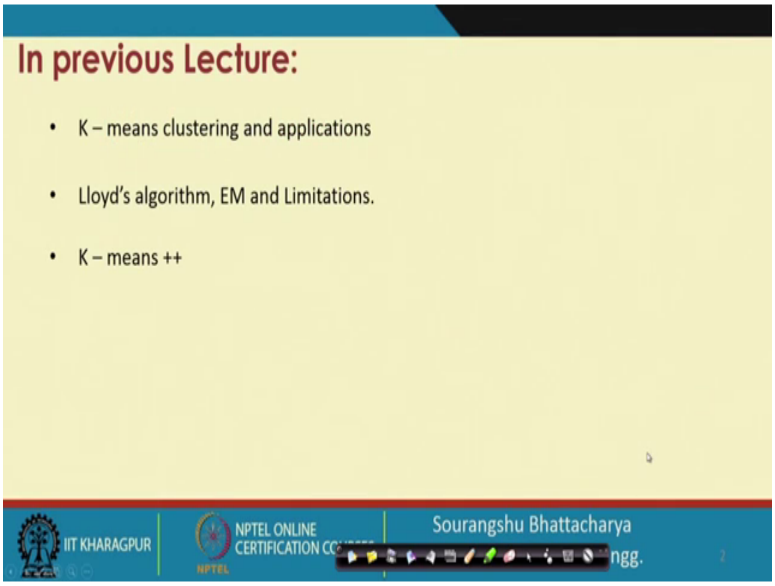
Scalable Data Science

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

Lecture – 23 a
Clustering (Contd.)

Hello everyone welcome back to the 23rd lecture of Scalable Data, NPTEL course on Scalable Data Science. Todays lecture is on clustering and this is the second part of that lecture. I am Professor Sourangshu Bhattacharya from Computer Science and Engineering at I I T Kharagpur.

(Refer Slide Time: 00:35)

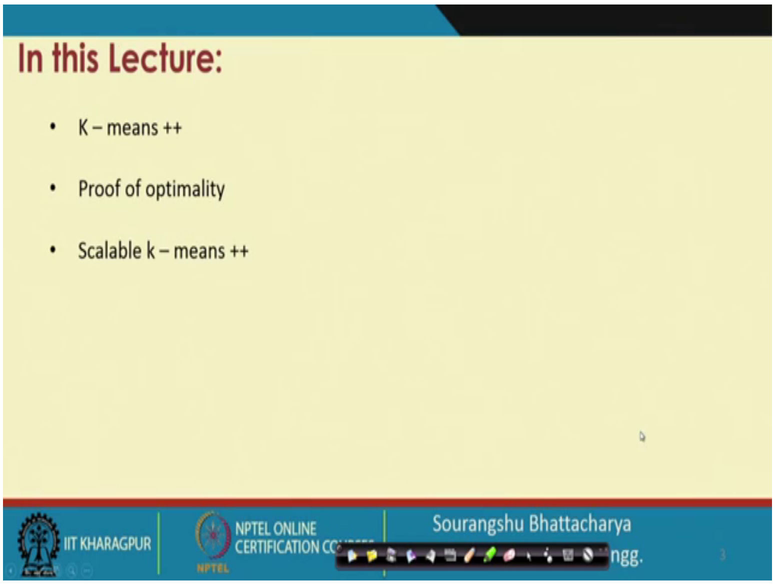


In previous Lecture:

- K – means clustering and applications
- Lloyd's algorithm, EM and Limitations.
- K – means ++

So, in the last lecture on clustering, we have seen the K means clustering algorithm and it is applications. And we have also seens seen the Lloyds algorithm or the Lloyds iteration for updating the clusters and it is connection with EM algorithms and also it is limitations that is to say that it does not give an optimal clustering. And in order to remedy that, we have introduced the K means plus plus algorithm which has optimal guarantees regarding the clustering objective function.

(Refer Slide Time: 01:13)

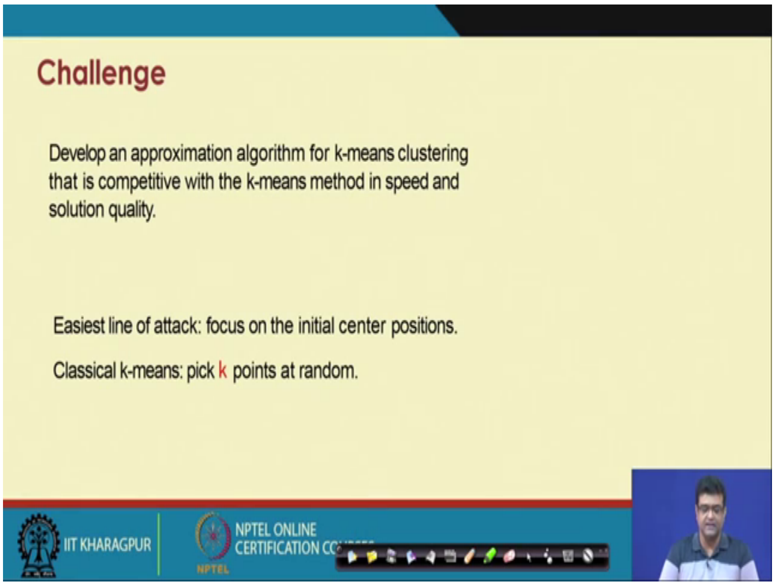


In this Lecture:

- K – means ++
- Proof of optimality
- Scalable k – means ++

So, in this lecture we have going to revisit the K means plus plus algorithm and we are going to show we are going to state the optimality guarantees and show parts of the proof for optimality. And then, we are going to see some drawbacks of K means plus plus algorithm and which are partially at least address by the scalability K means plus plus algorithm ok.

(Refer Slide Time: 01:51)



Challenge

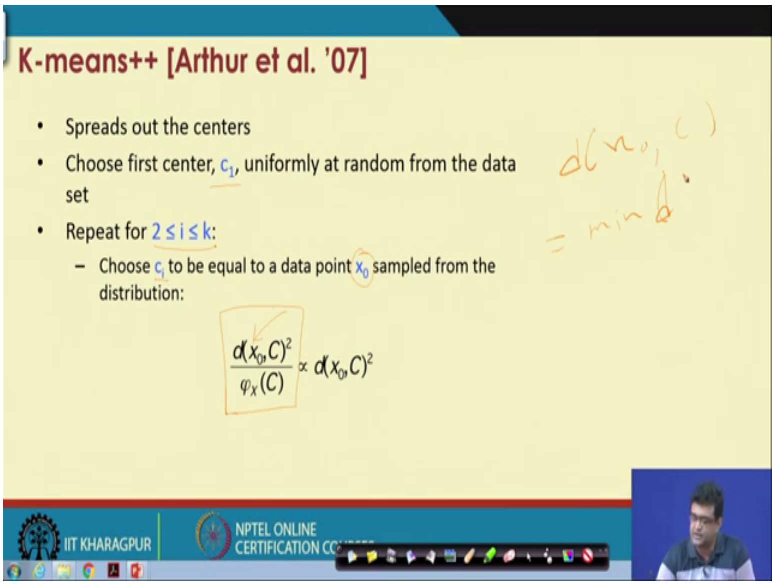
Develop an approximation algorithm for k-means clustering that is competitive with the k-means method in speed and solution quality.

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

Classical k-means: pick k points at random.

So, what is the scalable, so what is the K means plus plus algorithm? As you remember the problem with K means was that the initialization was random and the random initialization did not give any guarantees about the optimality of the final solutionSo, as we discussed, the best way to attack this problem of sub optimality is to focus on the initialization problem and K means plus plus is basically an algorithm to initialize the clustering of K means algorithm from where one can again perform the Lloyds iteration ok.

(Refer Slide Time: 02:41)



K-means++ [Arthur et al. '07]

- Spreads out the centers
- Choose first center, c_1, uniformly at random from the data set
- Repeat for $2 \le i \le k$:
  - Choose c_i to be equal to a data point x_0 sampled from the distribution:

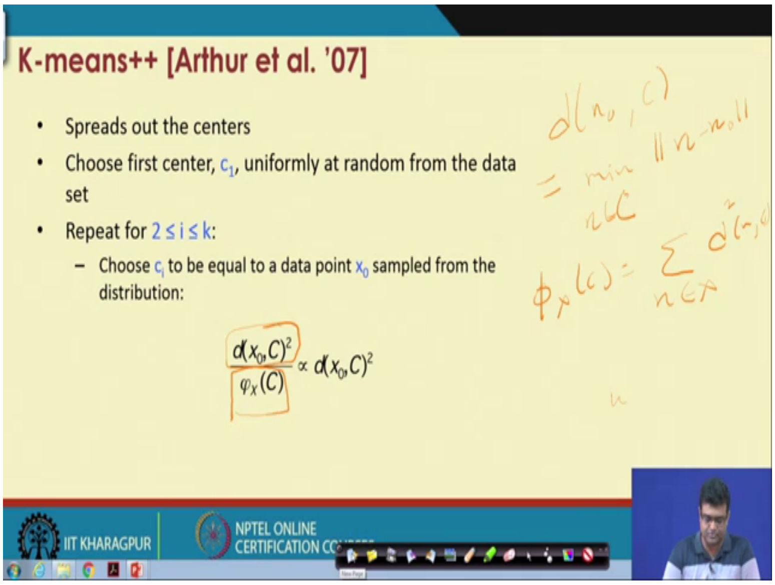
$$\boxed{\frac{d(x_0,C)^2}{\varphi_X(C)}} \propto d(x_0,C)^2$$

$d(x_0, C)$

$= \min d$

So, this is the K means plus plus algorithm just to recollect the K means plus plus algorithm. Basically, the algorithm goes like this that you first choose the cluster centers c 1 uniformly randomly. So, but then for the next few clusters that is for cluster index i from 2 to K, you have to choose a total of K clusters, you choose the cluster c i or rather you sample the cluster c i from the from the set of data points and the probability of sampling is this d. So, for any point let us say x 0, the probability of sampling this point is d of x 0 comma C which is the current clustering square by phi x of C. So, if just to remember, so d of x 0 comma C is nothing but the minimum distance from so minimum distance.

(Refer Slide Time: 04:25)



K-means++ [Arthur et al. '07]

- Spreads out the centers
- Choose first center, c_1, uniformly at random from the data set
- Repeat for $2 \leq i \leq k$:
  - Choose c_i to be equal to a data point x_0 sampled from the distribution:

$$\frac{\boxed{d(x_0,C)^2}}{\boxed{\varphi_X(C)}} \propto d(x_0,C)^2$$

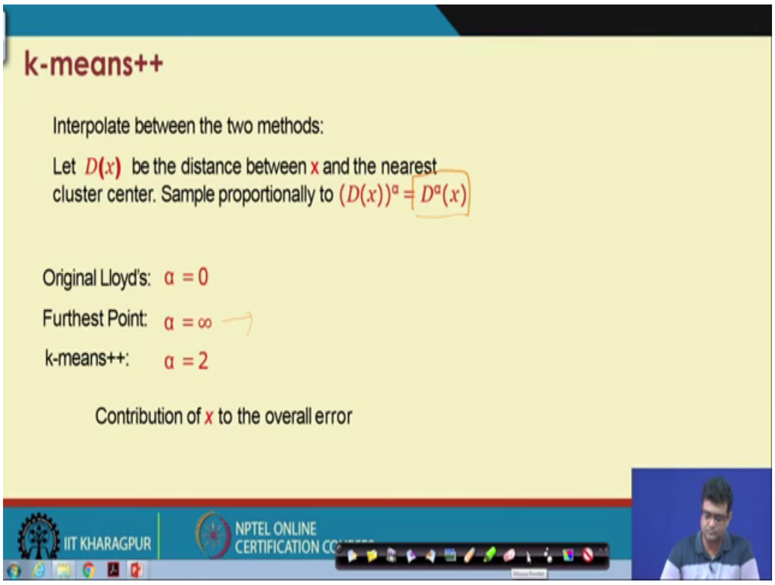
$$d(x_0, C) = \min_{x \in C} \|x - x_0\|$$

$$\phi_X(C) = \sum_{x \in X} d^2(x, C)$$

So, let me yeah, so d of x 0 comma C is the minimum distance, minimum over all x belonging to C, x minus x 0. So, this is the minimum distance and phi of x is of course the clustering objective for the clustering C which is nothing but so, you are phi of x comma C is some overall x belonging to the set x and d of x comma C whole square ok. So, this is what is phi of x comma C.

So, in other words, the points are getting sampled with the probability proportional to their contribution towards the K means objective cluster and this is also called the d square waiting because, actually you would take the square of the distance as the factor by which you contribute .

(Refer Slide Time: 05:50)



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: $\alpha = 0$

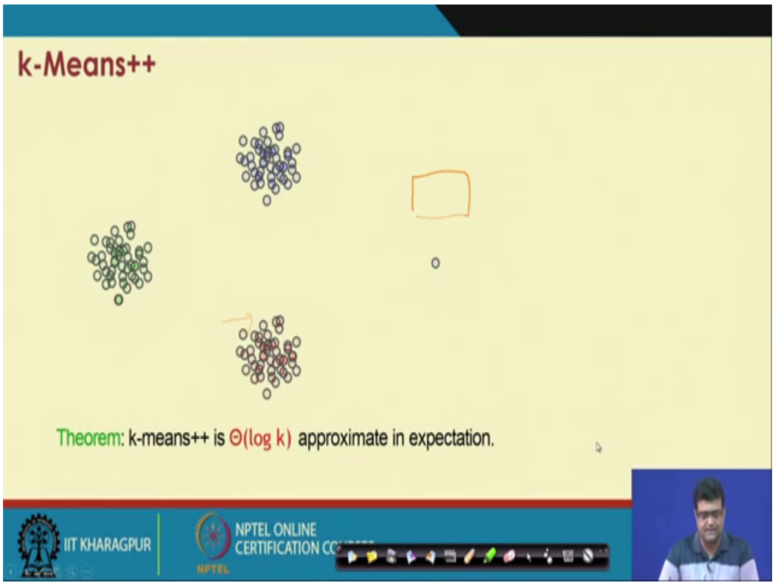
Furthest Point: $\alpha = \infty$

k-means++: $\alpha = 2$

Contribution of x to the overall error

So, we have also discussed that if instead of taking Dsquare if you take so, you can take any D to the power alpha waiting. Instead of taking D square if you take these D to the power 0 waiting, it becomes the original Lloyds algorithm which is the random selection and if you take alpha equal to infinity, then it becomes the furthest point selection which we have discussed ok.

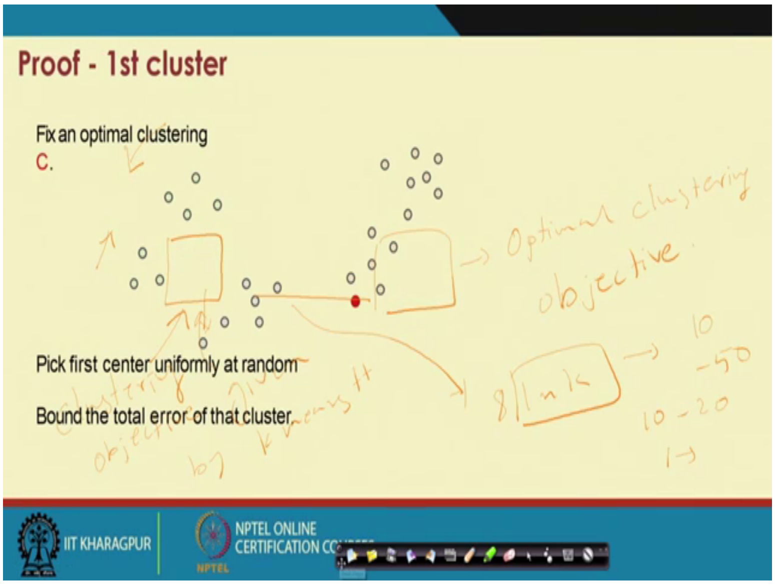
(Refer Slide Time: 06:31)



k-Means++

Theorem: k-means++ is $\Theta(\log k)$ approximate in expectation.

(Refer Slide Time: 06:39)



Proof - 1st cluster

Fix an optimal clustering C.

Pick first center uniformly at random

Bound the total error of that cluster.

Optimal clustering objective
clustering objective given by k means ++
8 ln k
10
-50
10 - 20
1 →
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, D square is kind of the intermediate between these two, so, now, so the main point here is that, so let me come here, so, this is the main theorem about K means plus plus ok.So, the theorem says that suppose you have constructed clustering C using the K means plus plus algorithm ok and then you compute the function phi which is the function we have just describe ok. The function phi of course, is a random function it is a random function because, if you remember every time in K means plus plus algorithm, you sample from a probability distribution.

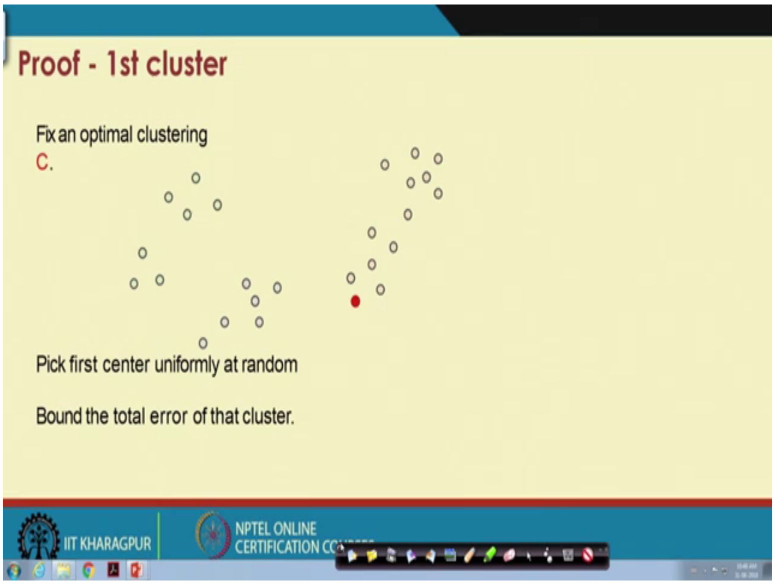
So, every run of the algorithm will give you a different initialization for the initial cluster centers. So, you are expected value of the objective phi with respect to this clustering which is what we are interested in or this is also the average value you can think is bounded by this factor 8 log K plus 2 ok times phi OPT .

So, phi OPT is the optimal clustering objective ok. So, phi OPT is the optimal clustering objective ok and expectation of phi, so, your phi is the clustering objective, given by K means plus plus and this is a random quantity. So, we are looking at it is expectation and this is upper bounded by roughly this factor can be thought to be 8 log K ok. So, it is upper bounded by 8 log K or simply order of log K ok.

It is also sometimes called the competitive ratio, so, sometimes it is said that a k means plus plus is 8 log k competitive ok. So that means, it can never do worst that 8 log k times the optimal objective. Now, k for typical applications you can think is maybe 10 to 50 ok. So, your k is the number of clusters,so, log k is a very small number ok.

So, you are essentially looking at something like 10 or 20 24 approximation. So, your objective can be at the very worst 10 or 24 times the optimal objective that you can ever achieve ok, so that is the guarantee that this initialization is providing. Of course, as you can understand this is this guarantee is a worst case guarantee, in practice it may be much better than it may be very close to it may be just one times the clustering objective .

(Refer Slide Time: 10:19)



Proof - 1st cluster

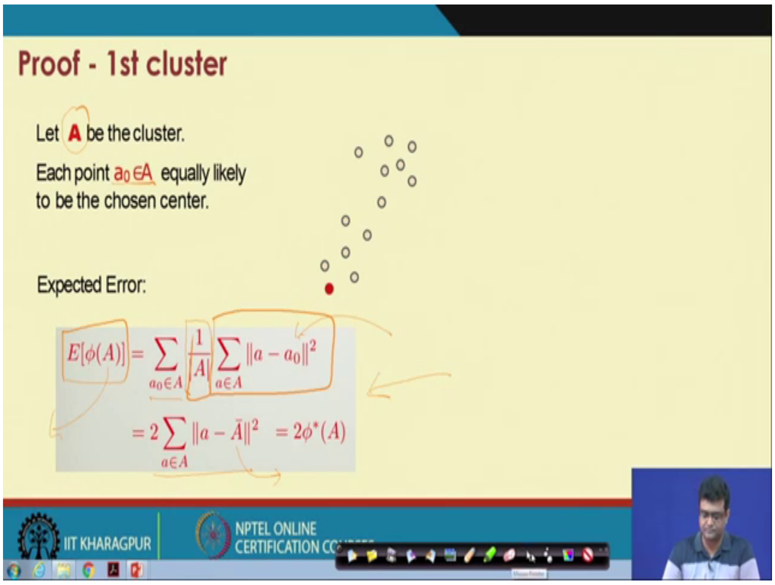
Fix an optimal clustering C.

Pick first center uniformly at random

Bound the total error of that cluster.

So, how do we go about proving this? So, the proof takes basically two steps, so, the proof actually takes three steps, so, the we will not go into the third step, but we will go into the first two steps. So, the first step is to bound the error the total error of the first cluster that is chosen ok.

(Refer Slide Time: 10:59)



Proof - 1st cluster

Let **A** be the cluster.

Each point $a_0 \in A$ equally likely to be the chosen center.

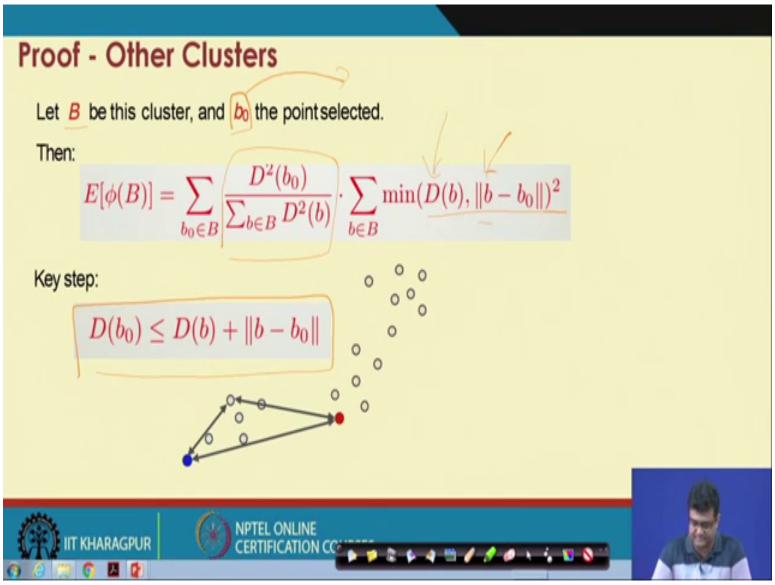
Expected Error:

$$\boxed{E[\phi(A)]} = \sum_{a_0 \in A} \boxed{\frac{1}{|A|}} \boxed{\sum_{a \in A} \|a - a_0\|^2}$$

$$= 2\sum_{a \in A} \|a - \bar{A}\|^2 = 2\phi^*(A)$$

So, since we are choosing the first cluster uniformly at random what we can say is that, so let us say we let us take a cluster A in the optimal clustering and let us take a cluster center a not belonging to A since all a naughts are equally likely, then the optimal or the expected value of the error function over this set of points A is basically or is basically this quantity ok. So, given that you have chosen the clustering objective a this is your loss function because, you have only one cluster centre and that is a naughtSo, this is your k means objective function and the chance that you select this a naught as the cluster centre is 1 by the size of the set A.So, and when you sum overall such possibilities of a naught belonging to A, you get this and now this A bar here is the centroid of the cluster A which for a single clustering is optimal. So, hence you can see that this expected error is exactly equal to two times the optimal clustering error on this cluster A. So, the first cluster that you select, you incurred an error of at most twice the optimal objective on that cluster.

(Refer Slide Time: 12:47)



Proof - Other Clusters

Let B be this cluster, and b_0 the point selected.

Then:

$$E[\phi(B)] = \sum_{b_0 \in B} \frac{D^2(b_0)}{\sum_{b \in B} D^2(b)} \cdot \sum_{b \in B} \min(D(b), \|b - b_0\|)^2$$

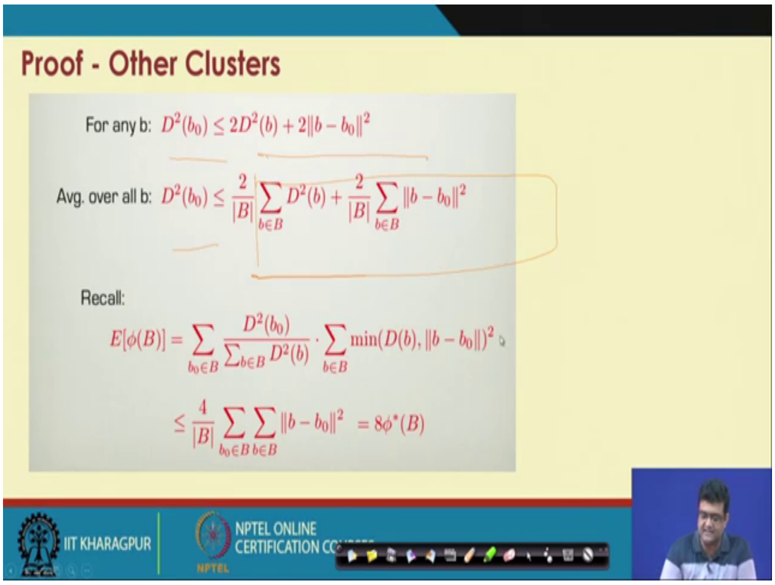
Key step:

$$D(b_0) \le D(b) + \|b - b_0\|$$

Now, the next cluster that you select, again let us say this next cluster that you select is the cluster BSo, and then from this cluster, so B is the cluster in the optimal clustering andthen from this cluster you are selected b naught as the point which is the centre of the cluster ok. So, the chance of selecting b naught is this ok, so, it is because you are selecting with D square waiting. So, the chance of selecting b naught is basically D square b naught by summation over all you know B all points b in B D square b ok and the cost that you incur is the minimum of these two numbers. So, either you are for every point let us say b, you it is either closest to this newly selected cluster b naught or it is closest to the existing point D of b ok.

So, that the cost is minimum of these two, more over from triangle in equality, we can say that this D of b naught is less than or equal to D of b plus the distance between b and b naught ok. So, whichever is the closest cluster? So, you can say that you can say that this holds ok.

(Refer Slide Time: 14:41)



Proof - Other Clusters

For any b: $D^2(b_0) \le 2D^2(b) + 2\|b - b_0\|^2$

Avg. over all b: $D^2(b_0) \le \frac{2}{|B|}\sum_{b\in B} D^2(b) + \frac{2}{|B|}\sum_{b\in B}\|b - b_0\|^2$

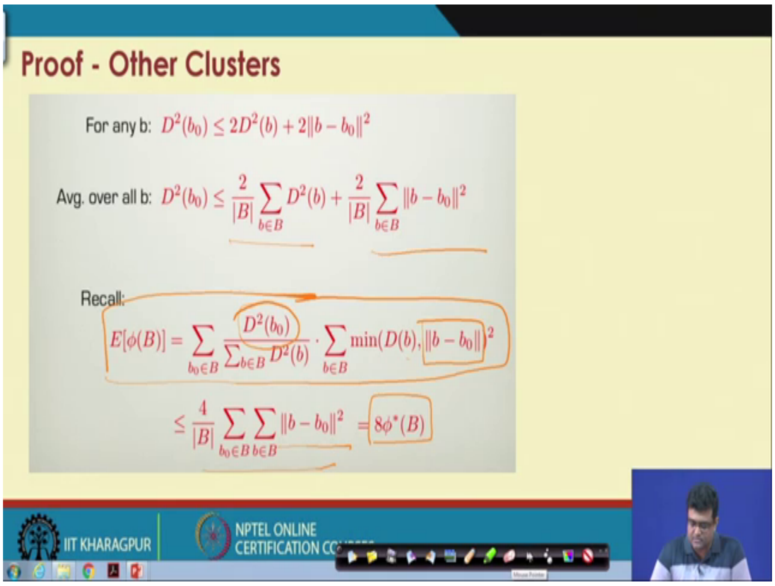
Recall:

$$E[\phi(B)] = \sum_{b_0\in B} \frac{D^2(b_0)}{\sum_{b\in B} D^2(b)} \cdot \sum_{b\in B} \min(D(b), \|b - b_0\|)^2$$

$$\le \frac{4}{|B|}\sum_{b_0\in B}\sum_{b\in B}\|b - b_0\|^2 = 8\phi^*(B)$$

So, furthermore you can just by expanding that you can see that actually this D square b naught is less than or equal to 2 times D square b plus 2 times b minus b not ok. So, this also holds ok and then you want to eliminate b, so, you take average overall B ok and you get that D square b naught is less than this particular quantity. So, this is just you would divide by the size of B and then sum over all elements in B ok.

(Refer Slide Time: 15:45)



Proof - Other Clusters

For any b: $D^2(b_0) \le 2D^2(b) + 2\|b - b_0\|^2$

Avg. over all b: $D^2(b_0) \le \frac{2}{|B|}\sum_{b\in B} D^2(b) + \frac{2}{|B|}\sum_{b\in B}\|b - b_0\|^2$

Recall:

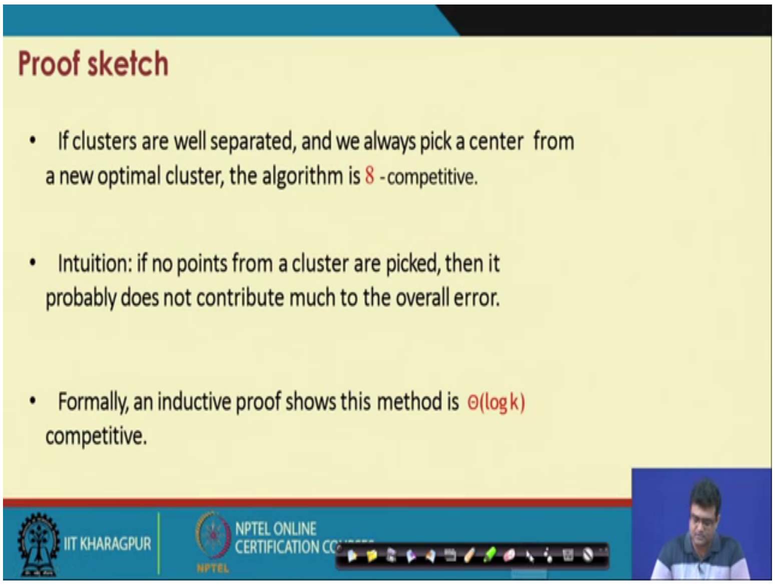
$$E[\phi(B)] = \sum_{b_0\in B} \frac{D^2(b_0)}{\sum_{b\in B} D^2(b)} \cdot \sum_{b\in B} \min(D(b), \boxed{\|b - b_0\|})^2$$

$$\le \frac{4}{|B|}\sum_{b_0\in B}\sum_{b\in B}\|b - b_0\|^2 = \boxed{8\phi^*(B)}$$

So, now given that you see that we had this recall that we had this , so recall that we had this equation in terms of expectation of phi B ok. So, now if you plug in this inequality in this D square b naught ok and in so, you get basically you get two terms, one is this term and another is this term ok. So, for the first term you for this minimum you just plug in as an upper bound this one, so, this minimum quantity is less than both size of b minus b naught square, so mode of norm of b minus b naught square and D of b .

So, for the first term, you plug in b minus b naught square and for the second term, you plug in D of b square, so, D of b square cancels in both cases and then in both cases the upper bound you are left with is b minus b naught square. And so, this is an upper bound over this and when you sum over this it become, so you can see that this b naught is just the centroid and hence this is nothing but 8 times the optimal clustering of b naught ok.

(Refer Slide Time: 17:30)



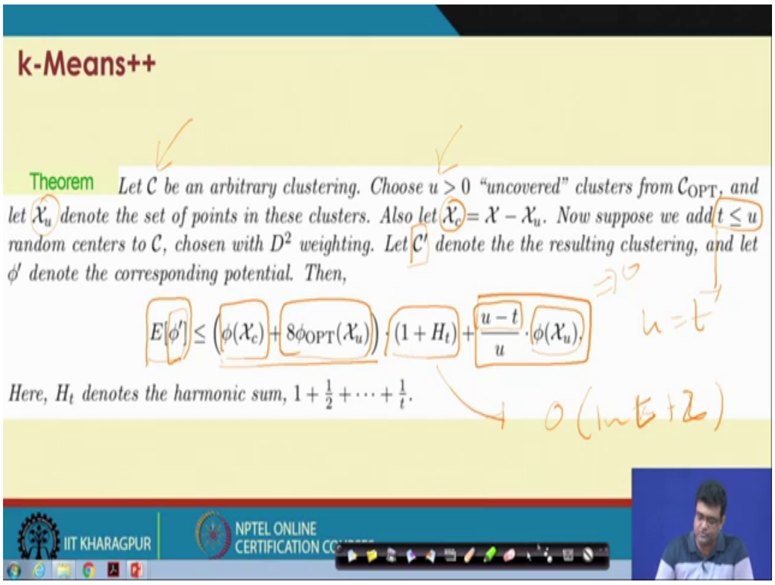
Proof sketch

- If clusters are well separated, and we always pick a center from a new optimal cluster, the algorithm is 8 -competitive.
- Intuition: if no points from a cluster are picked, then it probably does not contribute much to the overall error.
- Formally, an inductive proof shows this method is $\Theta(\log k)$ competitive.

So, what do we have so, in summary, we have that if the clusters are well separated, we can always pick a centre from a new optimal cluster and the algorithm means basically 8 competitive.

So, the intuition here is if no point from cluster is speaked, then it probably does not contribute much to the overall error ok. So, we will not we will not go into the proof as such, but using these two and the following theorems.

(Refer Slide Time: 18:15)



k-Means++

Theorem Let $\mathcal{C}$ be an arbitrary clustering. Choose $u > 0$ "uncovered" clusters from $\mathcal{C}_{\text{OPT}}$, and let $\mathcal{X}_u$ denote the set of points in these clusters. Also let $\mathcal{X}_c = \mathcal{X} - \mathcal{X}_u$. Now suppose we add $t \le u$ random centers to $\mathcal{C}$, chosen with D^2 weighting. Let $\mathcal{C}'$ denote the the resulting clustering, and let ϕ' denote the corresponding potential. Then,

$$E[\phi'] \le \left(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\right) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u).$$

Here, H_t denotes the harmonic sum, $1 + \frac{1}{2} + \cdots + \frac{1}{t}$.

$\Rightarrow 0$

$u = t$

$O(\ln t + 2)$

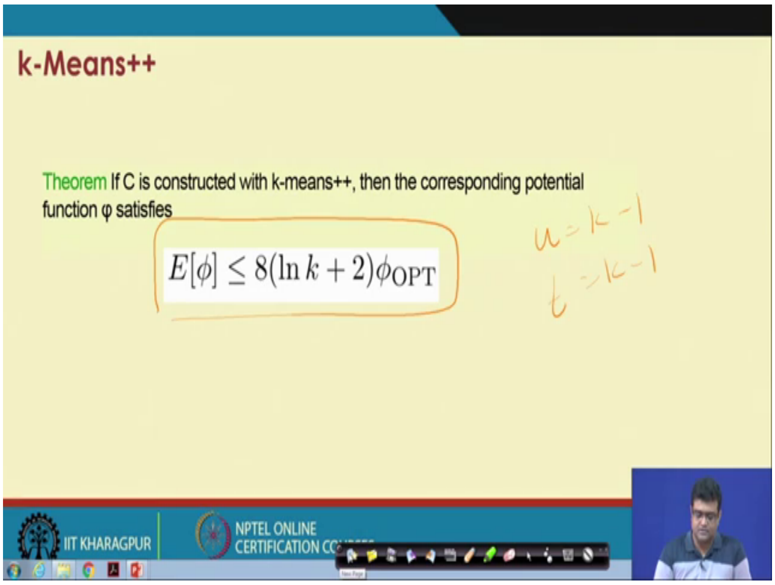
So, if you believe the following theorem to be true and the results this can be proved using the result that we have just shown, then you can get the optimal result. So, what does this theorem say? This theorem says that basically if you start with a cluster in C and then you choose you many you many uncovered clusters from the optimal clustering and let X u denote the set of points in this uncovered clusters and X c denote the set of points in the rest of the clusters ok.

And also now suppose you add t which is at most equal to u ok, that many points to the set of to the to the to the set of cluster centers ok. So, you add t many points to the setup cluster centers and then you get this new clustering C dashed ok. So, from the starting clustering C, you add t many clusters centers and get the new clustering C dashed, then and for this new clustering you get this objective phi dashed.

So, if this is the case, then the expected value of phi dashed is less than the cost that incurred on the cover clusters plus we have seen this already the 8 times the cost the 8 times the cost that you incurred on the uncovered clusters times this factor ok. This factor is nothing but you can think of is as a order log k factor actually order log k plus 2 factor.

So, this is order log k plus 2 or rather log t plus 2 and then, plus some term u minus t by u plus phi of u. So, the cost that you incur on the other clusters, but the k term here is u minus t, so, if you add exactly u many so if u is equal to t ok. So, t can be at most u, but if it is exactly equal to u which is the case in our proof ok, then this cost is just 0. So, this cost is 0, so, this term dominates the cost and this term is basically the cost that you incurred on the covered cluster plus 8 times the optimal cost that you incurred on the uncovered clusters and to go back to the result from here, all you have to do is you have to plug in.

(Refer Slide Time: 21:53)



k-Means++

Theorem If C is constructed with k-means++, then the corresponding potential function φ satisfies

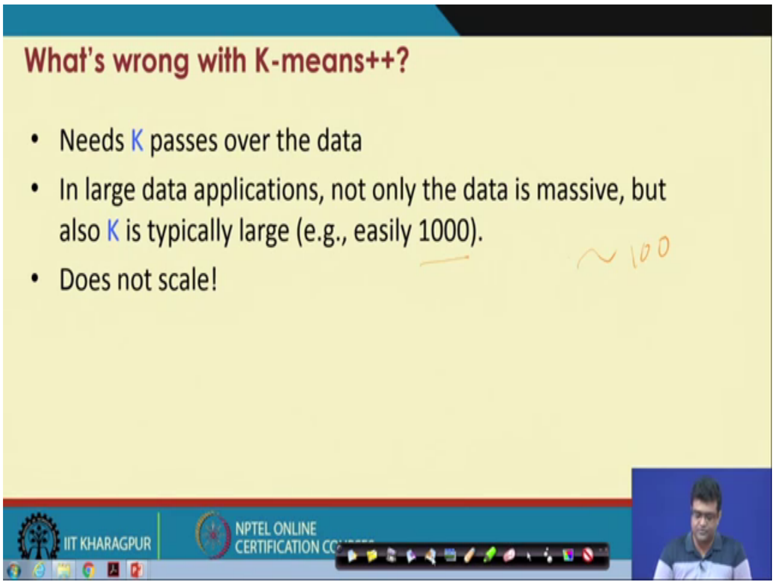
$$\boxed{E[\phi] \leq 8(\ln k + 2)\phi_{\text{OPT}}}$$

$u = k - 1$

$t = k - 1$

So, just to recall this result, all you have to do is you have to plug in u is equal to k minus 1 ok.So, the initial there is an initial covered cluster C which is the initial cluster and you have to add u is equal to k minus 1 and t is equal to k minus 1 and then you get this result from that result ok. So, hence we have shown and sort of given an integer as to y the new clustering is optimal.

(Refer Slide Time: 22:43)

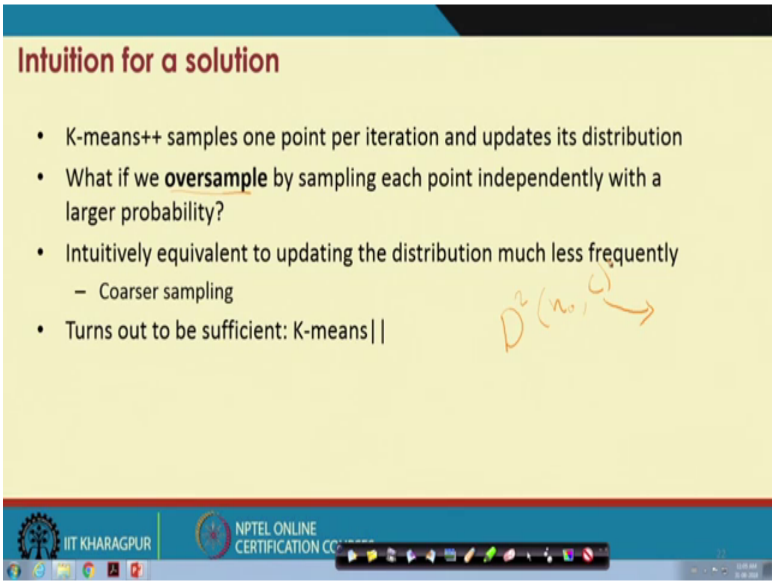


What's wrong with K-means++?

- Needs K passes over the data
- In large data applications, not only the data is massive, but also K is typically large (e.g., easily 1000). ~ 100
- Does not scale!

So, now this new clustering algorithm, so this K means clustering algorithm this has certain drawbacks ok. So, the first drawback is that it needs K passes over the data, this is because, in every pass you select a new cluster centre ok. So, in every pass you select a new cluster centre. So, you have to basically make K plus passes over the data and every time sample one point one cluster centre as your point, but, so, basically for many large applications, so let us say your K can be even 1000 or at least it can be of the order of 100. So, you may be looking for 100's of clusters so, that many a times does not scale. So, 100 passes over the data is not possible to make.

(Refer Slide Time: 23:54)



Intuition for a solution

- K-means++ samples one point per iteration and updates its distribution
- What if we **oversample** by sampling each point independently with a larger probability?
- Intuitively equivalent to updating the distribution much less frequently
  - Coarser sampling
- Turns out to be sufficient: K-means||

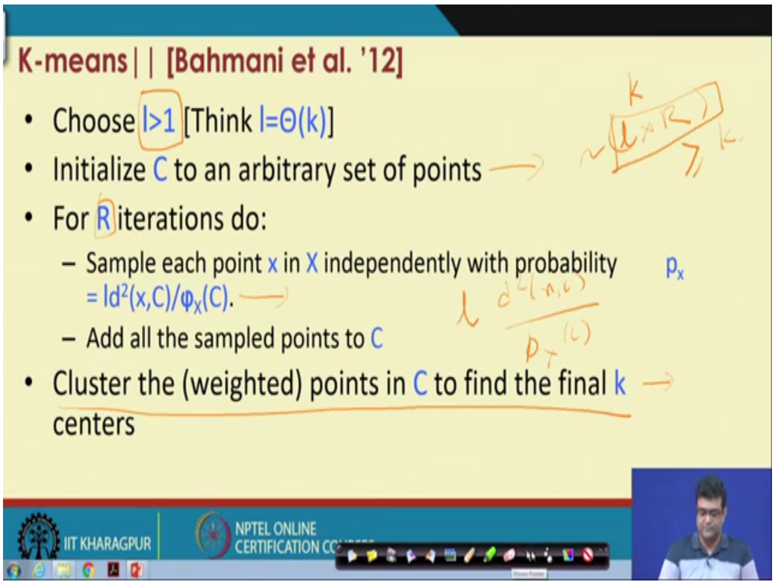
$D^2(x, C)$ →

Now, so,but what is the solution? So, the solution, so or rather the intuition for the solution is that note that K means plus plus is doing a sampling so, it is a randomize algorithm right.So, it is doing a sampling of the clusters, but why does it do not only that it is doing a sequential sampling. So, it is sampling the cluster centers one after the other ok. So, why does it do a sequential sampling or why does it sample one point after the other? The reason is that every time it is samples a point, the distribution from which it sample change it sample changes. The reason this distribution from which it samples changes is because, if you see the sampling probability depends on D square of x naught comma C ok. Now, this D square of x naught comma C it changes as your clustering changes.

So, the question is does it change a lot ok so, if it does not change a lot, can we do the following can we in over sample from the D square itself. So, instead of recalculating the clustering, let us say we sample let us say instead of one cluster center every time we sample 4 or 5 cluster centers every time. So, intuitively you are updating the distribution much less frequencely frequently or other. So, these are two equivalent we have thinking about it that you are selecting more sample more clusters centers per clusters in K means plus plus you are selecting one cluster centre per cluster, here your selecting more than one cluster centers per cluster, the equivalent way of thinking about it is that you are doing the sampling from a coarser distribution ok..

So, yeah, so without or rather without updating the distribution too much,so, it can be showns or it was shown in a recent paper called Scalable K means plus plus or K means parallel by Bahmani and et al, so that, so this particular algorithm actually achieves the same guarantees as the K means plus plus algorithm ok.

(Refer Slide Time: 26:49)



K-means| | [Bahmani et al. '12]

- Choose $l>1$ [Think $l=\Theta(k)$]
- Initialize C to an arbitrary set of points
- For R iterations do:
  - Sample each point x in X independently with probability $p_x = ld^2(x,C)/\varphi_X(C)$.
  - Add all the sampled points to C
- Cluster the (weighted) points in C to find the final k centers

k
$\sim (l \times R) \geq k$
$l \frac{d^2(x,c)}{\varphi_X(C)}$
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

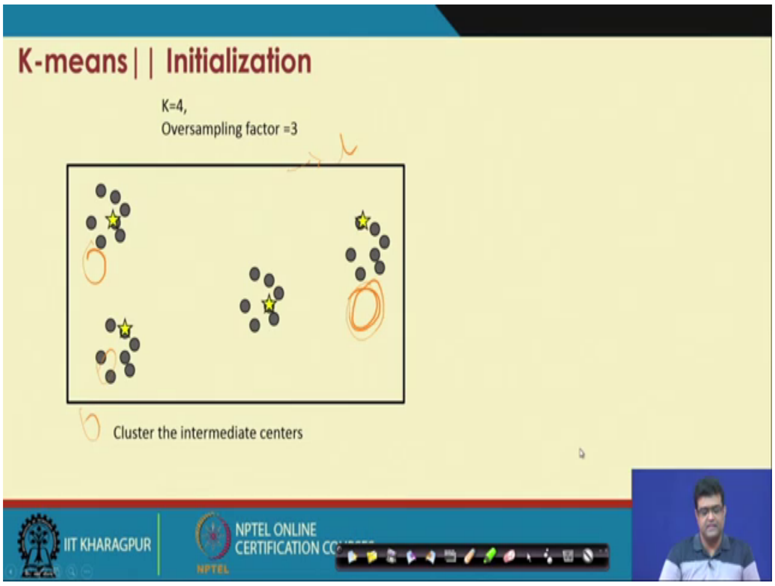
So, what is the algorithm ok? So, the algorithm is the following, .so, you choose a over sampling factor l you can think like l is theta k ok.So, every time you sample theta k points from the distribution instead of just one point ok. So, after this, you initialize C 2 an arbitrary set of cluster setup points and then for R iterations you run sampling from this distribution. So, instead of your you sample x independently with probability l d square by phi of x. So, earlier you were sampling with D square x c by phi of x c now you are sampling with l times this probability, so, you are basically over sampling by a factor of l .

Now, what will happen is at the end of it, you will get you are not guaranteed to so, you are not guaranteed to get k points. So, in k means plus plus, you are guaranteed to get k points a cluster centers because, you are selecting the first one randomly and then every iteration you are selecting one for k minus 1 iterations, but here you will at the end of all this you will get l times R points. So, if you have done R iterations, you will get roughly l times R points.

Now, you have to ensure that this l times R is greater than or equal to k and then once you have sampled is l times R points greater than equal to k, you just re cluster the weighted points in C that is l times R points using let us say something like a K means plus plus algorithm to find the final k clusters.

But this time, you are only going to have to make a pass over this l times R set of points rather than the full data set .So, this is much more efficient so, this is the algorithm ok.

(Refer Slide Time: 29:27)



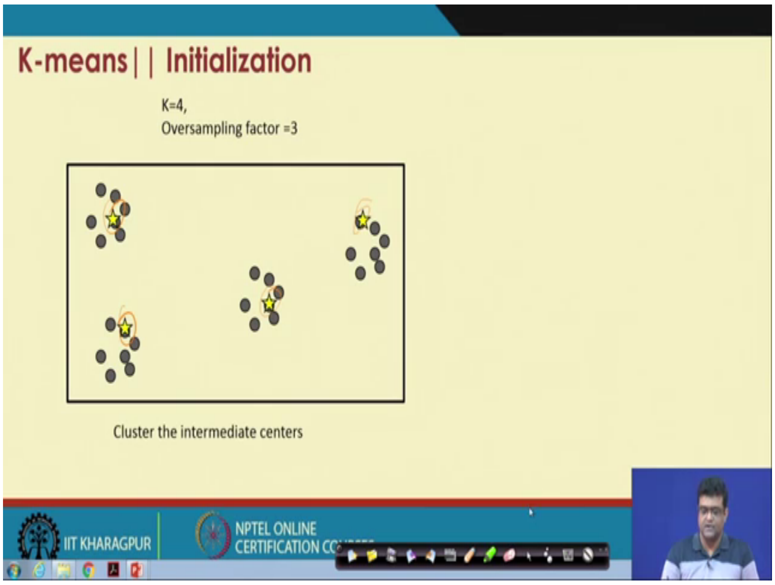
K-means | | Initialization

K=4,
Oversampling factor =3

Cluster the intermediate centers

So, just to describe what the algorithm is so, let us say this is your clustering. So, you first select, so you first select let us say this particular point ok, so, you select you select first this point and then let us say you have over sampling factor which was described as L as 3 and then you sample from D square distribution. So, D square is basically measuring the distance from this cluster centre and points which are far away are highly likely to get selected and you select three of them. So, let us say one you select here and two, you select here and you do not select a sample anyone here because all the points in this cluster are very close to this.

(Refer Slide Time: 30:47)



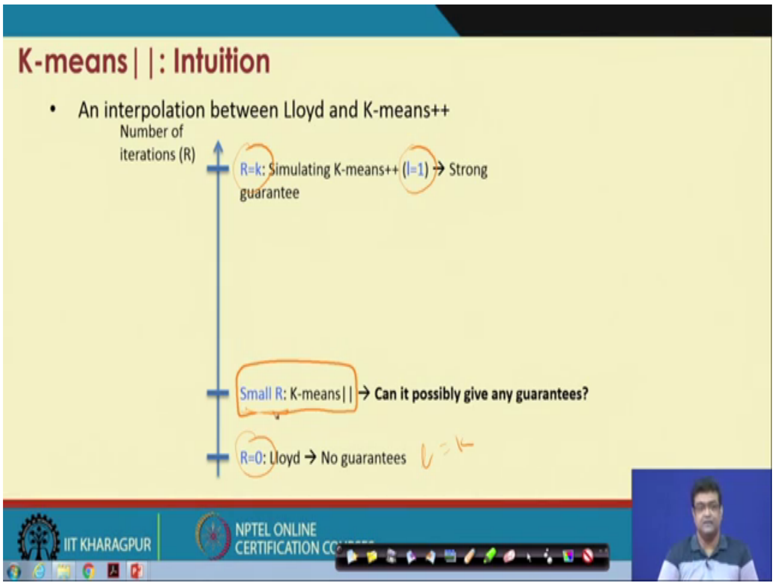
K-means| | Initialization

K=4,
Oversampling factor =3

Cluster the intermediate centers

So, next time, you again sample 3 ok. So, now, you get one here, one here and here and then you cluster the intermediate points ok. So, you cluster these intermediate points and then you get one point in this cluster, one point in this cluster and one point in every cluster like this. So, this the intuition behind the scalable K means plus plus algorithm .Now, what kind of so we have already seen this.

(Refer Slide Time: 31:09)



K-means| |: Intuition

- An interpolation between Lloyd and K-means++

Number of iterations (R)

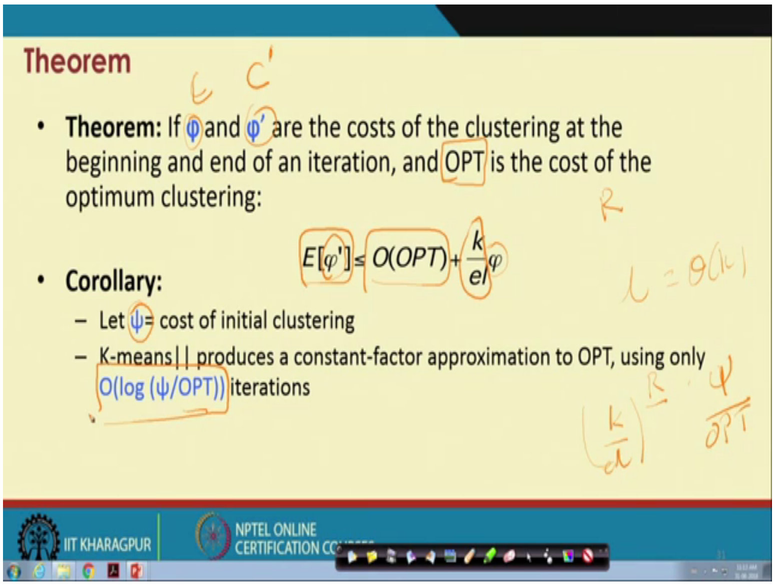
R=k: Simulating K-means++ (l=1) → Strong guarantee

Small R: K-means| | → Can it possibly give any guarantees?

R=0: Lloyd → No guarantees

So, this is one way of thinking about scalable K means plus plus. So, it is a generalization of K means plus plus algorithm,in the sense that if you set R is equal to k and l is equal to 1, you get the K means plus plus back .Now, normally what you want to do is you want to if you get R is equal to 0 and you select l is equal to k and you select l is equal to k, you get the original K means algorithm back because, you are selecting everything uniformly randomly. What you want is you want a small R ok, so, you want to select the small R and you want to select some l over sampling factor and then you want to give some guarantee ok.

(Refer Slide Time: 32:09)



Theorem

- **Theorem:** If φ and φ' are the costs of the clustering at the beginning and end of an iteration, and OPT is the cost of the optimum clustering:

$$E[\varphi'] \le O(OPT) + \frac{k}{el}\varphi$$

- **Corollary:**
  - Let ψ= cost of initial clustering
  - K-means| | produces a constant-factor approximation to OPT, using only O(log (ψ/OPT)) iterations

C C'
R
$l = \theta(k)$
$\left(\frac{k}{el}\right)^{R} \cdot \frac{\psi}{OPT}$
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

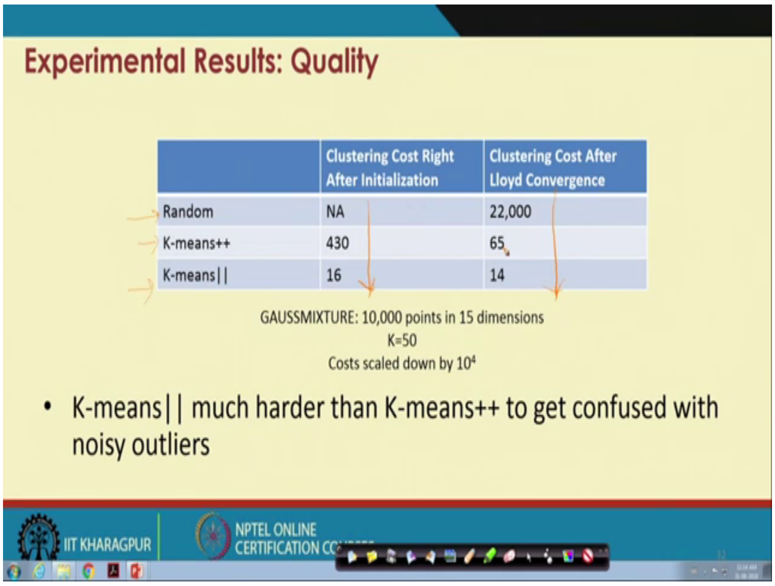
So, how do you want to do this ok? So, this is the main theorem that of the paper, so, what it says is that suppose you start with the let us say you start with a clustering C ok. So, you start with the clustering C and then after sampling new cluster centers, you go to a clustering C dashed and let their costs be phi and phi dashed.

Then and suppose you yeah, so then and suppose OPT is the cost of the optimal clustering, then of course, you are your phi dashed is going to be random ok. So, you take the expected value of phi dash, so, the expected value of phi dashed is less than is going to be less than or equal to the cost of the optimal clustering plus this k times l which is a constant. So, k times l if your l is theta k, then k times l is just a constant and E is a constant so, it is k times l times phi ok.

So, if you run this algorithm for R iterations ok, you reduce the cost by a factor of k times e to the power l to the power R ok.So, if let us say a psi was your initial cost of clustering and OPT is your final cost ok. So, in order to reduce the cost of a clustering initial clustering psi to OPT, so this is your ratio of reduction, so this is the factor by which you should reduce. So, every time the cost is getting reduced by this factor k times a k by e to the power l times R .So, you just have to run the iterations log O log of psi by OPT iterations ok.

So, that is, whatever is the ratio of the initial cost to the cost of the optimal solution ok, you have to run it log of that many times iterations ok. So, if you run it that many times you get a cost which is close to optimal which is order optimal cost. So, this the guarantee given by the a Scalable K means plus plus algorithm.

(Refer Slide Time: 35:23)



Experimental Results: Quality

	Clustering Cost Right After Initialization	Clustering Cost After Lloyd Convergence
Random	NA	22,000
K-means++	430	65
K-means||	16	14

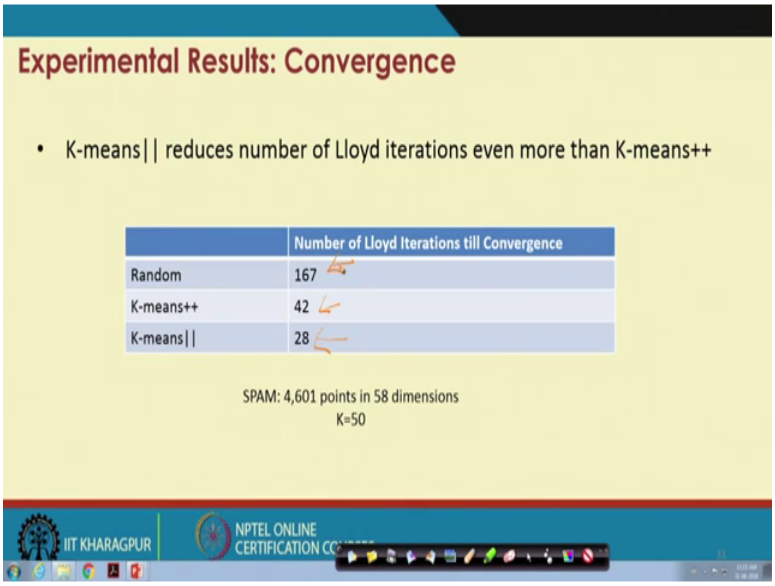
GAUSSMIXTURE: 10,000 points in 15 dimensions
K=50
Costs scaled down by 10^4

- K-means|| much harder than K-means++ to get confused with noisy outliers

So, some experimental results, so first they compare this random initialization which is the K means algorithm K means plus plus and K means parallel and basically what they show is that both if you measure cost just after initialization. So, this these are all initialization algorithms so, you can either try to measure the cost just after initialization or you can try to measure the cost you can run the Lloyds iterations after the initialization. And in both cases actually K means plus plus does much better than random and this K means parallel does slightly better than K means plus plus in terms of the cost itself.

So, the quality of the solution itself improves with this K means parallel and this is because basically you can think of it as also running many K means plus plus is in parallel because, you are doing a lot of sampling and then you are doing clustering for each of the samples.

(Refer Slide Time: 36:40)



Experimental Results: Convergence

- K-means|| reduces number of Lloyd iterations even more than K-means++

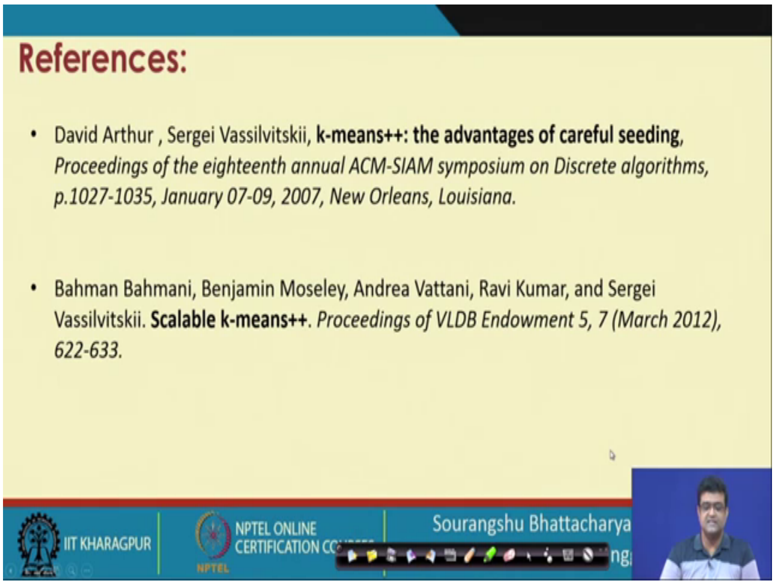
	Number of Lloyd Iterations till Convergence
Random	167
K-means++	42
K-means||	28

SPAM: 4,601 points in 58 dimensions
K=50

The second important result is that if you look at the number of iterations that number of Lloyds iteration that are taken till convergence. So, again the number of Lloyds iterations taken after the K means parallel initialization is the lowest followed by K means plus plus and of course, random initialization takes a lot of Lloyds iteration for convergence .

So, that concludes our x position on clustering and scalable clustering, so, it is easy to see that both K means plus plus and scalable rather both, so, both K means plus plus scalable K means plus plus are very accurate algorithms furthermore, K means plus plus is actually implementable in parallel and it requires much less number serial iterations ok.

(Refer Slide Time: 37:47)



References:

- David Arthur , Sergei Vassilvitskii, **k-means++: the advantages of careful seeding**, *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, p.1027-1035, January 07-09, 2007, New Orleans, Louisiana.*
- Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. **Scalable k-means++**. *Proceedings of VLDB Endowment 5, 7 (March 2012), 622-633.*

So, these are the references, so, the k means plus plus is was published by Arthur David Arthur and Sergei Vassilvitskii and scalable k means plus plus was by the authors Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar and Sergei Vassilvitskii and these are the reference.

Thank you.