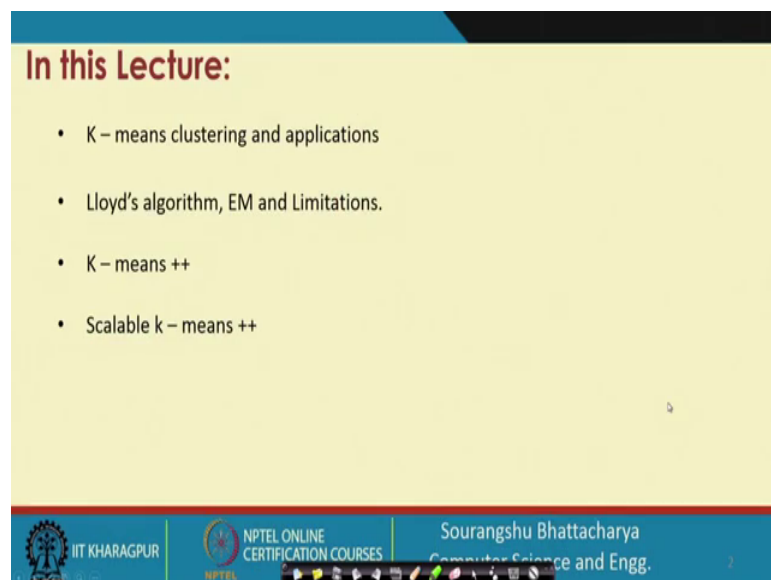


Scalable Data Science
Prof. Sourangshu Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 23
Clustering

Hello everyone, welcome to the 23rd lecture of NPTEL course on Scalable Data Science. I am Professor Sourangshu Bhattacharya from Computer Science and Engineering at IIT Kharagpur. And today we are going to discuss about Clustering ok.

(Refer Slide Time: 00:36)



In this Lecture:

- K – means clustering and applications
- Lloyd’s algorithm, EM and Limitations.
- K – means ++
- Scalable k – means ++

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Sourangshu Bhattacharya
Computer Science and Engg.

So, in this lecture we are going to discuss about k-means clustering and its application. So, we are going to talk about the general problem of clustering and then we are going to talk about the k-means clustering. And then we are going to discuss the Lloyd’s algorithm which is also sometimes called the k-means algorithm or k-means clustering algorithm, which is also same as the EM algorithm. We will also discuss a little bit not much. And then we will discuss; what are the limitations of this Lloyd’s algorithm.

And then we will discuss recent work which is the; actually so it is 2007 work, which is the k-means plus plus, which takes care of some of the limitations of the Lloyd’s algorithm. And then finally, we will see an algorithm which is called as scalable k-means plus plus, which takes care of some of the limitations of k-means plus plus which is to they say that it makes them more scalable ok.

(Refer Slide Time: 02:00)

Clustering

- Unsupervised learning
 - When your data doesn't have labels
- Useful for
 - Detecting patterns e.g. in image data, customer shopping results, anomalies...
 - For optimizing, e.g. distributing data across various machines, cleaning up search results, facility allocation for city planning...
 - when you "don't know" what it is exactly that we are looking for

[Image segmentation via clustering, James Hayes]

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, as we know clustering is basically an instance of unsupervised learning problem. So, that is when your data does not have labels ok. So, it can be used for detecting patterns like in image; so for example, in this particular representation you can see that: so in this representation clustering is being used to segment the images. That is the image has various portions, like the water body, the road, or the sand and then it has the grass and it has sky so, and it also has clouds. So, we are using a clustering algorithm to segment out this portions of the image.

Other pattern detection could for example, include problems regarding shopping results or so it you could try to segment the customers into different groups. And also maybe anomalies, like you could try to say try to group the different types of fault patterns in a particular transaction or in a particular machine using clustering. And then from that you can try to derive some inside.

So, it can be also use for optimizing for example, you could be used for distributing data across machines, it could be use for cleaning of search results, it could be use for facility allocation for city planning. So, in general it can be used for removing outliers and abnormal data points. So, and more generally unsupervised learning is very useful for exploratory data analysis, which is to say that you initially do not know what to do and then you are doing some data analysis.

So, for example, in this figure you can see that so this is some gene microarray data which has been clustered using dendrogram. And then you can see it finds groups of similar clusters. So, this is one group this is another group and so on and so forth. And maybe one can create a phylogenetic tree out of such clustering ok.

(Refer Slide Time: 05:03)

Clustering: basic idea

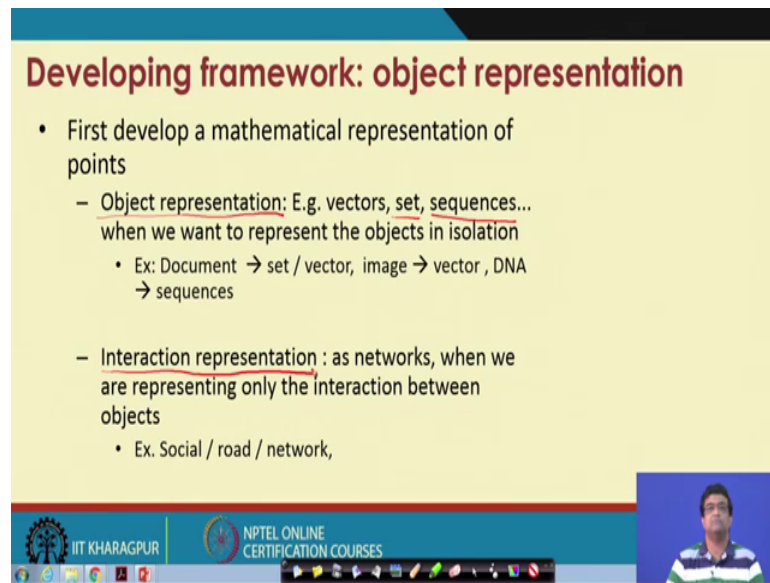
- Grouping objects into small number of meaningful groups
 - How to define similarity / distance between objects?
 - What is meaningful?
 - How many groups?
 - Typically there is no supervision

The slide contains two diagrams illustrating clustering ambiguity. The left diagram shows a set of points in a square with a dendrogram that groups them into three clusters. The right diagram shows the same set of points with a different dendrogram that groups them into two clusters. Red arrows point from the text questions to the corresponding dendrogram structures.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, what is the basic idea? The basic idea is that you have to group the objects that you want to cluster into a small number of basic groups ok. So, the main aspects here are what are the groups. So for example, how do you define similarity and distance between object? So this is first important thing. So, for example, so this is basically this figure illustrates the fundamental limitation of a clustering. So, you could try to cluster it using the first clustering which is this or you could try to cluster it using the second clustering which is this. And it is not clear which of them is the base so this is the basic problem of unsupervised learning that it is an impose problem ok. So, in order to make it concrete we have to define certain things. So, first of all you have to define a similarity measure or a distance between objects and then which will basically define what you mean by meaningful similarity. And then you may have to tell how many groups and so on and so forth ok.

(Refer Slide Time: 06:28)



Developing framework: object representation

- First develop a mathematical representation of points
 - Object representation: E.g. vectors, set, sequences... when we want to represent the objects in isolation
 - Ex: Document → set / vector, image → vector, DNA → sequences
 - Interaction representation: as networks, when we are representing only the interaction between objects
 - Ex. Social / road / network,

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, essentially, so the first thing is to develop an object representation ok. So, how do you develop an object representation? So, you can either depending on the situation you can represent the objects using vectors which is the most common case and which is also the case we will be looking at. But you could also represent the object using a set that is one way. For example, in case of images maybe you want to look at the objects as sets or you may want to look at the objects are sequences which is the case when let say you are trying to clusters streams or something like that or DNA sequences for example.

And the second thing that you have to decide is the interaction representation ok. So, basically you could either take all pairs of interactions which is the case in many cases or you could take the interactions on a network which is sometimes called clustering on network and so on and so forth.

(Refer Slide Time: 07:55)

Clustering framework: distance function

- In the object representation we need an appropriate distance function
 - L_p norms for vectors
 - Jaccard distance for sets
 - Edit distance for sequences
 - Divergences for probability distributions...
- Typically, nice to have the metric properties
 - $d(x, x) = 0, d(x, y) \geq 0$
 - $d(x, y) = d(y, x)$
 - $d(x, y) + d(y, z) \geq d(x, z)$
- Also nice if it is easy to calculate "average"
$$\min_x \sum_i d(p_i, x)$$

The slide includes a diagram showing a central point labeled 'centroid' with arrows pointing to several points labeled P_i . Handwritten red annotations include a curly brace around the first two metric properties and an arrow pointing from the centroid label to the central point.

Now, once you have defined these two; so once you have defined the object and the interaction. Then you need to come up with the distance function which is probably the most important thing. So, the distance function could for example, be the L_p norm between vectors or Jaccard distance for sets or edit distance for sequences and or divergences for probability distributions. So, these are all standard distance functions. So for this lecture we will mostly with dealing with the vectors and we will look at L_p or rather more precisely L_2 distance which is also called a Euclidean distance, but all the other measures are possible ok.

So, typically distances will have these three properties metric property; that is the distance should be positive, it should be symmetry, and it should obey what is called the triangle inequality which is that the distance between x and z will always have to be smaller than the sum of the distance between x and y , which is the third point and y and z ok. So, these are some of the properties under which our analysis we will define our algorithms and we will analyze them. The second interesting proper or important property is that we should be able to calculate an average of the data points.

For example, which minimizes; so for example, you could if you have data points P_i ok, you could try to calculate the average. So, this is the average or sometimes also called centroid of these data points P_i ok. Such that the x which minimizes the distance the or the sum over the distance from all such data points p_i to x ok. So, in other words if you

have a lot of such points, maybe your x should be somewhere here which minimizes all these pair wise distances or sum of all these pair wise distances ok. So, we should have a representation of x so this is another property which we will need ok.

(Refer Slide Time: 10:45)

Distance function: lp norms

- L2 norm/Euclidean distance

$$D(x,y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

Handwritten note: $\bar{x} = \frac{1}{N} \sum_{i=1}^N P_i$ (centroid)

- L1 norm
- L-infinity norm
- Easy to calculate averages.
- Also related is cosine distance

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, these are some of the formulas for the distances, most importantly we will be using the L 2 distance; which is the Euclidean distance and so one can also use L 1 distance and L infinity distance. And we can easily calculate the averages so, for example, in this case it is just the centroid. So, your basically x is going to be just 1 by the number of points, and then sum over i is equal to 1 to $N P_i$ so this is the formula for the centroid. So, in case of vectors and when the distance is L 2 distance it is very easy to calculate the centroid. Another related distance is the cosine distance.

(Refer Slide Time: 11:44)

Objective function

- Specifying number of clusters } →
 - K-means / K-median
- Specifying cluster separation / quality } →
 - e.g. radius of cluster, Dunn's index,...
- Graph based measures → spectral
- Working w/o an objective function →
 - Hierarchical clustering schemes

Now so one can use so, now, this objective function can be specified in many ways. So, for example, one way is to specify the number of clusters ok. So, if you have k clusters then you can you have a well defined objective function which you want to minimize for obtaining a good clustering we will see how. So, this so basically the k -means clustering algorithm which we will discuss is comes in this category.

Other than that sometimes you want to specify the cluster separation or cluster quality index ok. Something like the maybe a threshold on the radius of the cluster or maybe something like Dunn's index and things like that ok. So, this is another way of doing clustering for example threshold based clustering is one example. Yet other times you may want to specify some graph based measure ok.

So, these are sometimes called Spectral clustering techniques where you basically define a graph over all the data points and then you try to minimize a graph based metric over this set of data points. And finally, you may not want to specify any matrix; for example, in case of hierarchical clustering you just try to create hierarchies and try to create clustering at different levels which is also called the dendrogram. So, arguably the first one is the most widely used a clustering technique and in this lecture we will be discussing the first one which is the k -means clustering ok.

(Refer Slide Time: 13:53)

K-means

- Distance function is typically L2
- $C = \{c_1, c_2, \dots, c_k\}$, $\text{cost}(C) = \sum_x \min_{c_x} d(x, c_x)^2$
- Find C to optimize the above cost
 - Leads to a natural partitioning of the data
- Large amount of work, both from theory & data mining community
 - Great example of divergence between theory and practice and how that prompted new research directions for both

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, what is k-means clustering? So, K-means. So, as we have already discussed so basically the distance function in case of k-means clustering is typically L 2 distance which we have already design and describe which is also the Euclidean distance. And now we also specify a k; a value of k which is the number of cluster. Now, once we are specified all this, we need to find a set of cluster centers c which is c 1 till c k.

So, we need to find k cluster centers, one for each of these clusters such that the so, given all the data points x sum over all the data points the minimum over c x distance from x to c x whole square is minimized. So, what this minimum over c x does, is it calculates the distance of x from it is nearest cluster center. So, it basically calculates the distance of x from it is nearest cluster center ok. And the total such distance should be minimized ok. So, this is the k-means clustering formulation ok. So and the problem is that you have to find this set c.

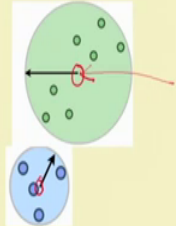
So, you are not given this cluster center you are only given the value of k and you have to finds c 1 till c k. That needs to the minimization of the above cost and this leads to a natural partitioning of the data. So, this algorithm has a large amount of work both from theory and data mining community and basically it is also used very extensively in practice.

(Refer Slide Time: 16:16)

k-means objective: alternate view


- Define “best” k-clustering of the data by
 - minimizing the “radius” of the each cluster

$\text{minimize } \sum_i \text{radius}(C_i)$



- minimizing the variance of each cluster
 - The mean $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ is the “expected” location of a point
 - Hence variance of $C_i = \sum_{x \in C_i} \|x - c_i\|^2$

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES



Another way of looking at the same problem is that the best k clustering of data can be obtained by minimizing the radius of each cluster ok. So, in other words another way of putting the same thing is once you specify a cluster center. So, once you specify a cluster center, you can find all the points that belong to this cluster those are precisely the points which are closer to this cluster center than any other cluster center which is there ok.

And now the problem of k-means is if you try to draw a circle something like this where all the points belonging to this cluster center will fall within the circle. Then the problem of k-means clustering is trying to minimize the radius of this circle. Again so we have already discussed this that once you know the clustering it is very easy to calculate the cluster center which is given by this formula and vice versa.

So, once you know the cluster center it is very easy to calculate the clustering that is which cluster which points belong to which cluster ok. So, so this minimizing radius can also be thought of as minimizing the variance, which is minimizing this quantity for each cluster center ok. So, we will see how this works out. So, now, we describe the Lloyd’s algorithm which is the algorithm for k-means clustering ok.

(Refer Slide Time: 18:32)

The canonical algorithm: Lloyd's algorithm

- Iterative algorithm
- Iterate
 - Find current centers of partitions
 - Assign points to nearest centers
 - Recalculate centers

Handwritten annotations: A red arrow points from the 'Iterate' bullet to the list of steps. A red bracket groups the last two steps, with an arrow pointing to the word 'Clustering'. Another red arrow points from the 'Recalculate centers' step to the word 'Centroid'.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, what is the Lloyd's algorithm it is an iterative algorithm it is a very simple algorithm ok. It says you initially randomly select the cluster centers ok. So, given some set of points you can randomly select some of the given points as cluster centers or you could choose randomly any other points as cluster centers ok. And then you do these two steps iteratively ok. So, you assign each point to or rather let me put it this way instead of saying these two steps you should do these two steps iteratively ok.

So, once you have the cluster centers you assign the points to the nearest cluster centers. In other words you actually calculate the clustering ok. So, you calculate which points belong to which clusters and once you have done this you recalculate the cluster centers which is the centroid calculation ok. So, you do the centroid calculation again and then you keep repeating these steps again and again. So, these recalculating the recalculating the centers gives you the current centers of the partitions and then you again repartition using now the current centers.

(Refer Slide Time: 20:32)

Lloyd's algorithm

- Iterative algorithm
- Iterate
 - Find current centers of partitions
 - Assign points to nearest centers
 - Recalculate centers
- Stopping criteria
 - when no (or small #) points change cluster →
 - when cluster centers don't shift much →
 -

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And so, this is an iterative algorithm so when do you stop this iteration. So, one time that you can stop this iteration is when no points change their clusters ok. So, every time the points, the clustering remains the same. So, every time the points are assigned to the same cluster or another way of stopping could be that you maintain the cluster centers from the previous cluster and then you see how much this cluster centers are shifting and if they do not shift much. Then you can stop the algorithm so both of these are equivalent ok.

(Refer Slide Time: 21:23)

Lloyd's Method: k-means

Initialize with random clusters

$k=3$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this is a run through of the algorithm. So, you first initialize and in this case we are showing all the data points and we have K is equal to 3 ok. So, we have K equal to 3. So, these are the three initialize clusters cluster centers rather.

(Refer Slide Time: 21:57)

Lloyd's Method: k-means

Assign each point to nearest center

Now, next is for every other point you assign it to the nearest clusters. So, basically this point gets assigned to this cluster, this point gets assigned to this cluster and all the other points gets assigned to the blue cluster.

(Refer Slide Time: 22:18)

Lloyd's Method: k-means

Recompute optimum centers (means)

And now you recompute the cluster center so, if you see now so earlier you had assigned this point to this point. Now because both of these points are in this cluster your new cluster center is this green point which is the centroid of these 2 points. Similarly your new cluster center for this cluster the red cluster is this point which is the centroid of these 2 points. And then the new cluster center for the blue cluster is this point which is the centroid of all these 5 points. And then you keep repeating this.

(Refer Slide Time: 23:09)

Lloyd's Method: k-means

Repeat...

(Refer Slide Time: 23:18)

Lloyd's Method: k-means

Repeat...Until clustering does not change

(Refer Slide Time: 23:21)

Lloyd's algorithm: analysis

- k centers, N points, d dimensions
- Time taken to calculate new cluster assignments : $O(kNd)$
- Time taken to calculate new centers : $O(Nd)$
- Number of iterations?

The slide includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES. A small video inset shows a presenter in the bottom right corner.

So, you keep repeating it and then it converges to a particular clustering ok. So, it converges so you can see that as you keep repeating things, it converges to a particular clustering.

(Refer Slide Time: 23:36)

Lloyd's Method: k-means

Repeat: Recompute centers

The diagram illustrates three clusters of points. Each cluster has a central point (center) and several peripheral points. Lines connect the peripheral points to their respective centers. The centers are colored green, red, and blue. The text 'Repeat: Recompute centers' is displayed above the diagram.

The slide includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES. A small video inset shows a presenter in the bottom right corner.

So, you see up till now this point was belonging to this cluster, but now it gets reassigned to this cluster because it is closer to this cluster center and then you get the final clustering ok. So, the analysis of this algorithm so basically suppose you have k centers, N points

and d dimensions Then you can see that time taken to compute new cluster assignments is $O(kNd)$.

Because d is the time taken to compute the distance between any 2 points, Nd is the time taken to compute the distance of all the points from k many cluster centers. So, for each cluster center and each point you have to compute the distance from each cluster center and then assign it to the closest cluster center. And then of course, ond is the time taken to recompute the new centers. And so, the question comes how many iterations do you need to run?

(Refer Slide Time: 25:06)

Lloyd's algorithm: convergence?

- For any current clustering, consider the objective function

$$\text{cost}(C) = \sum_x \min_{c_x} d(x, c_x)^2$$

At every step of the algorithm, this potentially decreases

Handwritten note: $d(x, c_x)$

The slide also features the IIT Kharagpur and NPTEL Online Certification Courses logos at the bottom, along with a small video inset of the presenter.

And so we have seen that this is the objective function for k-means clustering. So, so, another point is that you can see that this cost decreases at every step ok. This is because. So, if you see if you consider the. So, so for any point when it gets assigned to it is nearest point so, the $d(x, c_x)$ becomes minimum for that point. So, reassignment to the nearest point minimizes this for every point ok.

And hence after every iteration and similarly when you recompute the cluster center ok. So, for the nearest point the so, for the nearest point basically the or for any given cluster the distance of the centroid is or the centroid is by definition is the point which is at a minimum distance from all the points in that cluster.

Hence this sum over distances is also minimized when you recompute the cluster centers. So, both steps actually reduced the value of this total objective function, which is the sum total of cost of each cluster. So, one way of thinking about it is a sum total of cost of each cluster, another way of thinking about it is that it is sum total of cost of each point or the distance of each point to its closest neighbor ok.

(Refer Slide Time: 27:28)

Convergence

- It is known that in some datasets, Lloyd's algorithm can take exponential (2^n) number of steps
 - These tend to be unrealistic
- Bigger problem is where it converges to--- depends on initialization

Should have put single cluster here

and two here

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

So, since the objective function reduces in every iteration. So, this algorithm is basically guaranteed to converge and there are some analysis which shows that in the worst case it may take exponential number of steps, but typically you do not see so many steps. So, typically you see whenever you run the k-means clustering it converges within a finite number of steps ok. So, what is the problem? Ok. So, the problem is this that so, the problem is where do the cluster centers like after it has converged.

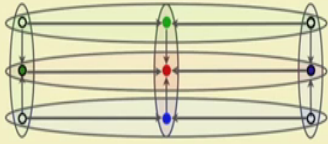
So, for example, this here I am showing one example output from a k-means clustering so you can see that actually here that 2 clusters are more well separated and here the 2 clusters are merged, but still the final solution here is that there are 2 cluster centers green and purple here and the corresponding clustered points are shown in the color. Whereas, there is only one blue cluster center here so this kind of solution is possible ok.

(Refer Slide Time: 29:09)

Convergence Analysis

Lloyd's Algorithm can be thought as a generalization of EM -algorithm for estimating mixtures of Gaussian distribution. } → Max Loglikelihood

Finds a local optimum



That is potentially arbitrarily worse than optimal solution

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So this figure actually gives you an idea a bigger idea of why this kind of solution is possible. So, before going into this figure let me describe. Let me tell you that this Lloyd's algorithm can be thought of as a generalization of the EM algorithm or the expectation maximization algorithm for estimating the mixture of Gaussians distribution ok. So, that also gives a kind of proof that that the algorithm will always converge, but both this so the EM algorithm is well known to be stuck in the local minima.

So, the expectation maximization algorithm is a supposed to maximize the log likelihood log likelihood function of this distribution ok. And it is known that the EM algorithm gets stuck in a local optima ok. So, let us see how it can get stuck in the local optima. So, this is one example that illustrates that so suppose you randomly choose suppose this is your data set and you randomly choose this point as the initial blue cluster centers, this point as the initial red cluster center and this point as the initial green cluster center ok.

So, so when you reassign the points these two points will become green ok. So, this will be the green cluster this will be the red cluster and this will be the blue cluster where as if you choose this to be the green cluster center this to be the red cluster center and this to be the blue cluster center

(Refer Slide Time: 31:38)

Convergence Analysis

Lloyd's Algorithm can be thought as a generalization of EM –algorithm for estimating mixtures of Gaussian distribution.

Finds a local optimum

That is potentially arbitrarily worse than optimal solution

$k=3$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Then this will be the green cluster, this will be the red cluster, and this will be the blue cluster ok. Now as the iterations progresses the cluster center won't change because this is in the middle of these two and similarly this is in the middle of these two. So, in both cases the cluster center will not change and both are 3 clustering's ok, but both are for k is equal to 3. But you can see that the first clustering has a much more distance between the points that are assigned to cluster then the second clustering.

So, here the distances at just this plus this whereas, here the distances at this plus this ok. So, and both of them are optimum ok. So, depending on the initialization you can arrive at either of these solutions ok. So, basically this is an this is an example of how the solution can come. Now, note that you can actually make this arbitrarily worse. So, it is one may ask that maybe it is a problem, but it is not too bad, but that is not the case because you can make this clusters as far apart from each other as you want ok.

So, you can make these points very far apart in this direction and still the logic that I give will work ok. So, not only is it and local optimum it is actually arbitrarily worse than the best solution or the global optimum ok. So, in order to solve these problems in 2007 Sergei Vassilvitskii and David Arthur they came up with an algorithm called k-means plus plus.

(Refer Slide Time: 33:51)

Challenge

Develop an approximation algorithm for k-means clustering that is competitive with the k-means method in speed and solution quality.

Easiest line of attack: focus on the initial center positions.

Classical k-means: pick k points at random.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The slide features a yellow background with a blue header. The text is in black, with the key phrases underlined in red. A small video inset of the speaker is visible in the bottom right corner.

So, the algorithm so, the line of attack is something like this that you. So, so clearly the initialization is the problem in case of k-means clustering ok. So, if you are starting with a bad initialization then there is no way you can get to a good solution ok. So, maybe we should focus on the initial center position and the classical k-means solution is that pick k points at random.

(Refer Slide Time: 34:34)

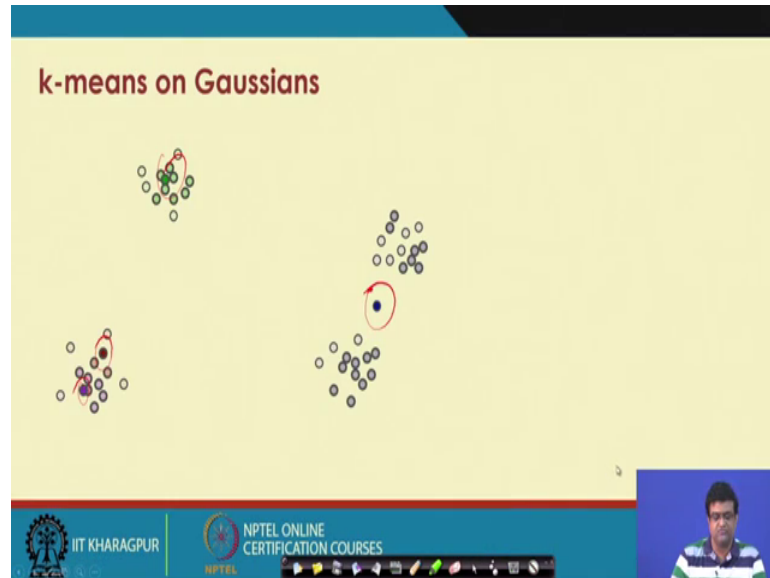
k-means on Gaussians

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The slide features a yellow background with a blue header. It displays four distinct clusters of data points, each represented by a group of small white circles. A small red dot is visible near the top-left cluster. A small video inset of the speaker is visible in the bottom right corner.

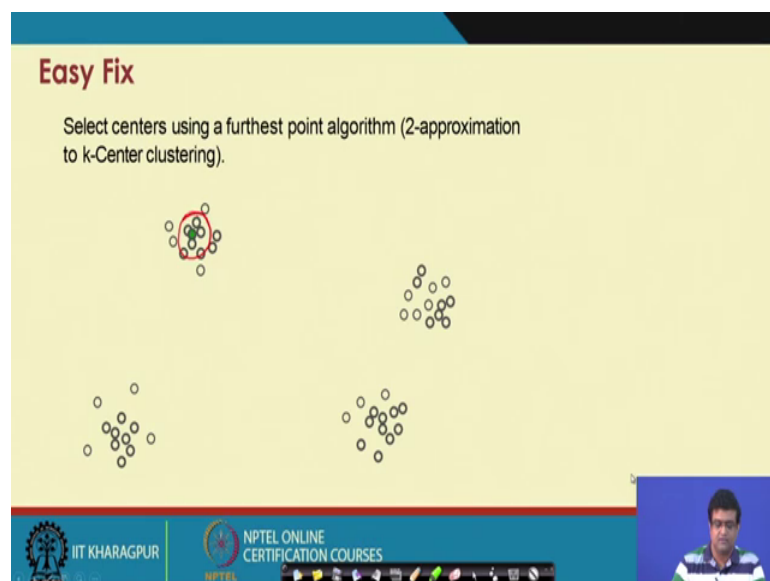
So, what the k-means plus plus proposes. So, so for simplicity we will take Gaussians clusters which are spherical clusters, but this will work the logical actually work for any cluster as we shall prove so.

(Refer Slide Time: 34:56)



So, one may select for example, like we mentioned the blue cluster center here, as a green cluster center here and a red here and a purple here and then it would always give sub optimal clustering ok.

(Refer Slide Time: 35:15)



But instead if we select a cluster centers using the furthest algorithm. So, what is the furthest algorithm? That is you first select 1 cluster center arbitrarily randomly ok.

(Refer Slide Time: 35:43)

Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

So, let us say this is your first cluster center. Then you select the next one as the point which is furthest away from the current cluster center ok.

(Refer Slide Time: 35:55)

Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

And then you select the third point which is furthest away from all whose sum of distances is furthest away from the first two points, that you have selected which is somewhere here ok.

(Refer Slide Time: 36:09)

Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

$\phi \leq 2\phi^*$

The slide features four diagrams illustrating the furthest point algorithm. The top-left diagram shows a cluster of points with one point highlighted in green. The top-right diagram shows the same cluster with two points highlighted in green and purple. The bottom-left diagram shows a cluster with three points highlighted in green, purple, and red. The bottom-right diagram shows a cluster with four points highlighted in green, purple, blue, and red. A handwritten red equation $\phi \leq 2\phi^*$ is written in the upper right area of the slide. The footer includes the IIT Kharagpur logo and the text 'NPTEL ONLINE CERTIFICATION COURSES'.

And finally, you select the fourth point which is furthest away from all the three points you have selected. Then you have a good starting selection of cluster centers. So, this actually gives a two approximation, which is to say that if ϕ^* is your optimal cost for the k-means clustering, then $\phi \leq 2\phi^*$.

Then you can be sure that the cost that you get here ϕ is less than or equal to twice the ϕ^* . So, you cannot do arbitrarily worse than the optimal clustering. You can always be less than twice the cost of the optimal clustering. But this also has a problem.

(Refer Slide Time: 37:13)

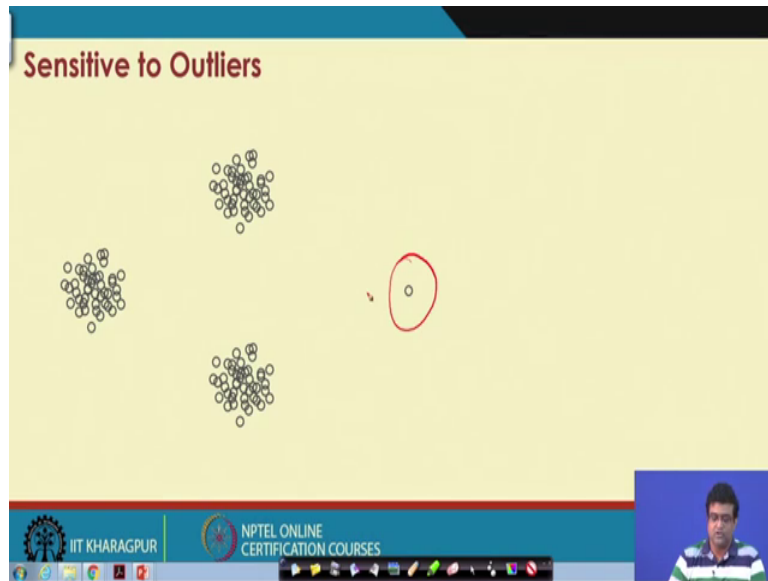
Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

$\phi \leq 2\phi^*$

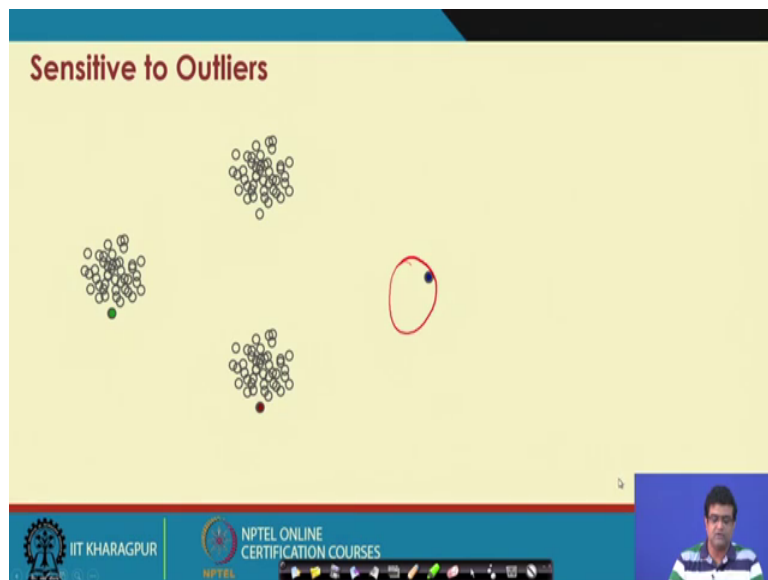
This slide is identical to the one above, showing the same diagrams and handwritten equation. The footer includes the IIT Kharagpur logo and the text 'NPTEL ONLINE CERTIFICATION COURSES'.

(Refer Slide Time: 37:16)



The problem is this suppose you have an outlier like this a single data point which is very very far away from all other data points so this outlier can cause the problems. So, in this case actually there are only three clusters and suppose you want to select three clusters.

(Refer Slide Time: 37:42)



So, then the outlier so the points that you select may be something like this ok.

(Refer Slide Time: 37:48)

Sensitive to Outliers

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And then your clusters will become something like this ok. So, this is not the correct clustering because this is just an outlier point ok.

(Refer Slide Time: 38:02)

k-means++

Interpolate between the two methods:
Let $D(x)$ be the distance between x and the nearest cluster center. Sample proportionally to $(D(x))^\alpha = D^\alpha(x)$

Original Lloyd's: $\alpha = 0 \rightarrow$
Furthest Point: $\alpha = \infty \rightarrow$
k-means++: $\alpha = 2 \rightarrow$

Contribution of x to the overall error

$(D(x))^2$
 $D(x)^2$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, the idea behind k-means plus is that basically all the selection methods of points are somehow using this D of x ; that is to say that the it is choosing the point x according to it is distance from existing center. So, if it is further away from existing centers then it is selected more ok. So, now, so you can think that instead of selecting so one can select proportionally to D of x to the power alpha.

Now if alpha is equal to 0; that means, D of x to the power alpha is always 1 that gives us the random initialization or the original Lloyd's algorithm ok. If alpha is equal to infinity ok, then you always select the furthest one this give the furthest point algorithm ok. So, what k-means plus plus does is it selects the points according to alpha is equal to 2. That is it selects the points according to D x square ok.

(Refer Slide Time: 39:49)

The slide is titled "k-Means++". It shows three clusters of points on a yellow background. A handwritten formula in red ink is shown on the right side of the slide:

$$n \sim \frac{D^2(n, c)}{\sum_n D^2(n, c)}$$

Below this formula, another handwritten equation is shown:

$$D(n, c) = \sum_{c' \in C} D(n, c')$$

The slide also features the IIT KHARAGPUR logo and NPTEL ONLINE CERTIFICATION COURSES text at the bottom. A small video inset of a presenter is visible in the bottom right corner.

So, if you select like this there is a so we will come to the algorithm.

(Refer Slide Time: 40:05)

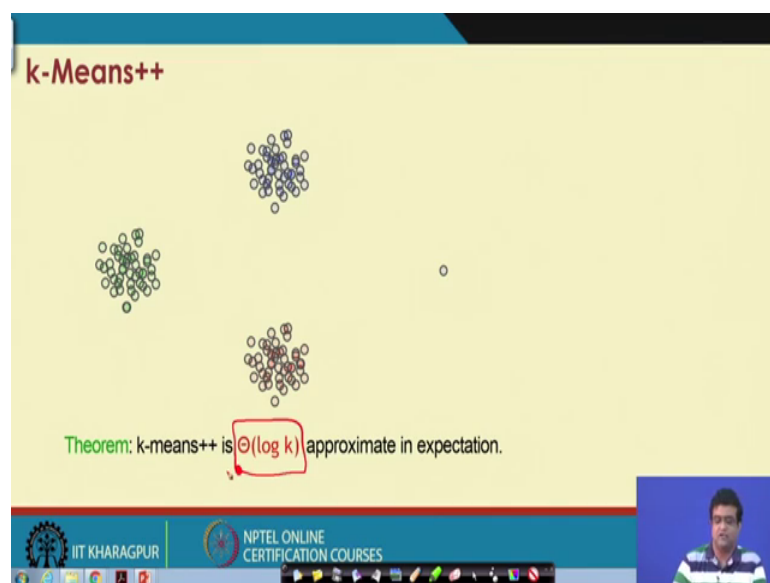
The slide is titled "Proof - 1st cluster". It contains the following text instructions:

- Fix an optimal clustering C .
- Pick first center uniformly at random
- Bound the total error of that cluster.

The slide also features a diagram of several points on a yellow background, with one point highlighted in red. The slide includes the IIT KHARAGPUR logo and NPTEL ONLINE CERTIFICATION COURSES text at the bottom. A small video inset of a presenter is visible in the bottom right corner.

So, let us go to the algorithm so the algorithm is the following that you first select a point. So, once you have selected set C . So, given a set C you select a new point x from the distribution $D^2 \times c$ by $\sum_{x \in D^2} \frac{1}{|D^2|} \frac{1}{|C|} \frac{1}{|C|} \sum_{c \in C} \frac{1}{|C|} \frac{1}{|C|} \dots$. So, that is you select the point which is having the. So, here $D \times c$ is nothing, but sum over all $c \times x$ in c distance between x and $c \times x$ ok. So, so, you basically select or rather sample points from this distribution which selects points which are furthest away from the points you have already selected ok. So, this is an example of how the algorithm operates ok. So, initially you select this green point, blue point, and red point.

(Refer Slide Time: 41:55)



And then you calculate you run the k-means iterations and you get cluster like this. So, this gets assigned to this cluster and this has a theta log k approximation guarantee over the actual k-means plus plus.

So, we will stop here. And in the next part of this lecture we will see why k-means plus plus is better than all the existing algorithms. And we will also see why it is not scalable and how to scale k-means plus plus through large data scale.

Thank you.