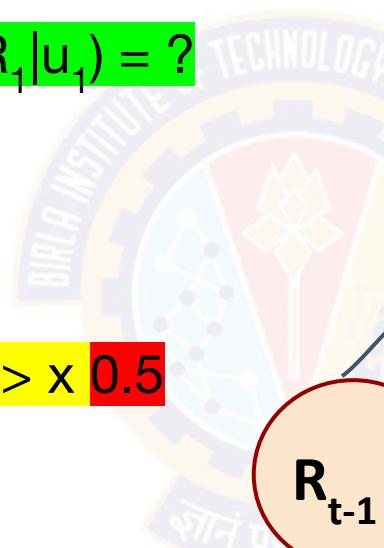
$$= <0.7, 0.3> \times 0.5 + <0.3, 0.7> \times 0.5$$

$$= <0.5, 0.5>$$

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8



State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

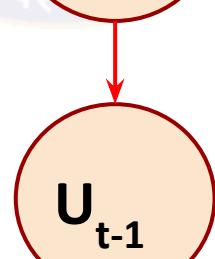
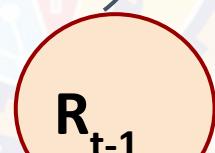
Umbrella appears. That is, $u_1=T$.

Computing the remainings

$$P(R_1|u_1) = \alpha P(u_1 | R_1) <0.5, 0.5>$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \boxed{\alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1})} \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

Umbrella appears. That is, $u_1=T$.

Computing the remainings

$$P(R_1|u_1) = \alpha P(u_1 | R_1) <0.5, 0.5>$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

R_{t-1}

U_{t-1}

R_t

U_t

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \boxed{\alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1})} \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

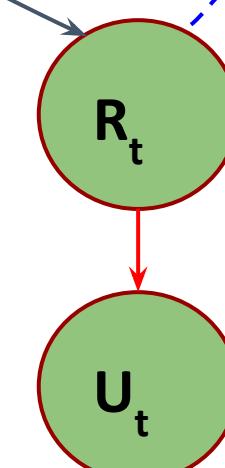
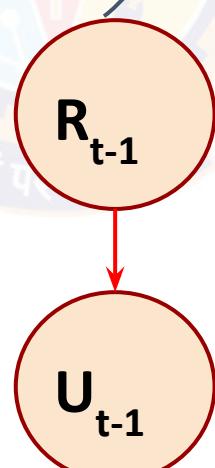
Umbrella appears. That is, $u_1=T$.

Computing the remainings

$$\begin{aligned} P(R_1|u_1) &= \alpha P(u_1 | R_1) <0.5, 0.5> \\ &= \alpha <0.9, 0.2> <0.5, 0.5> \\ &= \alpha <0.45, 0.1> \\ &= <0.45 / .55, 0.1 / 0.55> \\ &\approx <0.818, 0.182> \end{aligned}$$



	P(R_t=T)	P(R_t=F)
P(R_{t-1}=T)	0.7	0.3
P(R_{t-1}=F)	0.3	70.



	P(U_t=T)	P(U_t=F)
P(R_t=T)	0.9	0.1
P(R_t=F)	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \boxed{\alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1})} \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #1:

Umbrella appears. That is, $u_1=T$.

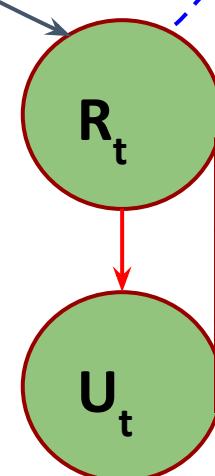
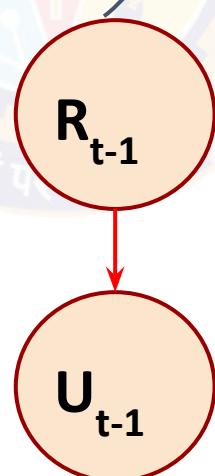
Computing the remainings

$$\begin{aligned} P(R_1|u_1) &= \alpha P(u_1 | R_1) <0.5, 0.5> \\ &= \alpha <0.9, 0.2> <0.5, 0.5> \\ &= \alpha <0.45, 0.1> \\ &= <0.45 / .55, 0.1 / 0.55> \\ &\approx <0.818, 0.182> \end{aligned}$$

When umbrella appears in day 1,
 $P(R_1|u_1) = <0.818, 0.182>$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



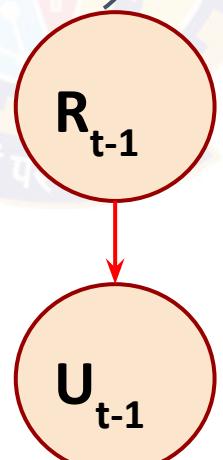
	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #2:

Umbrella appears. That is, $u_2=T$.



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

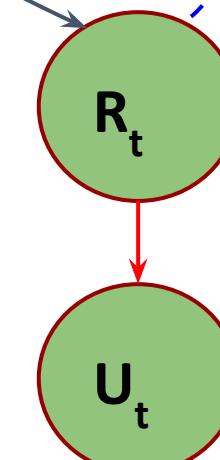
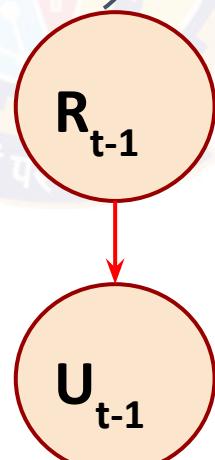
Day #2:

Umbrella appears. That is, $u_2=T$.

Compute $\sum_{r_1} P(R_2 | r_1) P(r_1 | u_1)$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

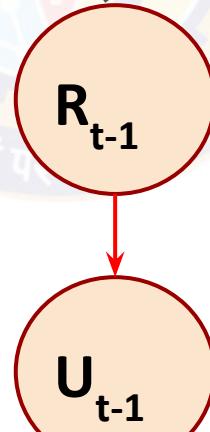
$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #2:

Umbrella appears. That is, $u_2=T$.

Compute $\sum_{r_1} P(R_2 | r_1) P(r_1 | u_1)$

We have just computed it.



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #2:

Umbrella appears. That is, $u_2=T$.

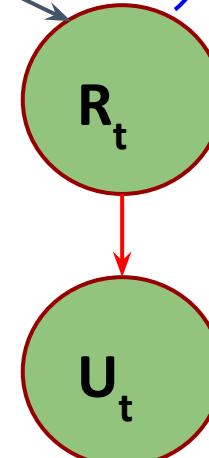
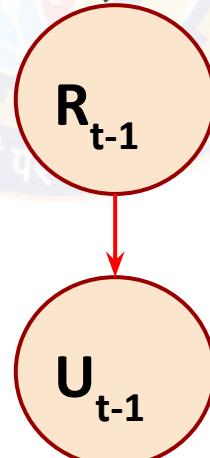
Compute $\sum_{r_1} P(R_2 | r_1) P(r_1 | u_1)$

$$\begin{aligned} &= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \\ &\approx \langle 0.627, 0.373 \rangle \end{aligned}$$

Stop here and verify that you can use the table entries to compute this !!!



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Complete this now.

Day #2:

Umbrella appears. That is, $u_2=T$.

Compute $\sum_{r_1} P(R_2 | r_1) P(r_1 | u_1)$

$$= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182$$

$$\approx \langle 0.627, 0.373 \rangle$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Complete this now.

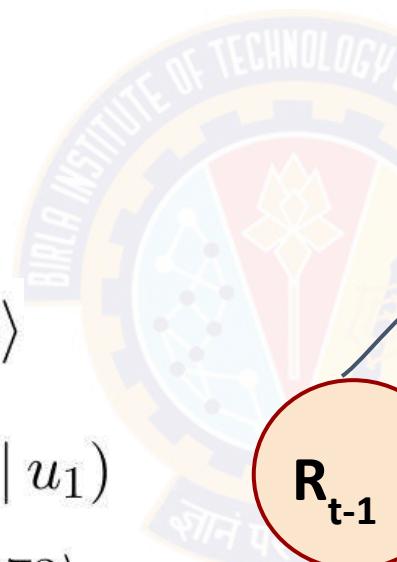
Day #2:

Umbrella appears. That is, $u_2=T$.

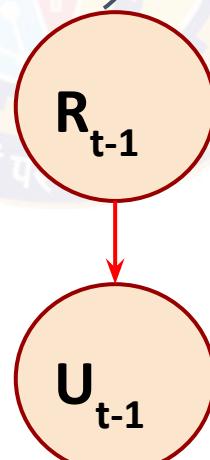
Compute $\sum_{r_1} P(R_2 | r_1) P(r_1 | u_1)$

$$\approx \langle 0.627, 0.373 \rangle$$

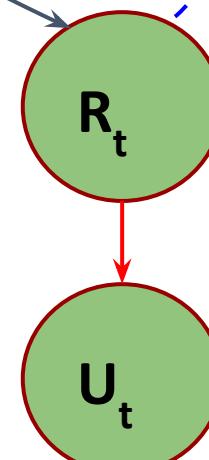
$$\begin{aligned} P(R_2 | u_1, u_2) &= \alpha P(u_2 | R_2) P(R_2 | u_1) \\ &= \alpha \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle \\ &\approx \langle 0.883, 0.117 \rangle \end{aligned}$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8



Stop here and verify that you can use the table entries to compute this !!!

State Estimation Problem (Filtering)

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Complete this now.

Day #0:

$$P(R_0) = <0.5, 0.5>$$

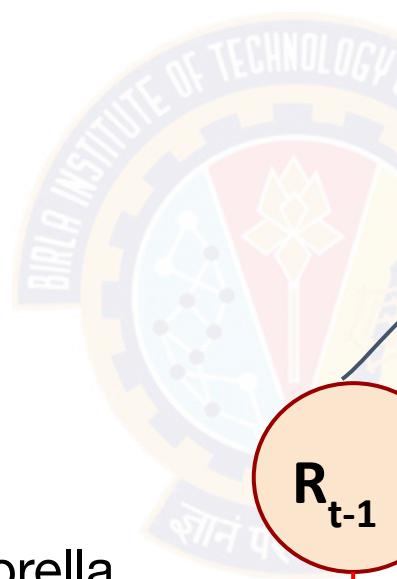
Day #1:

$$P(R_1|u_1) = <0.818, 0.182>$$

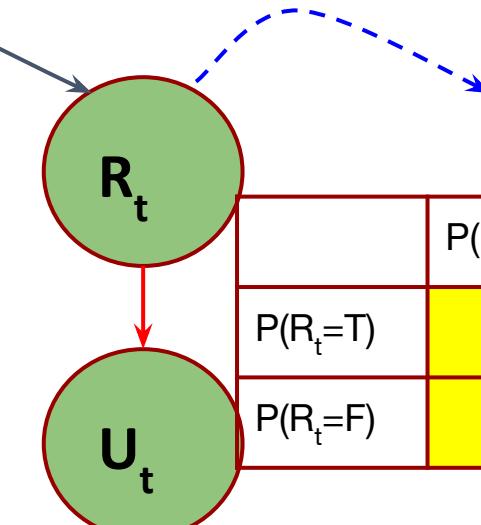
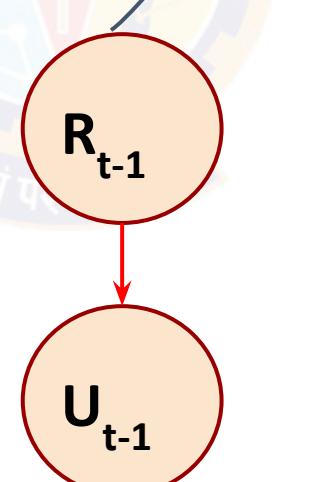
Day #2:

$$P(R_2|u_1u_2) = <0.883, 0.117>$$

With consecutive evidences with umbrella,
belief on rain increases !!!



	P(R _t =T)	P(R _t =F)
P(R _{t-1} =T)	0.7	0.3
P(R _{t-1} =F)	0.3	70.



	P(U _t =T)	P(U _t =F)
P(R _t =T)	0.9	0.1
P(R _t =F)	0.2	0.8

Stop here and verify that you can use the table entries to compute this !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.8: Prediction with HMM

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

- General (k-step) prediction as an extension of one step prediction



Revising Filtering

How good are the clusters obtained?

- Recollect that the filtering task included one step prediction as a part

$$\begin{aligned} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) &= \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t}) \\ &= \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t}) \end{aligned}$$

Revising Filtering

- Recollect that the filtering task included one step prediction as a part

$$\begin{aligned} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) &= \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \boxed{\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t})} \\ &= \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t}) \end{aligned}$$

One step prediction !!!

Prediction

- Can we extend one step prediction to predicting further into the future?

t	0	1	...	t	t+1	t+2	...	t+k+1
u	-	T	T	F	no evidences			
R	[0.5, 0.5] [Prior]				?	?	?	?

Prediction

- Can we extend one step prediction to predicting further into the future?

t	0	1	...	t	t+1	t+2	...	t+k+1
u	-	T	T	F	no evidences			
R	[0.5, 0.5] [Prior]				?	?	?	?

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

Generalizing one step prediction into k-steps ahead

$$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k}) P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t})$$

Prediction

$$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k}) P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t})$$

- As we keep predicting into the future without adding more evidences, the posterior converges to $\langle 0.5, 0.5 \rangle$
 - Ex. Predict the three states R_3, R_4, R_5 from the previous ex. on filtering [where we have computed $P(R_2|u_1u_2) = \langle 0.883, 0.117 \rangle$]
 - You can observe that the distribution over R slowly converges to stationary.

Prediction

$$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k}) P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t})$$

- As we keep predicting into the future without adding more evidences, the posterior converges to $\langle 0.5, 0.5 \rangle$
 - Ex. Predict the three states R_3, R_4, R_5 from the previous ex. on filtering [where we have computed $P(R_2|u_1u_2) = \langle 0.883, 0.117 \rangle$]
 - You can observe that the distribution over R slowly converges to stationary.
 - The amount of time a prediction takes to reach this stationary distribution is *mixing time*
 - It is not recommended, trying to keep predicting the future more and more into future without evidences (beyond a fraction of mixing time)



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.9: Smoothing with HMM - Forward-Backward Algorithm

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani



In this segment

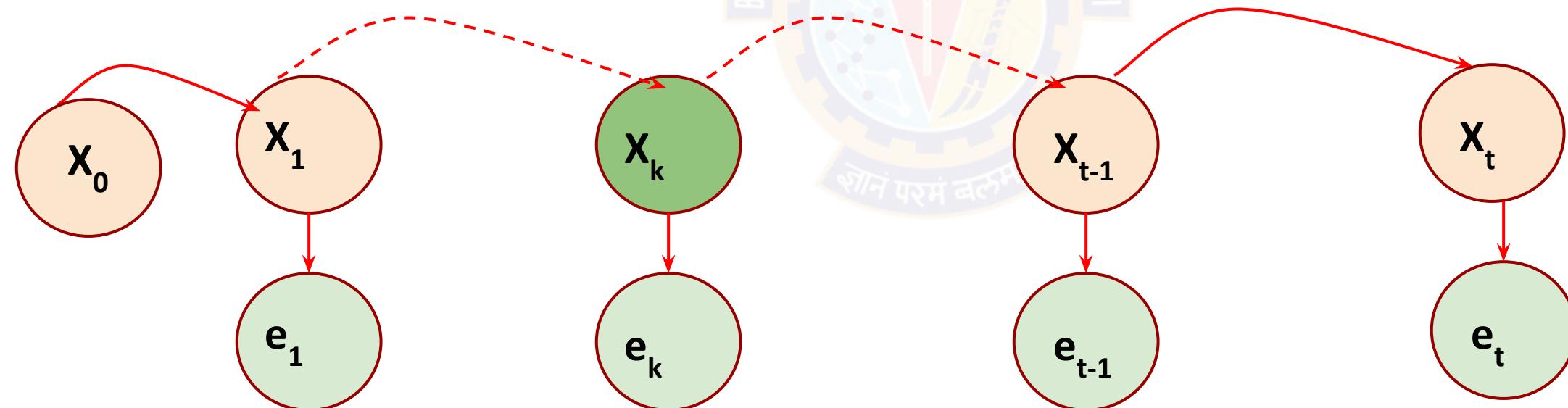
- Understanding smoothing
- Forward-Backward Algorithm



Smoothing / Most Likely State Estimation

Problem

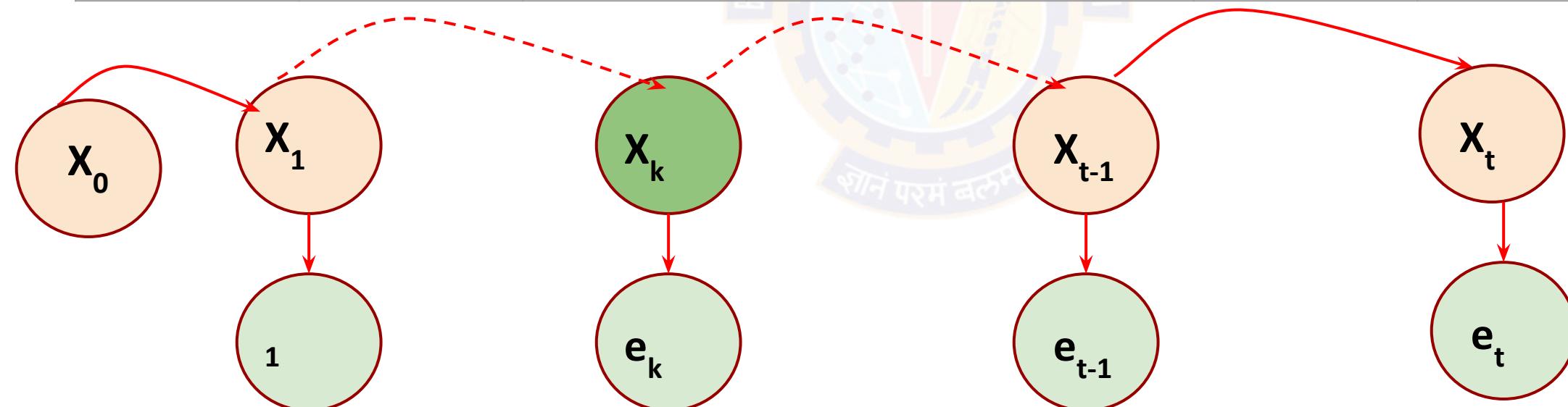
- Find the distribution over the past states given evidence up to the present
 - Smoothing / Most likely state estimation
- Given evidences $[e_1, e_2, e_3 \dots e_{t-1}, e_t]$:
Find $P(X_k | [e_{1:t}])$ for $0 \leq k < t$



Smoothing / Most Likely State Estimation

Example

t	0	1	2	3	4	5
u	-	T	T	T	F	F
R				?		



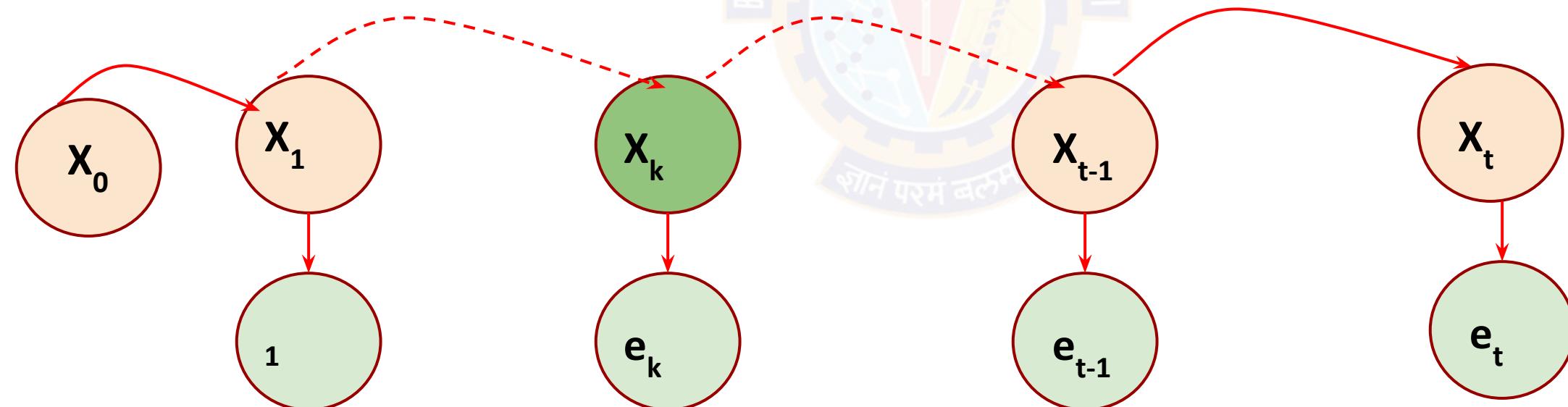
Smoothing / Most Likely State Estimation

Derivation

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$P(\mathbf{X}_k | \mathbf{e}_{1:t}) = P(\mathbf{X}_k | \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !



Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$P(\mathbf{X}_k | \mathbf{e}_{1:t}) = P(\mathbf{X}_k | \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha P(\mathbf{X}_k | \mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t} | \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

Computed from earlier Forward(.)

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$P(\mathbf{X}_k | \mathbf{e}_{1:t}) = P(\mathbf{X}_k | \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha P(\mathbf{X}_k | \mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t} | \mathbf{X}_k, \mathbf{e}_{1:k})$$

Apply Bayes Rule

State \mathbf{X}_k summarizes evidences upto k and given this, $\mathbf{e}_{k+1:t}$ is independent of $\mathbf{e}_{1:k}$

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \text{ Apply C.I.}$$

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)} \text{ Apply C.I.}$$

Let us call this term as backward estimation
(for some reasons !!!) and proceed to derive
this separately

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \text{ Apply C.I.}$$

$$= \alpha \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

Smoothing / Most Likely State Estimation

Strategy

- Need to estimate a state $[0, t]$ given all the evidence
 - Look for a recursive formulation

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})$$

You can split evidence such that first part is upto k & second is from k+1 !

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \text{ Apply Bayes Rule}$$

$$= \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \text{ Apply C.I.}$$

$$= \alpha \mathbf{f}_{1:k} \times \boxed{\mathbf{b}_{k+1:t}} \text{ Backward message !!!}$$

Smoothing / Most Likely State Estimation

Backward Algorithm

$\mathbf{b}_{k+1:t} = \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)$ Let us condition this on x_{k+1} .

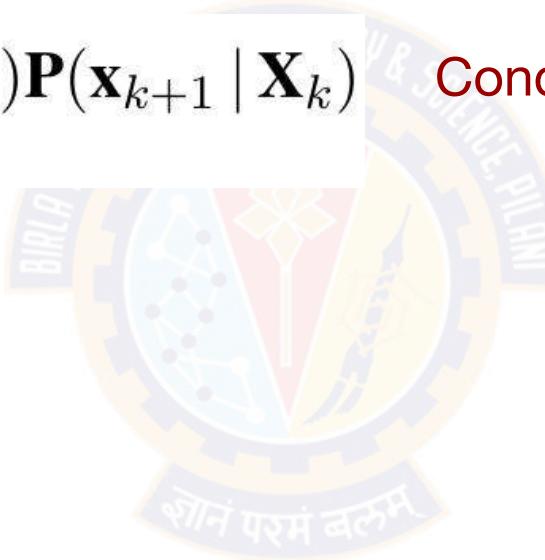


Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} \mid \underline{\mathbf{X}_k, \mathbf{x}_{k+1}}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

Conditioning on \mathbf{x}_{k+1} .



Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} \mid \underline{\mathbf{X}_k, \mathbf{x}_{k+1}}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \quad \text{Conditioning on } \mathbf{x}_{k+1}. \\ &= \sum_{\mathbf{x}_{k+1}} P(\underline{\mathbf{e}_{k+1:t} \mid \mathbf{x}_{k+1}}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \quad \text{Applying CI}\end{aligned}$$

Smoothing / Most Likely State Estimation

Backward Algorithm

$$\mathbf{b}_{k+1:t} = \mathbf{P}(\mathbf{e}_{k+1:t} | \mathbf{X}_k)$$

$$= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} | \mathbf{X}_k, \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} | \mathbf{X}_k) \quad \text{Conditioning on } \mathbf{x}_{k+1}$$

$$= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t} | \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} | \mathbf{X}_k) \quad \text{Applying CI}$$

$$= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}, \mathbf{e}_{k+2:t} | \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} | \mathbf{X}_k) \quad \text{Splitting evidences}$$

Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \quad \text{Conditioning on } \mathbf{x}_{k+1}. \\ &= \sum_{\mathbf{x}_{k+1}} \underbrace{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{x}_{k+1})}_{\text{Applying CI}} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \underbrace{\mathbf{P}(\mathbf{e}_{k+1}, \mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})}_{\text{Splitting evidences}} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \underbrace{\mathbf{P}(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})}_{\text{Applying Conditional Independence}} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

This is what we have got for backward message



Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} \underbrace{P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})}_{\text{Use sensor model directly}} \underbrace{\mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)}_{\text{Use transition model directly}}\end{aligned}$$

This is what we have got for backward message



Use sensor model
directly

Use transition model
directly

Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)} \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) \boxed{P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

Recursive Computation !!!



Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)} \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) \boxed{P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

Recursive Computation !!!

Written simply,

$$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$



Smoothing / Most Likely State Estimation

Backward Algorithm

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)} \xleftarrow{\text{Recursive Computation !!!}} \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) \boxed{P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})} \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

Written simply,

$$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$



Smoothed estimate / Most likely estimate is computed as

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \alpha \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

Given all the evidences , the entire smoothed state sequence can be computed in $O(t)$ time



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.9: Smoothing with HMM - Forward-Backward Algorithm (2)

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

- Understanding smoothing
- Forward-Backward Algorithm



Forward Backward Algorithm

Forward Backward Algorithm

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \alpha \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

$$\begin{aligned}\mathbf{f}_{1:t+1} &= \alpha \text{FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1}) \\ &= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})\end{aligned}$$

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1}) \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k)\end{aligned}$$

To smooth all the states:

- (1) Given n evidences, run filtering (forward) left to right once to compute smoothed estimates and save this.
- (2) Run backward algorithm in the other direction to get the smoothed estimate.

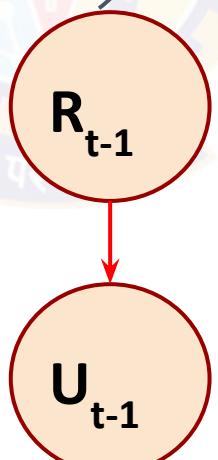
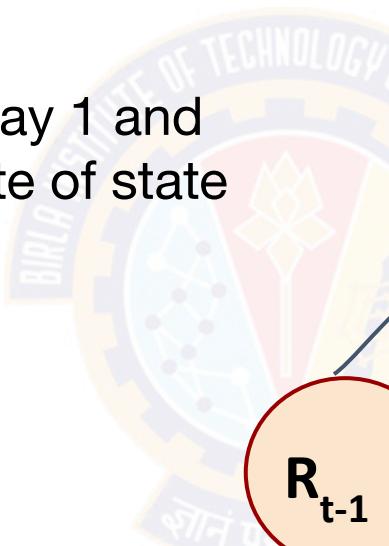
$O(t)$ Complexity .

Smoothing for a single state, or for the whole sequence has the same asymptotic complexity !!!

Filtering results for $[u_1=T, u_2=T]$

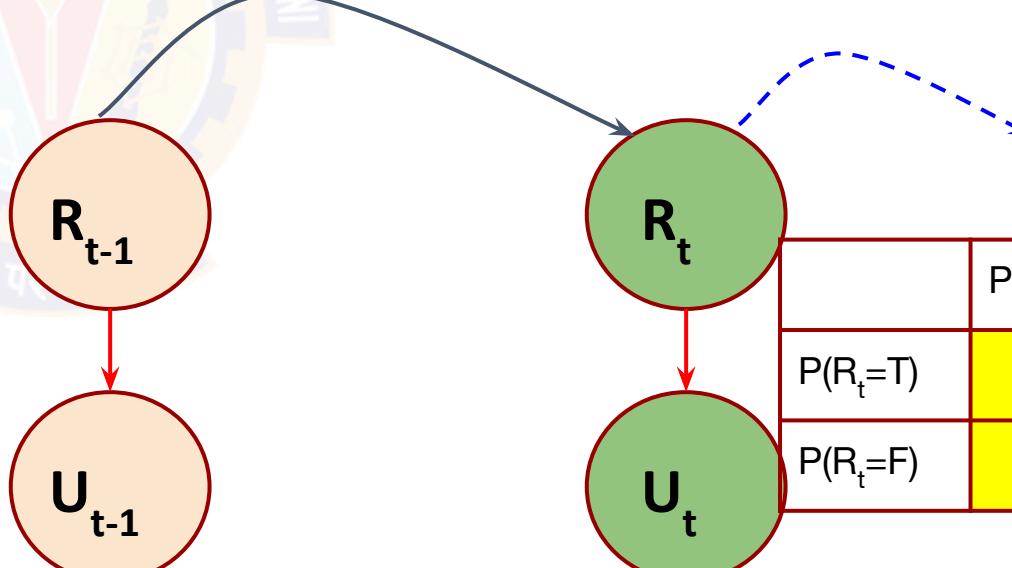
Task :

Given that umbrella is observed on day 1 and day 2, estimate the smoothed estimate of state for day #1



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8



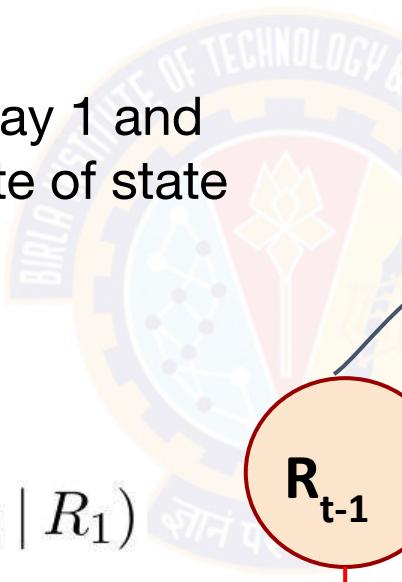
Filtering results for [u1=T, u2=T]

Task :

Given that umbrella is observed on day 1 and day 2, estimate the smoothed estimate of state for day #1

That is, find :

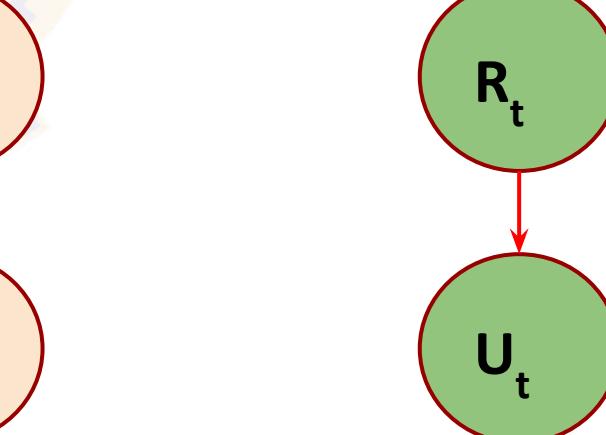
$$\mathbf{P}(R_1 | u_1, u_2) = \alpha \mathbf{P}(R_1 | u_1) \mathbf{P}(u_2 | R_1)$$



R_{t-1}

U_{t-1}

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



R_t

U_t

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Filtering results for [u1=T, u2=T]

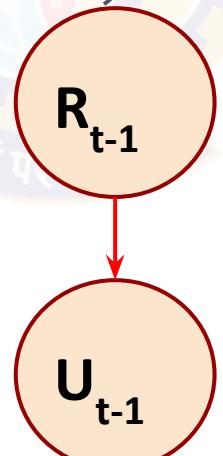
Task :

Given that umbrella is observed on day 1 and day 2, estimate the smoothed estimate of state for day #1

That is, find :

$$\mathbf{P}(R_1 | u_1, u_2) = \alpha \mathbf{P}(R_1 | u_1) \mathbf{P}(u_2 | R_1)$$

From forward algorithm



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Filtering results for $[u_1=T, u_2=T]$

$$P(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} P(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t})$$

Day #0: [Prior]

$$P(R_0) = <0.5, 0.5>$$

Day #1:

$$P(R_1|u_1) = <0.818, 0.182>$$

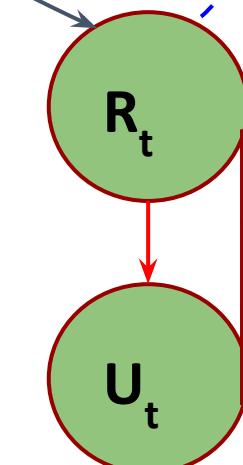
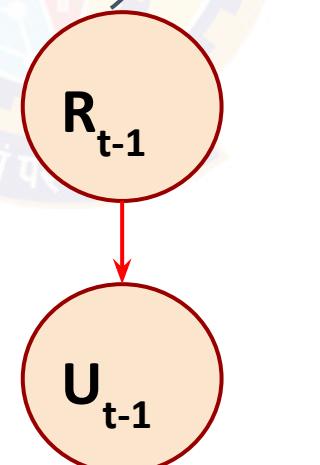
Day #2:

$$P(R_2|u_1u_2) = <0.883, 0.117>$$

With consecutive evidences with umbrella,
belief on rain increases !!!

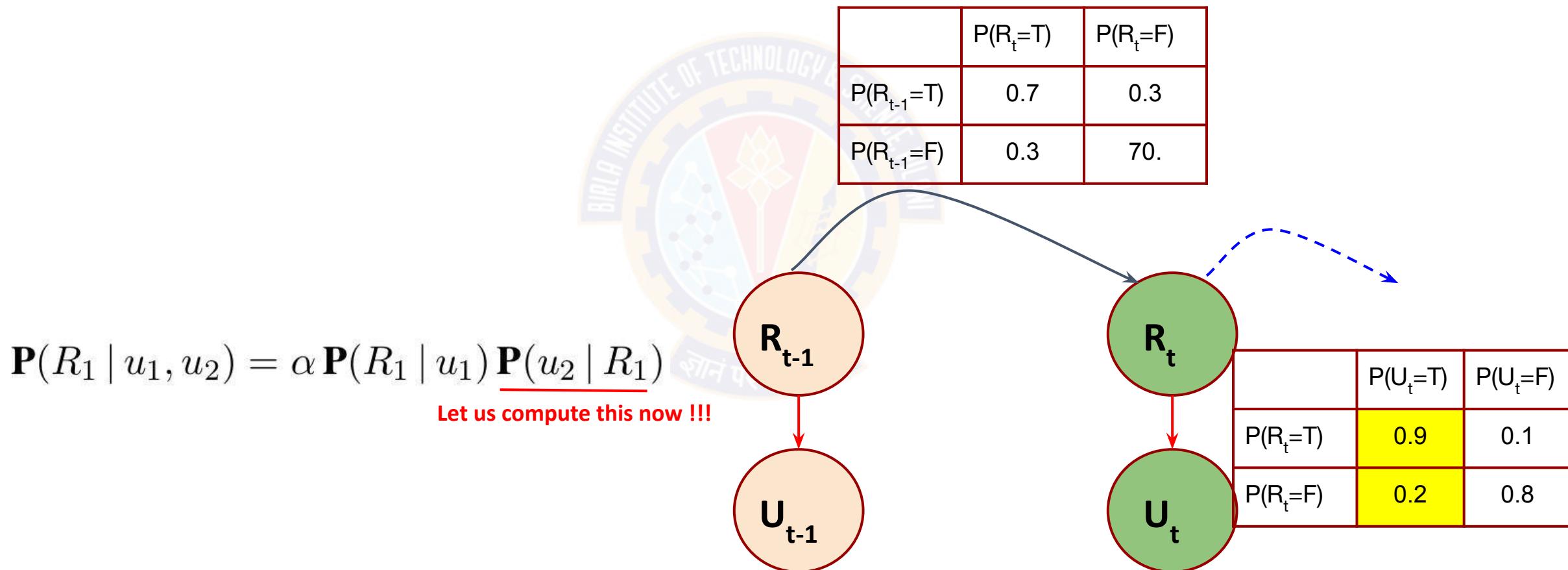
We have computed this
earlier !!!

	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.



	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Filtering results for [u₁=T, u₂=T]



Filtering results for [u1=T, u2=T]

$$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$

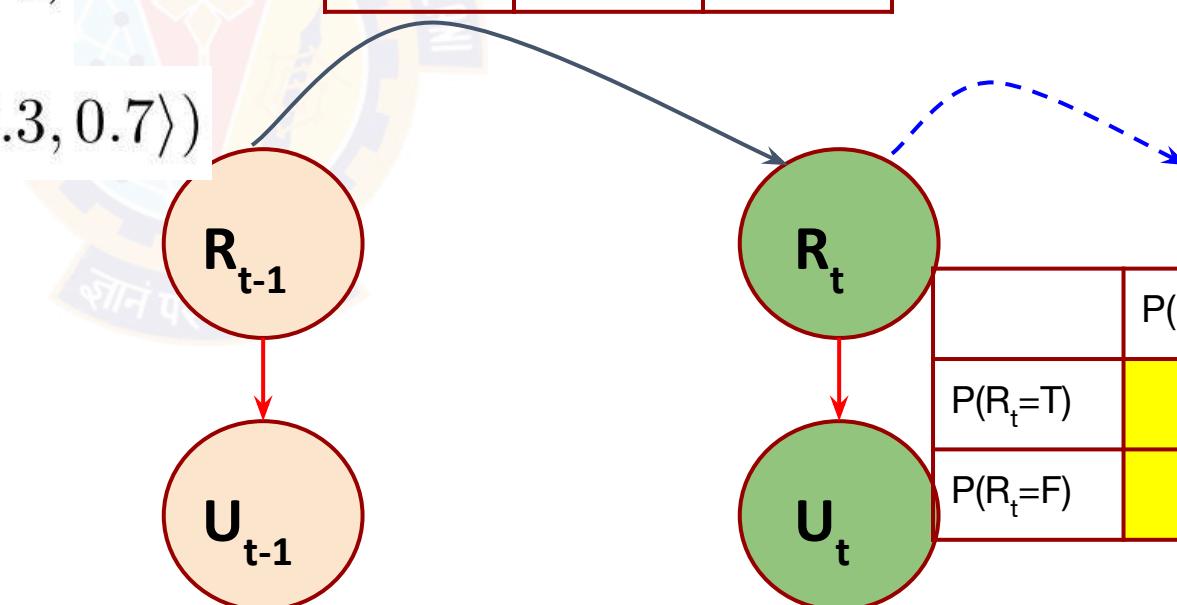
$$= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} | \mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t} | \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} | \mathbf{X}_k)$$

$$\mathbf{P}(u_2 | R_1) = \sum_{r_2} P(u_2 | r_2) P(r_2 | R_1)$$

$$= (0.9 \times 1 \times \langle 0.7, 0.3 \rangle) + (0.2 \times 1 \times \langle 0.3, 0.7 \rangle)$$

$$= \langle 0.69, 0.41 \rangle$$

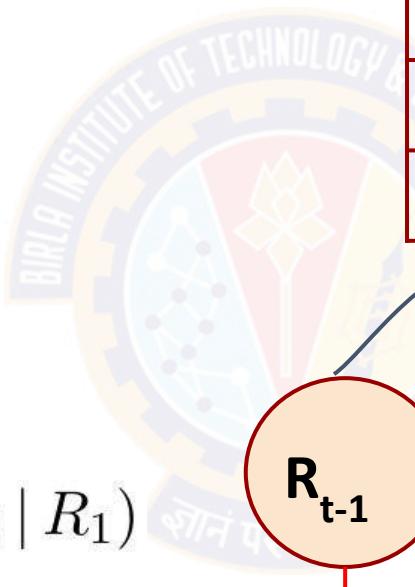
	P(R _t =T)	P(R _t =F)
P(R _{t-1} =T)	0.7	0.3
P(R _{t-1} =F)	0.3	70.



	P(U _t =T)	P(U _t =F)
P(R _t =T)	0.9	0.1
P(R _t =F)	0.2	0.8

Filtering results for [u1=T, u2=T]

$$\begin{aligned}\mathbf{P}(R_1 | u_1, u_2) &= \alpha \mathbf{P}(R_1 | u_1) \mathbf{P}(u_2 | R_1) \\ &= \alpha \langle 0.818, 0.182 \rangle \times \langle 0.69, 0.41 \rangle \\ &\approx \langle 0.883, 0.117 \rangle\end{aligned}$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

R_{t-1}

U_{t-1}

R_t

U_t

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

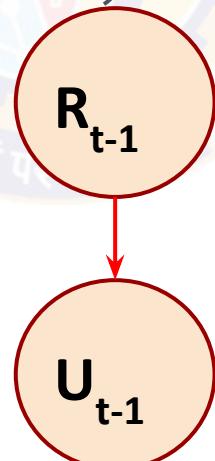
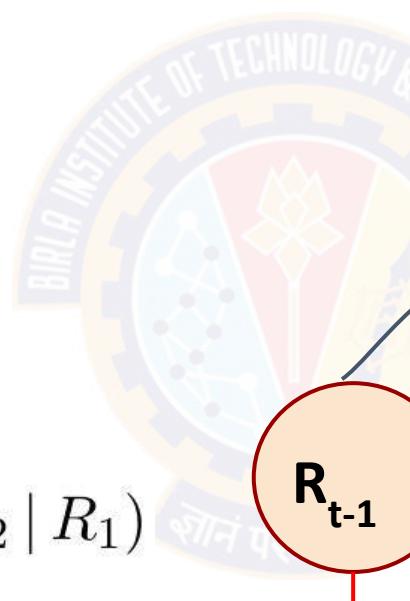
Filtering results for $[u_1=T, u_2=T]$

Earlier estimate while filtering

Day #1:

$$P(R_1 | u_1) = \langle 0.818, 0.182 \rangle$$

$$\begin{aligned} P(R_1 | u_1, u_2) &= \alpha P(R_1 | u_1) P(u_2 | R_1) \\ &= \alpha \langle 0.818, 0.182 \rangle \times \langle 0.69, 0.41 \rangle \\ &\approx \langle 0.883, 0.117 \rangle \end{aligned}$$



	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	70.

	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

While smoothing in the presence of more evidences, we are given to believe a little more that it rained on day 1.



Thank You!

Readings:
Section 15.2.2 of Artificial Intelligence - A Modern Approach, Russell & Norvig



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

8.11 Viterbi Algorithm - Most Likely Sequence in HMM

S.P.Vimal

Asst. Professor
WILP Division, Bits-Pilani

In this segment

- Viterbi Algorithm



Most Likely Sequence

What is it?

- Given a sequence of evidences what is the mostly likely sequence of states which could have generated this?



Most Likely Sequence

What is it?

- Given a sequence of evidences what is the mostly likely sequence of states which could have generated this?

For Ex:

t	1	2	3	4	5
Evidence Sequence (u)	T	T	F	T	T

Most Likely Sequence

What is it?

- Given a sequence of evidences what is the mostly likely sequence of states which could have generated this?

For Ex:

t	1	2	3	4	5
Evidence Sequence (u)	T	T	F	T	T
Possible State Sequence#1 [Rain]	T	T	T	T	T
Possible State Sequence#2 [Rain]	T	T	T	T	F
Possible State Sequence#3 [Rain]	T	T	T	F	T
....					

With more states, and state variables having more possible values the number of such sequences explodes exponentially

Most Likely Sequence

What is it?

- Given a sequence of evidences what is the mostly likely sequence of states which could have generated this?

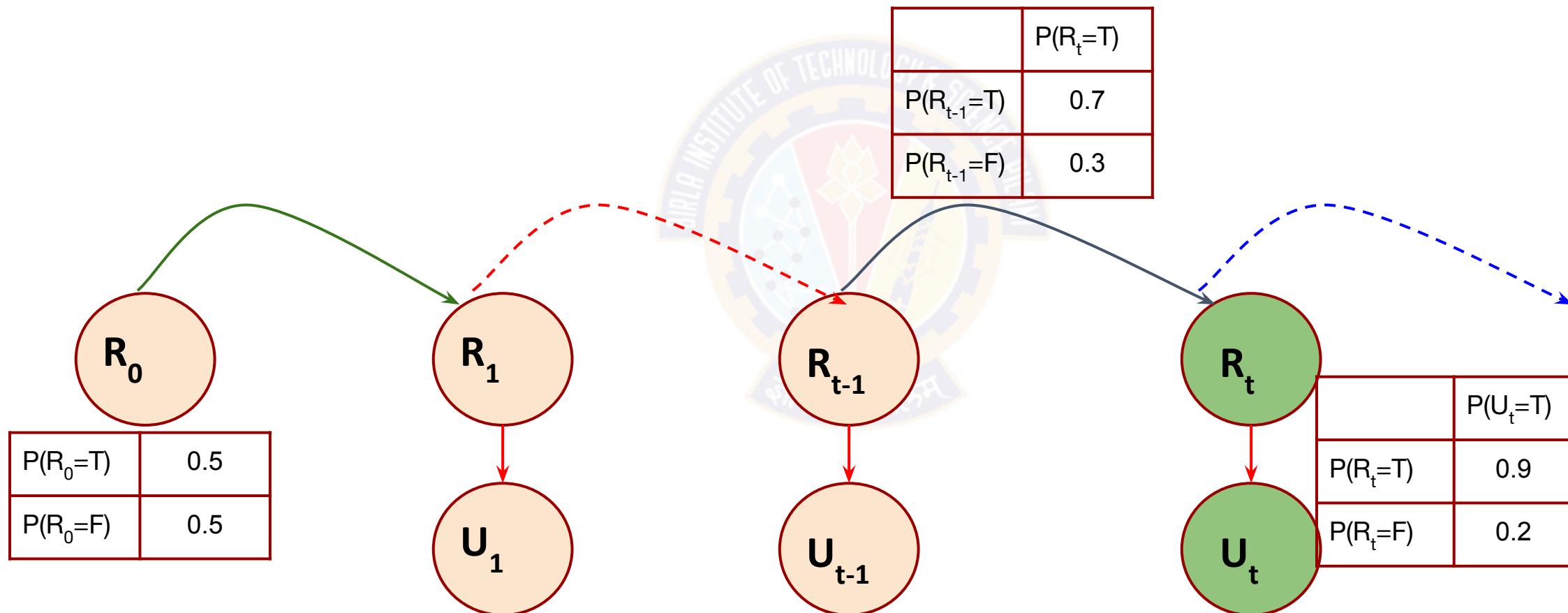
For Ex:

t	1	2	3	4	5
Evidence Sequence (u)	T	T	F	T	T
Possible State Sequence#1 [Rain]	T	T	T	T	T
Possible State Sequence#2 [Rain]	T	T	T	T	F
Possible State Sequence#3 [Rain]	T	T	T	F	T
....					

Let us understand the algorithm by first walking through the steps for a particular case of evidence sequence $u = [T, T, F, T, T]$

Most Likely Sequence

Recollect the running example



Viterbi Algorithm

$$U = [T, T, F, T, T]$$



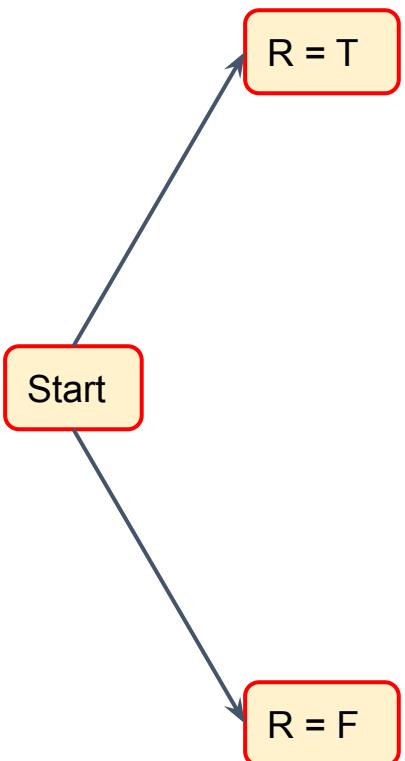
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



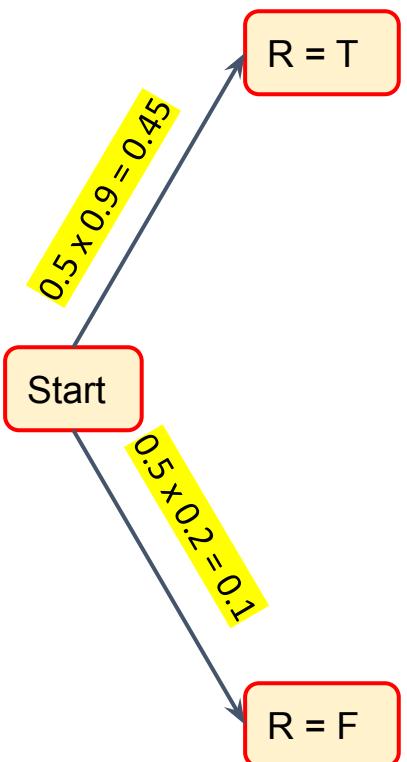
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



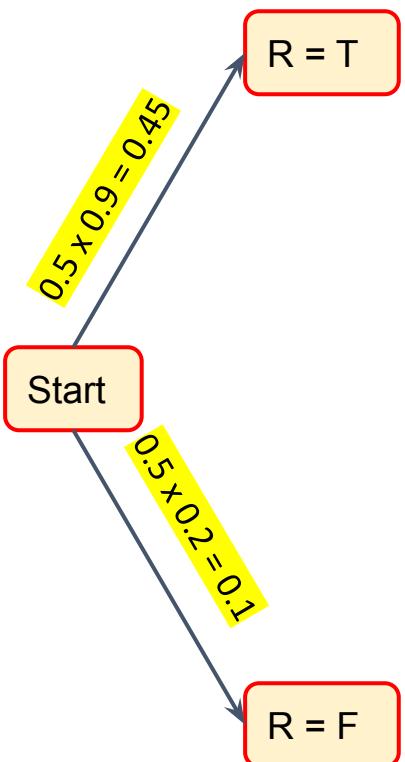
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

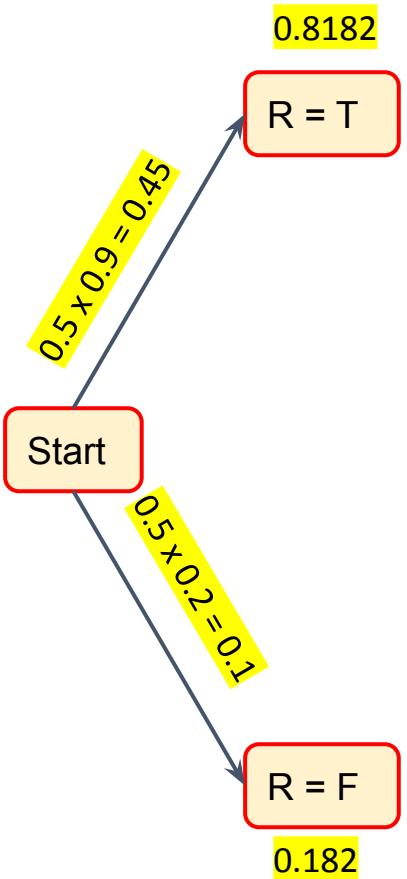
Normalizing this [not a necessary step, you can ignore this, but I take these scores]

0.45 becomes $0.45 / [0.45 + 0.1]$ which is 0.8182

0.1 becomes $0.1 / [0.45 + 0.1]$ which is 0.182

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



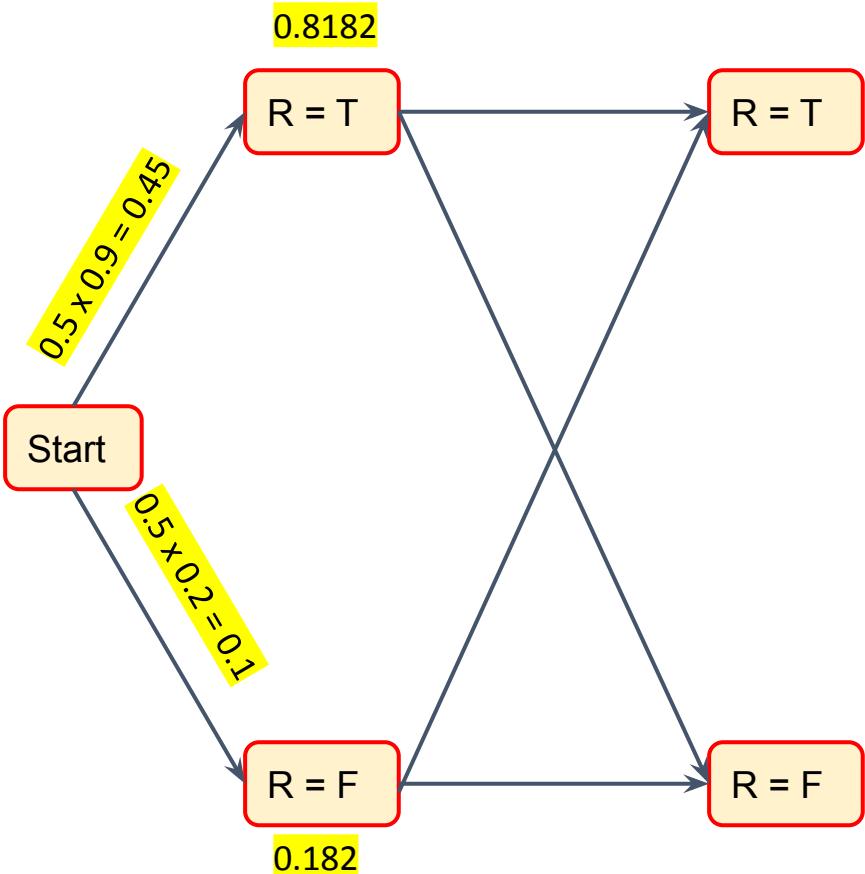
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



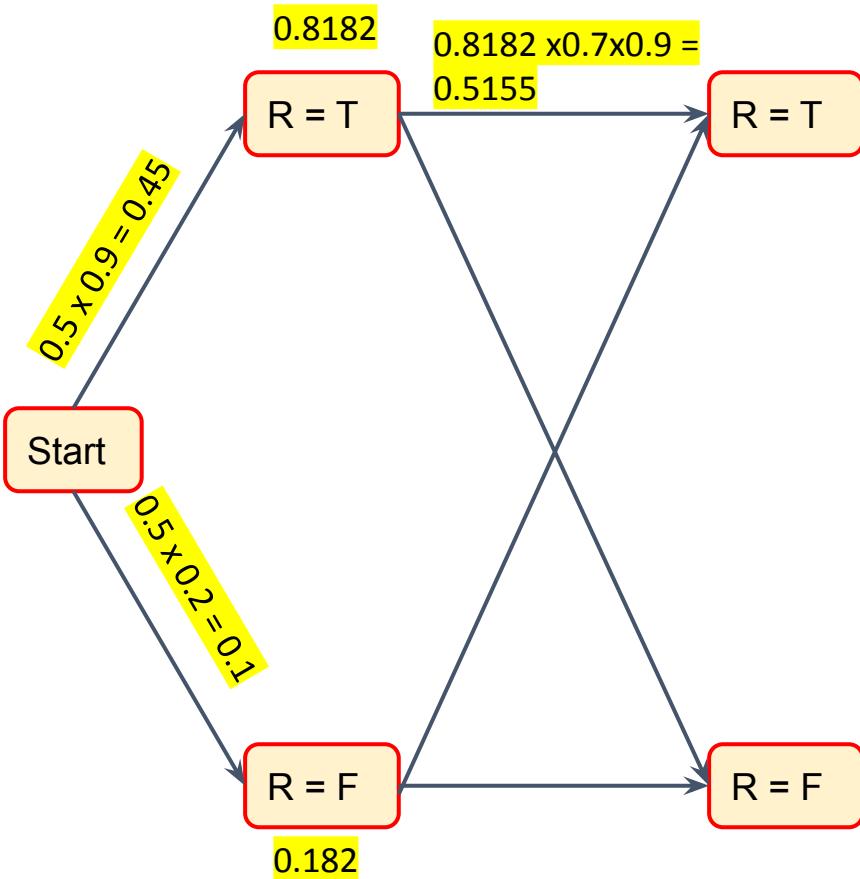
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



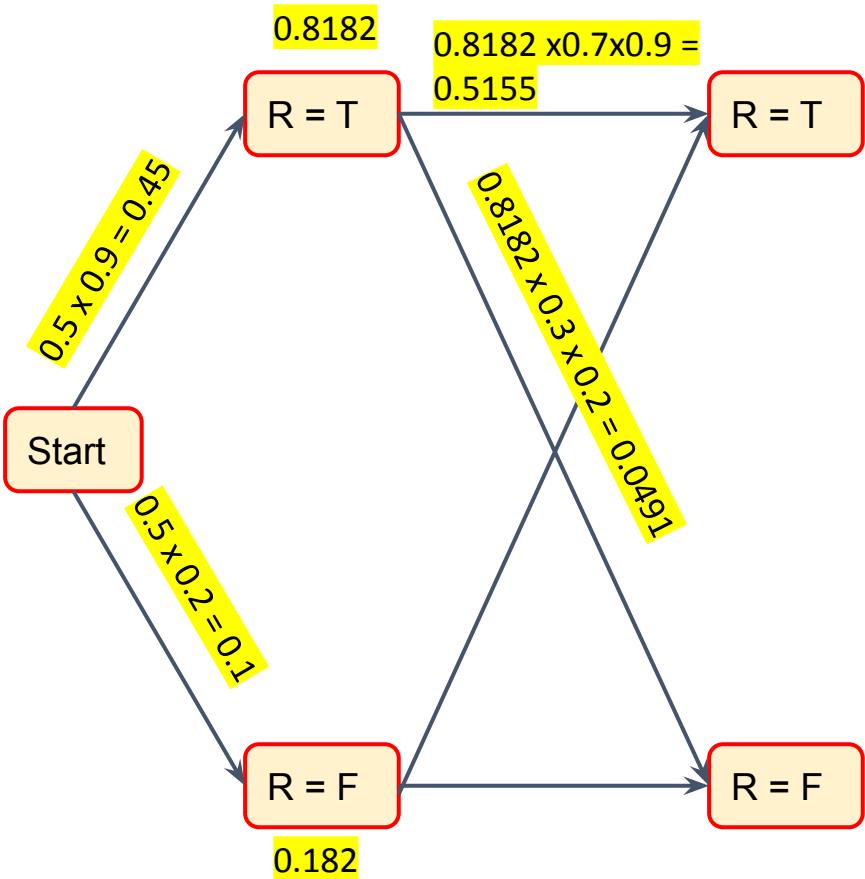
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



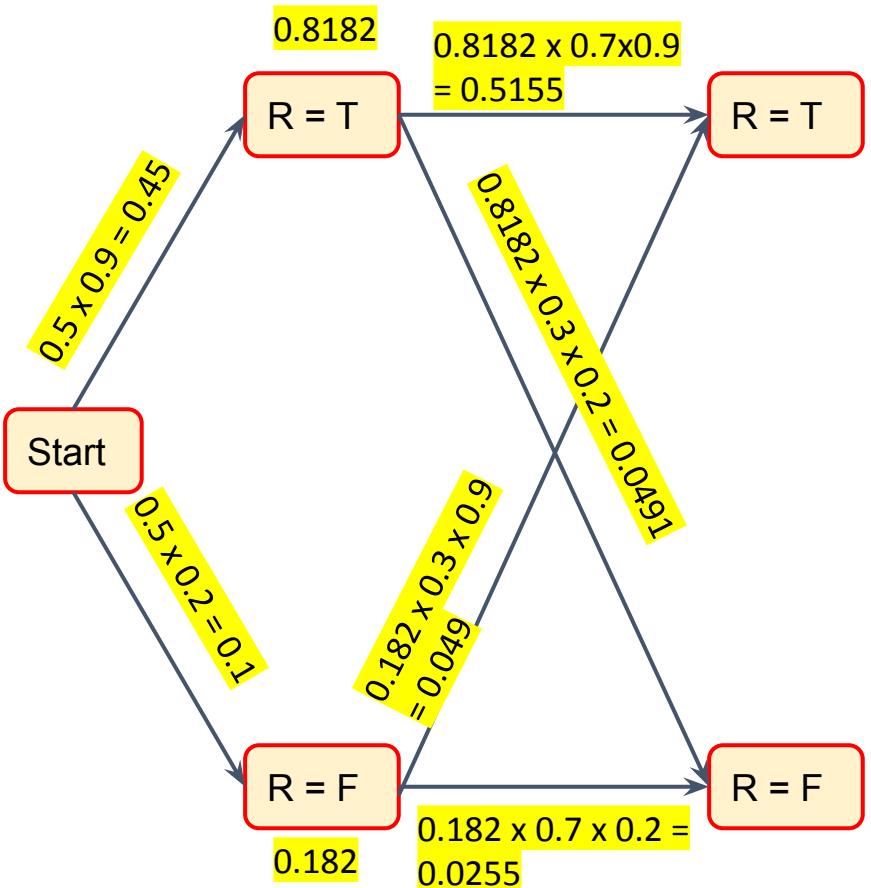
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



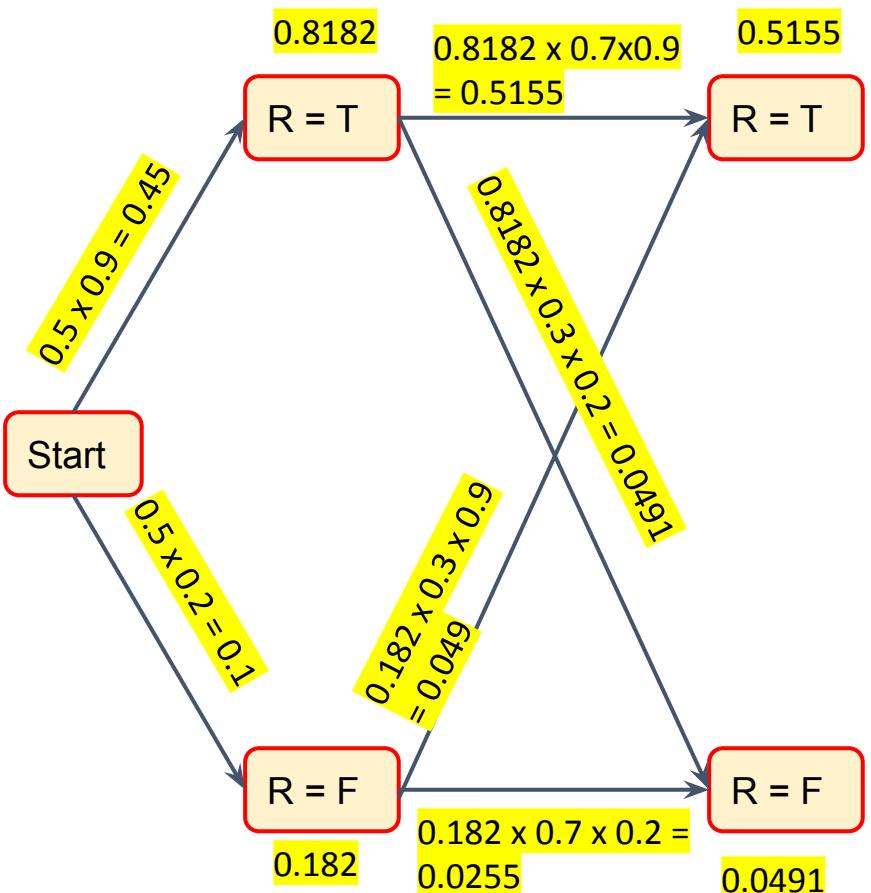
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



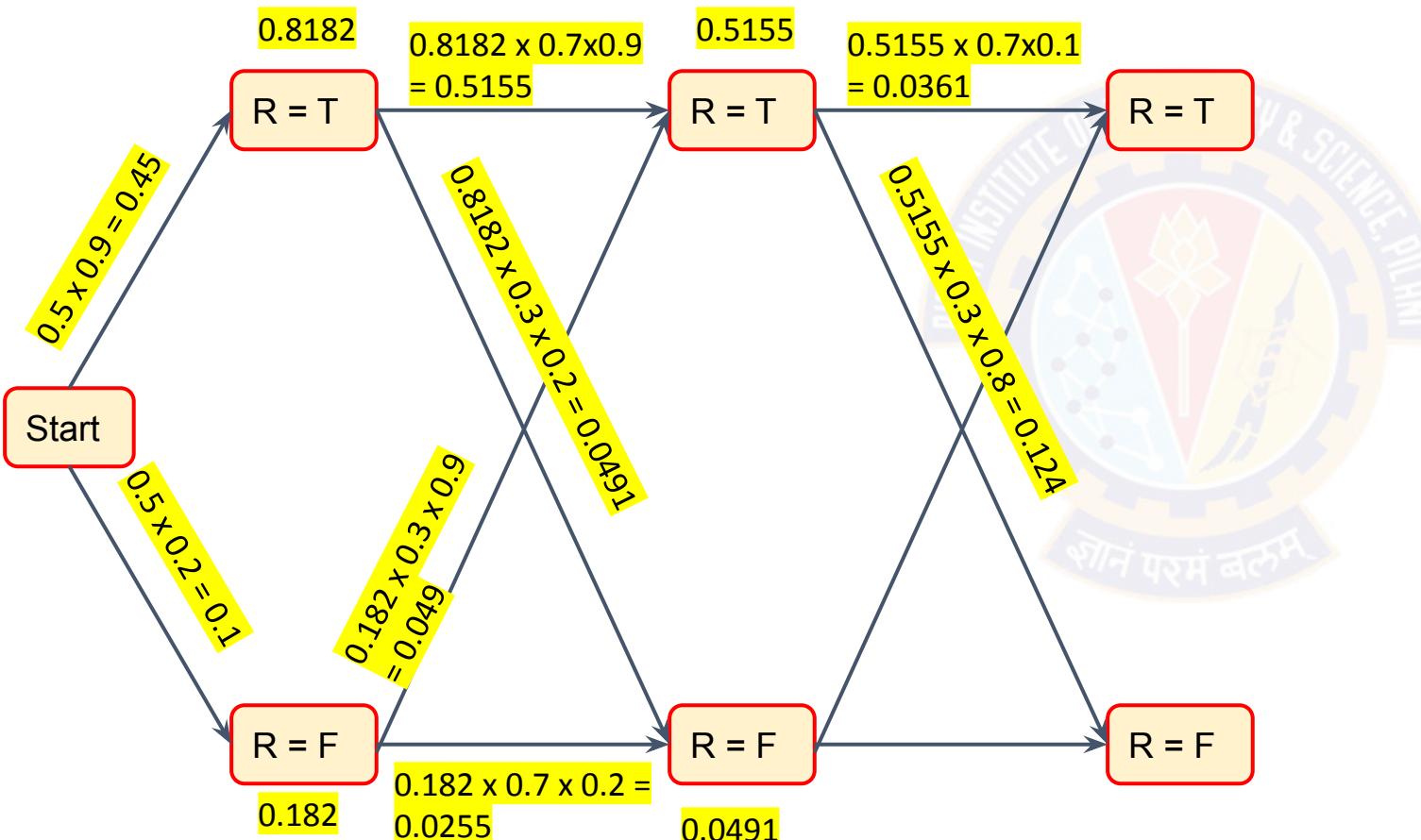
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



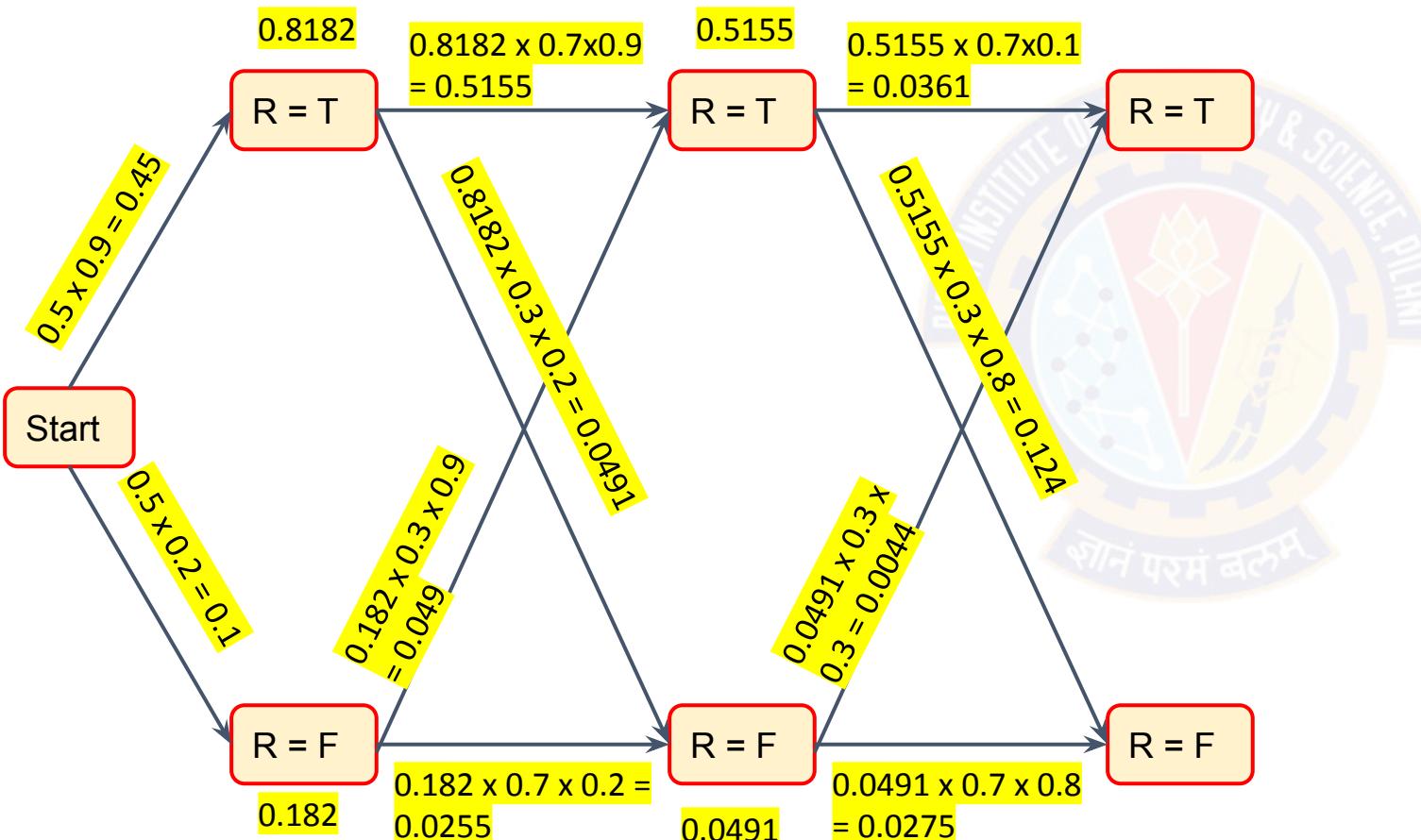
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



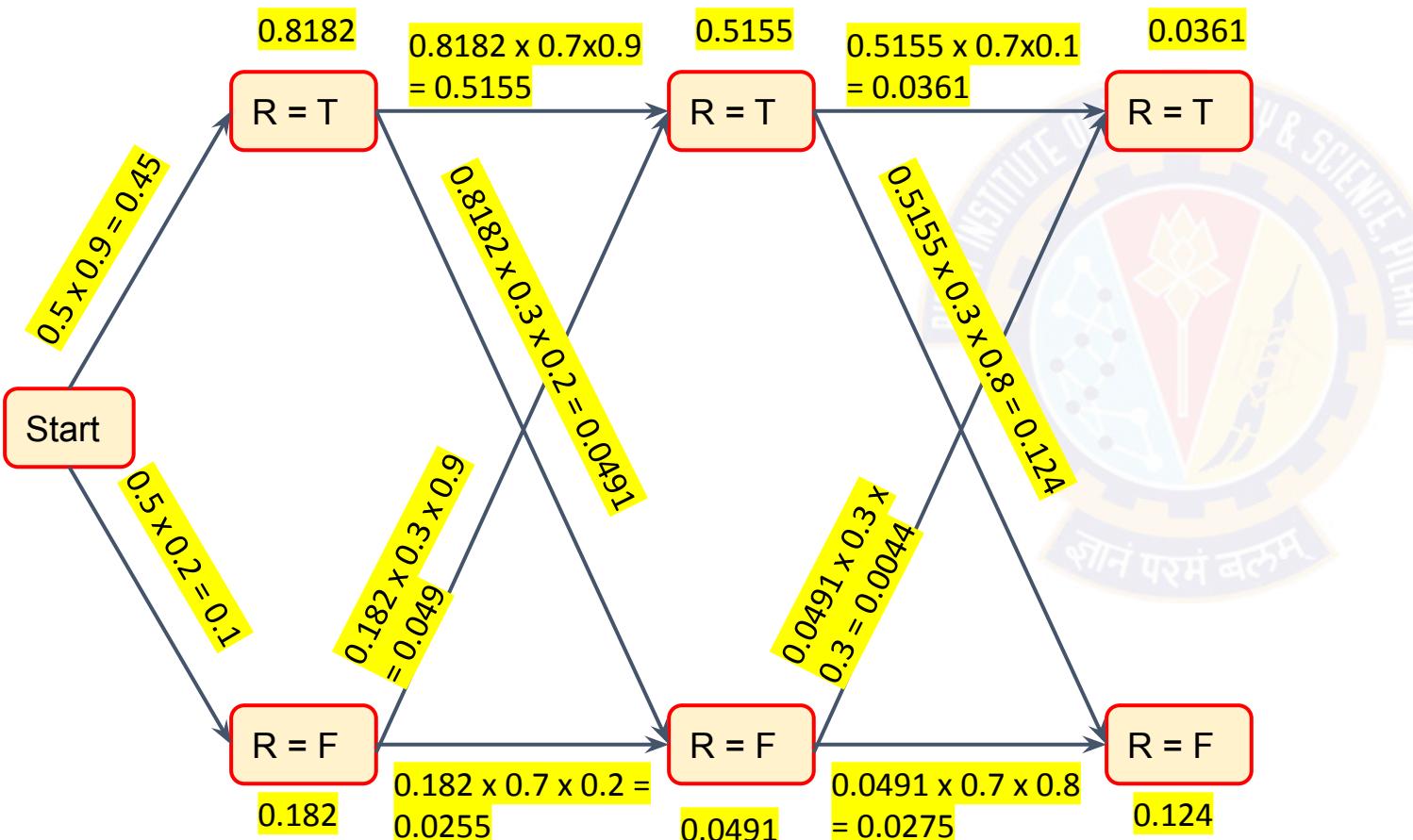
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



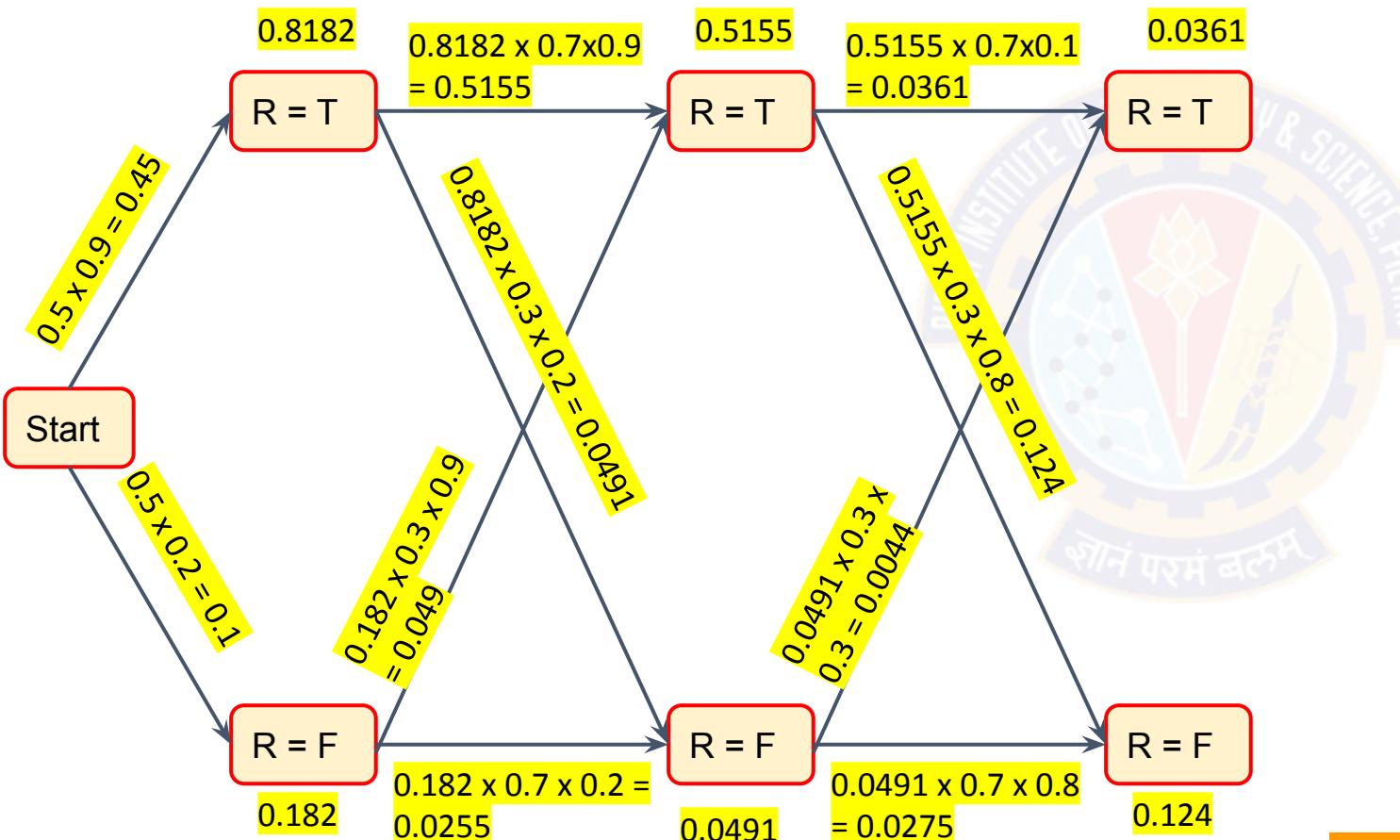
Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Viterbi Algorithm

$$U = [T, T, F, T, T]$$



Prior:	
$P(R_0=T)$	0.5
$P(R_0=F)$	0.5

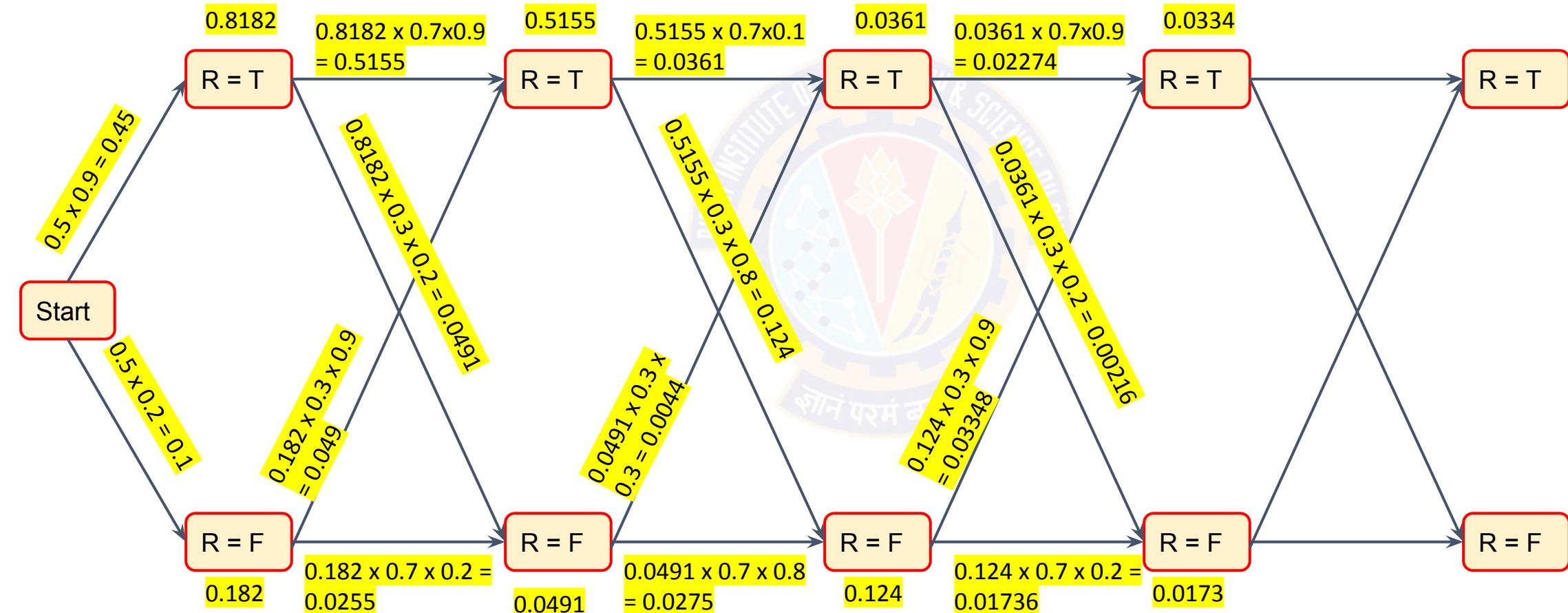
Transition	$P(R_t=T)$	$P(R_t=F)$
$P(R_{t-1}=T)$	0.7	0.3
$P(R_{t-1}=F)$	0.3	0.7

Sensor	$P(U_t=T)$	$P(U_t=F)$
$P(R_t=T)$	0.9	0.1
$P(R_t=F)$	0.2	0.8

Repeating this

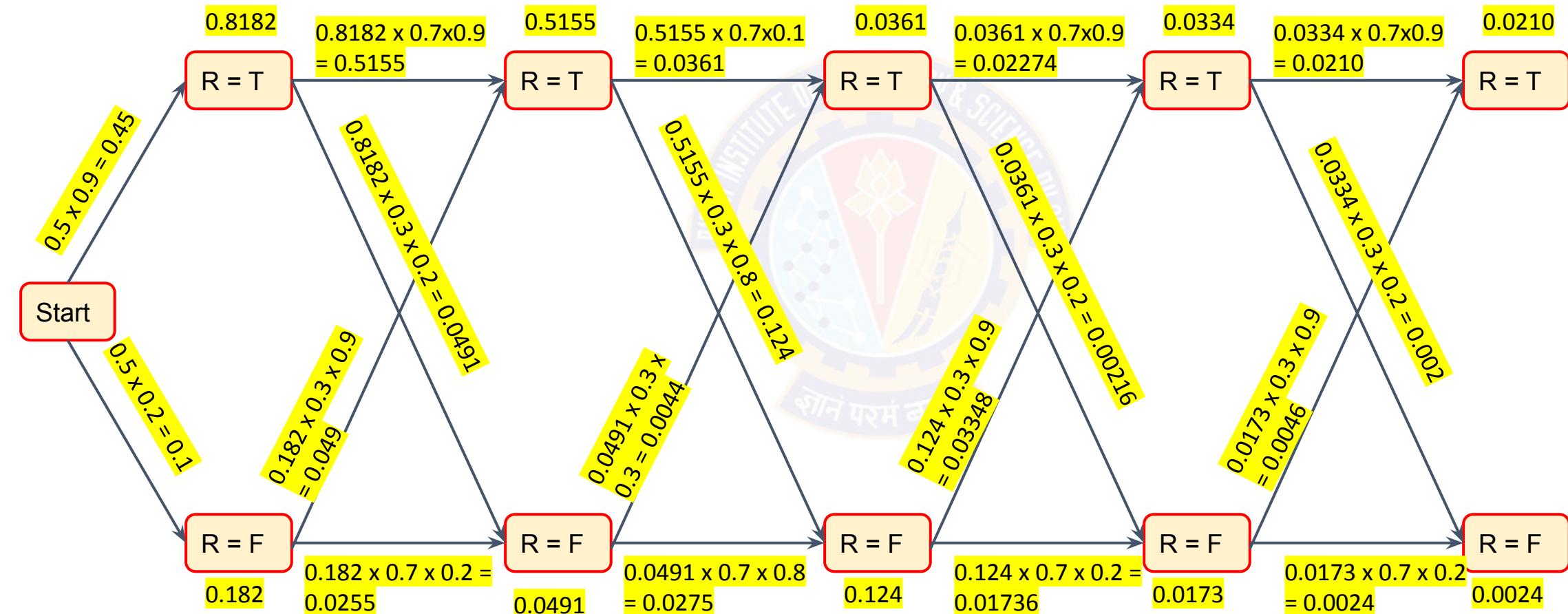
Viterbi Algorithm

$$U = [T, T, F, T, T]$$



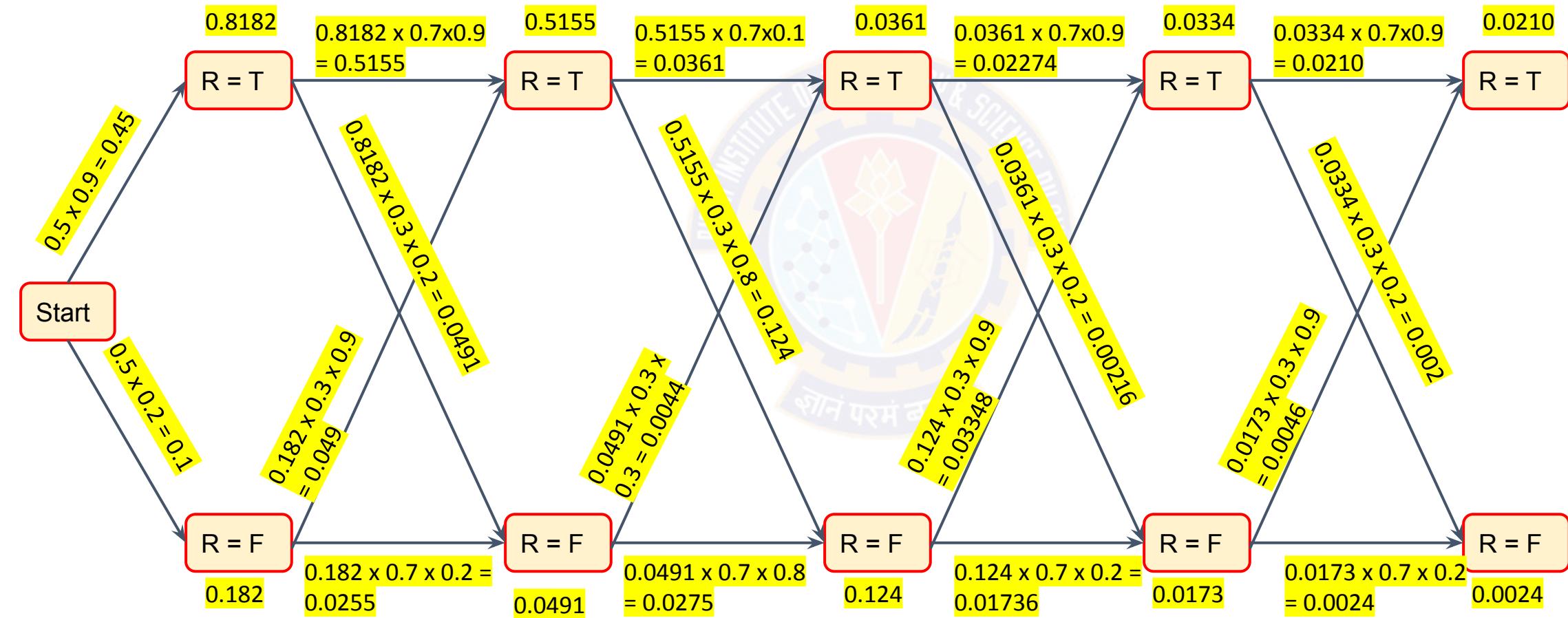
Viterbi Algorithm

$$U = [T, T, F, T, T]$$



Viterbi Algorithm

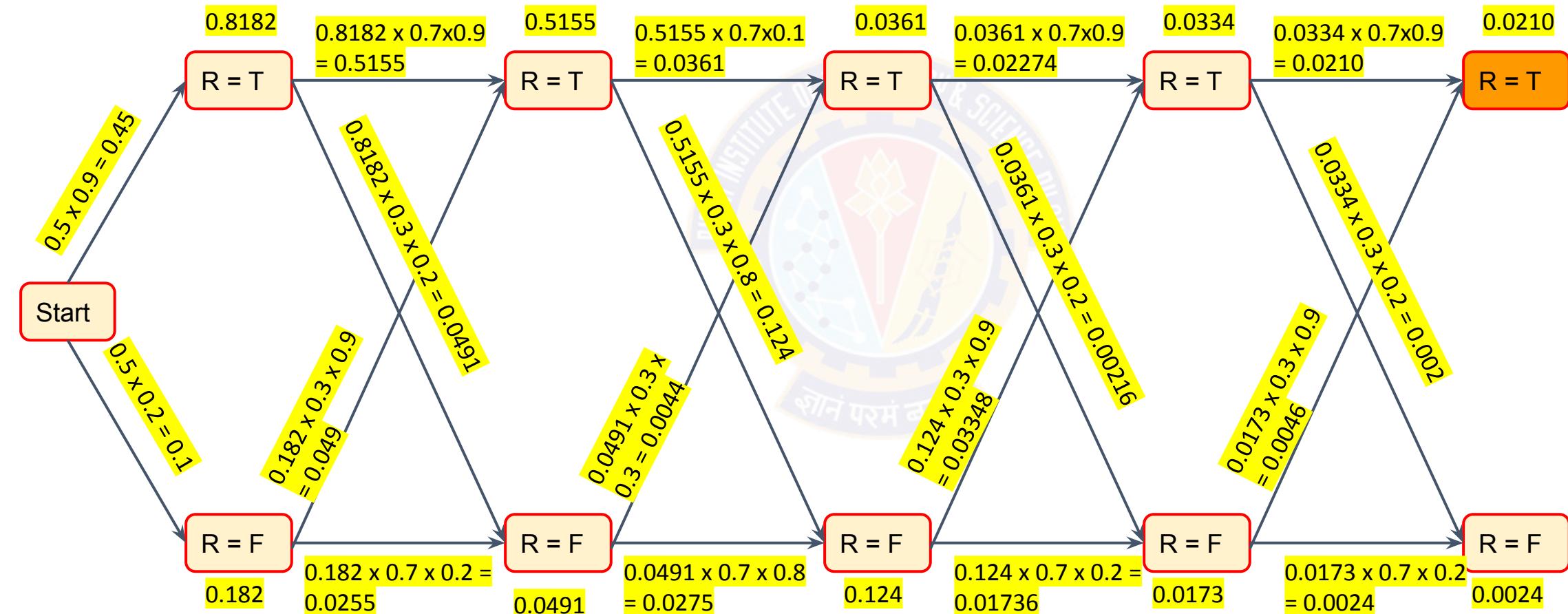
$$U = [T, T, F, T, T]$$



Backtrack to get the most likely state sequence now !!!

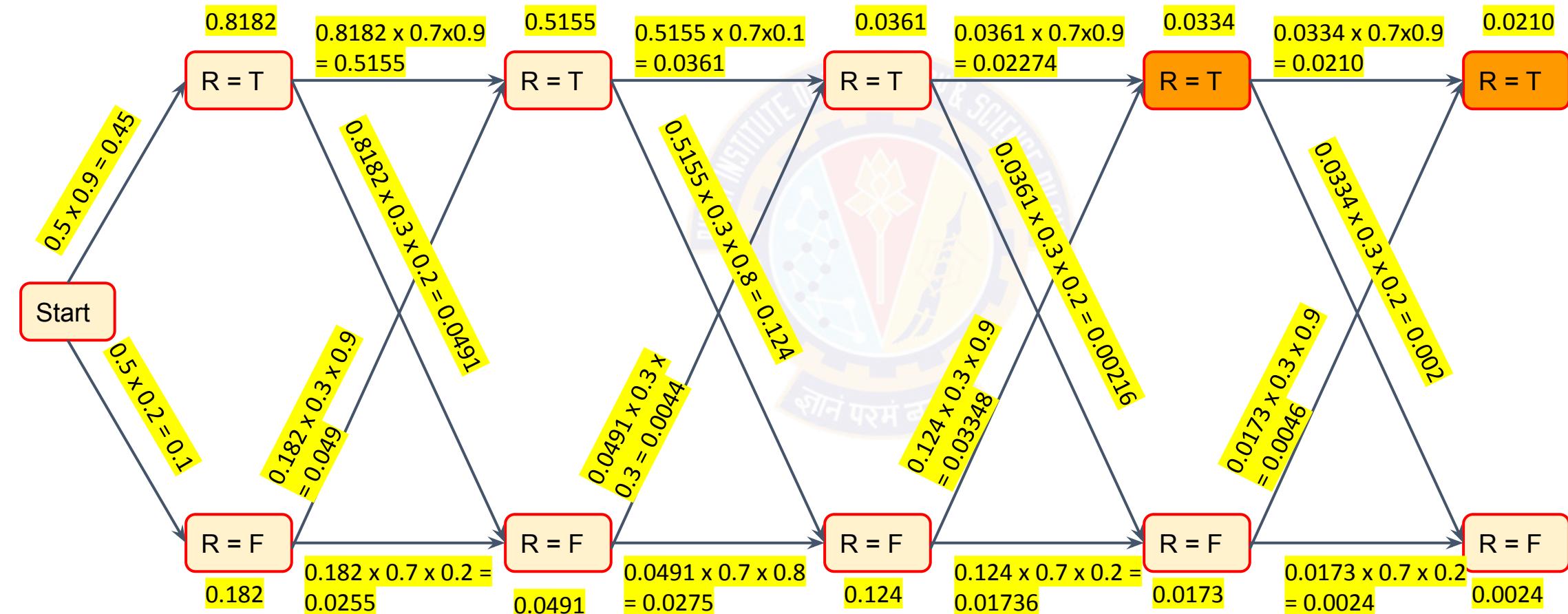
Viterbi Algorithm

$$U = [T, T, F, T, T]$$



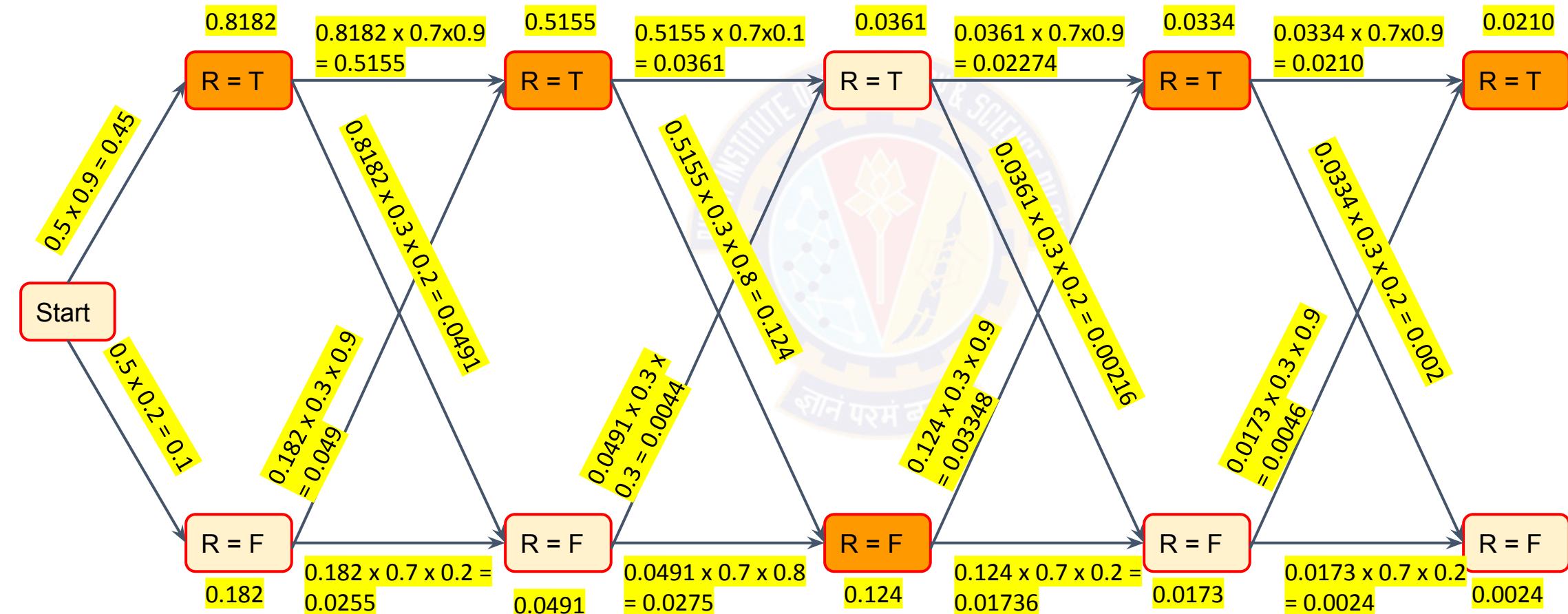
Viterbi Algorithm

$$U = [T, T, F, T, T]$$



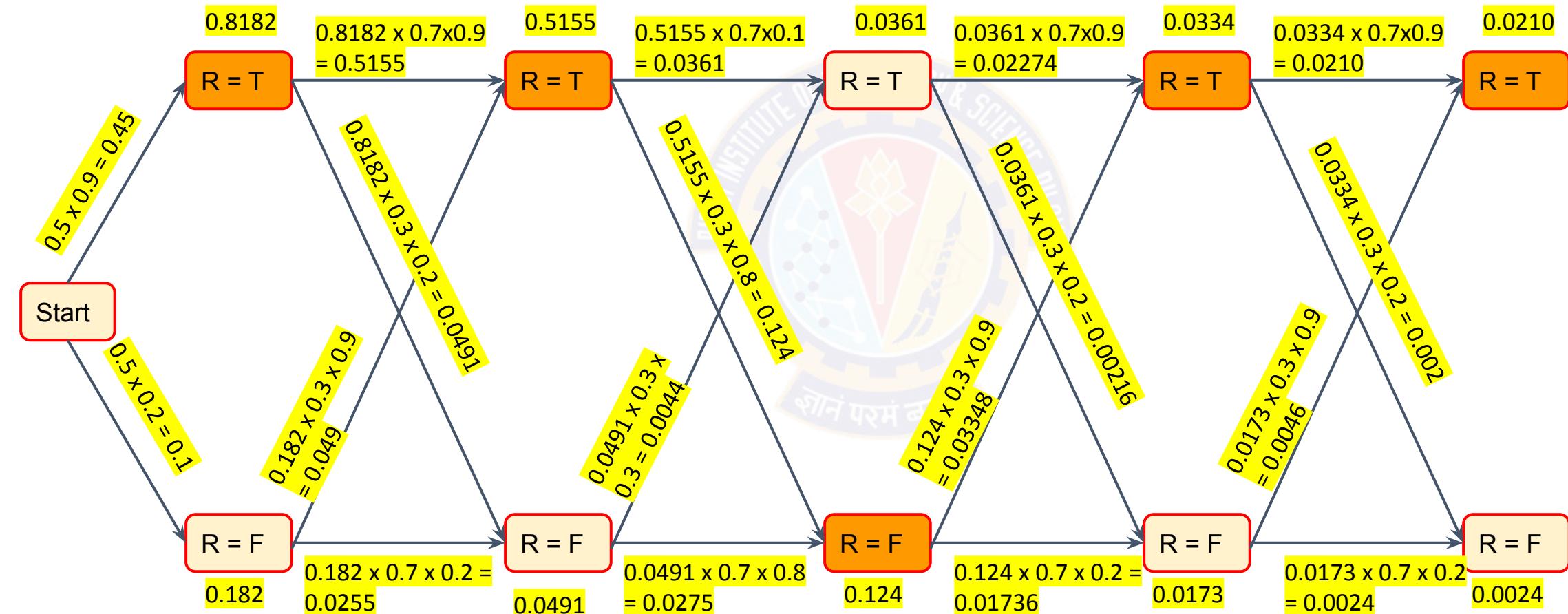
Viterbi Algorithm

$$U = [T, T, F, T, T]$$



Viterbi Algorithm

$U = [T, T, F, T, T]$

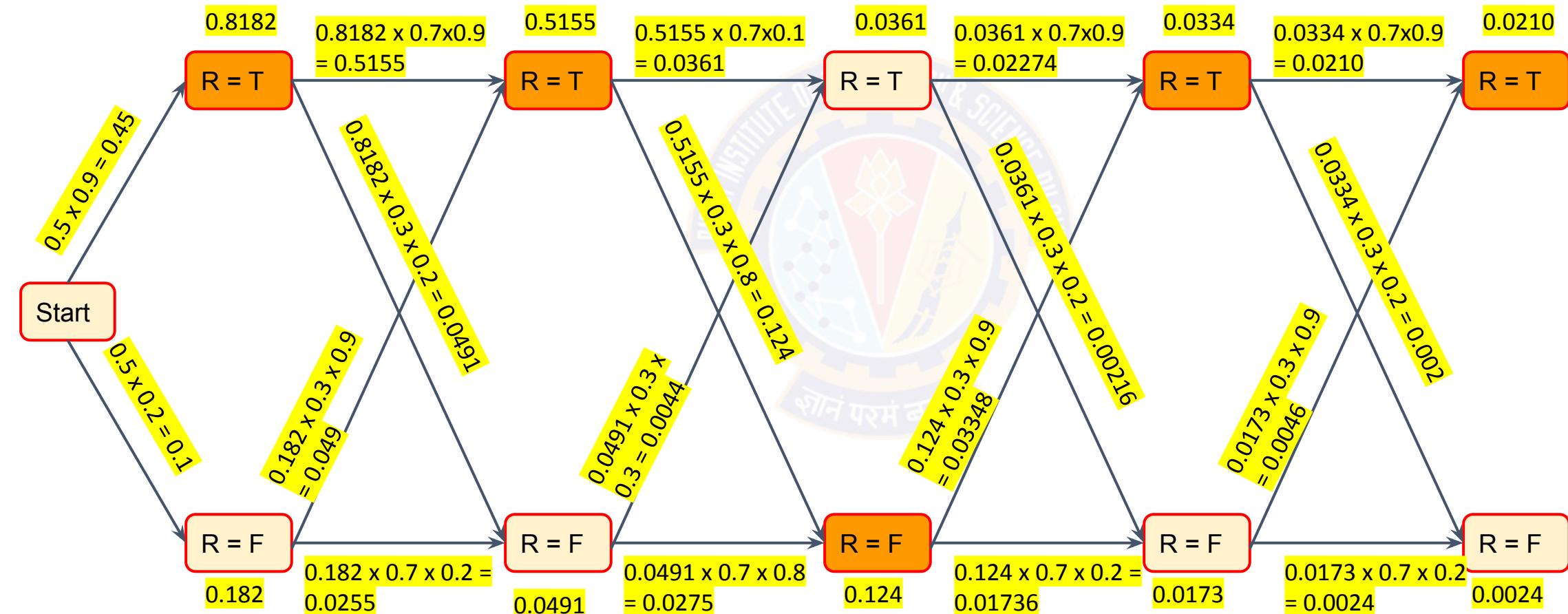


Most Likely State Sequence $R = [T, T, F, T, T]$

Viterbi Algorithm

$U = [T, T, F, T, T]$

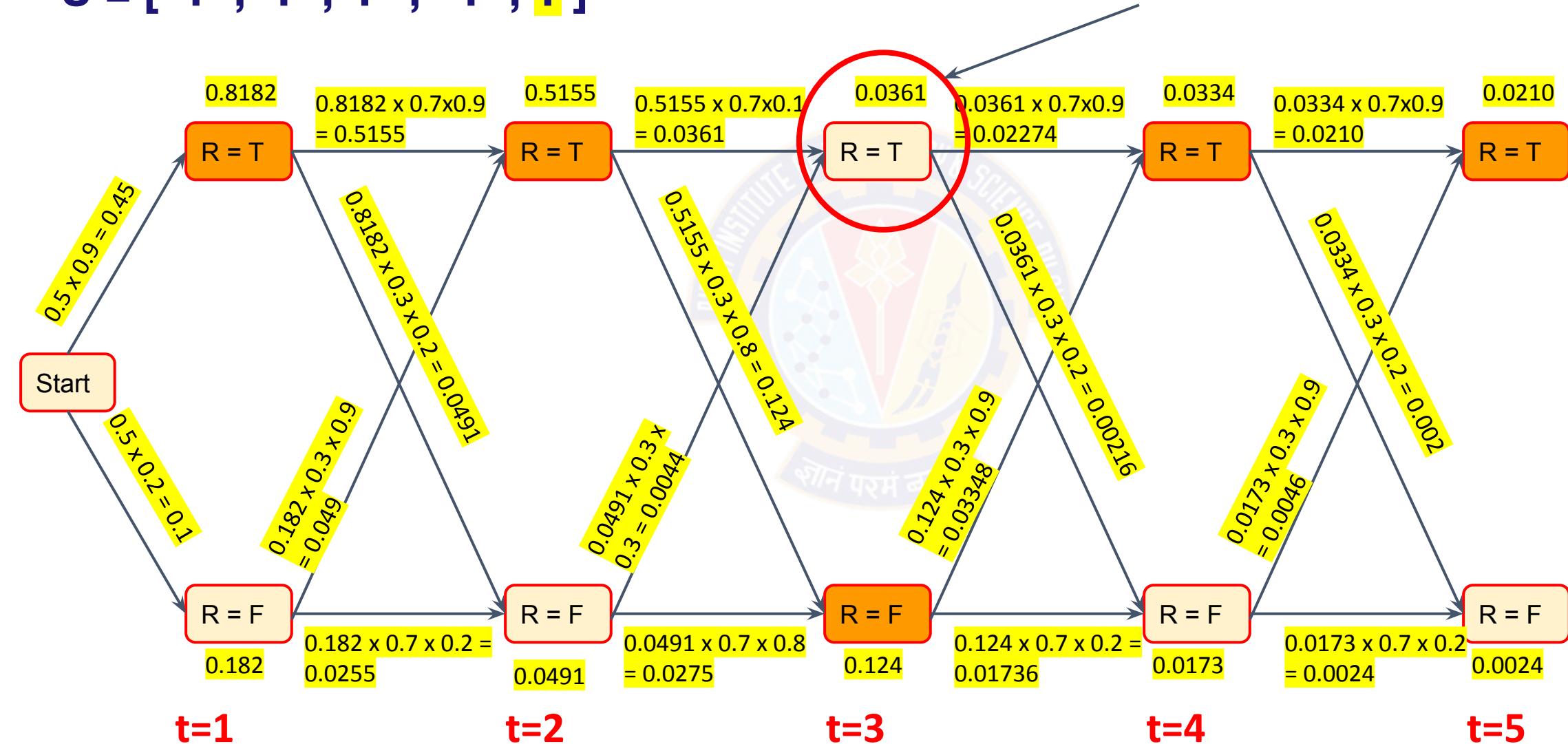
Let us try to understand what we did so far...



Viterbi Algorithm

$$U = [T, T, F, T, T]$$

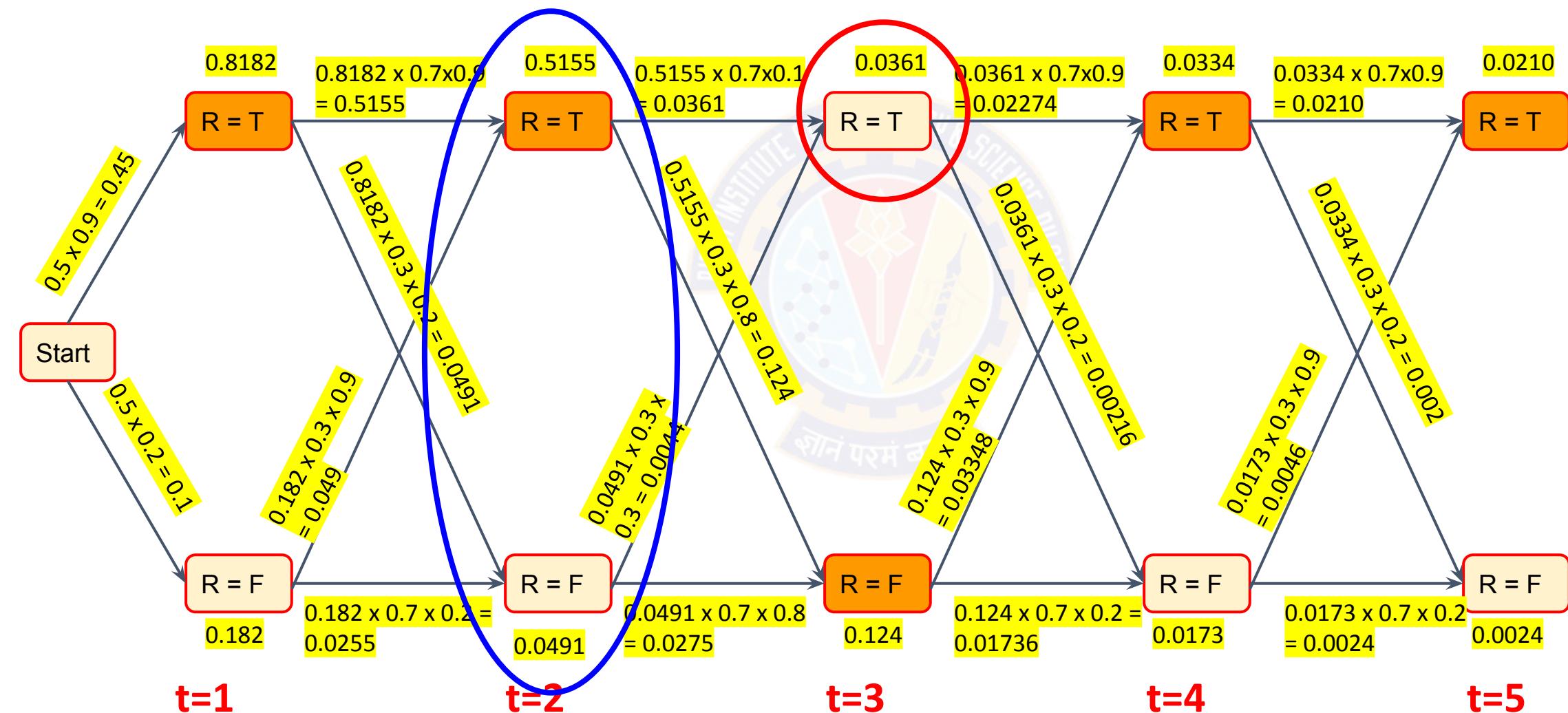
How to understand the most Likely path to this node?



Viterbi Algorithm

$$U = [T, T, F, T, T]$$

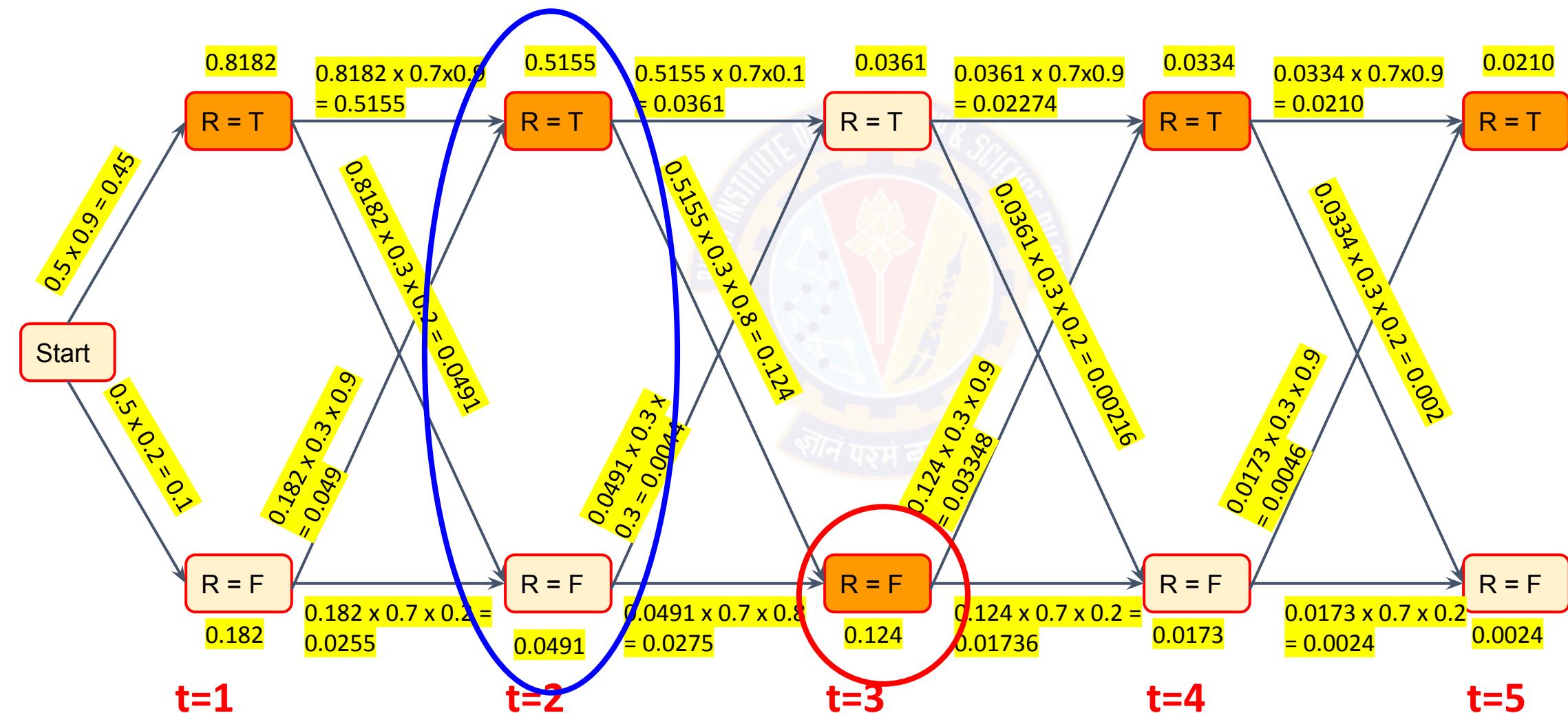
Most likely path to any state in the previous time & a transition here



Viterbi Algorithm

$$U = [T, T, F, T, T]$$

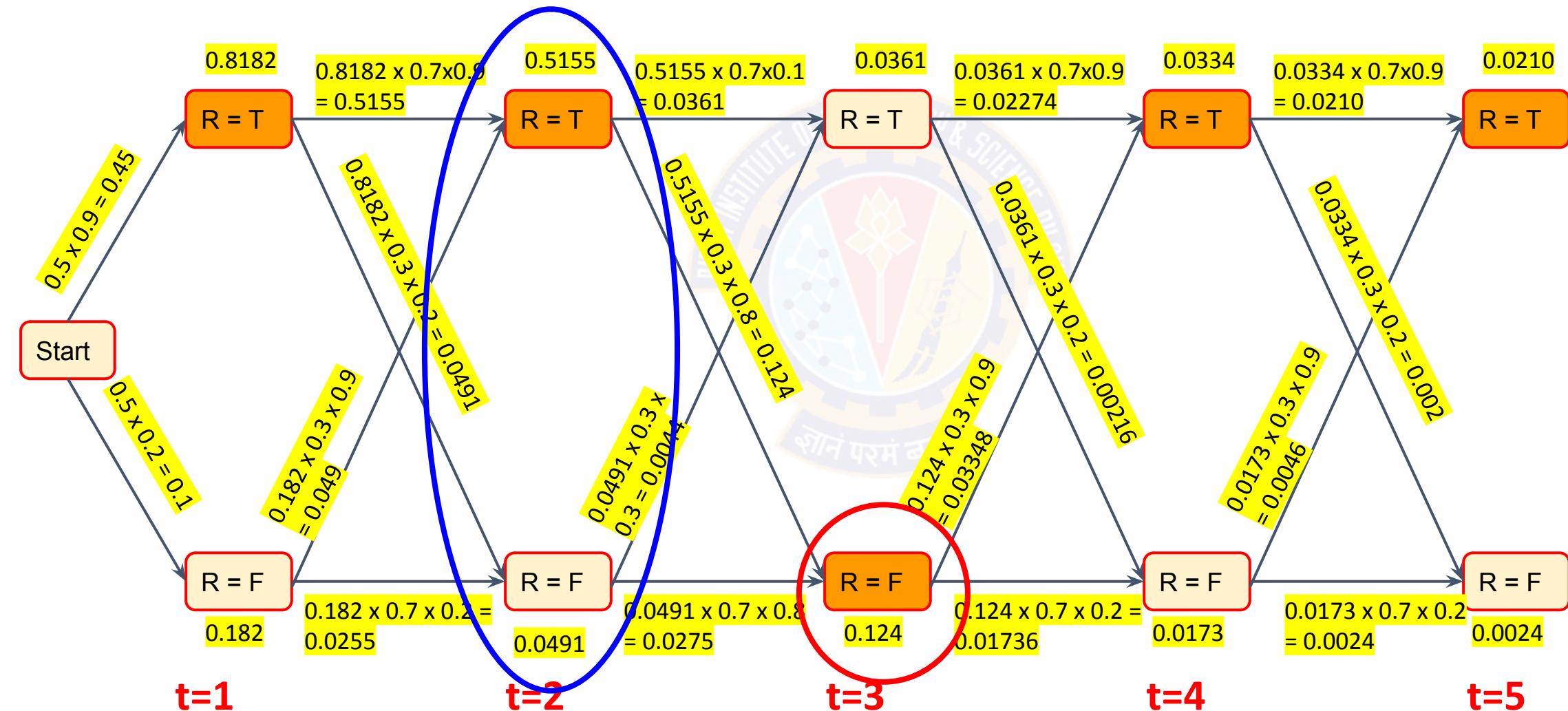
Most likely path to any state in the previous time & a transition here



Viterbi Algorithm

$$U = [T, T, F, T, T]$$

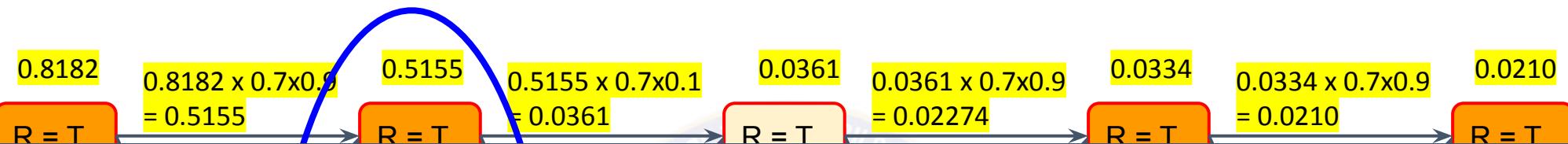
& Repeat this way !!!



Viterbi Algorithm

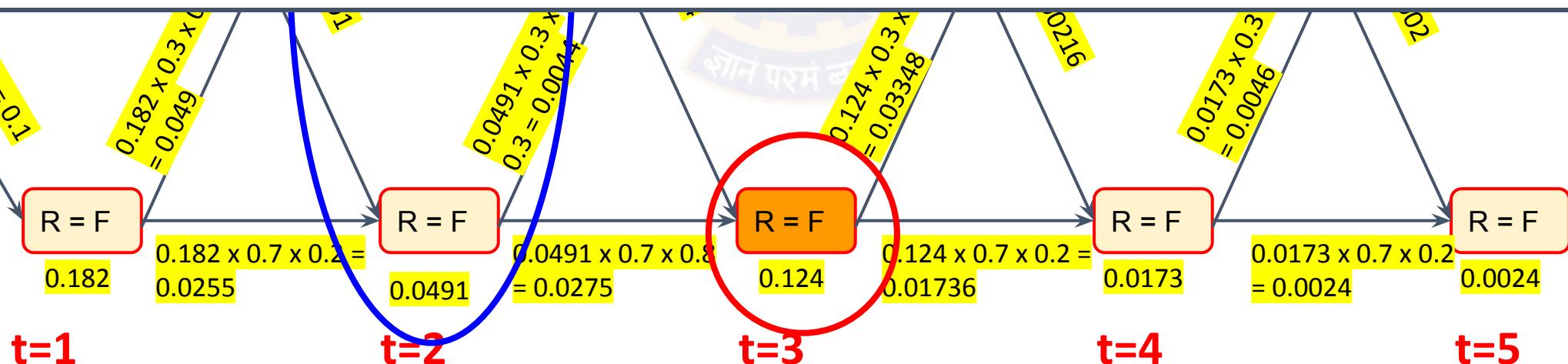
$$U = [T, T, F, T, T]$$

& Repeat this way !!!



$$\max_{\mathbf{x}_1 \dots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) \max_{\mathbf{x}_1 \dots \mathbf{x}_{t-1}} P(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t \mid \mathbf{e}_{1:t}) \right)$$



Viterbi Algorithm

Closing Points

- Named after Andrew Viterbi (1967)
- Linear time algorithm , $O(t)$ [both space and time]
 - We require details to help backtracking [to find sequences leading to each state]
- Wide applications in speech, language processing, analysing time series data

