

Computer Vision
Prof. Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 46
Clustering and Classification Part – I


We start a new topic in this course and that is on Clustering and Classification.

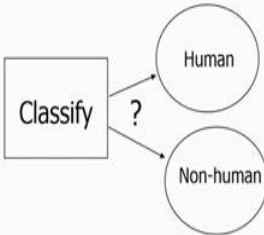
(Refer Slide Time: 00:22)

Classification

Classification:
Task of assigning a known category or class to an object.

- detection of pedestrian in an image patch.
- recognition of an alphabet given a 2-D pattern.
- assigning a pixel of an image to its foreground or background.

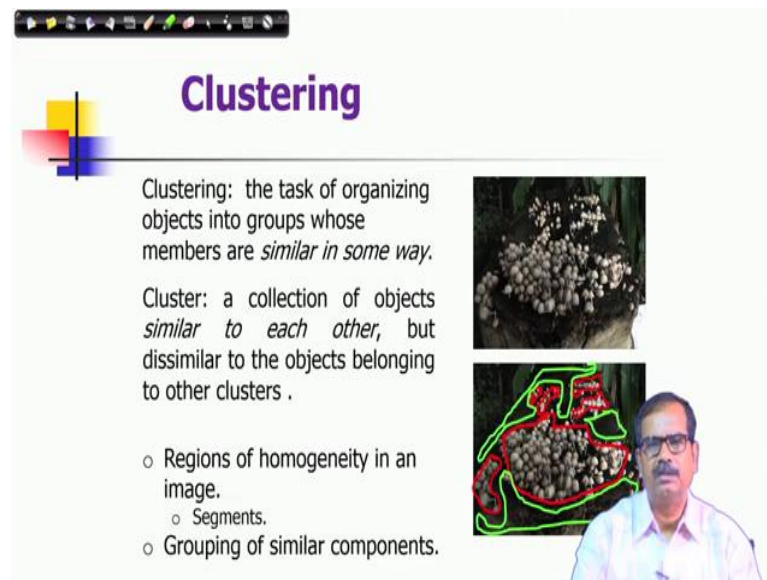




So, the classification is a task of assigning a known category or class to an object. For example, you take this image and you are asked to classify this image that which are the regions which contains human and or which contains which and whether it is human or non human. So, this is the classification task. So, suppose I have given this particular region of the image and I asked you to classify whether it contains human or not.

And in this way I can in this system I can give various patches from this images from this image and we can ask this ask to solve this problem. There could be other kinds of problems also like detection of pedestrian in an image patch, recognition of an alphabet given a 2-D pattern, assigning a pixel of an image to its foreground or background. So, these are different other classification problems and in this way you can define infinitely many types of problems. So, this is a nature of a classification problem.

(Refer Slide Time: 01:42)





Clustering

Clustering: the task of organizing objects into groups whose members are *similar in some way*.

Cluster: a collection of objects *similar to each other*, but dissimilar to the objects belonging to other clusters .

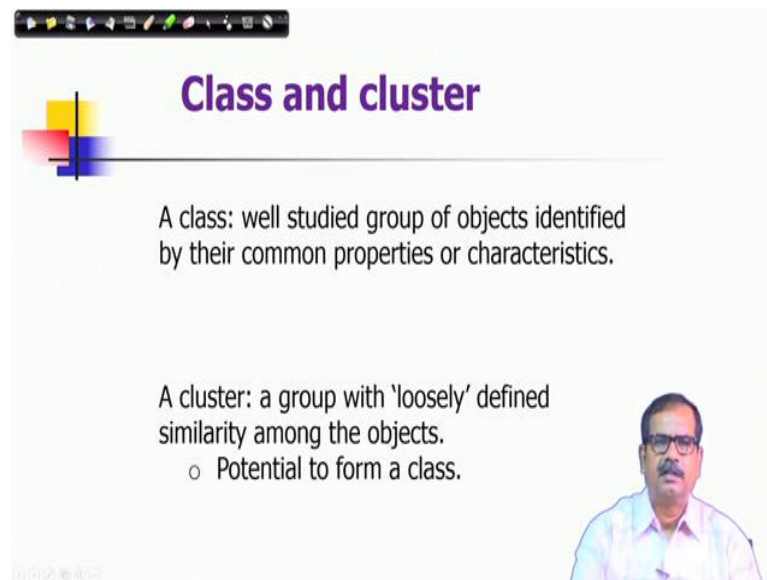
- Regions of homogeneity in an image.
 - Segments.
- Grouping of similar components.



Whereas a clustering it is a task of organizing objects into groups whose members are similar in some way. So, the cluster is a collection of objects which are similar to each other, but dissimilar to the objects belonging to other clusters. You can consider examples of some of the clustering as finding out regions of homogeneity in an image and which means you are deriving segments this is also a problem similar to clustering.

Or grouping of similar components and example we can see here suppose I have given an image of mushroom and in the background there is a humous substance and we would like to cluster the similar pixels or similar regions in this case. So, one result could be in this form here you can see that there are two primarily two regions are two types of clusters are there shown by the green contours and red contours. Again there are different regions of green contours and red contours which can be treated also as segments.

(Refer Slide Time: 03:01)



Class and cluster

A class: well studied group of objects identified by their common properties or characteristics.

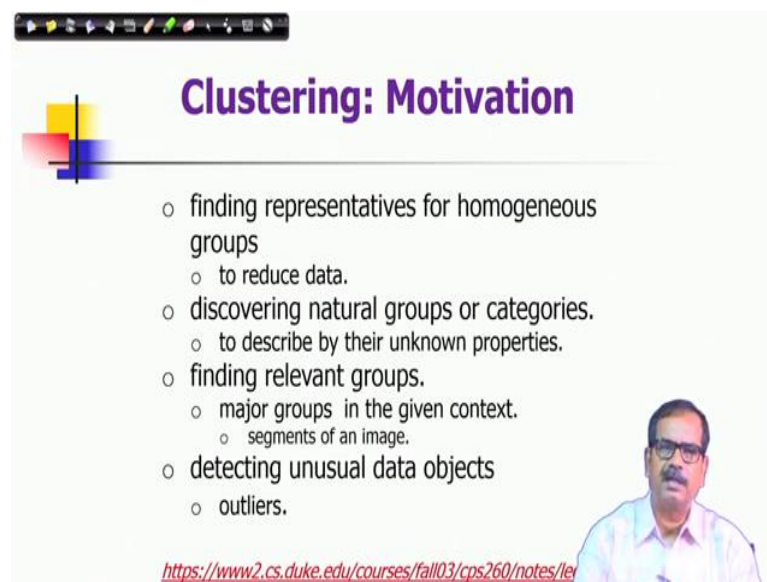
A cluster: a group with 'loosely' defined similarity among the objects.

- o Potential to form a class.

The slide features a decorative graphic on the left with overlapping yellow, red, and blue squares and a black crosshair. A small video inset in the bottom right corner shows a man with glasses and a mustache, wearing a light blue shirt, speaking.

So, what is the difference between class and cluster? A class is a well studied group of objects identified by their common properties or characteristics whereas a cluster is a group with loosely defined similarity among the objects it is potential to form a class.

(Refer Slide Time: 03:20)



Clustering: Motivation

- o finding representatives for homogeneous groups
 - o to reduce data.
- o discovering natural groups or categories.
 - o to describe by their unknown properties.
- o finding relevant groups.
 - o major groups in the given context.
 - o segments of an image.
- o detecting unusual data objects
 - o outliers.

<https://www2.cs.duke.edu/courses/fall03/cps260/notes/lec>

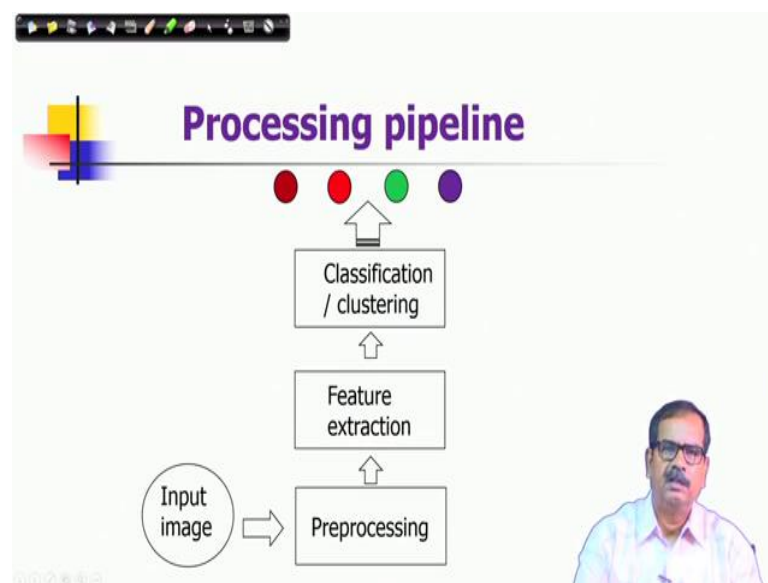
The slide features a decorative graphic on the left with overlapping yellow, red, and blue squares and a black crosshair. A small video inset in the bottom right corner shows the same man from the previous slide, speaking.

So, what are the motivations of clustering? I have given one example of its application like segmenting images will see different other motivations. Like you can find representatives for homogeneous groups and this would reduce your total data representations you can represent data by smaller set of representative samples we are in those characteristics of

data. Then discovering natural groups or categories that is also another motivation. So, that we can describe them by their unknown properties and finding relevant groups.

So, segmentation is one such example where we would try to draw our attention to relevant groups in the distribution, so in the images. So, it is a measure groups in the given context that we would like to identify like segments of an image. Then detecting unusual data objects, so usual like outliers in data. So, those are also can be detected using this kind of clustering techniques.

(Refer Slide Time: 04:25)

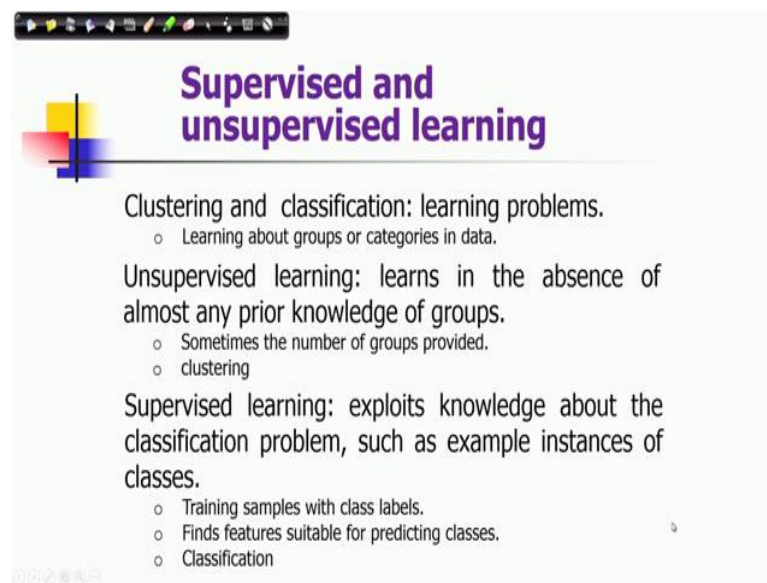


So, in the context of our image and video image processing computer visions also video processing. So, how clustering and classifications in that are placed in this particular processing of information. So, as you can see that it is a very high level process of information. So, at the lower level from the images we would like to derive certain representation of data in terms of feature extraction.

So, it should pass through the process of pre processing feature extraction and then you can get the represent the objects or images image patches in by feature descriptors or feature vectors that you have discussed on already in previous lectures. And we know different techniques of driving deriving the points of importance relevance in an image and then describing their neighbourhoods or also different patches or images you can describe by their corresponding feature vectors.

These feature vectors they are the representative of those corresponding class or groups or objects they are used for the purpose of classification and clustering. So, and then you have to though the task is that no you have to assign them certain known or unknown groups whatever that means they should be similar. For clustering groups are not well defined as we mentioned. So, in the example with different colors we are trying to represent those different kinds of classes or groups.

(Refer Slide Time: 06:02)



Supervised and unsupervised learning

Clustering and classification: learning problems.

- Learning about groups or categories in data.

Unsupervised learning: learns in the absence of almost any prior knowledge of groups.

- Sometimes the number of groups provided.
- clustering

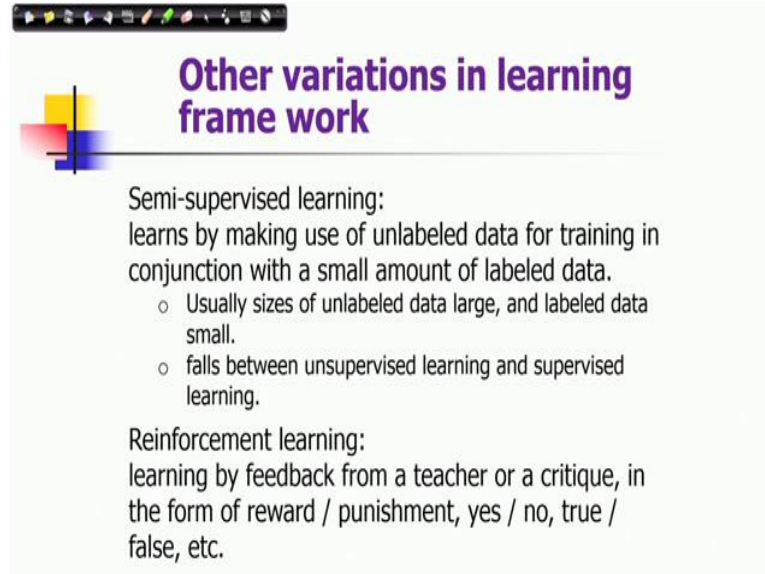
Supervised learning: exploits knowledge about the classification problem, such as example instances of classes.

- Training samples with class labels.
- Finds features suitable for predicting classes.
- Classification

So, we can see the approaches of you know clustering classification which are essentially learning problems. Now, there are two approaches supervised and unsupervised learning when you consider its a learning problem. So, here we are learned we are learning about groups or categories in data. And when we say it is an unsupervised learning then it learns in the absence of almost any prior knowledge of groups almost I told because sometimes know the number of groups that information is provided.

Now, this problem is like clustering. So, in clustering we used to unsupervised learning and in a supervised learning it exploits knowledge about the classification problem such as example instances of classes. Say in this case training samples with class levels are provided for solving this problem and it finds features suitable for predicting classes. So, this problem supervised learning is also used in the use solving classification problem.

(Refer Slide Time: 07:19)



Other variations in learning frame work

Semi-supervised learning:
learns by making use of unlabeled data for training in conjunction with a small amount of labeled data.

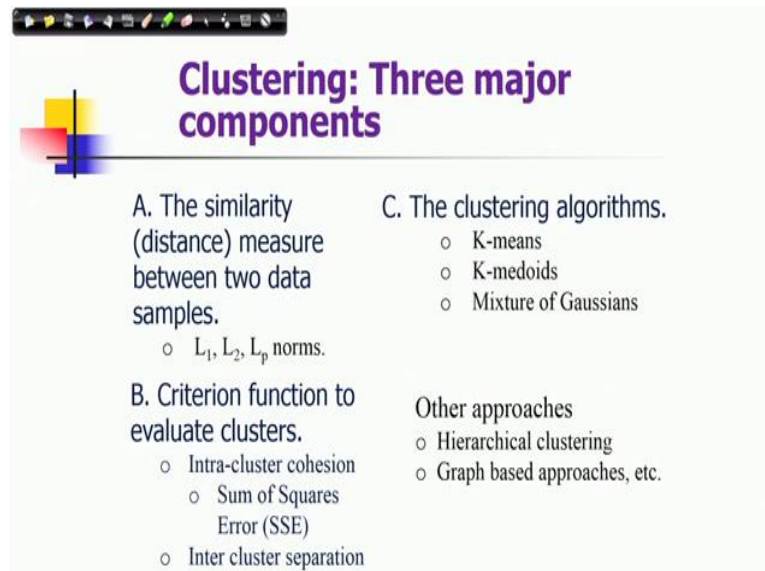
- Usually sizes of unlabeled data large, and labeled data small.
- falls between unsupervised learning and supervised learning.

Reinforcement learning:
learning by feedback from a teacher or a critique, in the form of reward / punishment, yes / no, true / false, etc.

Now, there are other variations in this learning framework you have semi supervised learning and reinforcement learning like in semi supervised learning it learns by making use of unlabeled data for training in conjunction with a small amount of labeled data. Usually the sizes of unlabeled data it is they are large and labeled data they are kept small.

And in this case this framework this learning mechanism it falls between unsupervised and supervised learning. In the reinforcement learning it is learning by feedback from a teacher or a critique in the form of reward or punishment yes or no true or false etcetera.

(Refer Slide Time: 08:03)



Clustering: Three major components

<p>A. The similarity (distance) measure between two data samples.</p> <ul style="list-style-type: none">○ L_1, L_2, L_p norms.	<p>C. The clustering algorithms.</p> <ul style="list-style-type: none">○ K-means○ K-medoids○ Mixture of Gaussians
<p>B. Criterion function to evaluate clusters.</p> <ul style="list-style-type: none">○ Intra-cluster cohesion○ Sum of Squares Error (SSE)○ Inter cluster separation	<p>Other approaches</p> <ul style="list-style-type: none">○ Hierarchical clustering○ Graph based approaches, etc.

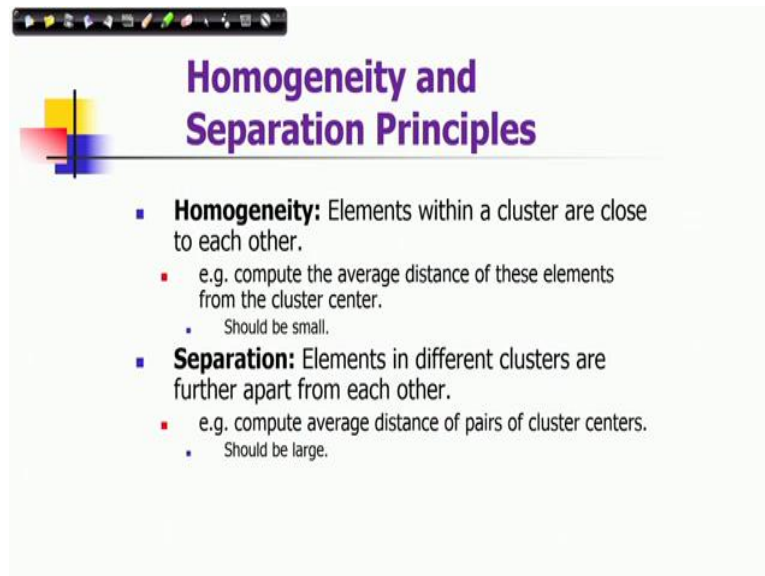
So, we will be discussing now the methods of clustering techniques, we will first discuss methods of clustering techniques and then we will discuss about the classification techniques. So, in clustering there are three measure components; the first it should have a distance measure or which measure similarity between two data or samples. And then there should be some criteria function to evaluate clusters and then of course, the methodology the algorithm by which you should compute you should solve the problem.

So, for defining similarity different distances could be used and some examples of distances we have already seen in various other applications in this particular course also in on various topics. Suppose the L_1 norm, L_2 norm and generalized L_p norm could be used and for criteria function or to evaluate clusters there are two particular properties which are looked at it which are looked for in having a good clustering solution.

One thing is that there should be a good intra cluster cohesion which means the members of the cluster they should have good homogeneity property. And one of the measures in this case could be sum of squares of error of deviations from that property or there should be inter cluster separation; that means, groups are also well separated well discriminated.

About the clustering algorithms in this particular lecture or in this course will be learning three different clustering algorithms; one is K means and then K medoids and mixture of Gaussians Gaussian technique. But there could be various other approaches like know there could be hierarchical clustering techniques graph based approaches etcetera will be considering only this three in this course.

(Refer Slide Time: 10:02)



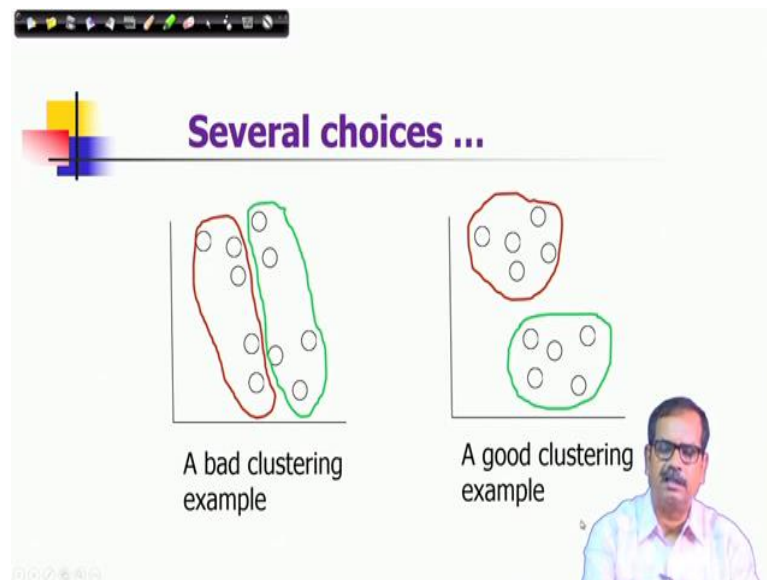
Homogeneity and Separation Principles

- **Homogeneity:** Elements within a cluster are close to each other.
 - e.g. compute the average distance of these elements from the cluster center.
 - Should be small.
- **Separation:** Elements in different clusters are further apart from each other.
 - e.g. compute average distance of pairs of cluster centers.
 - Should be large.

Homogeneity and separation principles that we discussed when we are considering evaluation of cluster clustering evaluation of clusters those are derived by technique. So, homogeneity as I mentioned it is elements within a cluster which should be close to each other. And for example, you can compute the average distance of these elements from the cluster center. So, if this distance is small then the cluster is good it preserves the homogeneity it has good homogeneity property.

Whereas this separation property in clusters are the elements in different clusters they should be further apart from each other. So, in this case may compute average distance of pairs of cluster centers and those centers should be no should be placed apart they should. So, this distance should be large for having a good separation property.

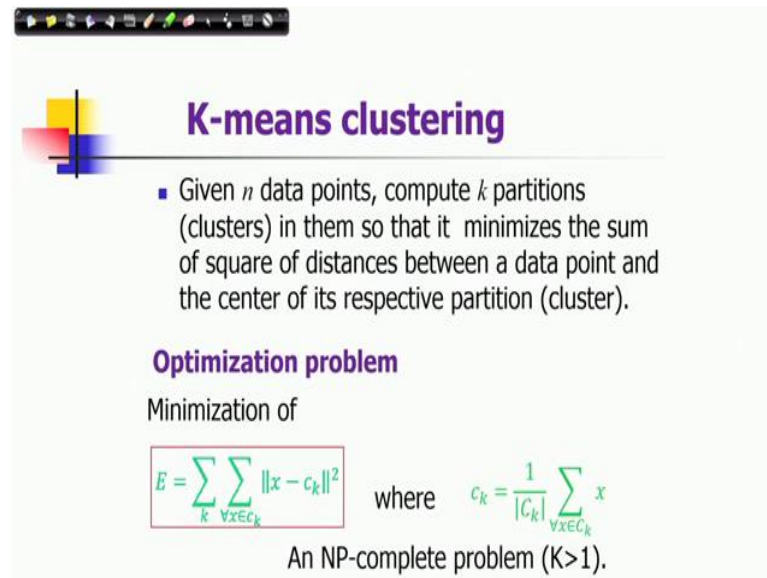
(Refer Slide Time: 10:58)



There could be a several choices of clustering or partitioning, so its a non trivial partitioning problem in that sense. Suppose I give you this data distribution its a 2 dimensional data points which are shown by this circles and locations of the data are shown in has a 2 dimensional coordinate space. So, these are the elements.

So, one could one kind of partition could be like this and the other kind of partitioning could be like this. Now, by considering the two property homogeneity and separation; the first one is a bad clustering example where this is the second one is the good clustering example.

(Refer Slide Time: 11:40)

A presentation slide titled "K-means clustering" with a decorative graphic of overlapping colored squares (yellow, red, blue) and a black crosshair. The slide contains a bullet point describing the problem, an "Optimization problem" section with the minimization of a cost function E, and a note that it is an NP-complete problem for K > 1.

K-means clustering

- Given n data points, compute k partitions (clusters) in them so that it minimizes the sum of square of distances between a data point and the center of its respective partition (cluster).

Optimization problem

Minimization of

$$E = \sum_k \sum_{x \in C_k} \|x - c_k\|^2 \quad \text{where} \quad c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$$

An NP-complete problem ($K > 1$).

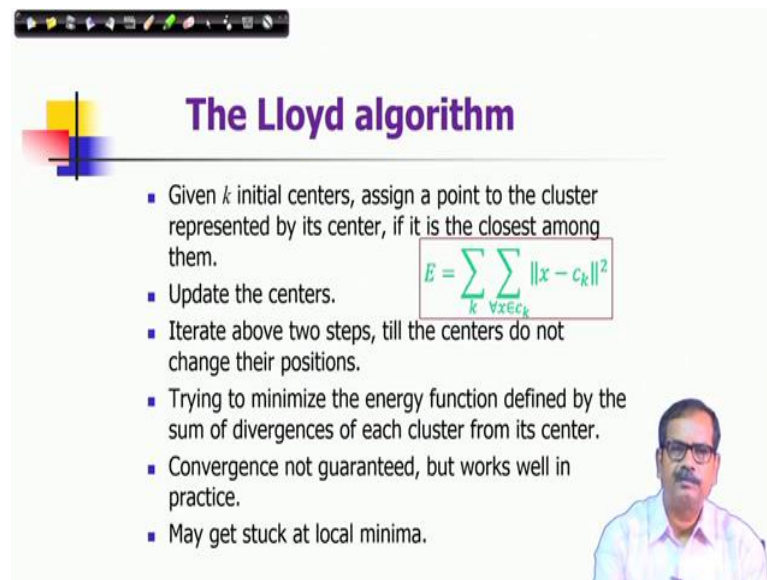
So, the first technique that I will be discussing here it is called K means clustering technique and in this case the problem is that if I give you n data points then you need to compute k partitions though partitions are actually clusters. So, that it they it minimizes the sum of square of distances between a data point and the center of its respective partition or cluster.

So, this is essentially an optimization problem where you can consider this mathematical formulation of the optimization function. So, you see that optimization function which you require to minimize this is actually sum of square of deviations or distances from the mean of a partition which is denoted here by c_k and to the points which are included in that partition. And if since there are k partitions you perform this job k times I mean you have summing all those components of k partitions.

$$E = \sum_k \sum_{x \in C_k} \|x - c_k\|^2$$

Now, this problem is an n NP complete problem provided K is greater than 1 because if there is if there is K it is just a centre of the cluster compute the centre of the data that itself will give you the solution.

(Refer Slide Time: 12:56)



The Lloyd algorithm

- Given k initial centers, assign a point to the cluster represented by its center, if it is the closest among them.
- Update the centers.
- Iterate above two steps, till the centers do not change their positions.
- Trying to minimize the energy function defined by the sum of divergences of each cluster from its center.
- Convergence not guaranteed, but works well in practice.
- May get stuck at local minima.

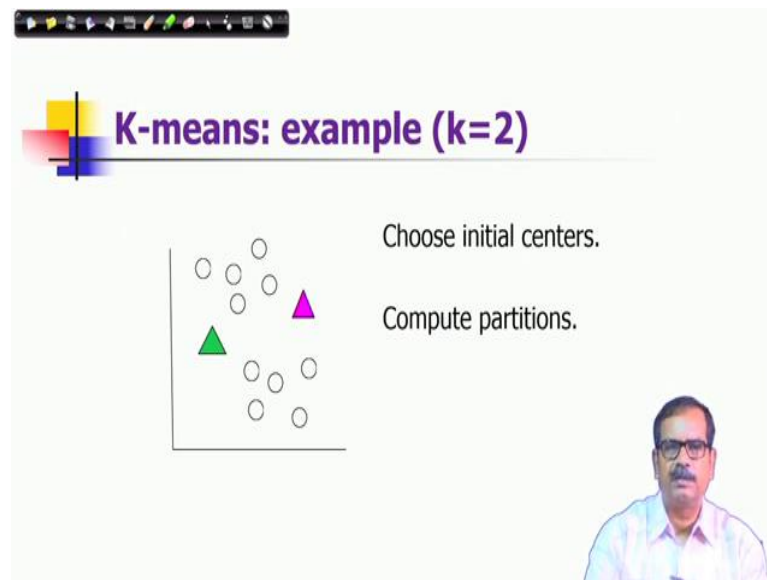
$$E = \sum_k \sum_{x \in c_k} \|x - c_k\|^2$$

So, the algorithm for K means clustering is a very famous algorithm and which is known as a Lloyd algorithm by the name of the inventor of this algorithm. So, it is given as k initial centers. Assigns given k first you consider randomly you can choose k initial centers and then you can based on that you can partition them by assigning the nearest center to a particular points.

So, the cluster of the by assigning that that cluster whose center is nearest to a point. So, it is a closest among them and then we can update the centers once again the partitions so, centers gets updated because once you get the partition and you can compute that center that would be definitely different from what you had earlier. And you can iterate these two steps till the centers do not change their position a very simpler approach, but very effective.

So, what it is trying to do it is trying to minimize the energy function defined by the sum of divergence says each cluster from its center it is the error function what I discussed in the previous slide. So, this is this function what we are trying to minimize. But this method the convergence not guaranteed, but works well in practice and it may get stuck at local minima that is another problem.

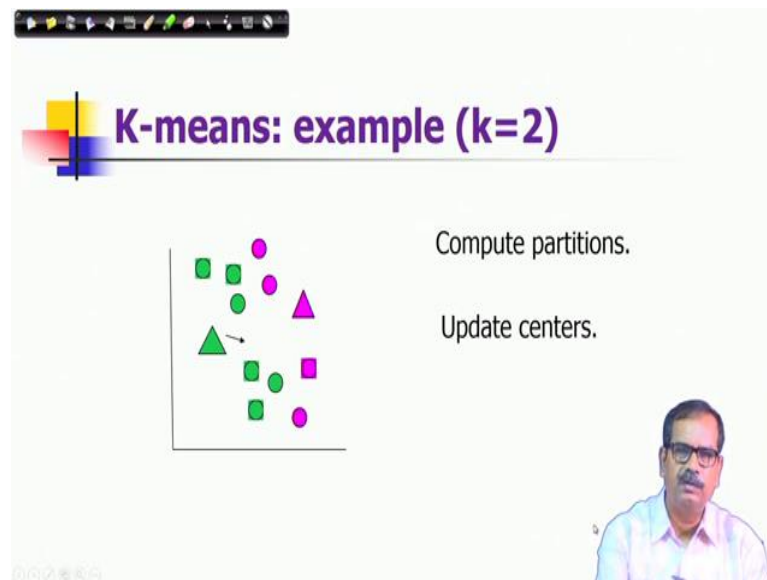
(Refer Slide Time: 14:25)



So, let me show you figuratively how this approach works the essence of this computation. So, we consider once again the distribution of the points in a 2 dimensional space and points are shown by circles their locations are shown because putting center of the circles. So, what we can do that we can choose initial cluster centers.

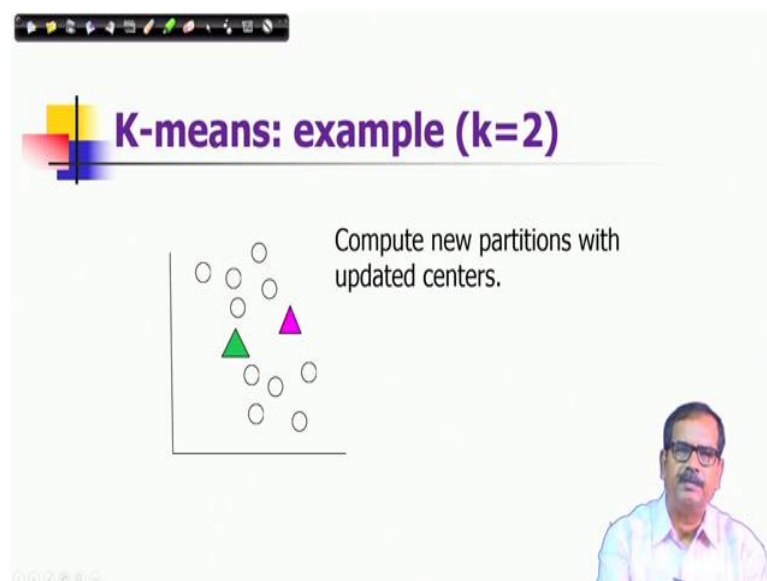
For example, say these are the two initial hypothesize has two cluster centers. Then what you should do? You have to now compute the partitions which means you have to consider the points which are closer to a particular cluster center. So, the corresponding cluster level should be given to those points. So, here the levels are shown by colors for the purpose of visualization.

(Refer Slide Time: 15:20)



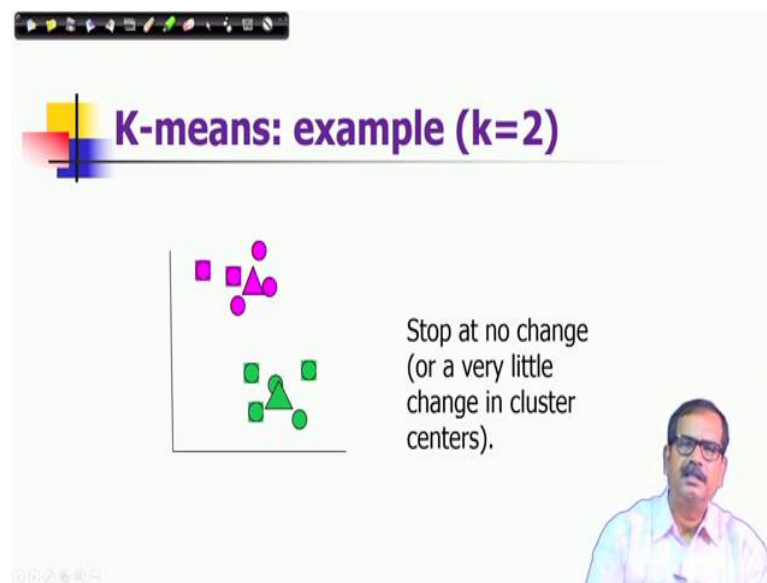
So, I will show the leveling by color. So, you can see these are the set of points which are found to be close to the cluster shown by the color pink and the other one by the color green. So, now we have a two partition once again best on those centers and now you have to you should again update this center. So, you can update this centers which means now these centers they should move inward because of the configuration here. And you get this is the updated positions again you perform the partitioning with these two new positions.

(Refer Slide Time: 15:49)



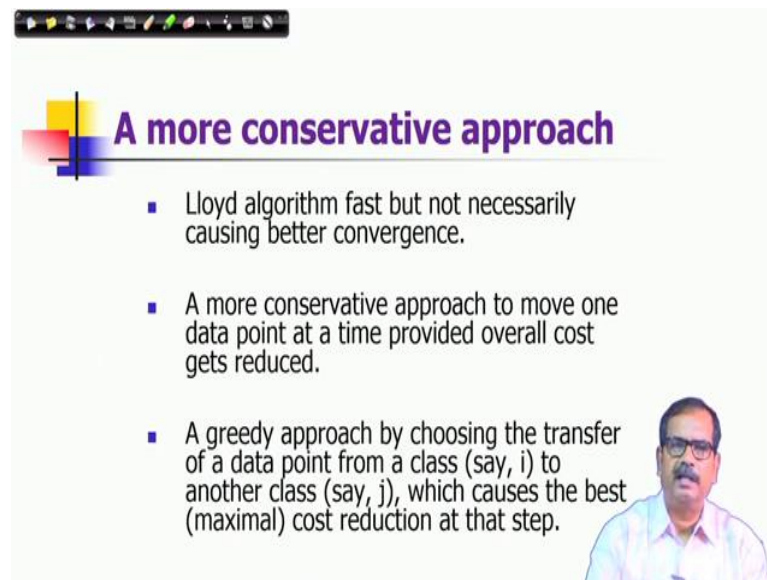
So, compute new partitions with updated centers. So, now, you will find that no these are the set of points which are being clustered which are closer to the corresponding centers and they are shown by their colors. So, you would repeat this operations you update centers and move the cluster centers no update no cluster centers once again you perform the partitions and you will find that you get a new partition; new partitions here and you can you should update the centers and it will be moved further.

(Refer Slide Time: 16:39)



And in this way you will find that after few iterations there is no change or a very little change in cluster centers and then you should stop.

(Refer Slide Time: 16:52)

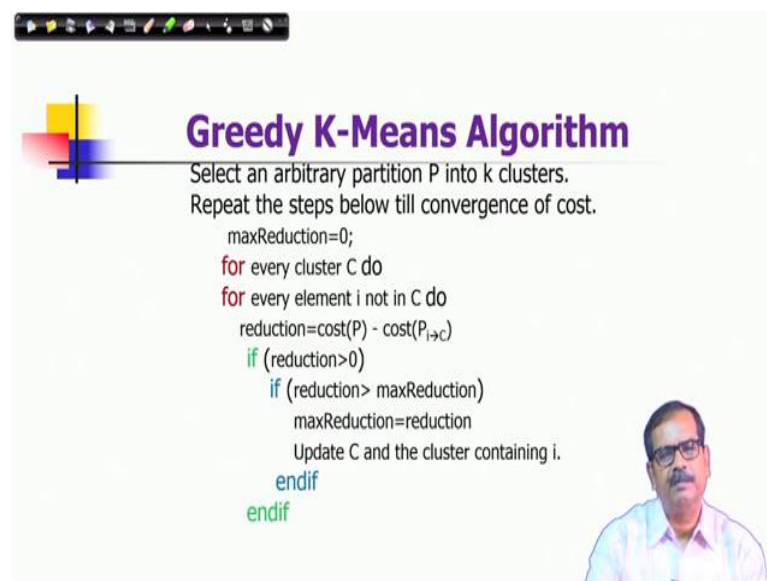


A more conservative approach

- Lloyd algorithm fast but not necessarily causing better convergence.
- A more conservative approach to move one data point at a time provided overall cost gets reduced.
- A greedy approach by choosing the transfer of a data point from a class (say, i) to another class (say, j), which causes the best (maximal) cost reduction at that step.

So, this is how the K means algorithm works. There is about conservative approach of this particular algorithm because Lloyd algorithm is a first is first algorithm, but its not necessarily causing better convergence. So, a more conservative approach is to move one data point at a time provided overall cost gets reduced. So, its a greedy approach and what it does in the principle of this approach is that it chooses the transfer of a data point from a class i to another class j which causes the best cost reduction that gives the maximal cost reduction at that step.

(Refer Slide Time: 17:27)



Greedy K-Means Algorithm

Select an arbitrary partition P into k clusters.
Repeat the steps below till convergence of cost.

```
maxReduction=0;
for every cluster C do
  for every element  $i$  not in  $C$  do
    reduction=cost( $P$ ) - cost( $P_{i \rightarrow C}$ )
    if (reduction>0)
      if (reduction> maxReduction)
        maxReduction=reduction
        Update  $C$  and the cluster containing  $i$ .
      endif
    endif
  endif
endif
```

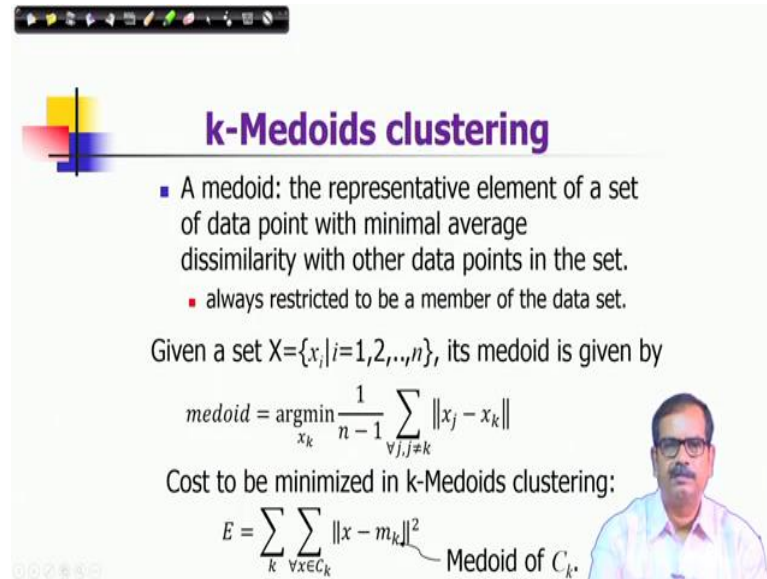

So, we can describe this computations in an algorithmic steps say select an arbitrary partition P into k clusters the like the K means clustering algorithm and you should repeat the steps till convergence of this cost. So, let us consider we keep track that what is the reduction at this step if there is no reduction called some iterations then we should not after some iterations then know we can stop.

So, initially it is initialized with 0 and then for every cluster C and for every element which is not in C you perform these operations. We perform the reduction that is a difference between cost of partitioning of P which was there earlier before the transfer and what is the cost after the transfer; that means, you have recomputed once again the centers and partitions.

Now, this can be done efficiently by simply considering the clusters which are affected by this transfer by recomputing their means. Rest other partitions levels of the rest of the assignment remains same and then you can quickly compute this cost. So, if this is if this there is a reduction and if this reduction is greater than the maximum reduction, then you I mean first thing if there is a reduction then you should move update C and the cluster containing.

So, which means in at every iteration we are considering for all elements what is the reduction of cost and choose that transfer which is a maximum and at that iteration only that transfer is used and go on doing this. So, its a very slow process its not so fast as Lloyds algorithm, but its convergence is better which means as I mentioned that K means algorithm may get stuck at local minima similarly this algorithm also it is not guaranteed that will get a global minima, but it should get a better local minima by this process.

(Refer Slide Time: 19:44)



k-Medoids clustering

- A medoid: the representative element of a set of data point with minimal average dissimilarity with other data points in the set.
 - always restricted to be a member of the data set.

Given a set $X = \{x_i | i=1, 2, \dots, n\}$, its medoid is given by

$$medoid = \underset{x_k}{\operatorname{argmin}} \frac{1}{n-1} \sum_{\forall j, j \neq k} \|x_j - x_k\|$$

Cost to be minimized in k-Medoids clustering:

$$E = \sum_k \sum_{x \in C_k} \|x - m_k\|^2$$

Medoid of C_k .

So, that was about K means clustering; a variation of K means clustering is called k medoids clustering. So, let us understand what is defined by a medoid.

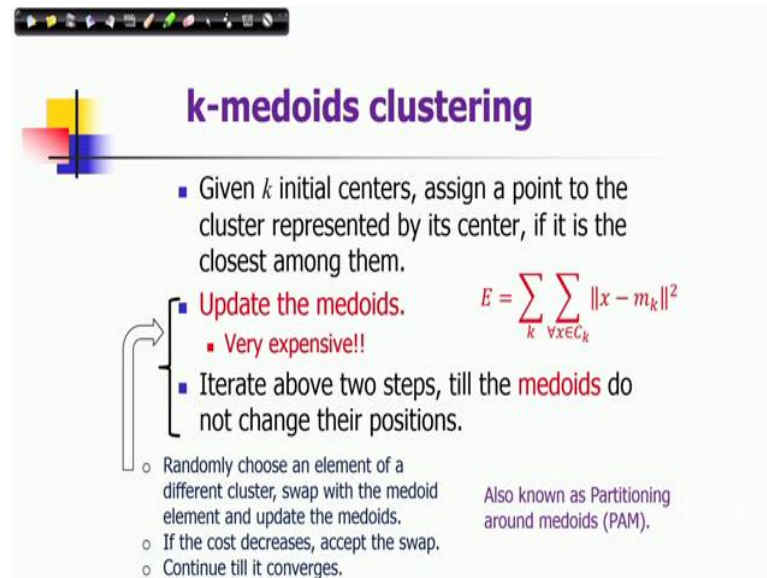
$$medoid = \underset{i, j \in C_k}{\operatorname{argmin}} \frac{1}{n-1} \sum \|x_j - x_k\|$$

A medoid is the representative element of a set of data point with minimal average dissimilarity with the other data points in the set you can consider the data points are there all vectors here. So, it is nothing, but the median vector what is defined conventionally in single processing community.

So, this definition mathematically can be represented in this way that given a set X, its medoid is given by this is what you compute the average of the distances from that point to other points and consider that element in that set which has the minimum such distance. So, that is how the medoid is defined.

The advantage here is that it is like median of a set what we get. So, in the median of the set it is one of the element of the set similarly medoid is also one of the member of this data set. So, the cost to minimize in k medoid clustering it is almost structure is same instead of the cluster means now we have replaced means by the cluster medoids in the expression. So, you can see this is almost similar expressions where these are the medoids or clusters k.

(Refer Slide Time: 21:12)



The slide is titled "k-medoids clustering" in purple. It features a small graphic of overlapping colored squares (yellow, red, blue) on the left. The main content is a list of steps in a blue box, with the second step "Update the medoids" and its sub-points highlighted in red. A red arrow points from the first step to the second. To the right of the list is the cost function equation $E = \sum_k \sum_{x \in C_k} \|x - m_k\|^2$. Below the list, there are three bullet points in a grey box. To the right of these is the text "Also known as Partitioning around medoids (PAM)." in purple.

k-medoids clustering

- Given k initial centers, assign a point to the cluster represented by its center, if it is the closest among them.
- **Update the medoids.**
 - **Very expensive!!**
 - Iterate above two steps, till the **medoids** do not change their positions.

$E = \sum_k \sum_{x \in C_k} \|x - m_k\|^2$

- Randomly choose an element of a different cluster, swap with the medoid element and update the medoids.
- If the cost decreases, accept the swap.
- Continue till it converges.

Also known as Partitioning around medoids (PAM).

So, k medoids clustering could be almost similar to k means clustering as you can see that this is the same algorithm except the fact that instead of updating cluster centers or means we are updating the medoids. So, iterate above two steps till the medoids do not change their position. So, the corresponding changes in this algorithm they are highlighted by this red color.

But the problem is that this updating of medoid is very computationally expensive it is not so simple like updating centers as we have seen for every cluster we have to compute the corresponding median vector by solving that problem. So, we need to know which we can have some other variations to make this computation faster; one of the approach is that we can consider computing medoid of only two clusters instead of all.

So, randomly choose an element of a different cluster and swap with the medoid element of one of the clusters and update the medoids if the cost decreases, then only you accept the swap and then you can continue this operations till it converges. So, this algorithm is also known as partitioning around medoids or PAM.

(Refer Slide Time: 22:35)

Mixture of Gaussians

- Each cluster center is augmented by a covariance matrix, whose values are re-estimated from corresponding samples.
 - Mahalanobis distance function:

$$d(x, \mu_k; \Sigma_k) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

Cluster center
Covariance matrix

Technique could be refined by computing probabilities of belongingness to a cluster.

Parametric PDF: $p(x|\{\pi_k, \mu_k, \Sigma_k\}) = \sum_k \pi_k N(x|\mu_k, \Sigma_k)$

$$N(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}d(x, \mu_k; \Sigma_k)}$$

Mixing coefficients

The last technique of the clustering methods which will be discussing in this particular no subject that is a use of mixture of Gaussians and by analyzing the probability density functions of the data in terms of representing it as a mixture of Gaussians. At the outset we can consider some similarities with the K means clustering technique. Say you consider a cluster center is augmented by co variance matrix, it is not a center is the mean of the data, but let us consider also your using its co variance matrix and know for every at every iterations you are updating corresponding means and co variance matrices.

$$d(x, \mu_k; \Sigma_k) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

And the distance function could be used like a Mahalanobis distance function. So, this is a example this is expression for Mahalanobis distance function as we can see it takes the takes into consideration of the co variance matrix also not only the mean and so this is the cluster center and this is the co variance matrix. So, you can apply like K means algorithm, but of course, this is not the technique of mixture of Gaussians, but these has some similarity. What we can do instead? We can refine it by computing probabilities of belonging into a cluster.

$$N(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}d(x, \mu_k; \Sigma_k)}$$

So, instead of crisply defining the membership of the of a cluster for every element at every iteration we maintain a probability of belongingness and continue doing this things till no we get a good probability density function which defines the you know which satisfy which describes the corresponding distribution of data consistently describes it.

So, what is this parametric probability density function is described by mixture of Gaussian distribution. So, this is a description here you can see that probability density function of a point data point x is given as a mixture of know normal distributions; so weighted sum of normal distribution.

While weights are given by the parameters π_k and each normal distribution has its own center and co variance matrix; that means, they are denoted by μ_k and Σ_k . Just for the completeness I am also providing you the expression called normal distribution in multi dimensional space. And the mixing co efficient are here it they are called this π_k .

(Refer Slide Time: 25:30)

Expectation Maximization (EM) Algorithm

→
$$z_{ik} = \frac{1}{Z_i} \pi_k N(x_i | \mu_k, \Sigma_k)$$

Normalizing factor $Z_i = \sum_k z_{ik}$

- Start with initial set : $\{\pi_k, \mu_k, \Sigma_k\}$.
- E-Step (Expectation stage)
 - Compute likelihood (z_{ik}) of x_i belonging to k th Gaussian cluster.
- Optional step. Decision to be taken at the end. → Assign x_i to the m th cluster whose *likelihood* is maximum.
- M-Step (Maximization Stage)
 - Re-estimate parameters ($\{\pi_k, \mu_k, \Sigma_k\}$) from class distribution
- Iterate above two steps till it converges

So, there is an algorithm which is known as expectation maximization algorithm and by which you can estimate this probability density functions and get those mixtures get those Gaussian distributions. And each Gaussian distribution component is representing like a cluster here and so elements which have higher probability due to that Gaussian mixture distribution those elements are assigned to those clusters. So, that is a process.

So, what we can do, we can start with an initial set of these parameters any arbitrary parameter you can choose and then the expectation stage what we do we compute the probabilities likelihood of x of the data to a particular known k th Gaussian cluster. So, for each cluster center we compute its likelihood which means now I have to compute simply the probability given that distribution and which has to be multiplied by.

So, this is the likelihood which we need compute and you can see that we are computing the probability of corresponding probability of x_i belonging to this distribution this is given by the probability distribution function, it needs to be multiplied by the mixture coefficient π_k and z_i is the normalizing coefficients because now you are considering k classes.

$$z_{ik} = \frac{1}{Z_i} \pi_k N(x_i | \mu_k, \Sigma_k)$$

So, we are considering the i th pixels and this is how we compute the likelihood. And then we assign the this pixel i th pixel or x_i we can assign it to the m th cluster whose likelihood is maximum.

However, now this assignment is an optional step it is it should not be taken at this at every iteration only at the final stage when you computed all the probabilities of likelihoods of every known points then only you assign the cluster which has the maximum likelihood and the maximization. So, this is the expectation stage.

So, you are computing the likelihoods and you from there know you get a redistribution of as redistribution of data. In the maximization stage once again with this likelihood of data you re estimate the parameters of the Gaussian distribution. So, you continue this process. So, this re estimation process let me explain more in an elaborated way.

(Refer Slide Time: 28:24)

Parameter re-estimation

$$z_{ik} = \frac{1}{z_i} \pi_k N(x|\mu_k, \Sigma_k)$$

← Normalizing factor

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

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

$N_k = \sum_i z_{ik}$
 Expected number of pixels in class k .

So, as we have seen that we can compute the likelihood of the data this is the expression by computing the probability using the probability density function called kth Gaussian distribution in the kth Gaussian component multiplying with the corresponding mixture coefficients and then normalizing with respect to all the classes or components clusters.

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

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

So, if I have these likelihood we can consider the strength it is as if it is representing the number of points in a cluster. So, it is not a discrete number as you can see it is sum of probabilities of all pixels whose sum of likelihood of all pixels which may which is not a not an integer, but it is an estimate of the number you mean probability estimate of the number.

So, if the likelihood is 1 and 0, then you get integers integer numbers, but in this case we will get an estimate of that number and this is how it is defined it is. So, it is expected

number of pixels in class k which is which has a fractional component also here in this framework. So, now you compute the mean of the distributions you compute the mean of the cluster k by considering the corresponding say likelihood as the as weights for a particular data points.

So, it is weighted know mean of the data points belonging to that cluster. So, the belongingness is defined by likelihood that is how you get the mean of the cluster in the same way you get the co variance matrix of the cluster. So, this is how the co variance matrix is also computed. So, now, you see that from here you get atleast two sets of parameters of mean and co variances and finally, the mixture coefficients that π_k those are also found as a fraction of total no N here N is the total number of pixels in the image.

So, N_k by N now total number of elements in the data points its not only image its a data points. So, N_k by N will give you the corresponding know probability of a or prior probability of a class k . So, it is acting like a mixture coefficients. So, you iterate this process and through iteration you know you can converge you can after convergence as I mentioned then you assign the highest assign the corresponding cluster which is represented by a corresponding Gaussian component to a pixel which as a highest likelihood to that component.

So, this is how the Gaussian mixture of Gaussian method could be used for clustering. So, let me stop here and will continue our discussion on classification in the next lectures.

Thank you very much for your attention.

Keywords: Clustering, K-means clustering, K-medoids, expectation maximization.