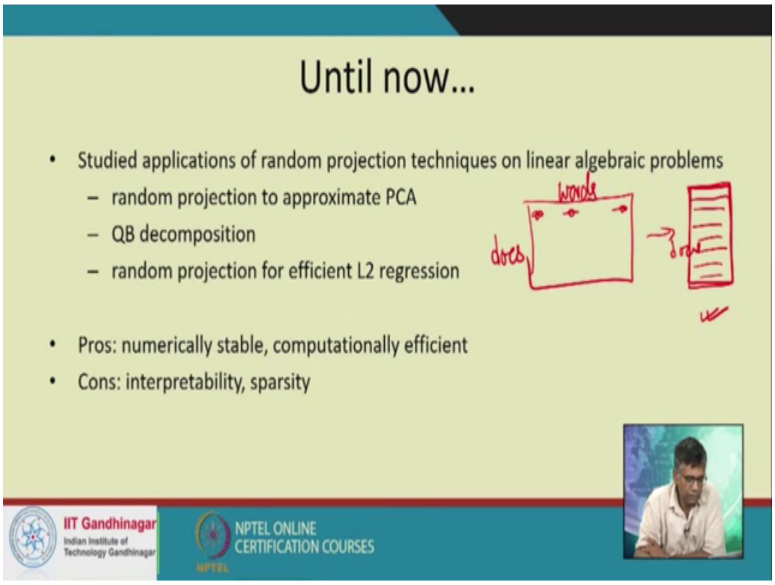
Scalable Data Science

Prof. Anirban Dasgupta
Department of Computer Science and Engineering
Indian Institute of Technology, Gandhinagar

Lecture - 16 b
Leverage Score & Applications

Hello and welcome to the course on Scalable Data Science. My name is Anirban I am from IIT, Gandhinagar and today we will study Leverage Score and its Applications.

(Refer Slide Time: 00:25)



Until now...

- Studied applications of random projection techniques on linear algebraic problems
  - random projection to approximate PCA
  - QB decomposition
  - random projection for efficient L2 regression
- Pros: numerically stable, computationally efficient
- Cons: interpretability, sparsity

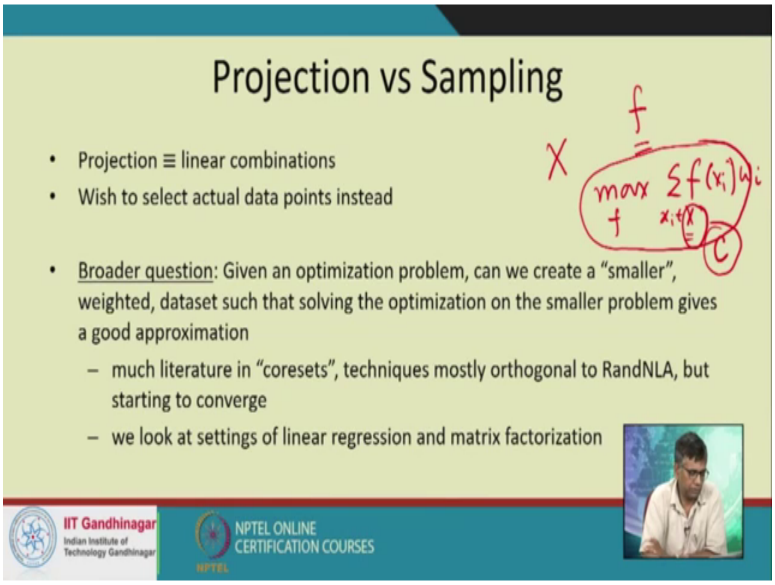
So, until now we have looked at different applications of the random projection techniques and various linear algebraic problems right. And some of these problems some examples of the things that we have studied have been number one application of random projection to approximate PCA to approximate matrix multiplication, we have looked at a particular decomposition low rank decomposition called a QB decomposition. We have also looked at how to apply random projection in order to do l two regression efficiently.

So, the pros of these of these techniques have been that that lot of them are numerically stable, and hence practical, they are also computationally efficient theoretically as well as practically, but the cons is that is that when you are computing random projection right the result might not always be very interpretable right. For instance imagine that we started with the matrix of documents versus words right and then we sort of took a projection of it, some random projection of it on the column space and we have some documents versus projected directions right. Now these projected directions are linear combinations of words right including both positive and negative weights, because the random number is going to have; I mean, the random variable is going to take both positive and negative values. And therefore, you are going to combine words the word counts by using coefficients that are both positive and negative.

And this does not have any direct interpretable meaning right because what does what does it really mean to take combinations of words; that is point number 1. Point number 2 is that I mean see each of the documents itself was fairly sparse right, but once we take linear combinations of the document I mean it is possible that each document had 10 words or an average. But once we take linear combinations because the linear combinations are dense, then a is essential at dense random projection, the matrix that we have at the end is fairly dense right. And this is the same issue that we had seen in the case of the of the dense random projection. And so, it is possible that the space taken by the by this by this dense projection although its low dimensional is actually much more than the space taken by the original sparse or document matrix.

Right and the and there is there is nothing special about what documents this could happen for any sparse data. That once you convert that is that once you use random projection right or once you or once we get a singular value decomposition of it, right the space taken even if we do a low rank decomposition of it the space taken could be much more than the space needed to represent original data, because a either the random projection or the low rank decomposition for instance the QB or the SVD. They are all dense I mean, they are essentially dense there is no sparsity constraint on them. So, let us keep these two issues in mind as we as we go through this lecture.

(Refer Slide Time: 03:42)



Projection vs Sampling

- Projection ≡ linear combinations
- Wish to select actual data points instead
- Broader question: Given an optimization problem, can we create a "smaller", weighted, dataset such that solving the optimization on the smaller problem gives a good approximation
  - much literature in "coresets", techniques mostly orthogonal to RandNLA, but starting to converge
  - we look at settings of linear regression and matrix factorization

X

$$\underline{\underline{f}} = \max_{f} \sum_{x_i \in \underline{\underline{X}}} f(x_i) w_i \quad C$$

IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

So, in order to handle these two issues, let us look at let us think a little bit about the relation between projection and sampling right. So, until now we have really been talking about projections of vectors right.

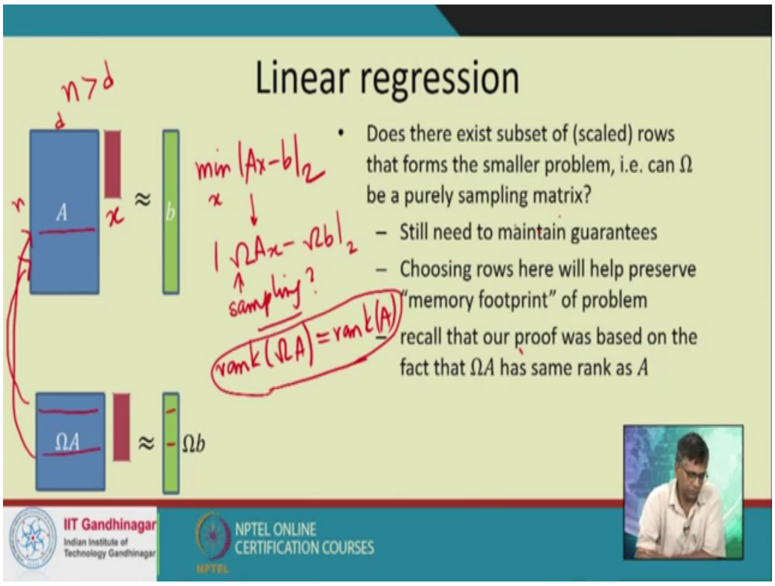
So, what is projection? Projection is essentially nothing, but a linear combination. So, so what if we wanted to choose actual data points instead of taking linear combinations of data points, can we do that right? And we again had faced this issue when we were trying to speed up random projections, and then we resort it to something like a sampling matrix and then we saw that we could not always do this right we could not always do this while retaining the properties of the random projection.

So, here again will welcome back to that question in a slightly different setting, but related setting. And this actually is part of a very broad I mean a much broader question in algorithms right and the broader question being is that given an optimization problem right. Can we create a smaller data set? And we want and we are solving the optimization problem over a data set x right that suppose we have we have a problem we have some function f, and we want to get let us say the max of summation f i right.

Where the max over f and the summation is over all xi belongs to x correct and my intent is to get the f that maximizes this quantity ok. So, the question is that instead of instead of taking the sum if the data set x is very large, instead of taking the sum over all points in the data set that can I find a much smaller data set let me call that c right; such that I take this sum over that smaller data set and maybe I sort of adding a weight to I sort of multiply each of these by a weight right because maybe I mean; for full generality let us just I mean put in a weight for each point and instead of taking this sum over the entire data set can I just do the sum compute the sum over the set c.

So, this is typically known as a core set and there is a lot of literature on this from the computational geometry perspective ok. The techniques are mostly orthogonal to the ones that we study have studied in this randomized numerically linear algebra techniques, but in a bunch of recent papers these directions are starting to converge. So, we would not look at the problem of corsets from the. So, we will look at the problem of corsets again from this rand and from this randomized numerical linear algebra setting and specifically we will look at two settings linear regression and matrix factorization ok. And we look at a one specific way of creating these corsets by using a sampling technique known as leverage core sampling ok.

(Refer Slide Time: 06:45)



Linear regression

- Does there exist subset of (scaled) rows that forms the smaller problem, i.e. can Ω be a purely sampling matrix?
  - Still need to maintain guarantees
  - Choosing rows here will help preserve "memory footprint" of problem
  - recall that our proof was based on the fact that ΩA has same rank as A

$n > d$
d
n
A
x
$\approx$
b
ΩA
$\approx$
Ωb
$\min_x |Ax-b|_2$
$|\Omega Ax - \Omega b|_2$
sampling?
$\text{rank}(\Omega A) = \text{rank}(A)$
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

So, let us sort of recall what linear regression the setting of linear regression that we had, that we have we are mostly talking about over constrained setting that is n is bigger than d. So, we have a set n your matrix A of that is of size n by d and were looking to find a vector x right; such that the error minimize for all possible x A x minus b 2 norm this is what we looking to minimize.

So, at this point we are not putting any constraint or any regularization of them on x right all that is all that has been done, but we are not covering that in this course and what well be doing in order to solve this problem more efficiently. We had our projection matrix omega right. And we were sort of we were shortening the matrix A converting it into a smaller matrix by multiply pre multiplying it with omega, and then we also find out a smaller version of b by pre multiplying it with omega and then we solve the smaller problem this is what we were doing right.

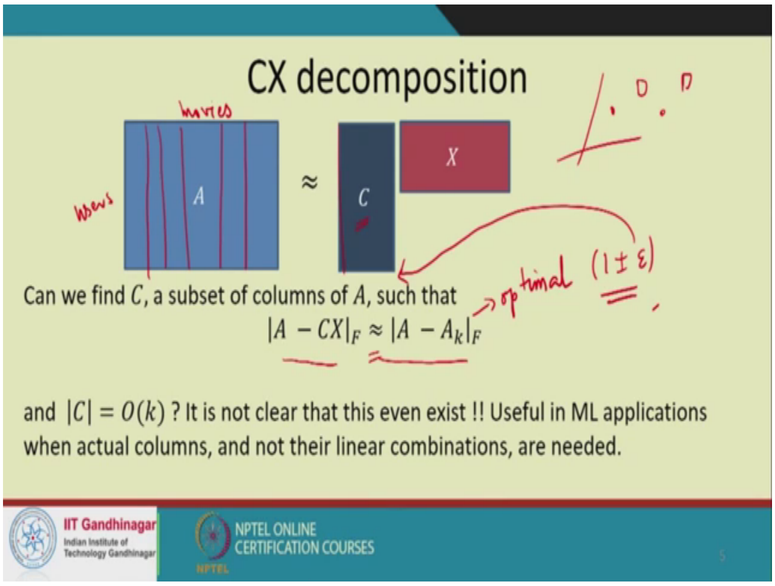
So, now, the question is that, that instead of projection can I replace omega by a sampling matrix right? So, what is replacing omega by A sampling matrix mean? What it means is that that the effect the omega A right should really be a matrix where each row in omega A right really comes from one of the rows in A and this should happen for all the rows. Each when omega I should really come from one of the rows from A right and typically we we tend to do sampling with replacement, because that is much easier to analyse ah, but there is nothing sacrosanct about it. And some and that is the same and that the same for each element in b ok.

So and why would this be helpful? Ok one of the reasons is; obviously, clear that if each row in A is really just a scaled version. So, this potential is scaling if each row in A is really just a scale if its omega A is really just a scale version of the of one of the rows in A, then the size taken to store omega A cannot be much more than the size taken to store a right.

So, the question is that does such omega really exist ok? Because remember what what the property that we what was the property that we needed out of omega? We needed the fact that the rank of omega A right; we need we needed the fact that the rank of omega A. It is really equal to the rank of A right. In fact, we need it a little bit stronger than this what we needed was that with mega preserves the omega preserves the singular value structure of U of A right which is the left singular vector of which are the left singular vectors of a ok, but let us even forget that let us even look at the smaller constraint that we want rank of omega A to equal the rank of A and then it is not entirely clear right that we can actually find such an omega.

So, of course, here is an easy solution right that if we were allowed to look at A and if we were allowed to sort of find out d linearly independent rows of A right, then we might be able to do this well easily. But doing that is really again to just solving this problem this regression problem by itself right. So, the question is that can I quickly find out an omega in a manner that is computationally much less efficient, that is computationally much more efficient than actually solving the regression problem and can I still guarantee this ok.

(Refer Slide Time: 10:41)



CX decomposition

Can we find C, a subset of columns of A, such that

$$\underline{|A - CX|_F} \approx \underline{|A - A_k|_F} \rightarrow \text{optimal} \quad (1 \pm \varepsilon)$$

and $|C| = O(k)$? It is not clear that this even exist !! Useful in ML applications when actual columns, and not their linear combinations, are needed.

So, the next question and one that we just mentioned before is that of finding a CX decomposition. So, what is the CX decomposition? Here we are saying that suppose we have users versus movies and what I really want is a low rank decomposition of this matrix?

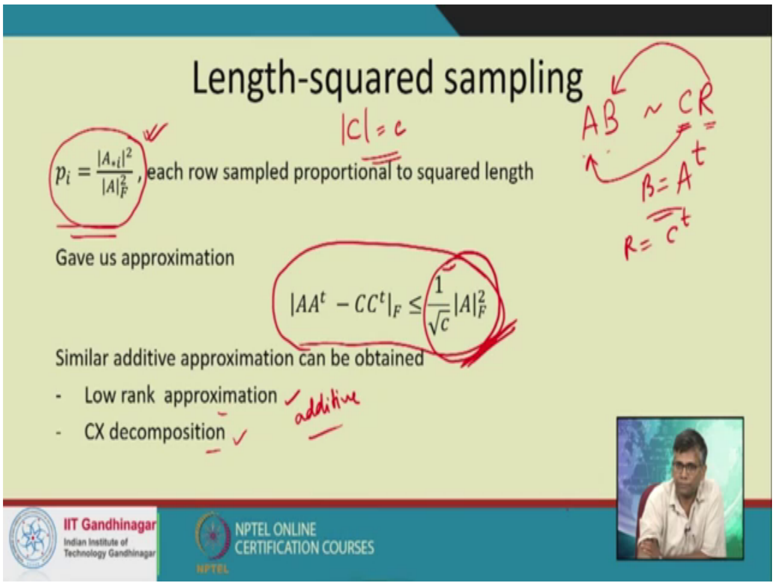
Because, we believe that a low rank decomposition will expose the different genres or the topics that the users. And the move and the movies that the users are interested in and that movies belong to right. So, the low rank decomposition is going to put users and movies in the same in the same plane right and then and then we can use this representation this joint representation in order to do recommendation and so, on and so, forth right.

So, but now here interestingly what we want is that, we want this low rank decomposition in a manner that the I mean instead of instead of a general QB decomposition, we want it to be a form such that we first get to choose a set of columns C that are C I mean each of the columns needs to belong to one of the each of the columns of C has to come from one of the columns of A modulo some scaling right. So, C is really a subset of the columns of A and I still want to guarantee right that I can find such an C and such an X and the corresponding x.

Such that the error A minus CX frobenius is within a small bound a small factor bound of A minus A k frobenius which is the optimal error ok. And this approximation really I mean that I mean theoretically it should be within some small 1 plus minus epsilon and of course, epsilon is going to control the size of the size of c.

So, does such a C really exist even that is not entirely clear. However, if we could do this then we have convinced ourselves that this will be useful in a lot of machine learning applications right. Because now the columns of C are really interpretable in addition to not being in addition to having a smaller foot that of a the columns of C are also interpretable because they are actually in this case for instance they are actually examples of movies themselves ok.

(Refer Slide Time: 12:57)



Length-squared sampling

$|C| = c$

$AB \sim CR$

$B = A^t$

$R = C^t$

$p_i = \frac{|A_{*i}|^2}{|A|_F^2}$, each row sampled proportional to squared length

Gave us approximation

$$|AA^t - CC^t|_F \leq \frac{1}{\sqrt{c}}|A|_F^2$$

Similar additive approximation can be obtained

- Low rank approximation ✓
- CX decomposition ✓

additive

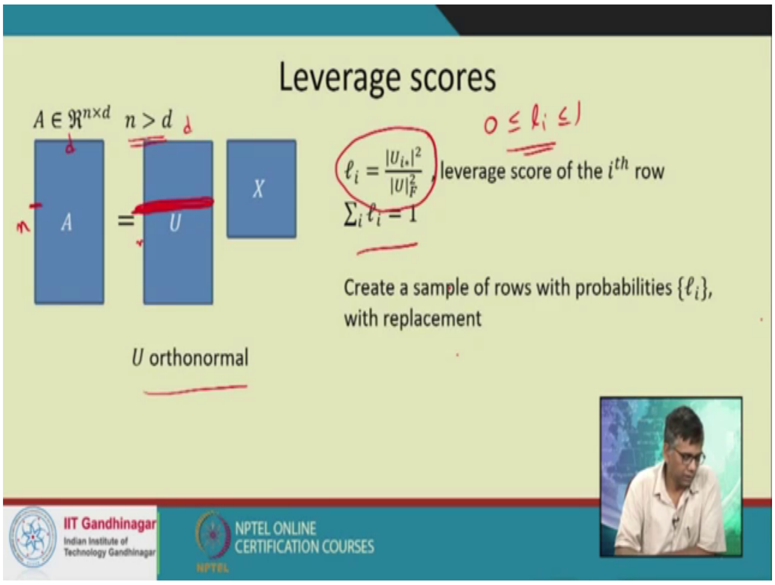
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

So, here is one possible way right. So, we have remember we have looked at the problem of matrix multiplication, that we have that we are trying to I I mean in order to multiply a times b two matrixes A times B, we created two samples a sample A from sample C from A and A sample R from B using some sort of a length square sampling right. In particular one of the one of the special cases that we looked at is as follows when B equal to A transpose right. In the case that B equal to A transpose, then the length square sampling that people interested in looked turned out to be the following right; that I choose C that I create the matrix C. So, the matrix R in a uv equal to n transpose then the matrix R is a is again equal to c transpose right. So, I need to create only the matrix C and one of the ways we discussed in order to create a matrix C was to choose the ith column of Awith probability that is proportional to the length square of the ith column right.

So, basically every column of c right is a IID random variable right and that IID random variable takes the value of the of the ith column of a with probability proportional with probability pi. So, it is a and it takes a for instance it takes value A 1 it takes value A 1 with probability A star 1 with probability b 1 it takes value a star two with probability p 2 and so, on and so, forth ok. And we are what we discussed was that you know I mean that once we choose this values of pi we get this particular approximation right. That if the happen to choose C columns right im overloading the the notation C im using it both as a set and as and as a matrix. But if I if I happen to choose C columns in small c columns in C, then I get the guarantee that A transpose minus CC transverse frobenius is not very big.

In the sense that it is frobenius sum of a square by square root c and this particular sampling itself and this particular decomposition itself can then be used to obtain low rank approximation and CX decompositions with additive error with additive error. And the additive error that we get is really the one that comes out here and in each of this cases basically the idea is that you choose a set of columns C using this particular sampling technique right and then you do a low rank decomposition of that set of columns right or you get a CX decomposition using that set of columns. And the bound on the error that you get in the data t bounded error that you get in these two cases is comes really from the bound that we have calculated here and we are not going to go into the details of this ok.

(Refer Slide Time: 12:53)



Leverage scores

$A \in \Re^{n \times d}$ $n > d$

$\ell_i = \frac{|U_{i*}|^2}{|U|_F^2}$, leverage score of the i^{th} row

$0 \leq \ell_i \leq 1$

$\sum_i \ell_i = 1$

Create a sample of rows with probabilities $\{\ell_i\}$, with replacement

U orthonormal

A
=
U
X
n
d
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

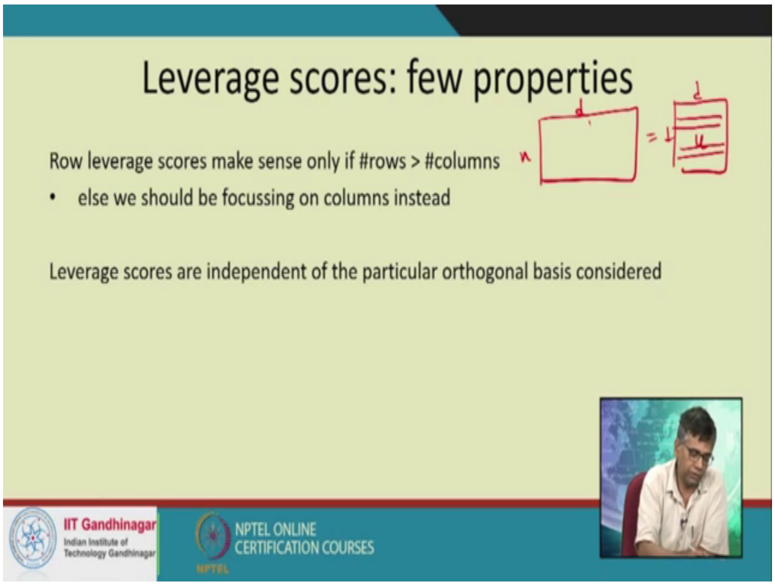
Because what we are interested in is today is known as the leverage scores. So, let me define the leverage scores first ok.

So, let us take a matrix a let us consider the setting when n is bigger than d right that the number of rows is bigger than the number of columns, and let us take an orthonormal decomposition of the set set of columns right. So, let u be an orthogonal basis of the set of columns and let me write A as u times x right. So, use of size n by d without loss of generality assume that A has rank d.

So, in that case I defined the I defined the ith leverage score the leverage score after of the ith row right, to be the rho norm the norm of the ith row of U square right. So, I take the so, if I take ith row of U and I take the l two square norm of that right and in order to make it a probability I divide it by the sum of these norms right, which is essentially the frobenius norm of U ok. So, this is a is a quantity between 0 and 1 and the sum of these equals o1 exactly ok.

And and this is known as the leverage score of the ith row or we will call it the row average scores. So, so one of the things that we will do is that what we will typically once we have defined this leverage scores, we will typically create a sample of rows using the leverage scores li with replacement ok. So, just like we were using the probabilities pi in the previous in the previous slide, we I mean here instead of the probabilities pi will use this leverage score probabilities li right and will create and we will do sampling with replacement with I mean by taking these probabilities ok.

(Refer Slide Time: 17:45)



Leverage scores: few properties

Row leverage scores make sense only if #rows > #columns

- else we should be focussing on columns instead

Leverage scores are independent of the particular orthogonal basis considered

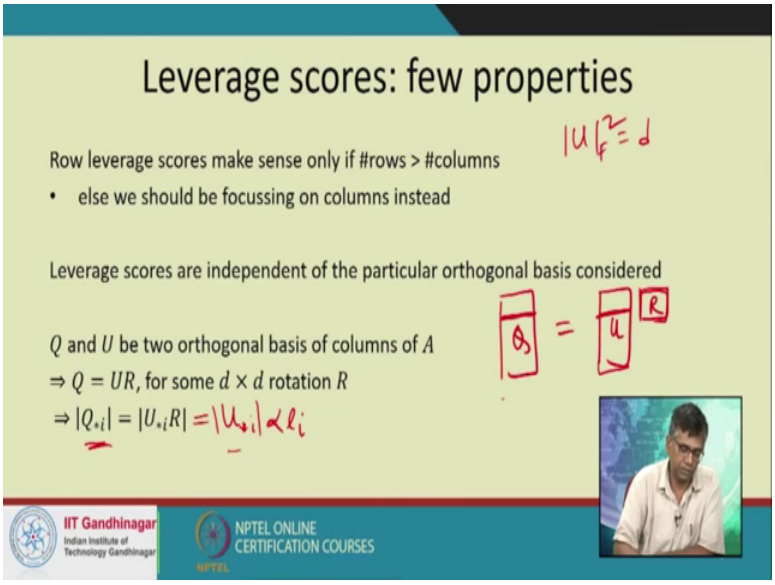
d
n
=
d
d
U
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

So, before we show its use let us look at a few of its properties see. So, one of the things should be fairly obvious the row leverage scores only makes sense that the number of rows is equal is greater than the number of columns.

Because else the form of u looks as follows U looks like I mean U looks like because if number of a if then, if a looks like this number of if the number of columns is bigger than the number of rows right then U is a square matrix right U is a d by d orthonormal matrix and in that case all the row leverage scores all the row norms of U are all 1 right. So, in that case it does not make sense to talk about it and it sort of is intuitive because if the number of columns is bigger, then you should not really be talking about sampling rows you should really be talking about sampling columns ok.

So, and also an important quantity and an important thing to realize is that, while we define the leverage scores in terms of in terms of a specific ortho[normal]- I mean an arbitrary orthonormal basis the actual definition is independent of the orthonormal basis that you choose right.

(Refer Slide Time: 18:51)



Leverage scores: few properties

Row leverage scores make sense only if #rows > #columns

- else we should be focussing on columns instead

$|U|_F^2 = d$

Leverage scores are independent of the particular orthogonal basis considered

Q and *U* be two orthogonal basis of columns of *A*

$\Rightarrow Q = UR$, for some $d \times d$ rotation R

$\Rightarrow |Q_{*i}| = |U_{*i}R| = |U_{*i}| \propto \ell_i$

$\boxed{Q} = \boxed{U}\,\boxed{R}$

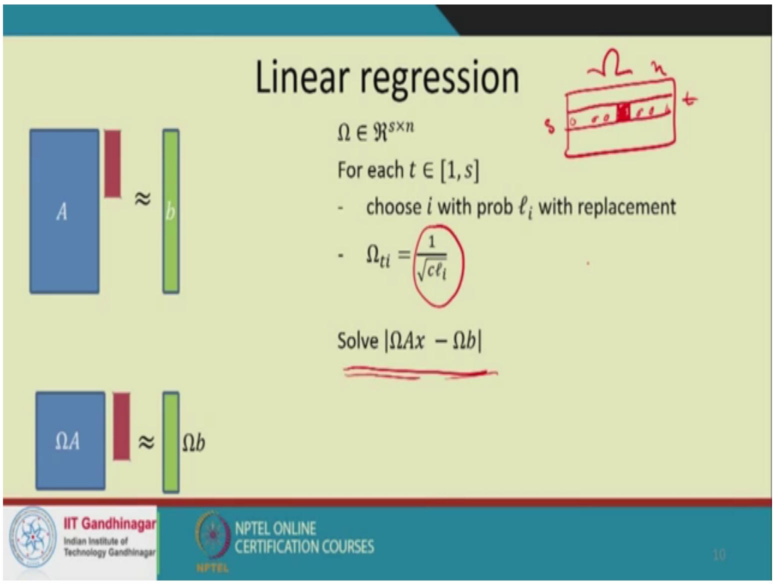
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

The in order to say this let us just look at this calculation, that suppose Q and u are two different orthonormal basis of the of the columns of the columns of A ok.

And because there are two different or thermal bases of the columns of A, then there exists some rotation matrix R right which is I mean rotation which means an orthonormal matrix d by the orthonormal matrix R such that Q equal to U times R right what; that means, is that, the ith row of Q is really equal to the I at row of u times R.

So, the norm of the of the two norm of the ith row Q equals the two norm of the ith row of U right and this is true for any two basis any two orthonormal basis Q and u and therefore, and because this is what we define as the leverage score as the and this divided by the normalization, and remember that the normalization the frobenius norm of u is always d the frobenius norm squared always d and that is independent of the basis that you choose. Because both th enumerator and the denominator are independent of the basis that you choose an leverage score itself is independent of the basis.

(Refer Slide Time: 20:00).


A
≈
b
ΩA
≈
Ωb
Ω
n
s
t
IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES
10

Linear regression

$\Omega \in \Re^{s \times n}$

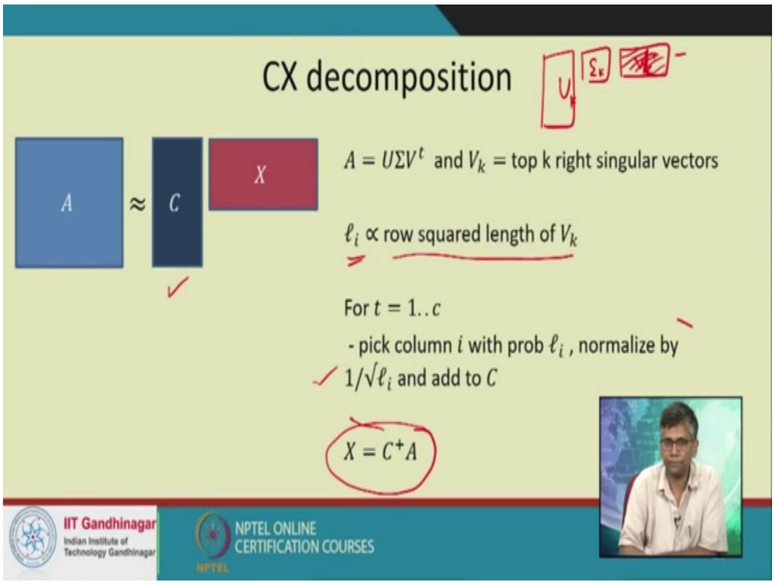
For each $t \in [1, s]$

- choose i with prob ℓ_i with replacement
- $\Omega_{ti} = \frac{1}{\sqrt{c\ell_i}}$

Solve $|\Omega Ax - \Omega b|$

So,. So, here is I mean, once you define the leverage score here is going to be my algorithm for linear regression. What we will do is that we will create the matrix will create the matrix omega as follows. So, omega is going to be s by n matrix right and in order to create the in order to create that the tth row we will choose we will choose I with probability li with replacement. So, therefore,. So,. So, and and suppose for the tth row, we happen to choose the we happen to choose the I specific I right. So, therefore, for tth row of omega will place zero everywhere else,except for the except for the tth position and in that position we will place we will place one over c square root of li ok. Basically we are normalizing by the by the o[ne]- by the square root of the probability that we chose it, with and then we solve this.. And once we have defined this particular omega we use it as we were using it before basically we solve omega Ax minus omega b and then we return the result as my approximation.

(Refer Slide Time: 21:13)



CX decomposition

$A \approx C\,X$

$A = U\Sigma V^t$ and V_k = top k right singular vectors

$\ell_i \propto$ row squared length of V_k

For $t = 1..c$

- pick column i with prob ℓ_i, normalize by $1/\sqrt{\ell_i}$ and add to C

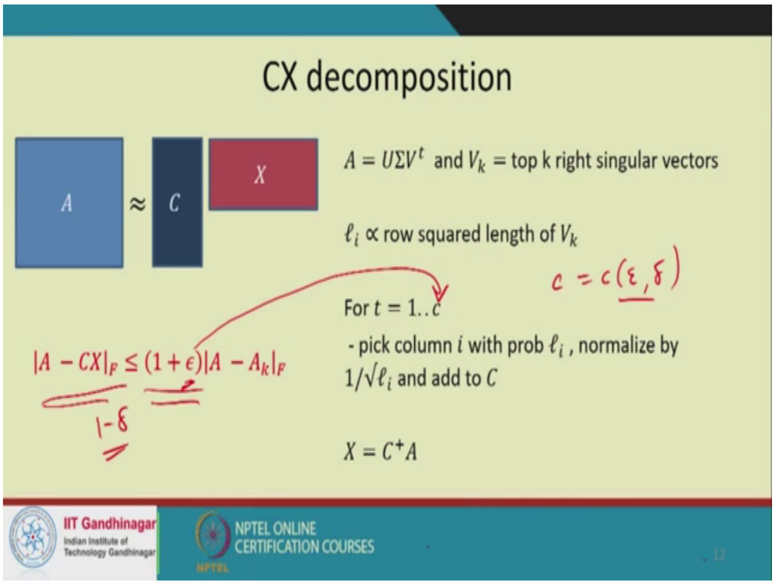
$X = C^+A$

IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

In order to do this CX decomposition here is what we will do right? We will first we will first decompose A as U sigma V transpose U sigma V transpose right and then we will take the top rank a U and V. So, so it will be a thin SVD and now what well do? Well define a lis because we because now we want to sample columns of a we will look at V transpose instead of instead of U ok. And we will and we will take the rows squared length of V k or the of the column square length of the of the Vk transpose and that and that will give me the corresponding leverage scores right.

And we will do exactly the same; that will pick column I with probability in order to define C we will pick column I with probability li normalize it by by 1 over square root of li and add it to the to the tth column of as the th column of c and this will keep on doing with replacement which means that the same column of a can be pick multiple times and we put in C and then we will define X to be the pseudo inverse of C times A that is it ok.

(Refer Slide Time: 22:27)



CX decomposition

$A = U\Sigma V^t$ and V_k = top k right singular vectors

$\ell_i \propto$ row squared length of V_k

$c = c(\underline{\varepsilon}, \delta)$

For $t = 1..c$

- pick column i with prob ℓ_i, normalize by $1/\sqrt{\ell_i}$ and add to C

$X = C^{+}A$

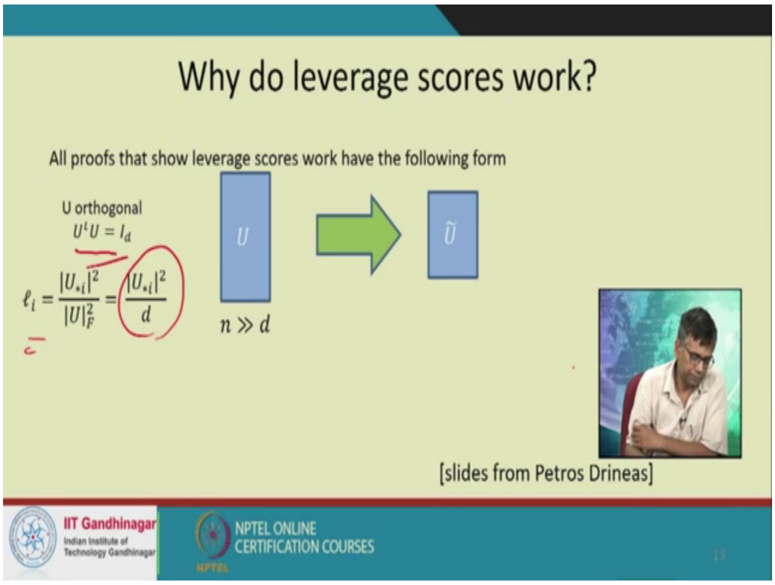
$$\underbrace{|A - CX|_F}_{1-\delta} \le \underline{(1+\epsilon)}|A - A_k|_F$$

IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

And what and what will be able to show is that both in the I mean both in the linear regression case that, we saw before and in the in the in the CX decomposition case the that were seeing now is that well get 1 plus epsilon approximation right. So, of course, this I mean the epsilon that you get you have to be in order to show such a result the epsilon the that you get has to figure in the number of samples that you choose right.

So, so only when the number of samples is right I mean when number of samples depends on on some on this on this epsilon and delta, you will be able to show that with probability 1 minus delta you get a you get a guarantee that a minus x is is not more than 1 plus epsilon of times the optimal error, and will be able to get a similar guarantee for linear regression.

(Refer Slide Time: 23:17)



Why do leverage scores work?

All proofs that show leverage scores work have the following form

U orthogonal

$U^tU = I_d$

$$\ell_i = \frac{|U_{*i}|^2}{|U|_F^2} = \frac{|U_{*i}|^2}{d}$$

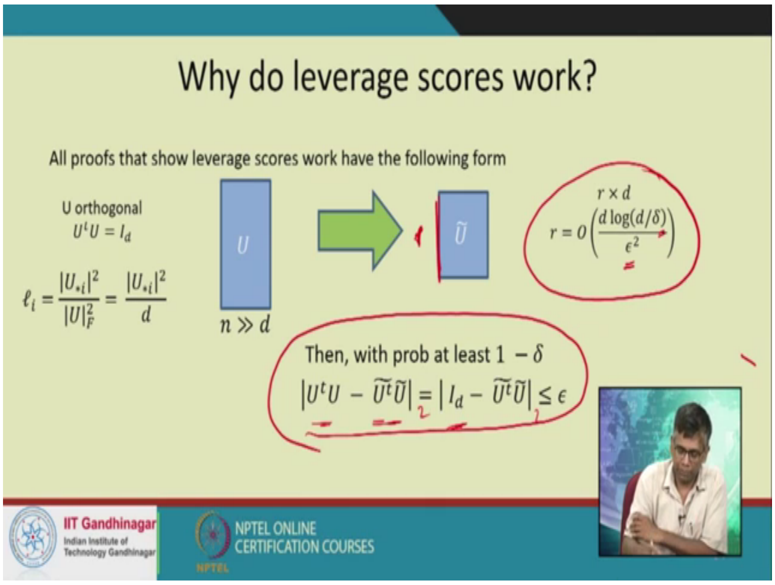
[slides from Petros Drineas]

IIT Gandhinagar Indian Institute of Technology Gandhinagar

NPTEL ONLINE CERTIFICATION COURSES

So, just to give some intuition as to why leverage scores work; so, all proofs that show leverage scores have basically of the following form right then consider U which is which is an orthonormal basis of a right and therefore, and because of that U transpose U is actually the d by d identity matrix. It is not very hard to see and these are the leverage scores right and these are the leverage scores that come out. So, using this leverage scores wego from U to U tilde right. So, so using the leverage the sampling using the leverage scores like I described in the previous two slides as well as using normalization we go from U to U tilde right.

(Refer Slide Time: 23:57)



Why do leverage scores work?

All proofs that show leverage scores work have the following form

U orthogonal

$U^tU = I_d$

$$\ell_i = \frac{|U_{*i}|^2}{|U|_F^2} = \frac{|U_{*i}|^2}{d}$$

$$r = O\left(\frac{d \log(d/\delta)}{\epsilon^2}\right)$$

Then, with prob at least $1 - \delta$

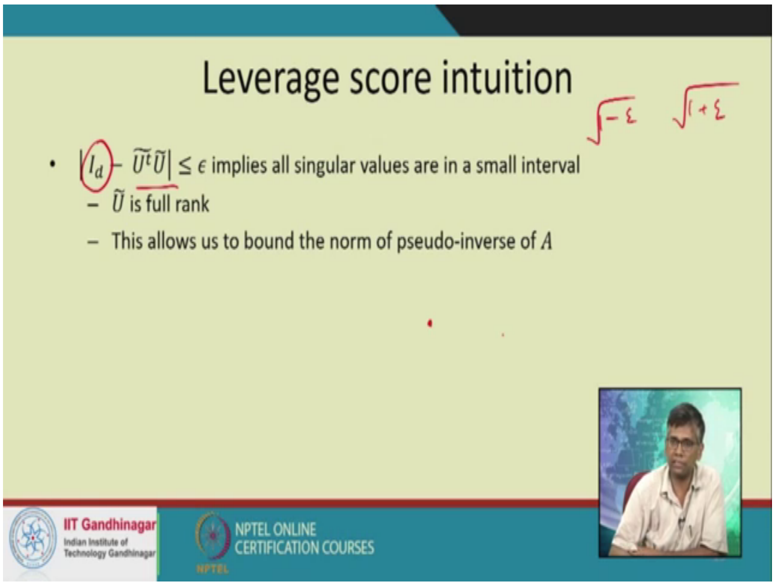
$$|U^tU - \tilde{U}^t\tilde{U}|_2 = |I_d - \tilde{U}^t\tilde{U}|_2 \leq \epsilon$$

IIT Gandhinagar Indian Institute of Technology Gandhinagar

NPTEL ONLINE CERTIFICATION COURSES

And suppose we choose suppose the number of samples that we choose that is the that is the number of rows of U tilde is some r where r is at least d log d by delta by epsilon square. So, this epsilon is the error is my error parameter again and delta is my confidence. Suppose we have chosen so, many rows so, many samples from u right. In that case well be able to show that with probability 1 minus delta right the u transpose u remember u transpose u is really the identity matrix t. So, u transpose U minus the U tilde transpose U tilde the two norm the spectral norm of this matrix is less than epsilon ok.

(Refer Slide Time: 24:39)



Leverage score intuition

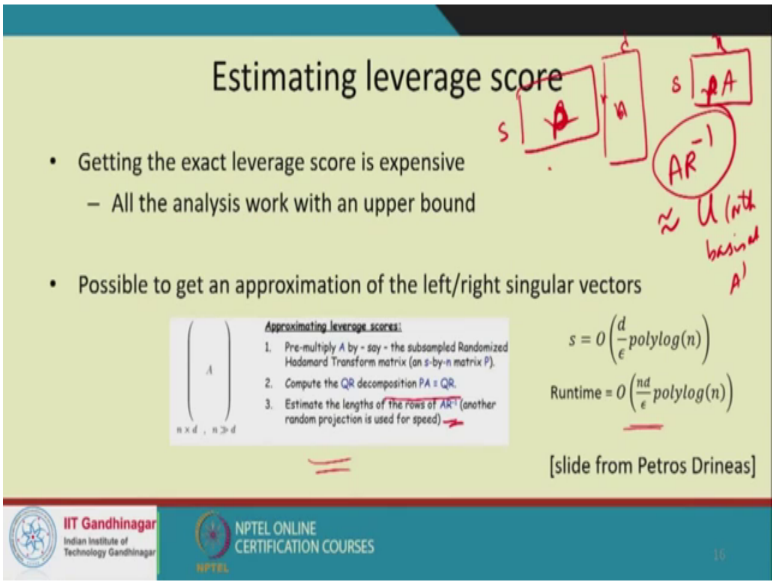
$\sqrt{1-\epsilon}$ $\sqrt{1+\epsilon}$

- $|I_d - \tilde{U}^t\tilde{U}| \leq \epsilon$ implies all singular values are in a small interval
  - $\tilde{U}$ is full rank
  - This allows us to bound the norm of pseudo-inverse of A

So, let me interpret what; that means,. So, what; that means, first of all is that see the I mean the singular values of I are all 1 right and therefore, what this means is that the singular values of U tilde transpose U tilde lie between. In fact, the singular values of U tilde transpose U tilde lie between one 1 minus epsilon and 1 plus epsilon, which means the singular values of u lie between square root of 1 minus epsilon.

And square root of 1 plus epsilon is specifically what it means is that U tilde is full rank because all the singular values line this line in this interval right. And what this will allow us to do later is in the proofs instead it allow it allows us to bound the norm]the pseudo in the norm of the pseudo inverse of a right and this and this pushes helps us push the proof through ok.

(Refer Slide Time: 25:23)



Estimating leverage score

- Getting the exact leverage score is expensive
  - All the analysis work with an upper bound
- Possible to get an approximation of the left/right singular vectors

Approximating leverage scores:

1. Pre-multiply A by - say - the subsampled Randomized Hadamard Transform matrix (an s-by-n matrix P).
2. Compute the QR decomposition PA = QR.
3. Estimate the lengths of the rows of AR^{-1} (another random projection is used for speed)

$$s = O\left(\frac{d}{\epsilon}polylog(n)\right)$$

$$\text{Runtime} = O\left(\frac{nd}{\epsilon}polylog(n)\right)$$

[slide from Petros Drineas]

.

So,. So, how do we estimate leverage scores right? Because if you were sort of paying attention, in order to I mean sample using the leverage scores we I in every case we needed to get we needed actually needed to get the singular value decomposition right and as we saw that singular value decomposition is fairly is fairly expensive.

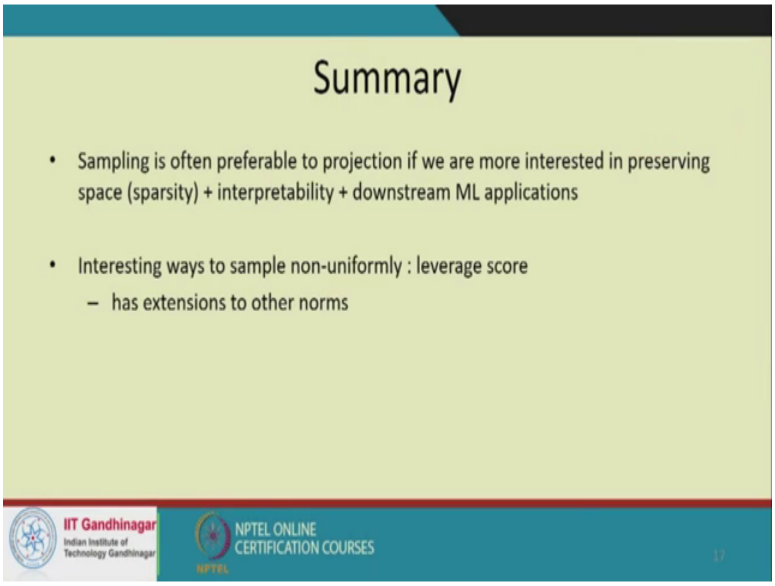
In fact, it is at least as expensive as, as the linear regression is doing solving the linear regression and also I mean getting a CX approximation. So, the CX approximation is not so, obvious because it is not clear that such a without the without the proof without the without our proof using the similar using the without our proof using the leverage scores.

Leverage scores it was not even clear that the CX decomposition existed right with the small case. So, therefore, for the CX decomposition we can still say that the leverage score has its use, but for the linear regression it is not entirely clear; however, luckily it is easy to estimate leverage scores although, I mean while we cannot get an exact value of it we can estimate it to some bound. And it turns out that the entire analysis can work if we can approximate it to some particular bound and. In fact, here is a here is an algorithm for approximating it and. So, first it is going to take time I mean which is proportional to n nb by epsilon times log factors instead of the instead of the nd square and its going to use the randomized Hadamard transformations that we have seen right basically has the idea that we have a right. So, so first we transform a using using the using randomizedha Hadamard transformation matrix right and then we get a much smaller a much smaller a right ok.

So, we get a which is s by; so, a is n by d and using using a PHD matrix omega A we get omega which is s by n. So, this is omega A right and then we do a QR decomposition of this tI am calling it p how right till and therefore, I will rename it to say that this is p ok. So, so PA is of size s by n and then, we do a QR decomposition of PA which is actually cheap now because PA is of size only s by n right and then we get see the point is that the r that. We get out here is a very good approximation of the r in that we could have gotten if we had done the QR decomposition of the original a.

So, therefore, once we look at the matrix AR inverse right that is a very good approximation to that is a very good approximation to an orthogonal basis of A right. So, AR inverse is a very good approximation to the u which is a orthogonal basis of A right. And therefore, by using the matrix AR inverse in fact, we can do another I mean in fact, because we only need the length of the rows of AR inverse we can we can do another trick with the random projection just for speed, but basically using the matrix AR inverse we can get a very good approximation of that leverage scores.

(Refer Slide Time: 28:21)



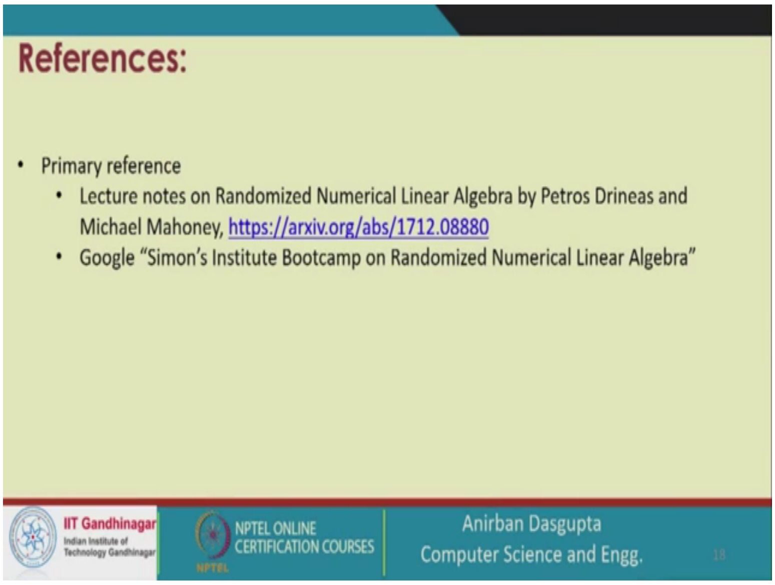
Summary

- Sampling is often preferable to projection if we are more interested in preserving space (sparsity) + interpretability + downstream ML applications
- Interesting ways to sample non-uniformly : leverage score
  - has extensions to other norms

IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES

So, just to summarize that in this in this technique,I mean in this lecture; we discussed the role of sampling versus projection and we discussed that sampling is more preferable to projection if you are interested in preserving things like the sparsity of the matrix, the total memory footprint the interpretability as well as its often useful in downstream machine learning applications. We did see at least one interesting I mean basically two interesting ways one is the length square sampling and the other is the leverage score sampling, in which we are doing length square sampling with respect to the an orthogonal basis right. And we saw that at this particular leverage scored has very interesting applications in solving linear regression in solving linear regression as well as getting a CX decomposition approximately. There is an extension of leverage scores to other norms ah; however, the there are while that theoretical results the I mean the extension of leverage scores to other norms is not really very practical yet right.

(Refer Slide Time: 29:17)



References:

- Primary reference
  - Lecture notes on Randomized Numerical Linear Algebra by Petros Drineas and Michael Mahoney, https://arxiv.org/abs/1712.08880
  - Google "Simon's Institute Bootcamp on Randomized Numerical Linear Algebra"

IIT Gandhinagar Indian Institute of Technology Gandhinagar | NPTEL ONLINE CERTIFICATION COURSES | Anirban Dasgupta Computer Science and Engg.

Just for the just to give you some references lot of the slides as I have been gotten from Michael Mahoney and petros stock from their boot camp that you can Google, that you can obtain by Goggling this. And it is really a very nicely set of lectures and we have been following this lecture notes by both of them and.

Thank you.