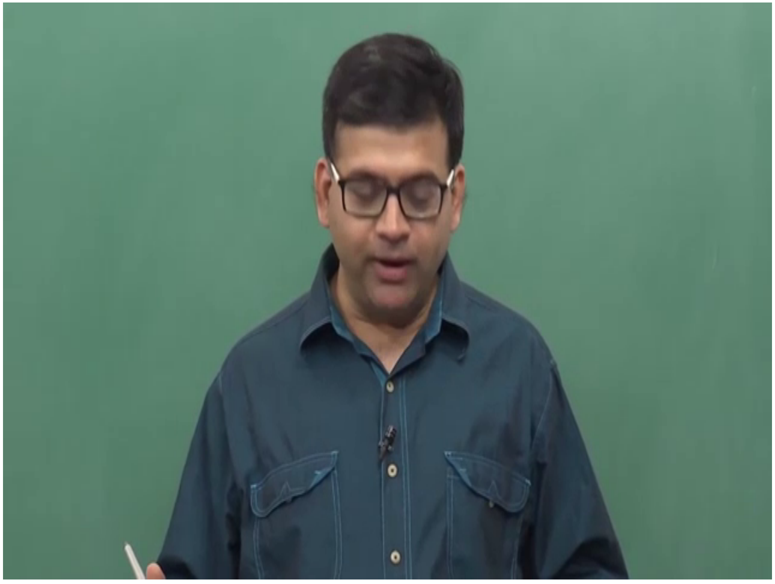
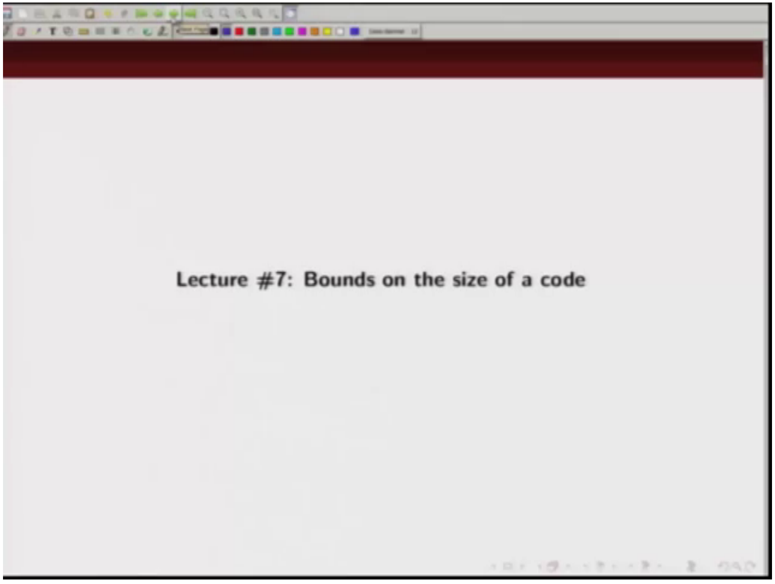
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 03
Lecture Number 13
Bounds on the Size of Code

(Refer Slide Time 00:14)



Today we are going to discuss on bounds on the size of the code

(Refer Slide Time 00:18)



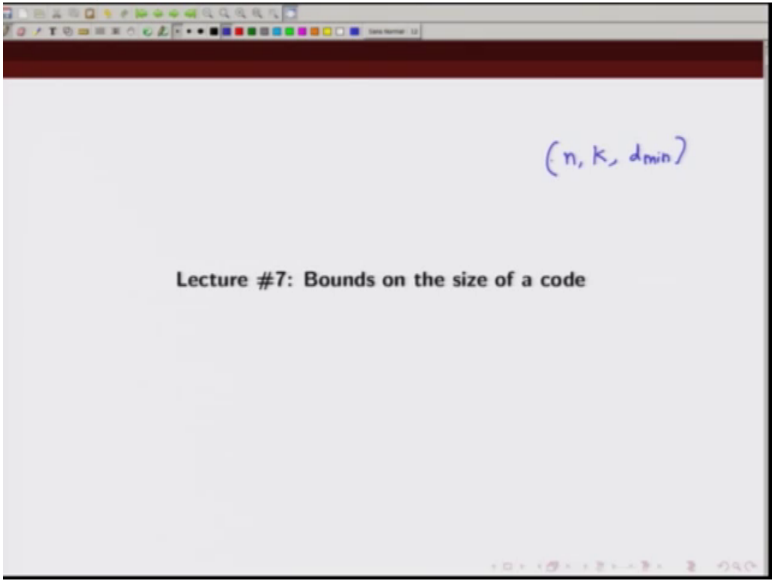
Lecture #7: Bounds on the size of a code

So let's say you know the code

(Refer Slide Time 00:20)

dimension information sequence length and you know the minimum distance of the code. You would like to know, for example, what is the minimum number of parity bits required so that you get that guaranteed minimum distance of the code. So a code as is known can be described by parameters n, k and let's say minimum distance of the code. So if we specify any two of

(Refer Slide Time 00:48)



(n, k, d_{min})

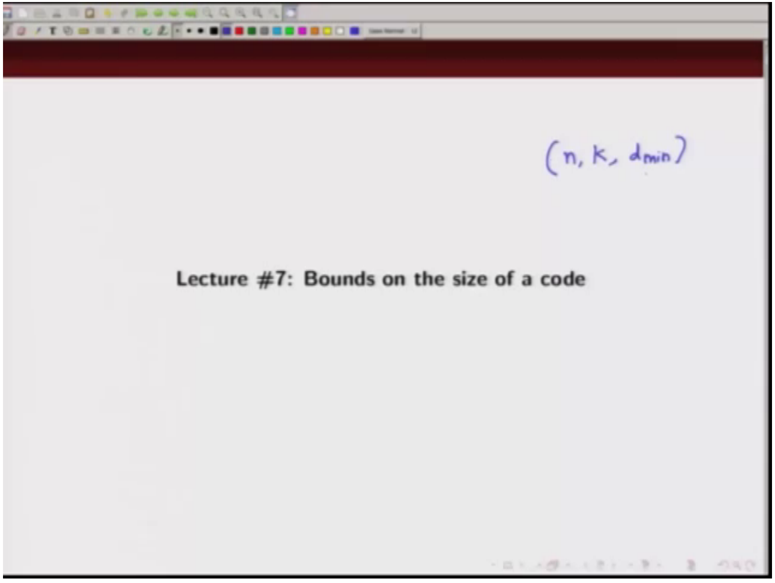
Lecture #7: Bounds on the size of a code

these parameters we would like to know

(Refer Slide Time 00:51)

what would be the third parameter. For example if I specify

(Refer Slide Time 00:55)



(n, k, d_{min})

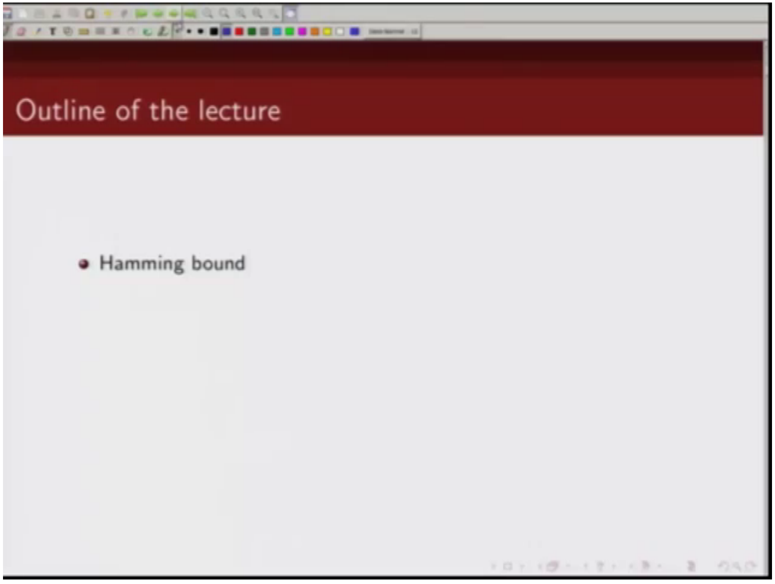
Lecture #7: Bounds on the size of a code

the minimum distance of the code and the information sequence length, I am interested in knowing what is the minimum number of, minimum n required such that I get this k and d, Ok. Or let's say if n and k are specified, I am interested in finding out what is the maximum minimum distance I can get. So fixing 2 parameters I am interested to know about the third

(Refer Slide Time 01:23)

parameter. And in this lecture we are going to talk about bounds that link these 3 parameters. So in particular

(Refer Slide Time 01:35)

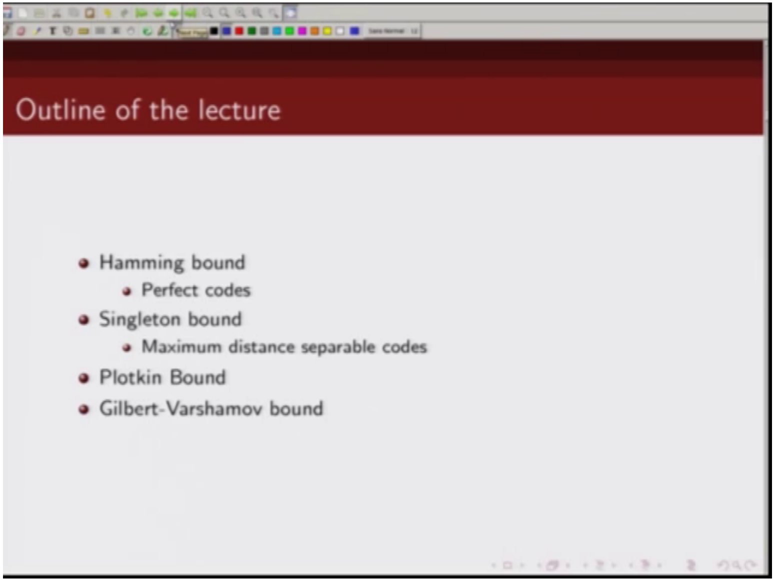


Outline of the lecture

- Hamming bound

we will be talking about Hamming bound and we will introduce the concept of perfect codes. Then we will talk about singleton bound and the codes that satisfy singleton bound are known as maximum distance separable codes, we will talk about them. Then we will talk about Plotkin bounds and Gilbert-Varshamov bound.

(Refer Slide Time 01:59)

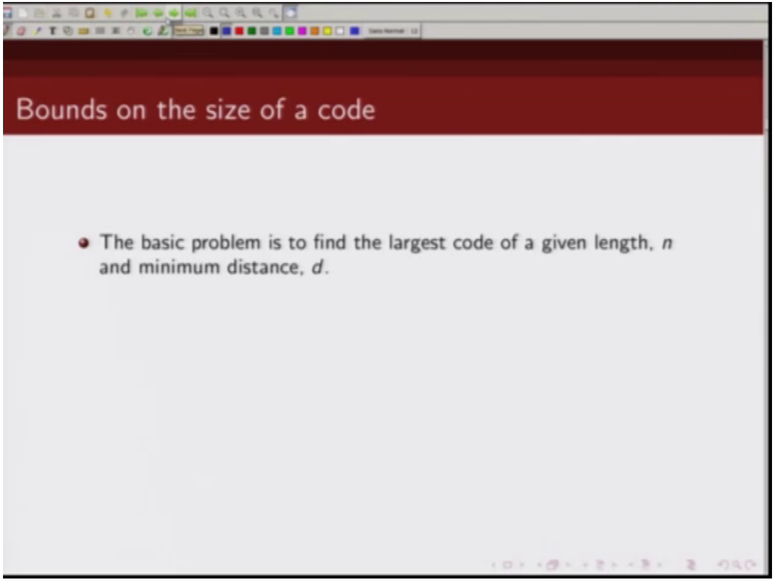


Outline of the lecture

- Hamming bound
  - Perfect codes
- Singleton bound
  - Maximum distance separable codes
- Plotkin Bound
- Gilbert-Varshamov bound

So as I said

(Refer Slide Time 02:02)



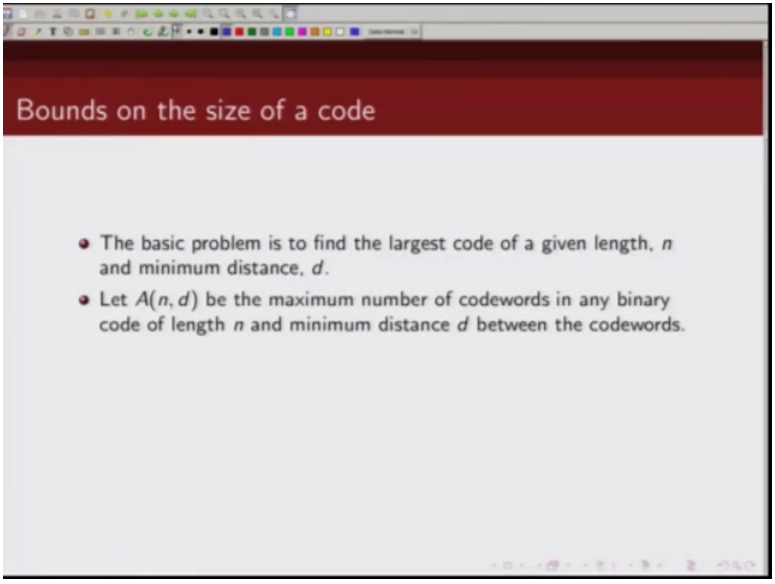
Bounds on the size of a code

- The basic problem is to find the largest code of a given length, n and minimum distance, d.

we are interested to find the largest codeword of length n and minimum distance d. So knowing these two parameters we are interested to know what would be the third parameter and in this talk we are going to talk about bounds which will give us upper bound and lower bounds on those parameters.

(Refer Slide Time 02:24)

(Refer Slide Time 02:26)

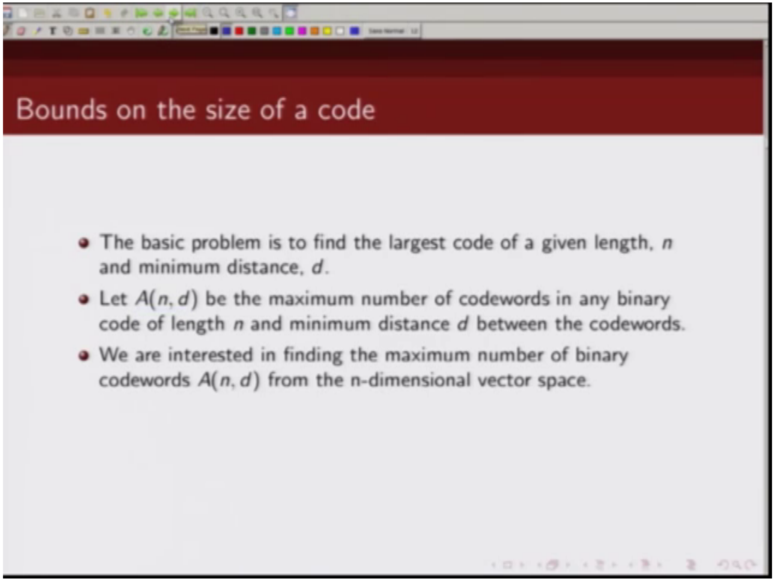


Bounds on the size of a code

- The basic problem is to find the largest code of a given length, n and minimum distance, d.
- Let $A(n, d)$ be the maximum number of codewords in any binary code of length n and minimum distance d between the codewords.

So let a and d denotes the number of codewords of length n and minimum distance d. So we would

(Refer Slide Time 02:37)

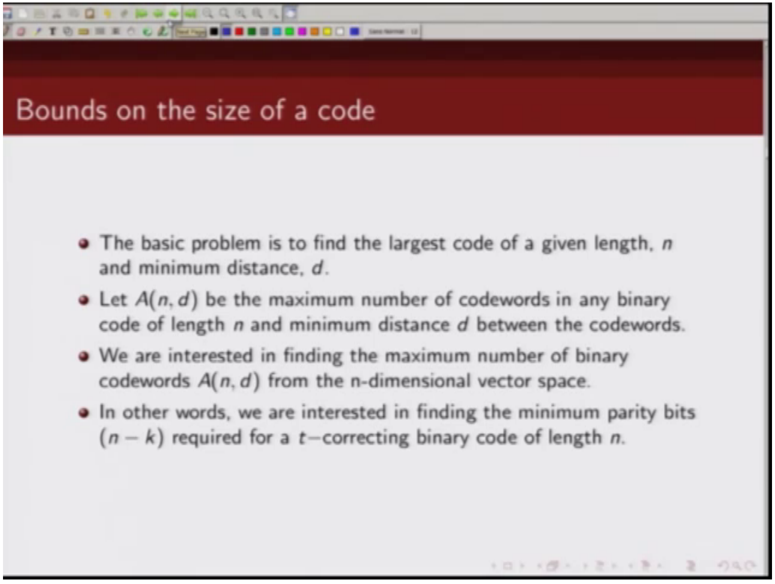


Bounds on the size of a code

- The basic problem is to find the largest code of a given length, n and minimum distance, d.
- Let $A(n, d)$ be the maximum number of codewords in any binary code of length n and minimum distance d between the codewords.
- We are interested in finding the maximum number of binary codewords $A(n, d)$ from the n-dimensional vector space.

like to know how many such codewords exist which have codeword length n or minimum distance d and minimum distance d or in other

(Refer Slide Time 02:48)

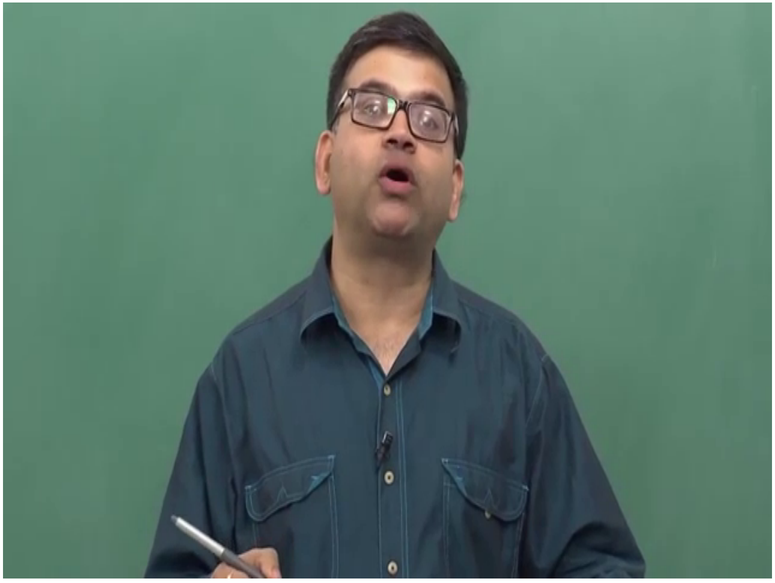


Bounds on the size of a code

- The basic problem is to find the largest code of a given length, n and minimum distance, d.
- Let $A(n, d)$ be the maximum number of codewords in any binary code of length n and minimum distance d between the codewords.
- We are interested in finding the maximum number of binary codewords $A(n, d)$ from the n-dimensional vector space.
- In other words, we are interested in finding the minimum parity bits $(n - k)$ required for a t−correcting binary code of length n.

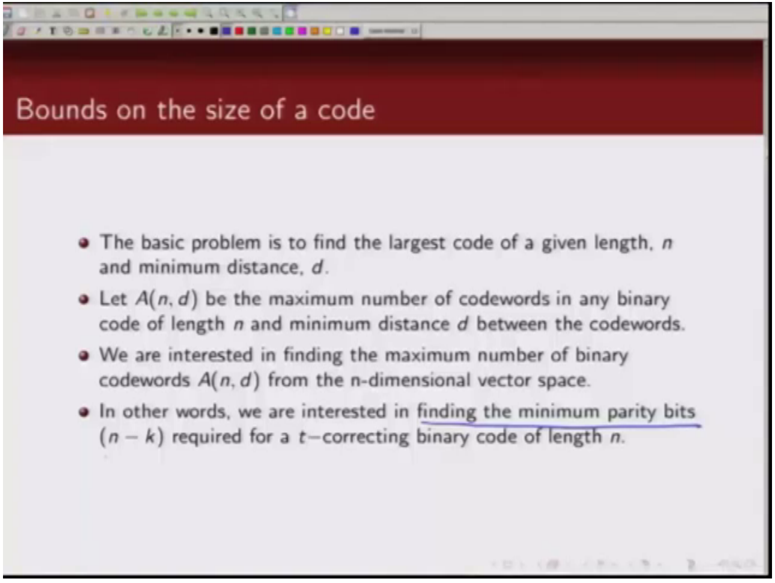
words we could also pose this problem like this. We are interested to find the minimum number of parity bits required. So given that we know k and d, we would like to know what is the minimum

(Refer Slide Time 03:01)



n required such that we can get those k and d. So

(Refer Slide Time 03:08)



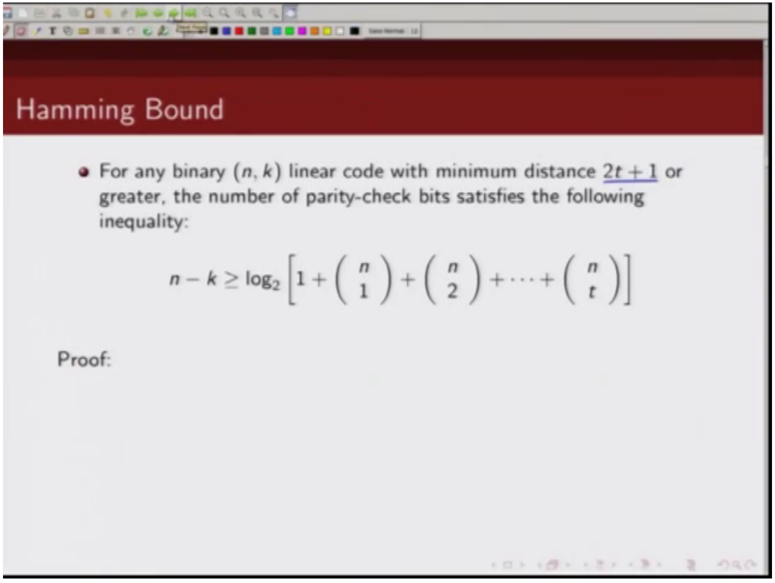
Bounds on the size of a code

- The basic problem is to find the largest code of a given length, n and minimum distance, d.
- Let $A(n, d)$ be the maximum number of codewords in any binary code of length n and minimum distance d between the codewords.
- We are interested in finding the maximum number of binary codewords $A(n, d)$ from the n-dimensional vector space.
- In other words, we are interested in finding the minimum parity bits $(n - k)$ required for a t−correcting binary code of length n.

answer to these questions basically we will pose and we will try to get some bound on those parameters. So if I give you 2 parameters you should be able to, we are interested to know what is the bound on the third parameter. So what are the permissible values for the third parameter?

So we will start

(Refer Slide Time 03:28)



Hamming Bound

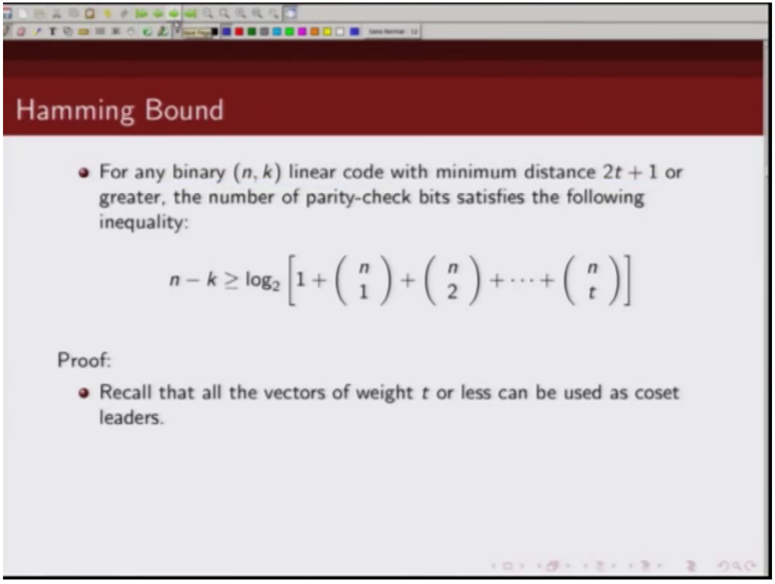
- For any binary (n, k) linear code with minimum distance $2t+1$ or greater, the number of parity-check bits satisfies the following inequality:

$$n-k \geq \log_2\left[1+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{t}\right]$$

Proof:

with Hamming bound. So what is Hamming bound? So the Hamming bound says, for a binary n k linear code whose minimum distance is at least 2 t plus 1, Ok the number of parity check bits satisfies this relationship. So if we have a linear code whose minimum distance is at least 2 t plus 1, then number of parity bits must satisfy this. So number of parity bits are lower bounded by this. So how do we prove this?

(Refer Slide Time 04:14)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t+1$ or greater, the number of parity-check bits satisfies the following inequality:

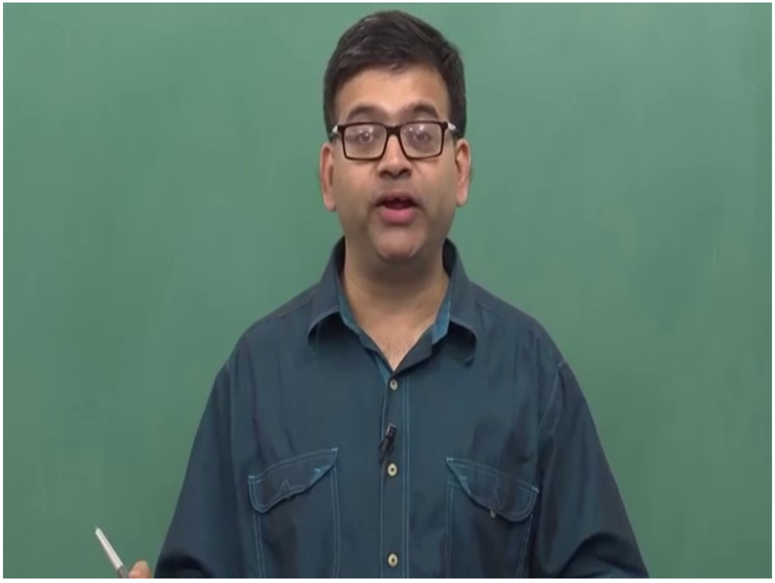
$$n-k \geq \log_2\left[1+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{t}\right]$$

Proof:

- Recall that all the vectors of weight t or less can be used as coset leaders.

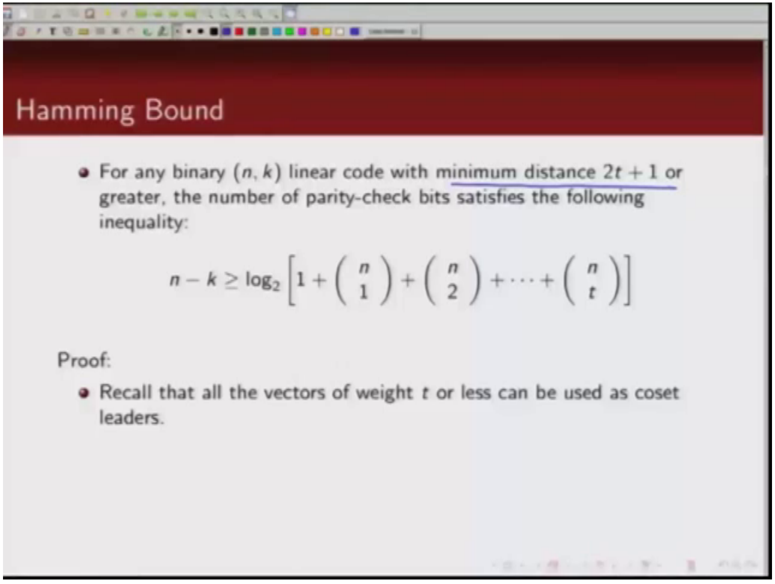
Now if we are interested in a linear block code whose minimum distance is at least 2 t plus 1, we know from the property of linear block code

(Refer Slide Time 04:26)



that this code can correct t errors, Ok. A code which has minimum distance of at least 2 t plus 1 can correct t errors. Now if we look at our syndrome decoding, recall if we want to correct t errors, all these error patterns of weight up to t should be the coset leaders. Then we can correct these error patterns.

(Refer Slide Time 04:58)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t + 1$ or greater, the number of parity-check bits satisfies the following inequality:

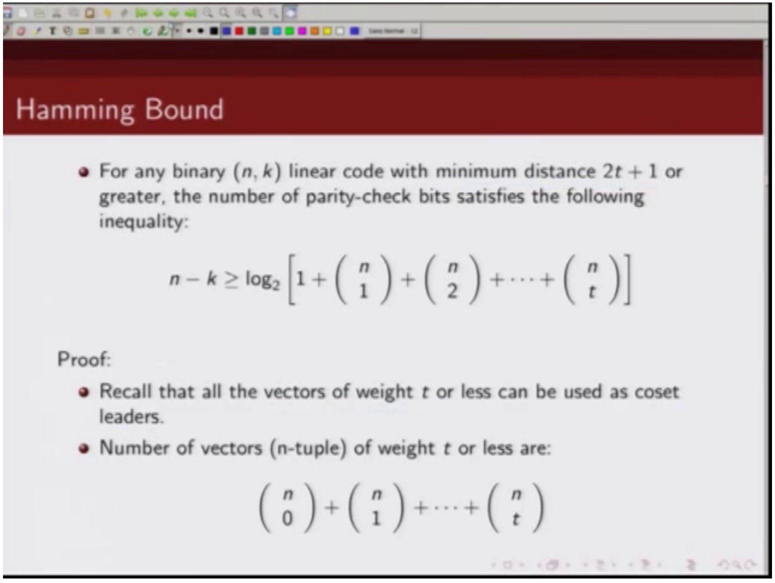
$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t} \right]$$

Proof:

- Recall that all the vectors of weight t or less can be used as coset leaders.

So if a linear code has a minimum distance of 2 t plus 1 which means all error patterns of weight up to t are correctable and hence we can use all weight patterns of t or less weight as coset leaders; so all error patterns up to weight t can be used as coset leaders. Now let

(Refer Slide Time 05:29)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t + 1$ or greater, the number of parity-check bits satisfies the following inequality:

$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right]$$

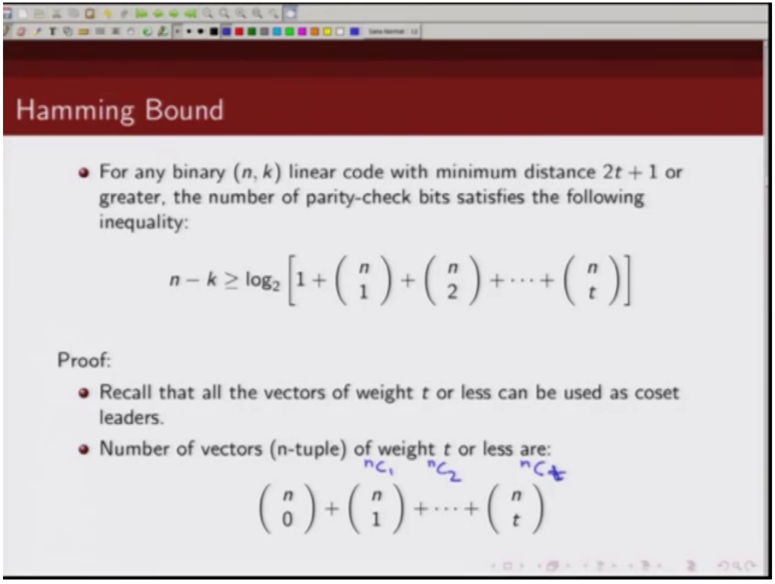
Proof:

- Recall that all the vectors of weight t or less can be used as coset leaders.
- Number of vectors (n-tuple) of weight t or less are:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$$

us count how many such error patterns are there. So let us look at how many error patterns of weight 0 are there. That is given by n C 0. How many error patterns of weight 1 are there? That is given by n C 1. So how many error patterns of weight 2 that is n C 2. And similarly how many error patterns of weight t, that is n C t.

(Refer Slide Time 05:59)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t + 1$ or greater, the number of parity-check bits satisfies the following inequality:

$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right]$$

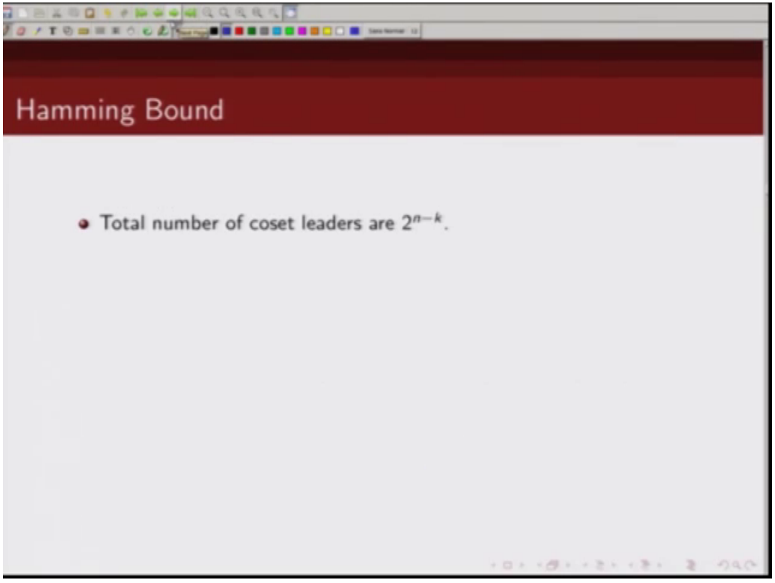
Proof:

- Recall that all the vectors of weight t or less can be used as coset leaders.
- Number of vectors (n-tuple) of weight t or less are:

$$\binom{n}{0} + \overset{^nC_1}{\binom{n}{1}} + \overset{^nC_2}{\cdots} + \overset{^nC_t}{\binom{n}{t}}$$

So these are the total number of error patterns of weight 0 1 2 3 up to weight t. Now note all of these should be the coset leaders. Then only we can correct them. But

(Refer Slide Time 06:18)



Hamming Bound

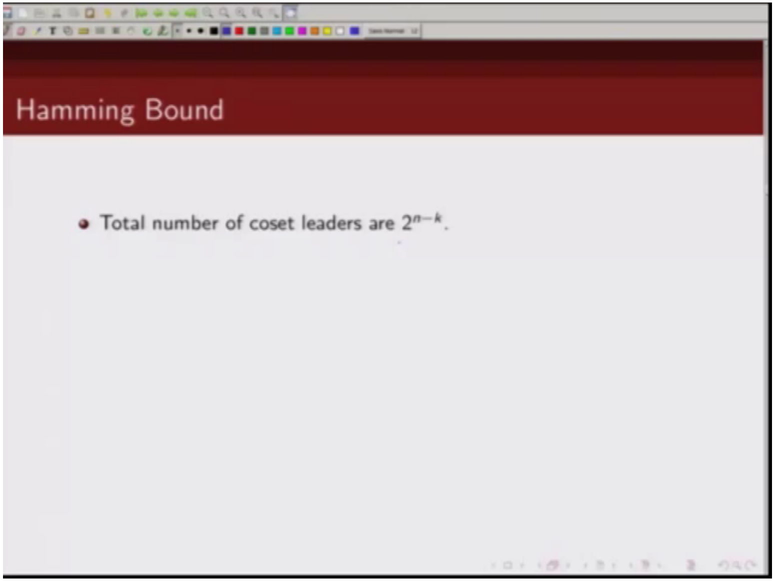
- Total number of coset leaders are 2^{n-k}.

what is the maximum number of coset leaders possible? Now

(Refer Slide Time 06:22)

that number we know is given

(Refer Slide Time 06:24)

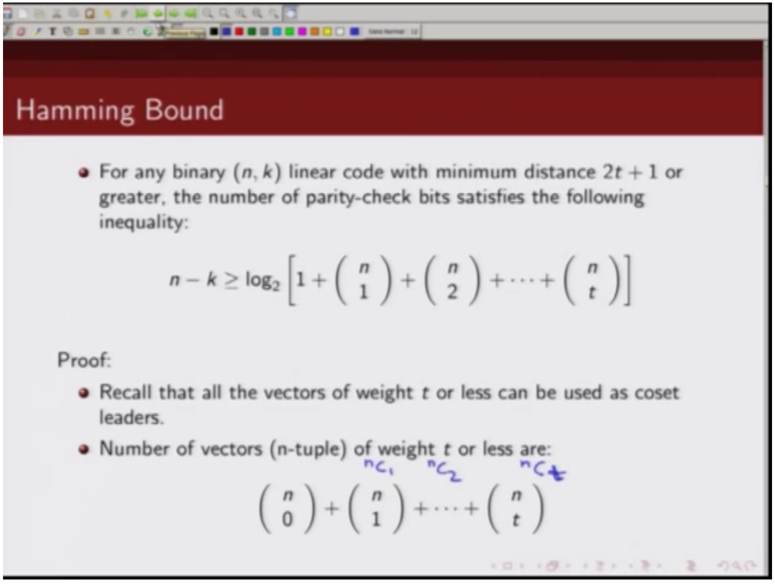


Hamming Bound

- Total number of coset leaders are 2^{n-k}.

by 2 raised to power n minus k. So these are the total number of coset leaders we have. Now

(Refer Slide Time 06:33)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t + 1$ or greater, the number of parity-check bits satisfies the following inequality:

$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t} \right]$$

Proof:

- Recall that all the vectors of weight t or less can be used as coset leaders.
- Number of vectors (n-tuple) of weight t or less are:

$^nC_1$ $^nC_2$ ^nC_t

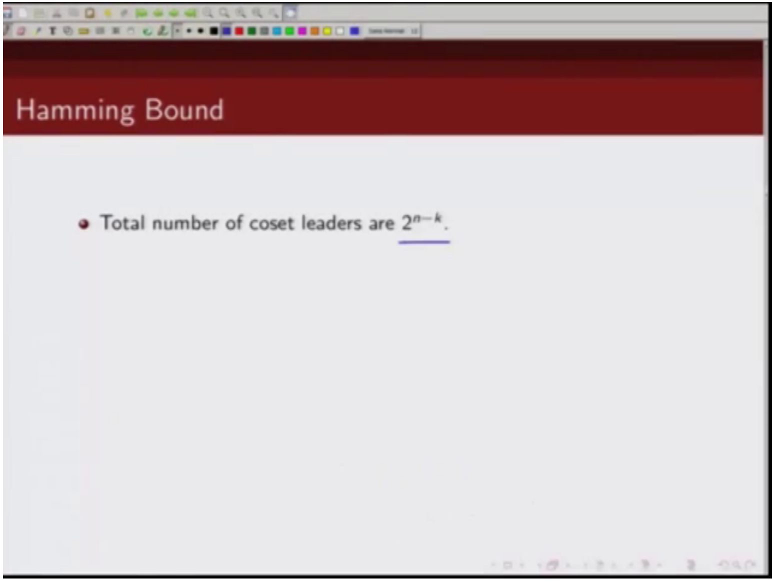
$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$$

we want all of these error patterns

(Refer Slide Time 06:36)

to be coset leaders. Then this number should be less than

(Refer Slide Time 06:43)

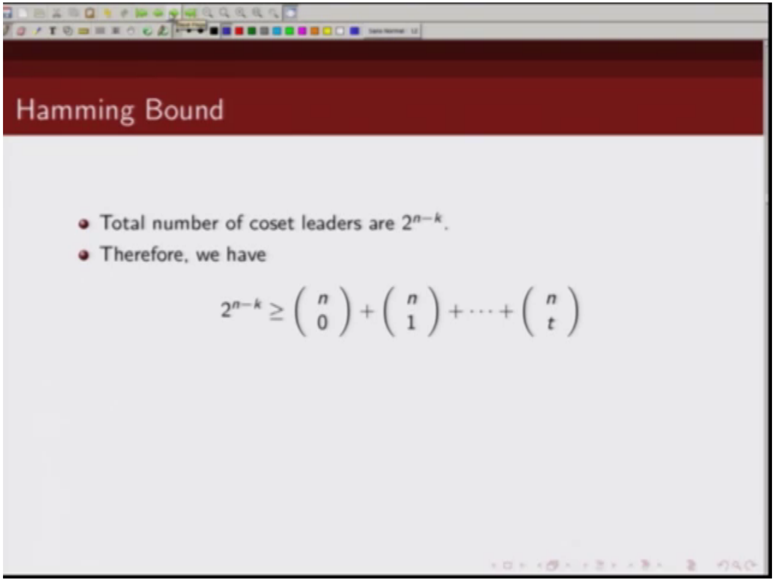


Hamming Bound

- Total number of coset leaders are 2^{n-k}.

2 raised to power n minus k. Hence

(Refer Slide Time 06:47)



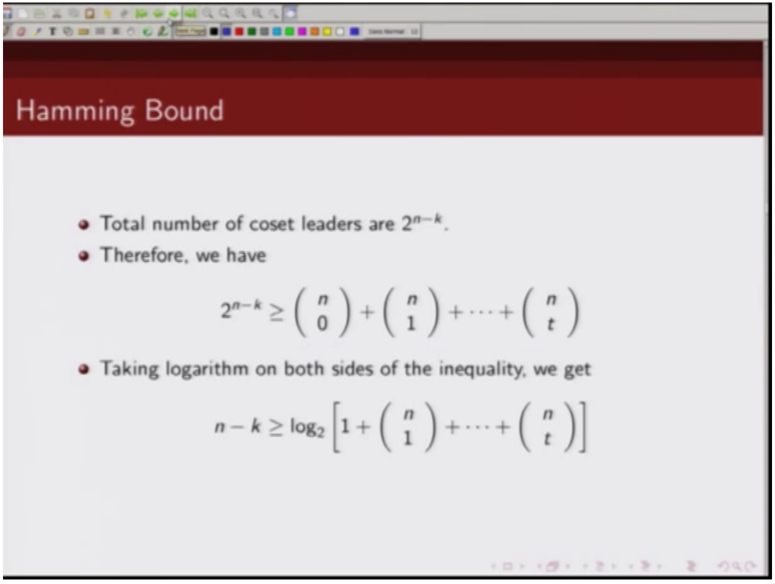
Hamming Bound

- Total number of coset leaders are 2^{n-k}.
- Therefore, we have

$$2^{n-k} \geq \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$$

we get this condition that total number of coset leaders should be more than all error patterns of weight up to t. And hence we take

(Refer Slide Time 07:01)



Hamming Bound

- Total number of coset leaders are 2^{n-k}.
- Therefore, we have

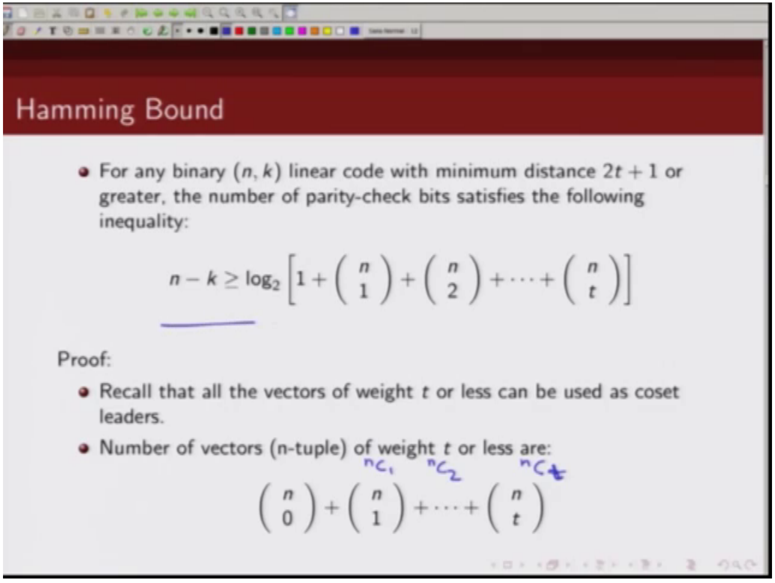
$$2^{n-k} \geq \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$$

- Taking logarithm on both sides of the inequality, we get

$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \cdots + \binom{n}{t}\right]$$

the log of this, we get this condition that n minus k is less than, is greater than equal to log of this, Ok and that's basically our

(Refer Slide Time 07:15)



Hamming Bound

- For any binary (n, k) linear code with minimum distance $2t+1$ or greater, the number of parity-check bits satisfies the following inequality:

$$\underline{n-k} \geq \log_2\left[1+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{t}\right]$$

Proof:

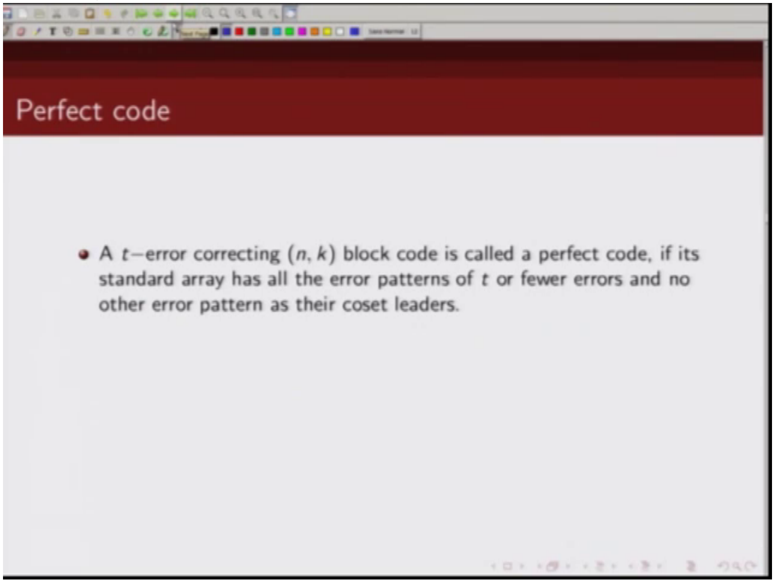
- Recall that all the vectors of weight t or less can be used as coset leaders.
- Number of vectors (n-tuple) of weight t or less are:

$$\binom{n}{0}+\overset{^nC_1}{\binom{n}{1}}+\overset{^nC_2}{\cdots}+\overset{^nC_t}{\binom{n}{t}}$$

Hamming bound.

Now

(Refer Slide Time 07:20)

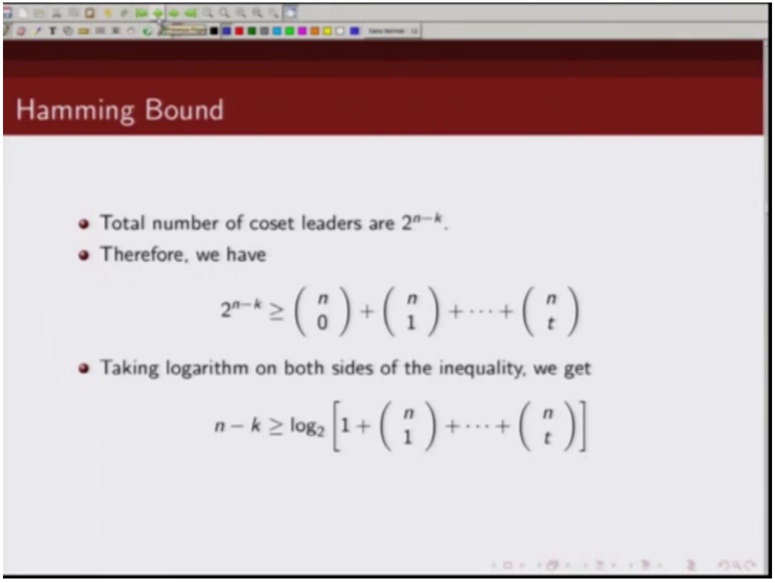


Perfect code

- A $t-$error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.

when is our Hamming bound satisfied with equality?

(Refer Slide Time 07:27)



Hamming Bound

- Total number of coset leaders are 2^{n-k}.
- Therefore, we have

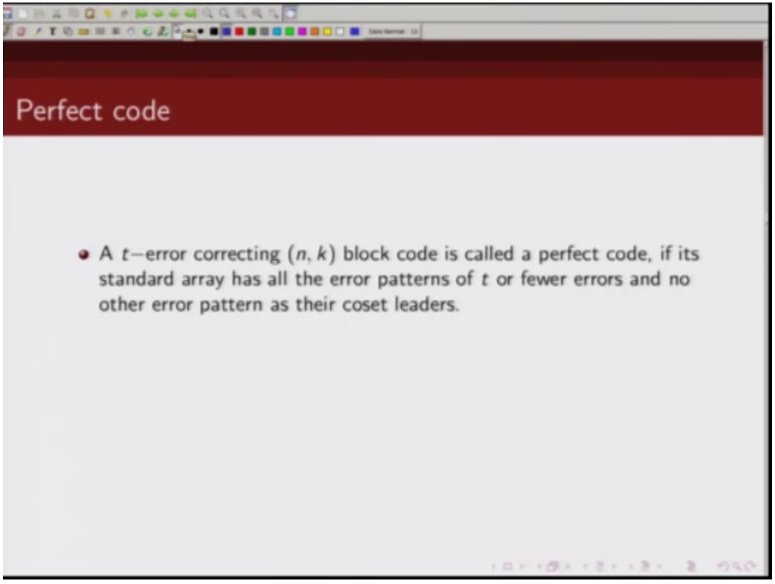
$$2^{n-k} \geq \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$$

- Taking logarithm on both sides of the inequality, we get

$$n-k \geq \log_2 \left[1 + \binom{n}{1} + \cdots + \binom{n}{t}\right]$$

Our Hamming bound will be satisfied with equality when all error patterns of weight up to t are coset leaders and no other error pattern is coset leader. So when this is satisfied with equality, when this equation satisfied then this inequality satisfied with equality then we have Hamming bound satisfied with

(Refer Slide Time 07:53)



Perfect code

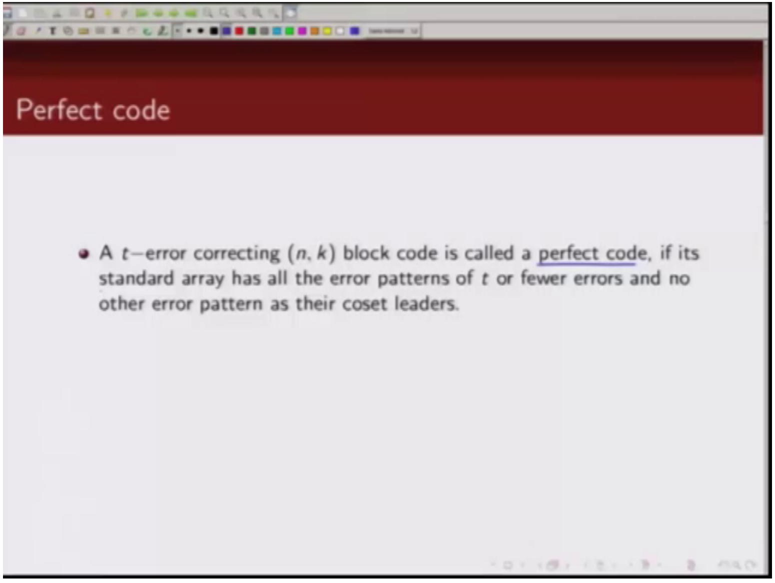
- A $t-$error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.

equality. So a t error correcting code is called perfect code if it satisfies Hamming bound

(Refer Slide Time 08:04)

with equality. And when will it satisfy Hamming bound with equality? When

(Refer Slide Time 08:10)



Perfect code

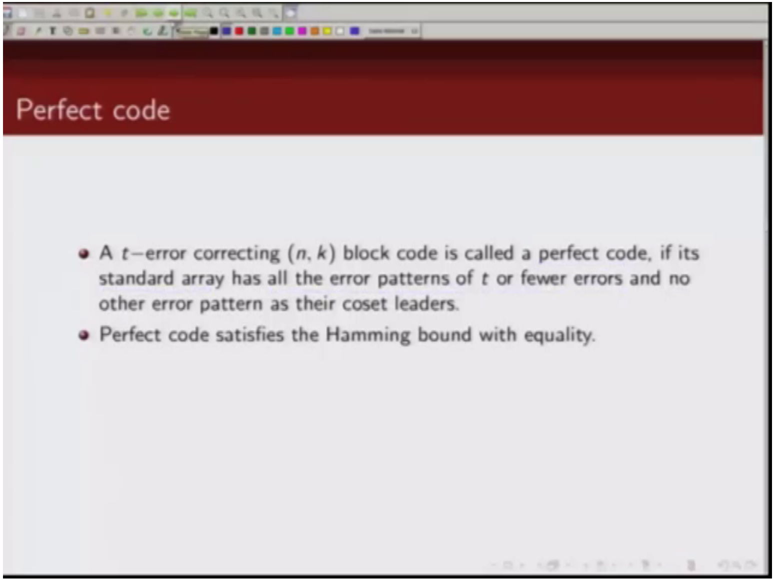
- A t−error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.

its standard array has all error patterns of t or fewer errors and no other error pattern as their coset leader. So it is important. No other error pattern except all error pattern up to weight t should be the coset leader.

(Refer Slide Time 08:33)

So if Hamming bound is satisfied with equality it is known as perfect

(Refer Slide Time 08:40)

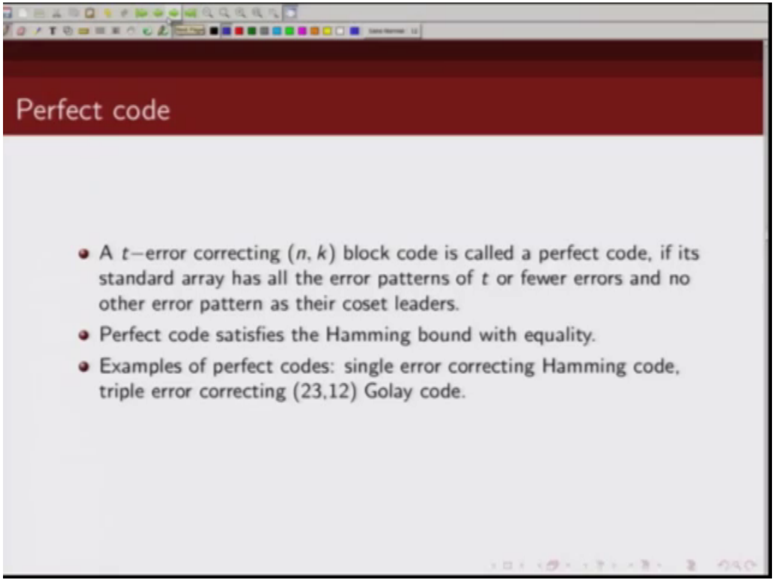


Perfect code

- A t−error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.
- Perfect code satisfies the Hamming bound with equality.

code. Now

(Refer Slide Time 08:42)

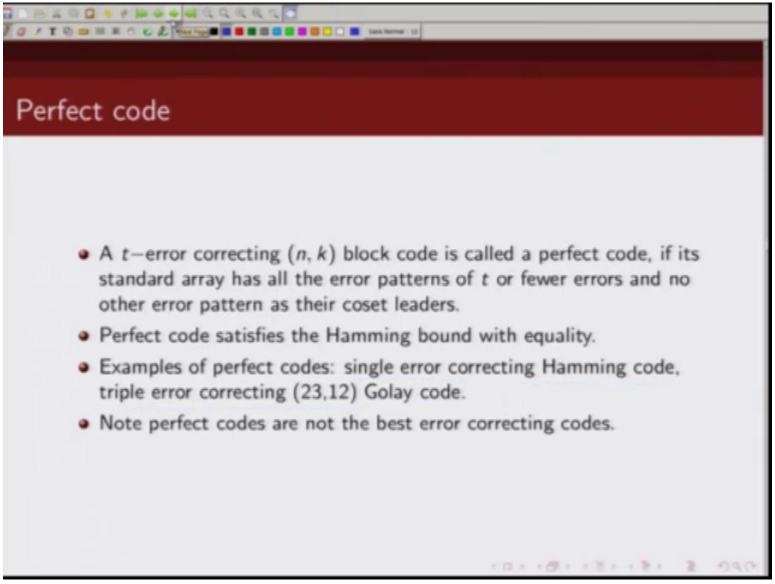


Perfect code

- A t−error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.
- Perfect code satisfies the Hamming bound with equality.
- Examples of perfect codes: single error correcting Hamming code, triple error correcting (23,12) Golay code.

examples of perfect code is, for example single error correcting Hamming code or triple error correcting 23 12 Golay code. Now I just want

(Refer Slide Time 08:56)

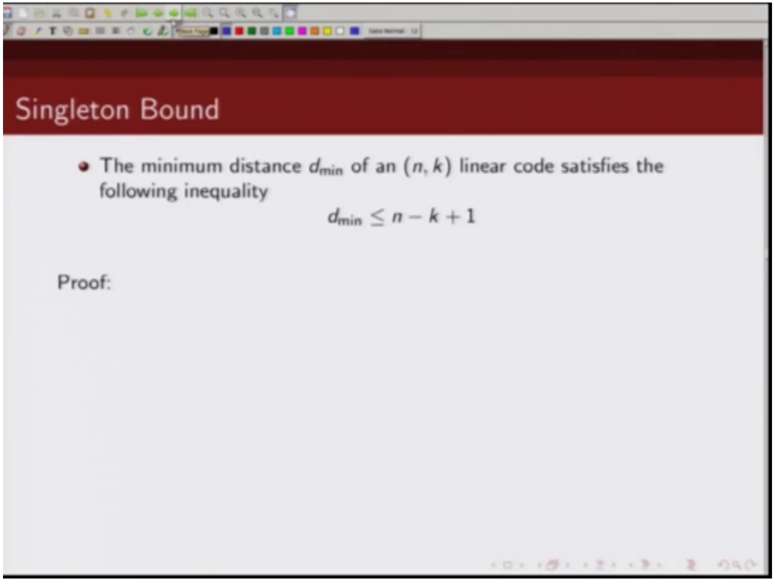


Perfect code

- A t−error correcting (n, k) block code is called a perfect code, if its standard array has all the error patterns of t or fewer errors and no other error pattern as their coset leaders.
- Perfect code satisfies the Hamming bound with equality.
- Examples of perfect codes: single error correcting Hamming code, triple error correcting (23,12) Golay code.
- Note perfect codes are not the best error correcting codes.

to caution you that perfect code does not mean these are the best possible codes, Ok. So don't confuse perfect code as the best possible error correcting codes.

(Refer Slide Time 09:11)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

$$d_{\min} \leq n - k + 1$$

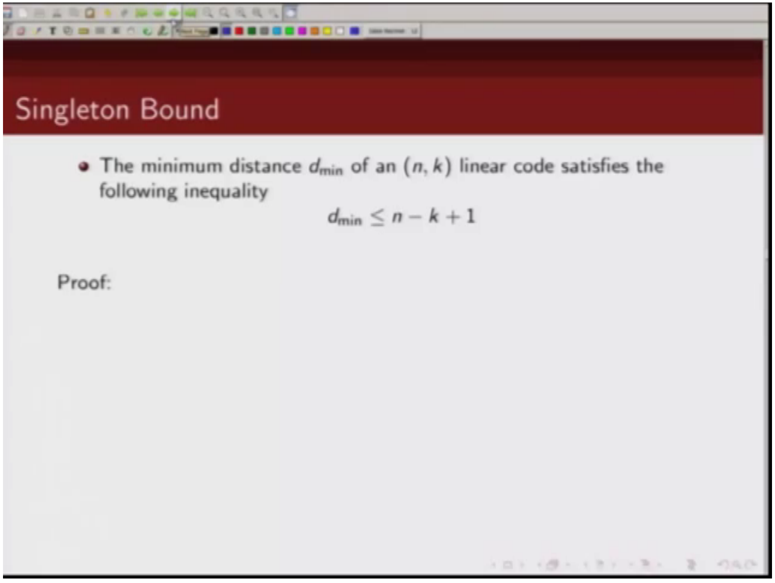
Proof:

The next bound that we will talk about is Singleton bound which gives an upper bound of, on

(Refer Slide Time 09:19)

minimum distance. So it says, the singleton bound says

(Refer Slide Time 09:23)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

$$d_{\min} \leq n - k + 1$$

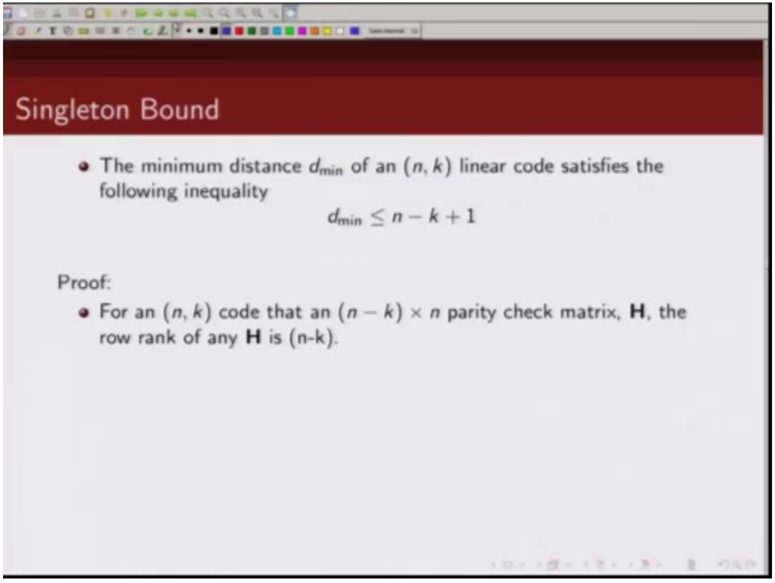
Proof:

that a minimum distance of n k linear block code must satisfy this inequality; so minimum distance is less than equal to n minus k

(Refer Slide Time 09:38)

plus 1. Now how do we prove this?

(Refer Slide Time 09:44)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

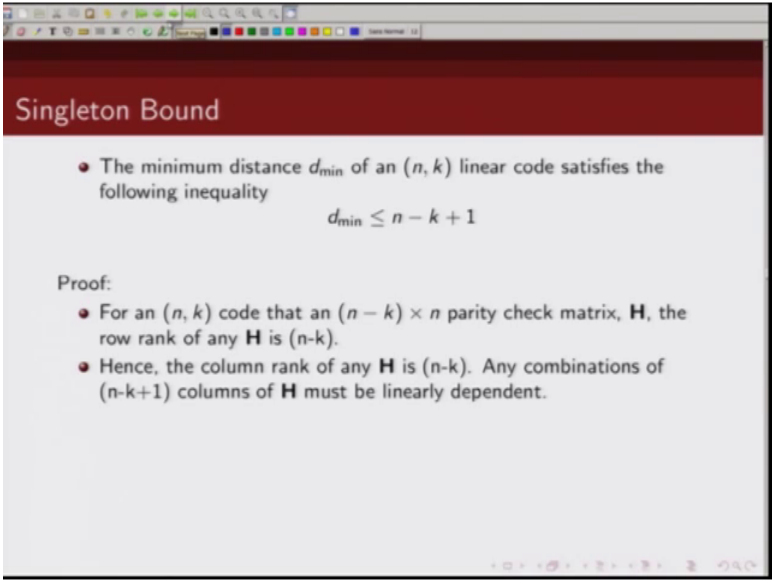
$$d_{\min} \leq n - k + 1$$

Proof:

- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).

So for an n-k linear block code, we have n minus k cross n parity check matrix. And what is the row rank of the parity check matrix? That is n minus k. Now

(Refer Slide Time 10:05)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

$$d_{\min} \leq n - k + 1$$

Proof:

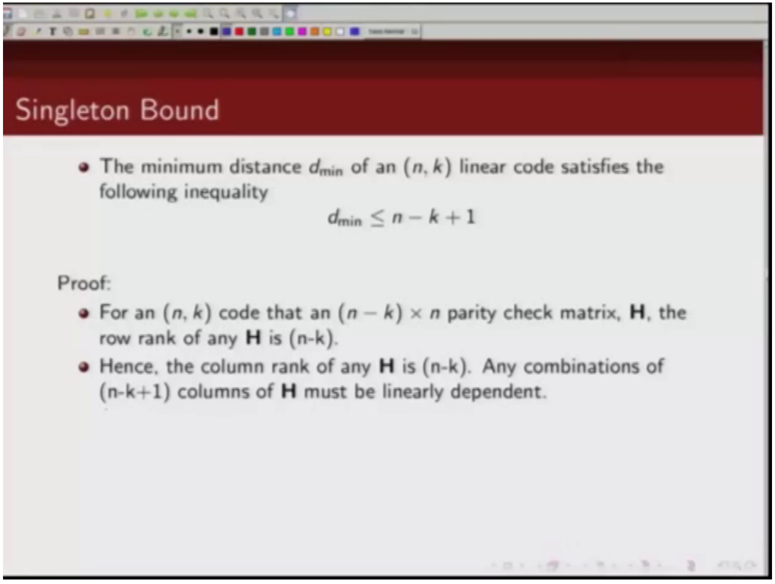
- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.

row rank is n minus k then column rank

(Refer Slide Time 10:08)

is also n minus k. Now if the column rank of parity check matrix H is n minus k that means any combinations of n minus

(Refer Slide Time 10:18)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality
$$d_{\min} \leq n - k + 1$$

Proof:

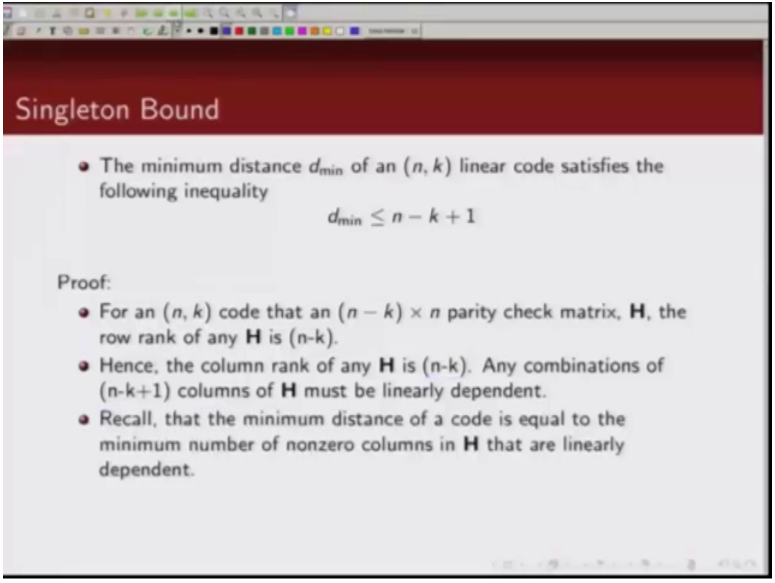
- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.

k plus 1, because the row rank is n minus k, so if we take n minus k plus 1 columns they must be linearly dependent, right. So they must be linearly dependent. And what do we know, what is the relationship

(Refer Slide Time 10:37)

between the columns of the parity check matrix and minimum distance of a code?

(Refer Slide Time 10:43)



Singleton Bound

- The minimum distance d_{min} of an (n,k) linear code satisfies the following inequality
$$d_{\text{min}} \leq n - k + 1$$

Proof:

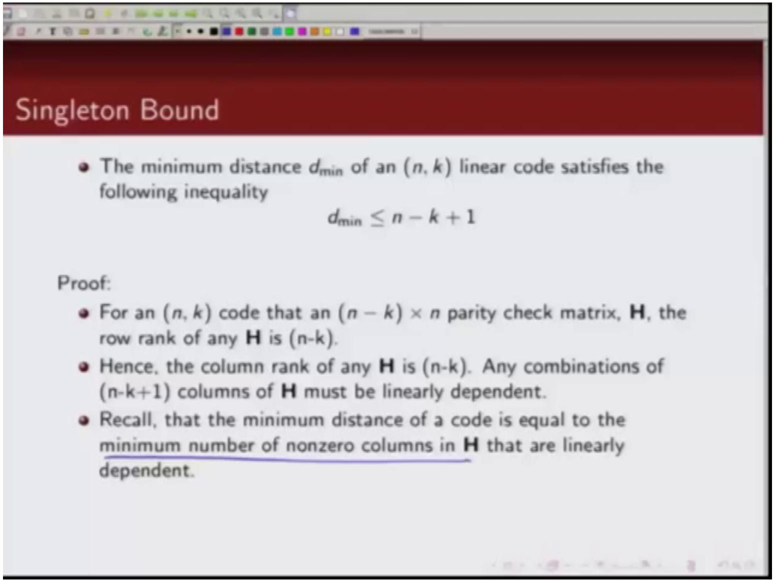
- For an (n,k) code that an $(n-k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.
- Recall, that the minimum distance of a code is equal to the minimum number of nonzero columns in **H** that are linearly dependent.

Now we know that a minimum distance of the code is equal to the minimum number of non-zero columns in the parity check matrix that are linearly dependent. If you recall, we have proved

(Refer Slide Time 11:00)

that the minimum number of the columns of the parity check matrix add up to zero, then there exists the code of that weight. Now so

(Refer Slide Time 11:14)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality
$$d_{\min} \leq n - k + 1$$

Proof:

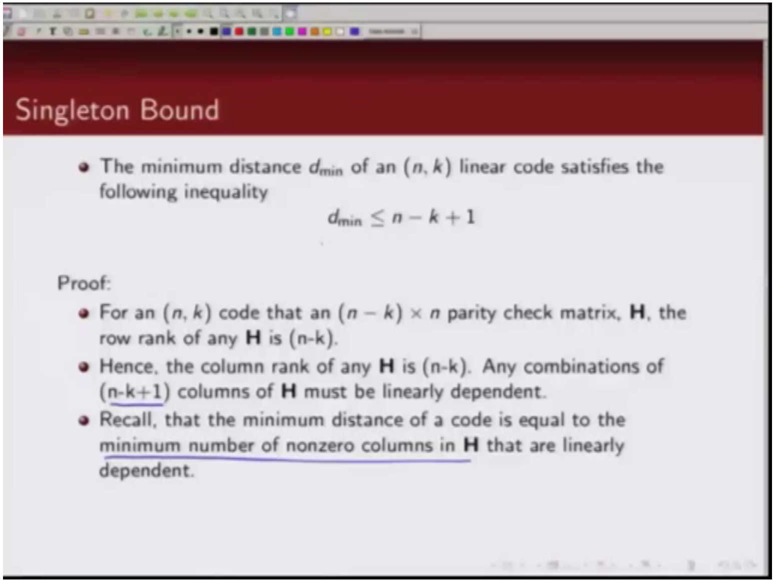
- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.
- Recall, that the minimum distance of a code is equal to the minimum number of nonzero columns in **H** that are linearly dependent.

what we have seen is then n minus k plus one columns of H matrix, they add up to zero because they are linearly dependent. Why, because the column rank is n minus k. And if n minus k

(Refer Slide Time 11:30)

plus 1 columns add up to zero that means the minimum distance can be at most n minus k plus 1. Hence we prove that

(Refer Slide Time 11:43)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

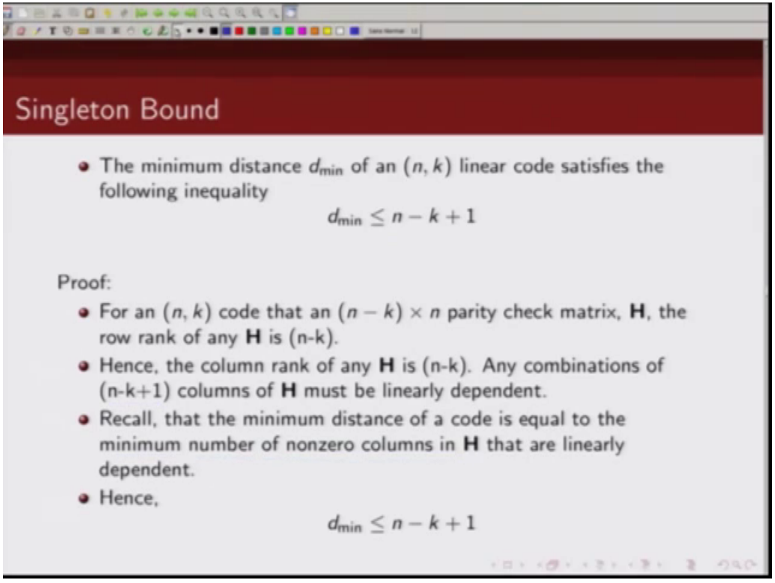
$$d_{\min} \leq n - k + 1$$

Proof:

- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.
- Recall, that the minimum distance of a code is equal to the minimum number of nonzero columns in **H** that are linearly dependent.

that minimum distance of a linear n k code is upper bounded by n minus k plus 1.

(Refer Slide Time 11:56)



Singleton Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

$$d_{\min} \leq n - k + 1$$

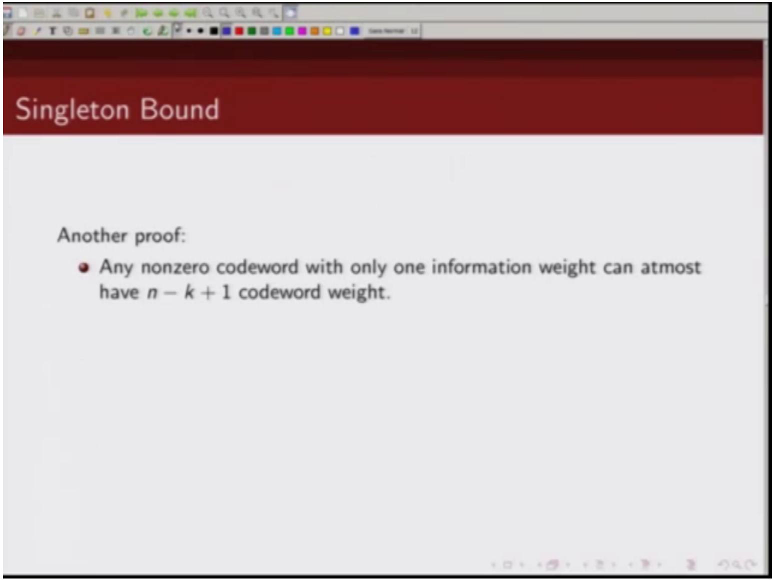
Proof:

- For an (n, k) code that an $(n - k) \times n$ parity check matrix, **H**, the row rank of any **H** is (n-k).
- Hence, the column rank of any **H** is (n-k). Any combinations of (n-k+1) columns of **H** must be linearly dependent.
- Recall, that the minimum distance of a code is equal to the minimum number of nonzero columns in **H** that are linearly dependent.
- Hence,

$$d_{\min} \leq n - k + 1$$

Now the same thing we can prove in a

(Refer Slide Time 11:58)



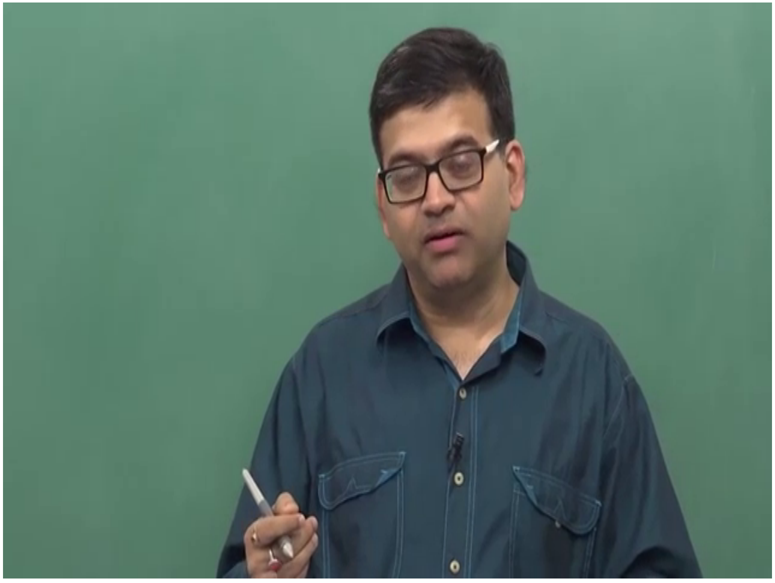
Singleton Bound

Another proof:

- Any nonzero codeword with only one information weight can atmost have $n - k + 1$ codeword weight.

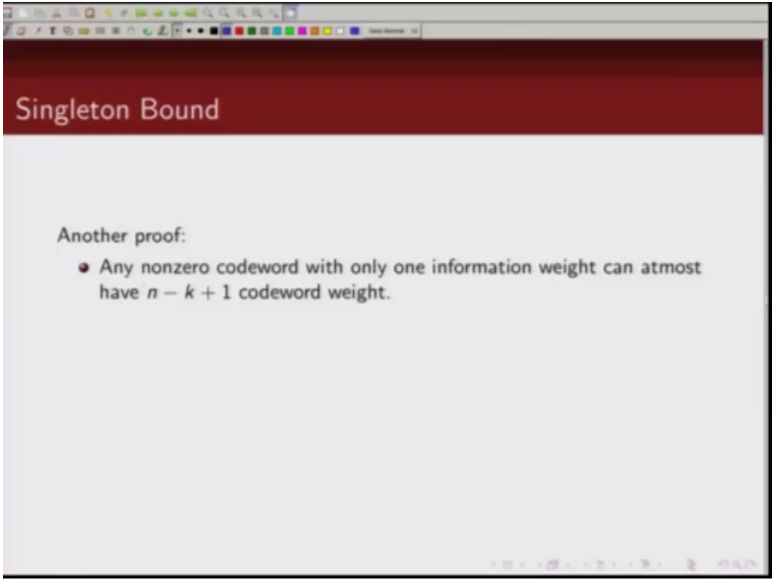
different way. I will just give you an alternative proof of the same result. So what is a minimum weight

(Refer Slide Time 12:10)



information sequence, non zero information sequence? So if we consider

(Refer Slide Time 12:14)



Singleton Bound

Another proof:

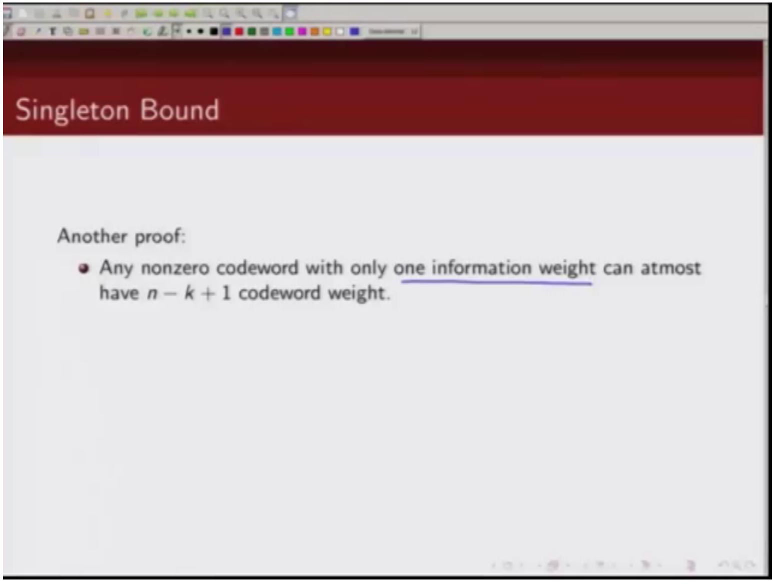
- Any nonzero codeword with only one information weight can atmost have $n - k + 1$ codeword weight.

minimum weight information sequence; that would be weight 1, right? So the minimum non zero information sequence weight is

(Refer Slide Time 12:25)

weight 1, right? Now if we consider minimum non zero information sequence which is of weight 1, now how many number of parity bits we have; n minus k. Now let's assume all n minus k of these parity bits are 1.Then what is the maximum possible minimum distance? That is n minus k plus 1, which is the weight of the information sequence. So we cannot have

(Refer Slide Time 12:58)



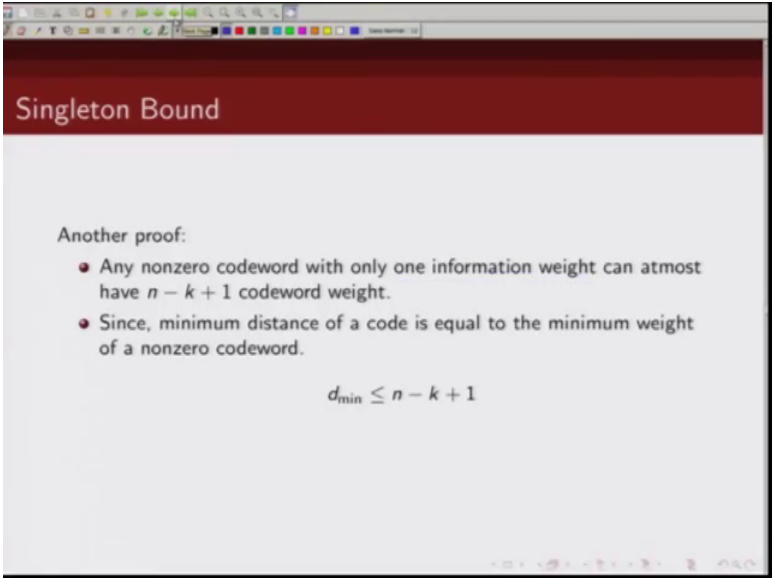
Singleton Bound

Another proof:

- Any nonzero codeword with only one information weight can atmost have $n-k+1$ codeword weight.

weight more than n minus k.

(Refer Slide Time 13:01)



Singleton Bound

Another proof:

- Any nonzero codeword with only one information weight can atmost have $n-k+1$ codeword weight.
- Since, minimum distance of a code is equal to the minimum weight of a nonzero codeword.

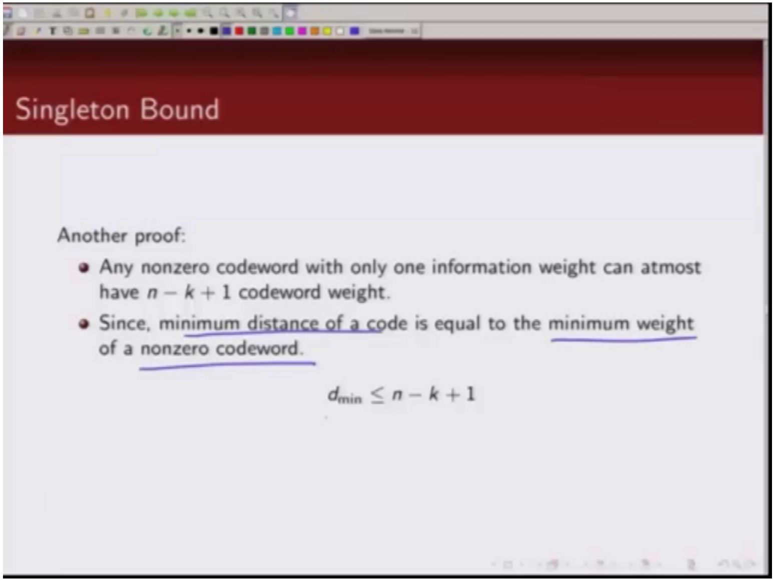
$$d_{\text{min}} \leq n-k+1$$

And since the minimum distance of the code is equal to the minimum weight of the non zero codeword, right? So if you feed in a non zero information sequence, the maximum output weight

(Refer Slide Time 13:16)

that you can get is n minus k plus 1. And hence the minimum

(Refer Slide Time 13:22)



Singleton Bound

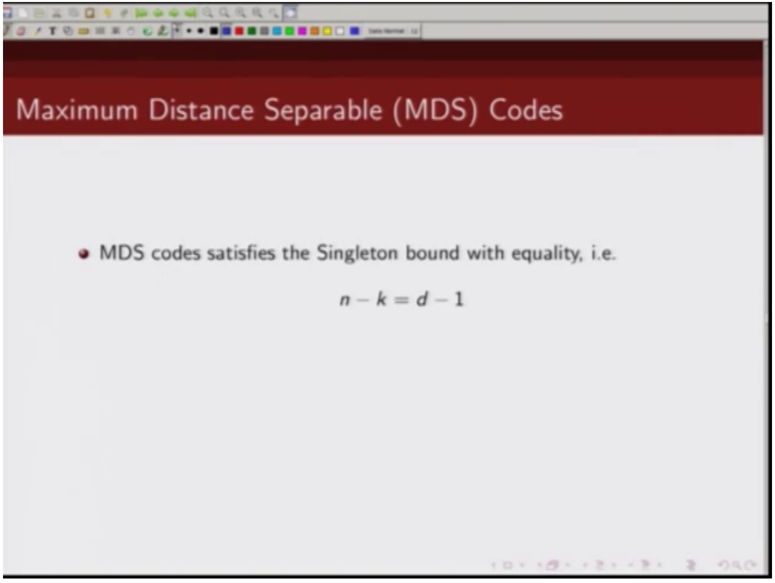
Another proof:

- Any nonzero codeword with only one information weight can atmost have $n-k+1$ codeword weight.
- Since, minimum distance of a code is equal to the minimum weight of a nonzero codeword.

$$d_{\min} \leq n-k+1$$

distance of the code cannot be more than n minus k plus 1. Now

(Refer Slide Time 13:31)



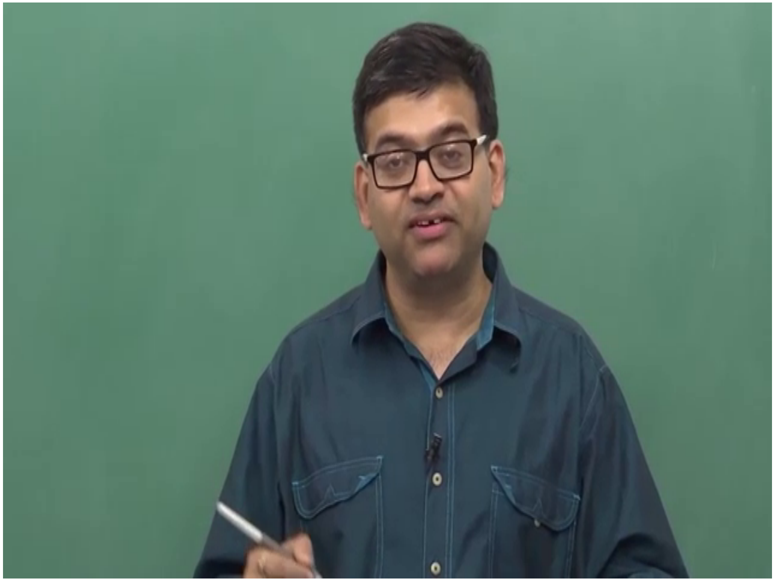
Maximum Distance Separable (MDS) Codes

- MDS codes satisfies the Singleton bound with equality, i.e.

$$n-k=d-1$$

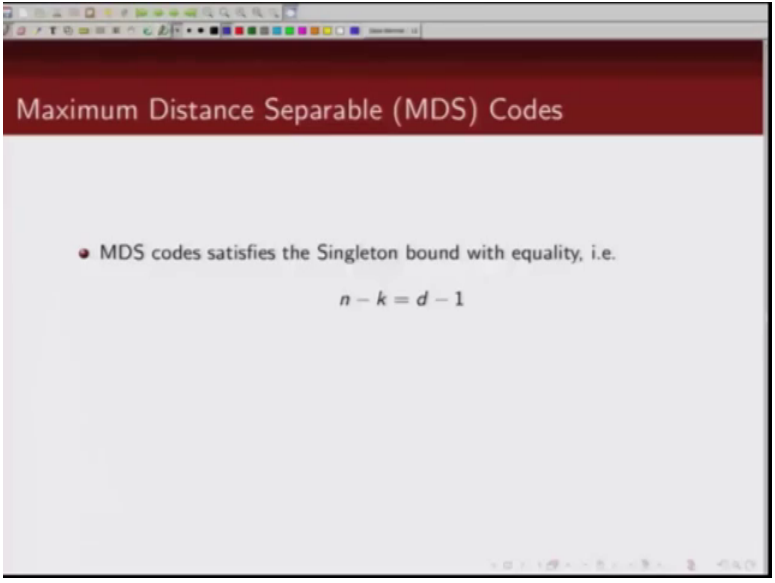
the code that satisfies singleton bound with equality are known as maximum distance separable code. So maximum distance separable code will have

(Refer Slide Time 13:43)



the property that minimum distance is equal to n minus k plus 1.

(Refer Slide Time 13:51)

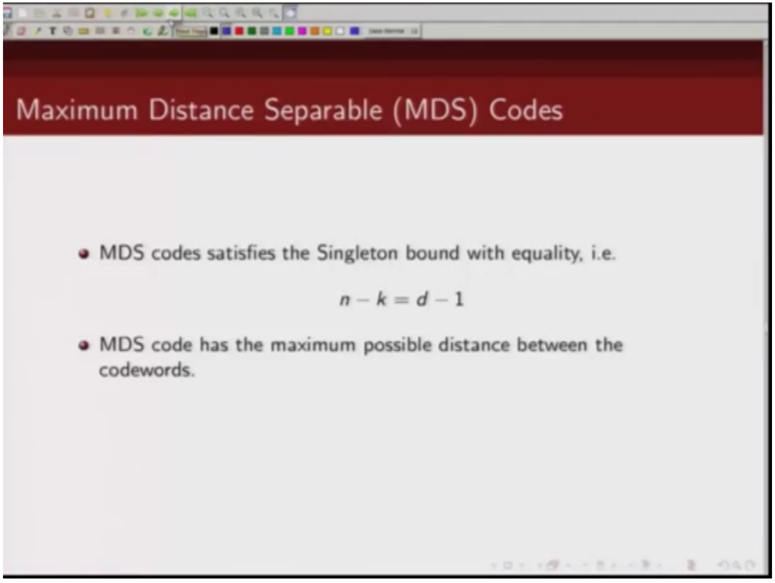


Maximum Distance Separable (MDS) Codes

- MDS codes satisfies the Singleton bound with equality, i.e.

$$n - k = d - 1$$

(Refer Slide Time 13:52)



Maximum Distance Separable (MDS) Codes

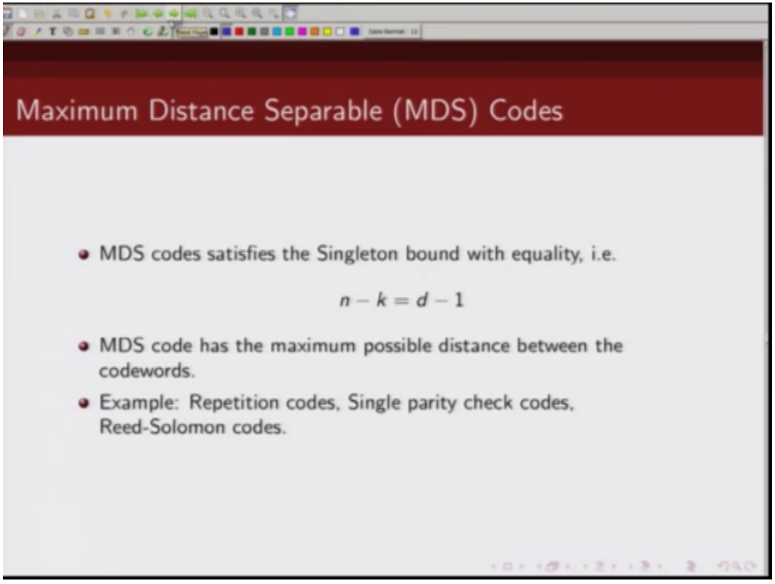
- MDS codes satisfies the Singleton bound with equality, i.e.

$$n - k = d - 1$$

- MDS code has the maximum possible distance between the codewords.

And these are very good code. They have the maximum distance possible between the set of codewords. Examples

(Refer Slide Time 14:00)



Maximum Distance Separable (MDS) Codes

- MDS codes satisfies the Singleton bound with equality, i.e.

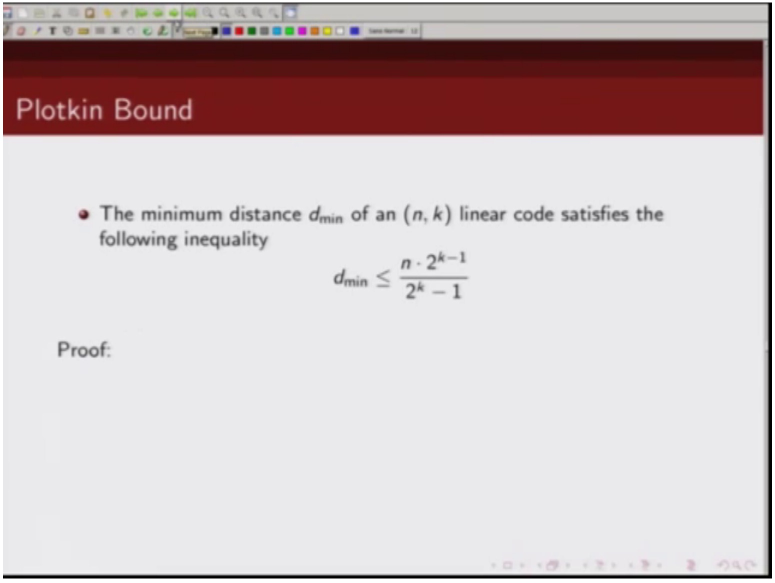
$$n - k = d - 1$$

- MDS code has the maximum possible distance between the codewords.
- Example: Repetition codes, Single parity check codes, Reed-Solomon codes.

of maximum distance separable codes are repetition code, single parity check codes, Reed-Solomon codes.

The third bound

(Refer Slide Time 14:11)



Plotkin Bound

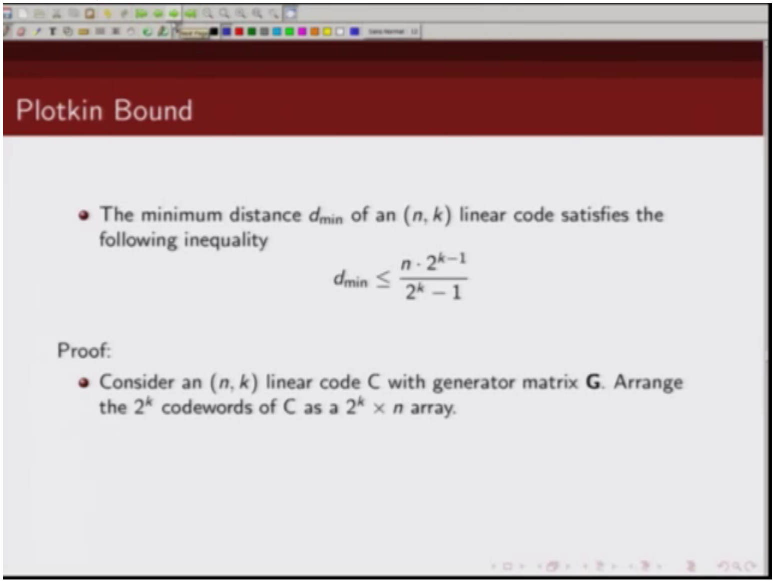
- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Proof:

that we are going to prove is what is known as Plotkin bound. So what does Plotkin bound says? That minimum distance of an n k linear code satisfies this inequality. So the minimum distance of the code is upper bounded by this quantity. Now to prove this,

(Refer Slide Time 14:34)



Plotkin Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

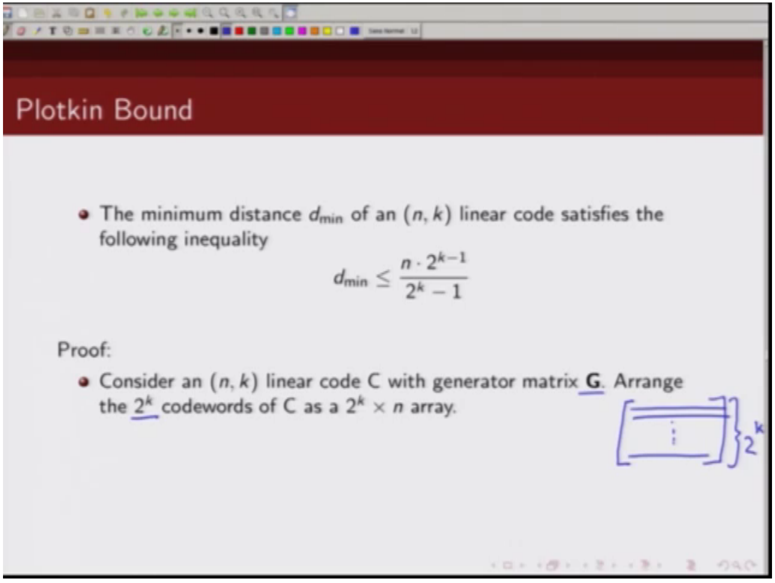
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Proof:

- Consider an (n, k) linear code C with generator matrix **G**. Arrange the 2^k codewords of C as a $2^k \times n$ array.

we will first consider a linear n k code whose generator matrix is G. Since it is a linear n k code the total number of codewords are 2 k codewords. So we will arrange these 2 k codewords as n array. So these are your n bit codewords and we will arrange all of these 2 k codewords as rows

(Refer Slide Time 15:03)



Plotkin Bound

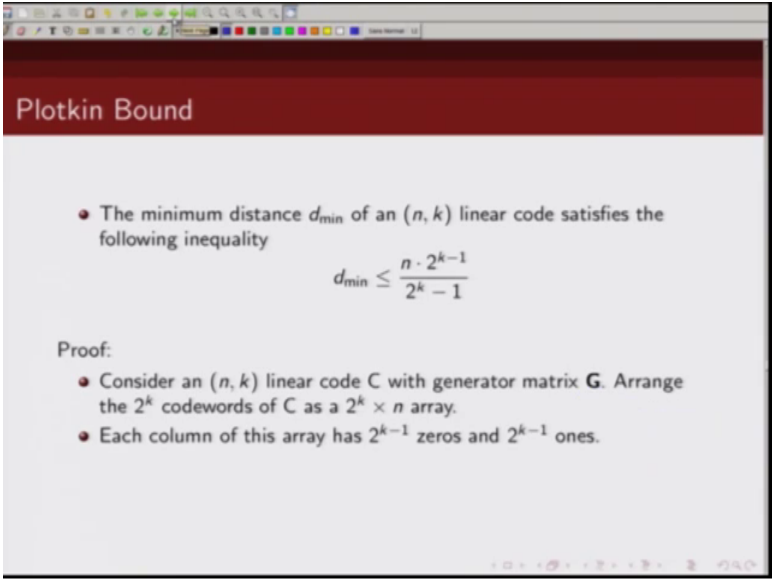
- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Proof:

- Consider an (n, k) linear code C with generator matrix $\underline{\mathbf{G}}$. Arrange the $\underline{2^k}$ codewords of C as a $2^k \times n$ array.

of this array.

(Refer Slide Time 15:07)



Plotkin Bound

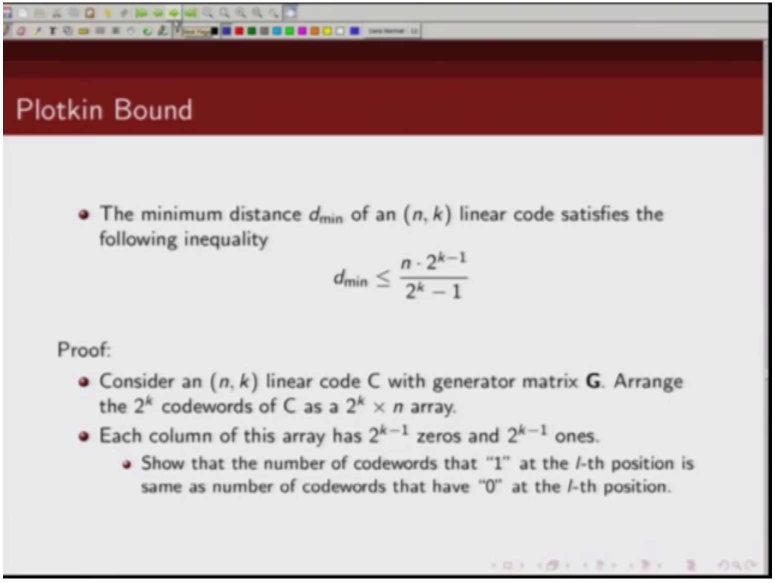
- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Proof:

- Consider an (n, k) linear code C with generator matrix $\mathbf{G}$. Arrange the 2^k codewords of C as a $2^k \times n$ array.
- Each column of this array has 2^{k-1} zeros and 2^{k-1} ones.

Now what we are going to show is each in this array, there are equal number of zeroes and equal number of 1's.

(Refer Slide Time 15:21)



Plotkin Bound

- The minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality

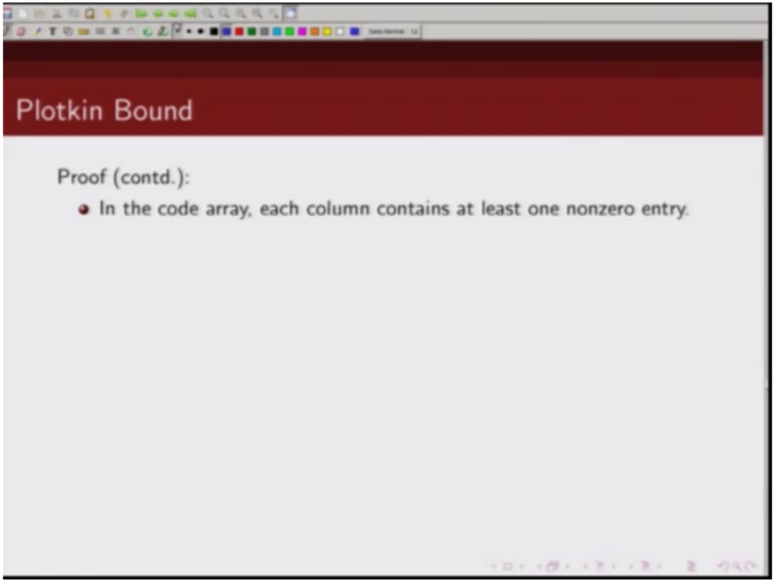
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

Proof:

- Consider an (n, k) linear code C with generator matrix **G**. Arrange the 2^k codewords of C as a $2^k \times n$ array.
- Each column of this array has 2^{k-1} zeros and 2^{k-1} ones.
  - Show that the number of codewords that "1" at the l-th position is same as number of codewords that have "0" at the l-th position.

So how do we show there are equal number of zeroes and equal number of 1's? We show it by showing that number of codewords that have 1 at the ith position is same as number of codewords that have zero at ith position. And in this way we will show that there are equal numbers of zeroes and 1s in this array.

(Refer Slide Time 15:46)



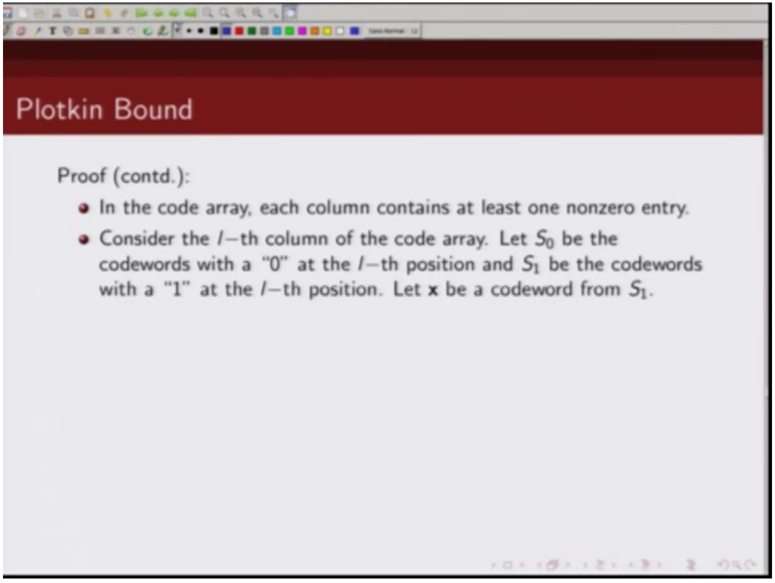
Plotkin Bound

Proof (contd.):

- In the code array, each column contains at least one nonzero entry.

So we have a code array where at least one non zero element.

(Refer Slide Time 15:51)



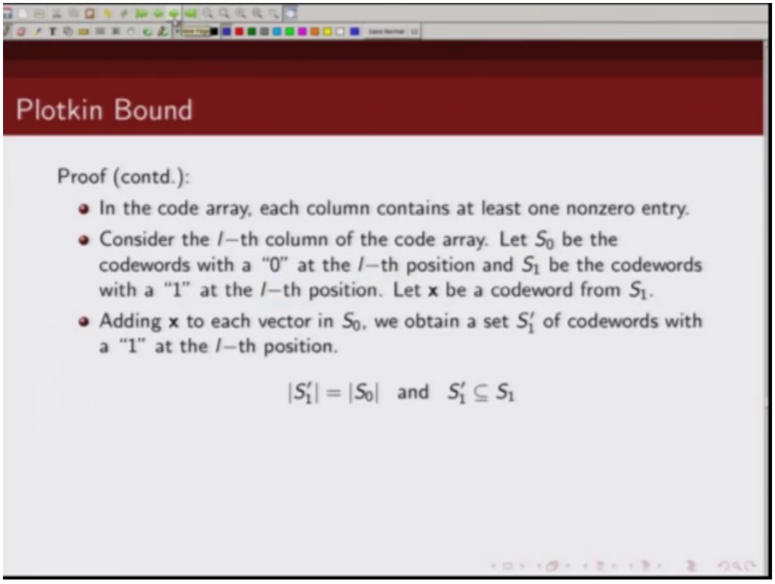
Plotkin Bound

Proof (contd.):

- In the code array, each column contains at least one nonzero entry.
- Consider the l−th column of the code array. Let S_0 be the codewords with a "0" at the l−th position and S_1 be the codewords with a "1" at the l−th position. Let $\mathbf{x}$ be a codeword from S_1.

Let us denote by S 0, the codewords that have zero at the ith location and S 1 as the set of codewords which have 1 at the ith location. And let us pick up a codeword x which belongs to this set S 1 which has 1 at the ith location.

(Refer Slide Time 16:19)



Plotkin Bound

Proof (contd.):

- In the code array, each column contains at least one nonzero entry.
- Consider the l−th column of the code array. Let S_0 be the codewords with a "0" at the l−th position and S_1 be the codewords with a "1" at the l−th position. Let $\mathbf{x}$ be a codeword from S_1.
- Adding $\mathbf{x}$ to each vector in S_0, we obtain a set S_1' of codewords with a "1" at the l−th position.

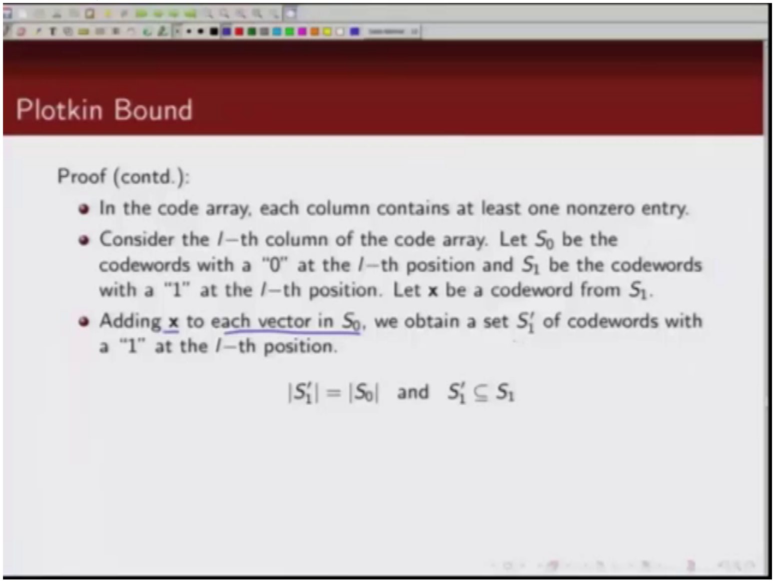
$$|S_1'| = |S_0| \quad \text{and} \quad S_1' \subseteq S_1$$

Now if we add x, if we add x to each vector in this set S 0 which has zero at the lth location and x, remember has 1 at lth location; if we add these 2 vectors,

(Refer Slide Time 16:36)

because both of them are codewords, some of them will be another valid codeword which will have 1 at the lth location. Why? Because x has 1 at lth location and this set S 0 has zero at the lth location. So we will get a new set

(Refer Slide Time 16:54)



Plotkin Bound

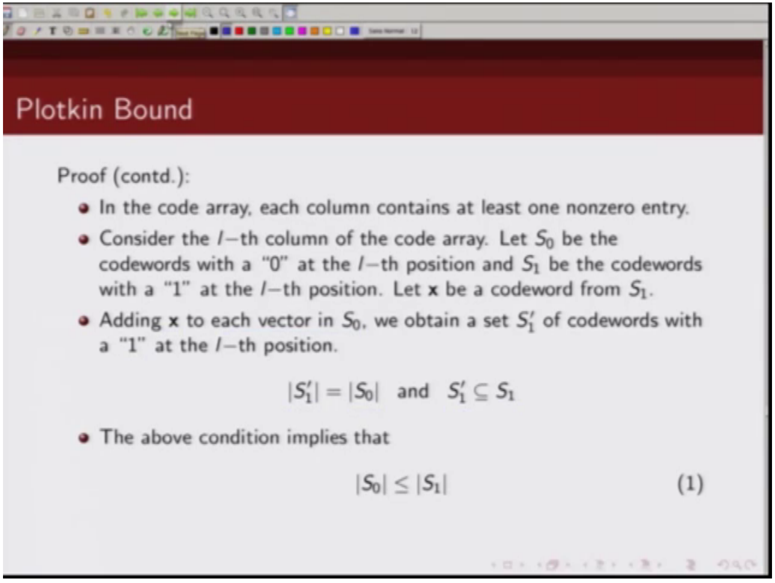
Proof (contd.):

- In the code array, each column contains at least one nonzero entry.
- Consider the $l-$th column of the code array. Let S_0 be the codewords with a "0" at the $l-$th position and S_1 be the codewords with a "1" at the $l-$th position. Let $\mathbf{x}$ be a codeword from S_1.
- Adding $\mathbf{x}$ to each vector in S_0, we obtain a set S_1' of codewords with a "1" at the $l-$th position.

$$|S_1'| = |S_0| \quad \text{and} \quad S_1' \subseteq S_1$$

of codewords, let's call it S prime, S 1 prime which will have 1 at the lth location. So clearly number of codewords which has 1 at that location is same as the original set S 0 and this set S1 prime is the subset of the set of codewords which has 1 at lth location. Now from this condition

(Refer Slide Time 17:31)



Plotkin Bound

Proof (contd.):

- In the code array, each column contains at least one nonzero entry.
- Consider the l−th column of the code array. Let S_0 be the codewords with a "0" at the l−th position and S_1 be the codewords with a "1" at the l−th position. Let $\mathbf{x}$ be a codeword from S_1.
- Adding $\mathbf{x}$ to each vector in S_0, we obtain a set S_1' of codewords with a "1" at the l−th position.

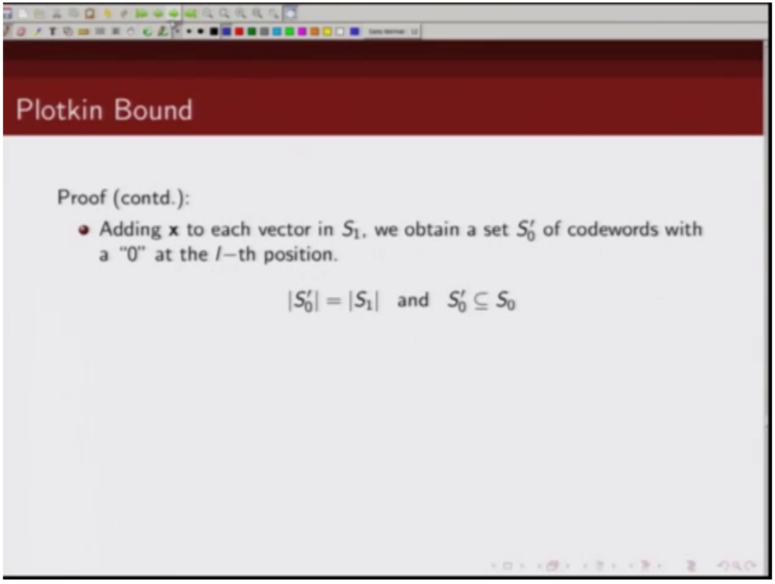
$$|S_1'| = |S_0| \quad \text{and} \quad S_1' \subseteq S_1$$

- The above condition implies that

$$|S_0| \leq |S_1| \qquad (1)$$

we can conclude that number of codewords which have zero at lth location is, must be less than equal to number of codewords which have 1 at lth location. Next what we do is

(Refer Slide Time 17:50)



Plotkin Bound

Proof (contd.):

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

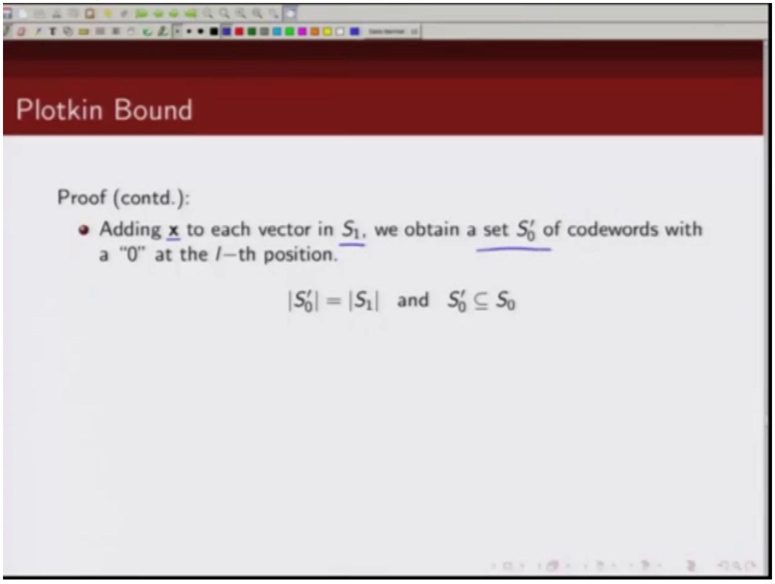
$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

we add the same vector x, x now, now to set S 1. Earlier we added them to set S 0. In the process we generated new set of codewords which has 1 at lth location. Now what we are doing is we are adding this codeword

(Refer Slide Time 18:07)

x to the set S 1. Now x has 1 at the lth location. S 1 has 1 at lth location. So when we add them together we get a new set of codewords which will have zero at lth location. So we get a new set of codewords

(Refer Slide Time 18:27)



Plotkin Bound

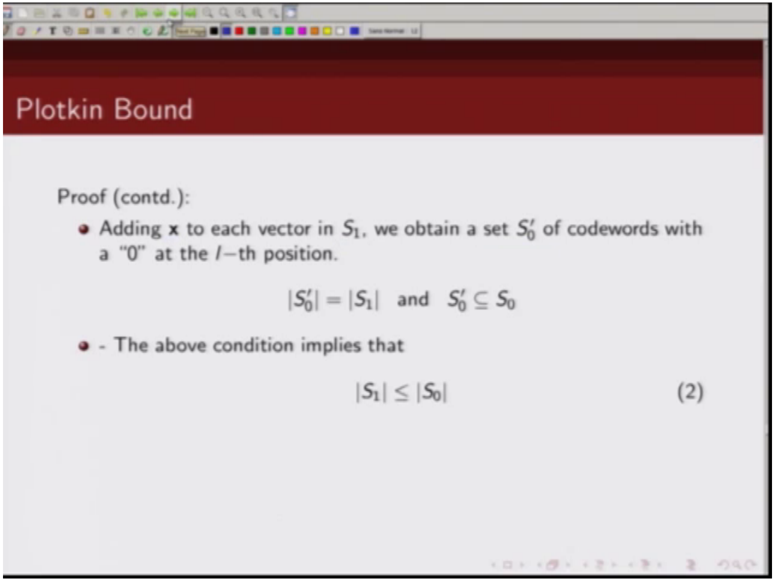
Proof (contd.):

- Adding $\underline{\mathbf{x}}$ to each vector in $\underline{S_1}$, we obtain a set $\underline{S_0'}$ of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

which we are denoting by S 0 prime which will have zero at the lth location. And number of codewords is, which are in this S o, S 0 prime will be same as number of codewords which have 1 at lth location. So from here we can and this S o prime will be subset of this S 0, the set which has zero at lth location. So from this we can conclude

(Refer Slide Time 19:02)



Plotkin Bound

Proof (contd.):

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

- - The above condition implies that

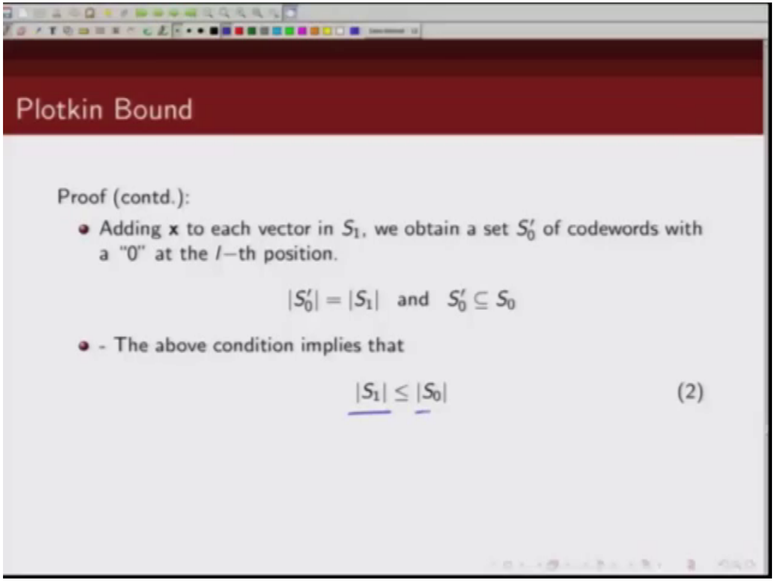
$$|S_1| \leq |S_0| \qquad (2)$$

then that number of codewords which have 1 at lth location

(Refer Slide Time 19:09)

is a subset of, is less than number of codewords which have zero at lth location.

(Refer Slide Time 19:17)



Plotkin Bound

Proof (contd.):

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

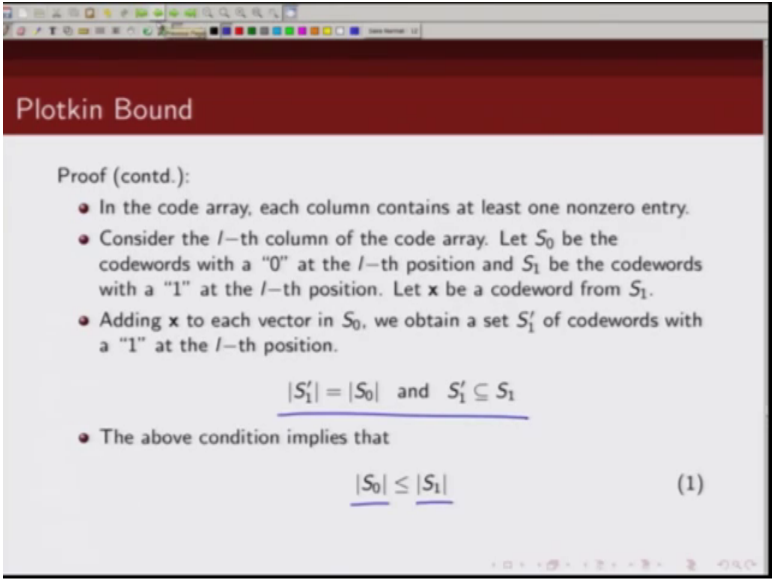
$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

- - The above condition implies that

$$\underline{|S_1|} \leq \underline{|S_0|} \qquad (2)$$

Now if we look at this relation 2 and relation 1,

(Refer Slide Time 19:23)



Plotkin Bound

Proof (contd.):

- In the code array, each column contains at least one nonzero entry.
- Consider the l−th column of the code array. Let S_0 be the codewords with a "0" at the l−th position and S_1 be the codewords with a "1" at the l−th position. Let $\mathbf{x}$ be a codeword from S_1.
- Adding $\mathbf{x}$ to each vector in S_0, we obtain a set S_1' of codewords with a "1" at the l−th position.

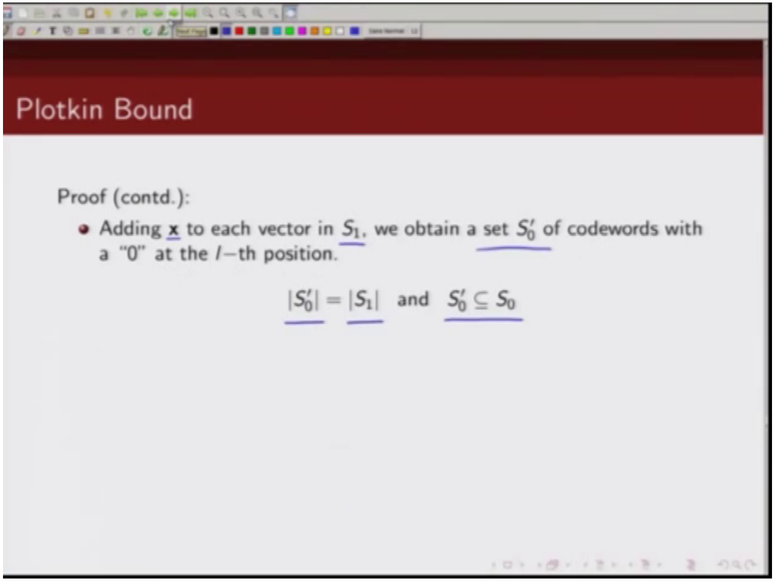
$$\underline{|S_1'| = |S_0| \quad \text{and} \quad S_1' \subseteq S_1}$$

- The above condition implies that

$$\underline{|S_0|} \leq \underline{|S_1|} \qquad (1)$$

here we got the condition that number of codewords which are zero at the lth location is less than number of codewords which have 1 at lth location.

(Refer Slide Time 19:34)



Plotkin Bound

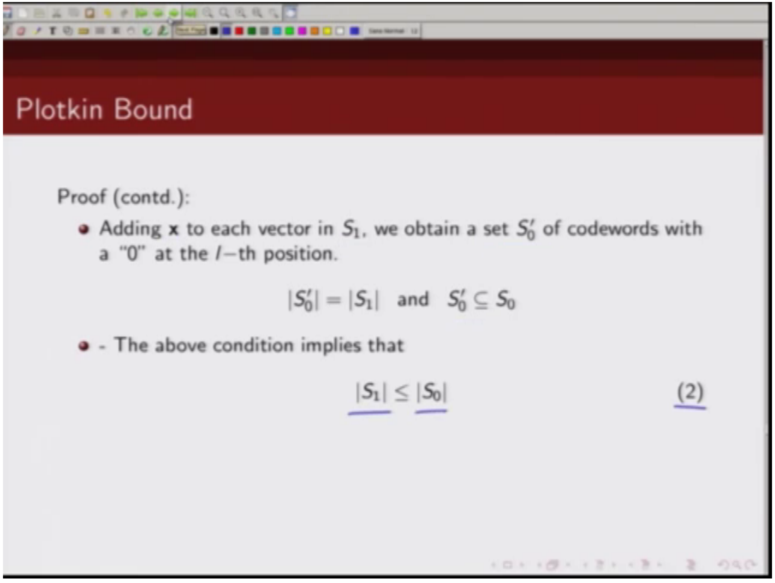
Proof (contd.):

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

And here we got this condition that

(Refer Slide Time 19:36)



Plotkin Bound

Proof (contd.):

- Adding $\mathbf{x}$ to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the $l-$th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

- - The above condition implies that

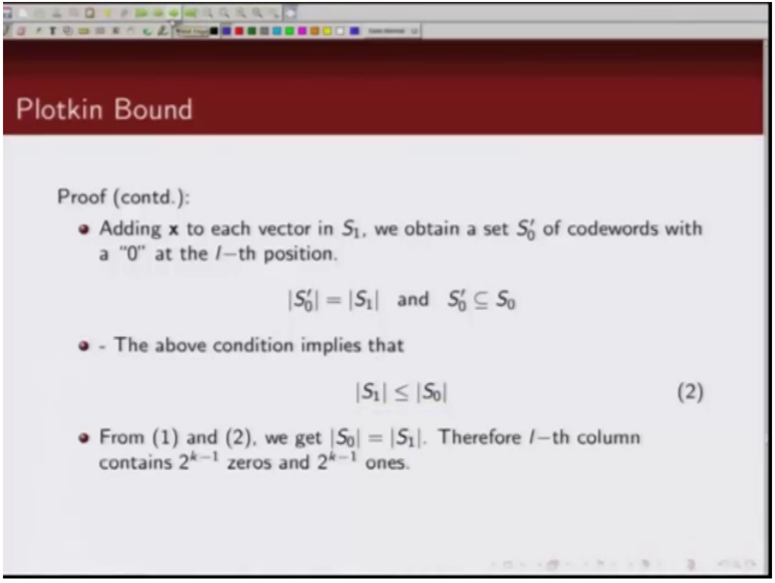
$$|S_1| \leq |S_0| \qquad (2)$$

number of codewords which have 1 at lth location is less than equal to number of codewords which have zero at lth location. Now these two conditions will be simultaneously satisfied only if

(Refer Slide Time 19:51)

both are same. So from 1 and 2

(Refer Slide Time 19:55)



Plotkin Bound

Proof (contd.):

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

- - The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

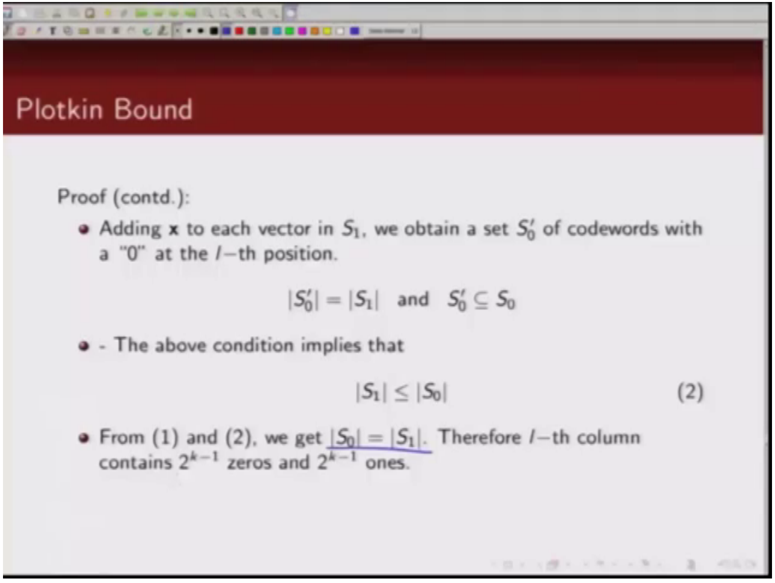
- From (1) and (2), we get $|S_0| = |S_1|$. Therefore l−th column contains 2^{k-1} zeros and 2^{k-1} ones.

we can conclude that number of codewords that has zero at lth location is same as number of codewords which have 1 at lth location. So then you have a 2 k, 2 k cross n,

(Refer Slide Time 20:14)

this array. So half of them basically are zero, half of them are 1. So you have total,

(Refer Slide Time 20:22)



Plotkin Bound

Proof (contd.):

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \quad \text{and} \quad S_0' \subseteq S_0$$

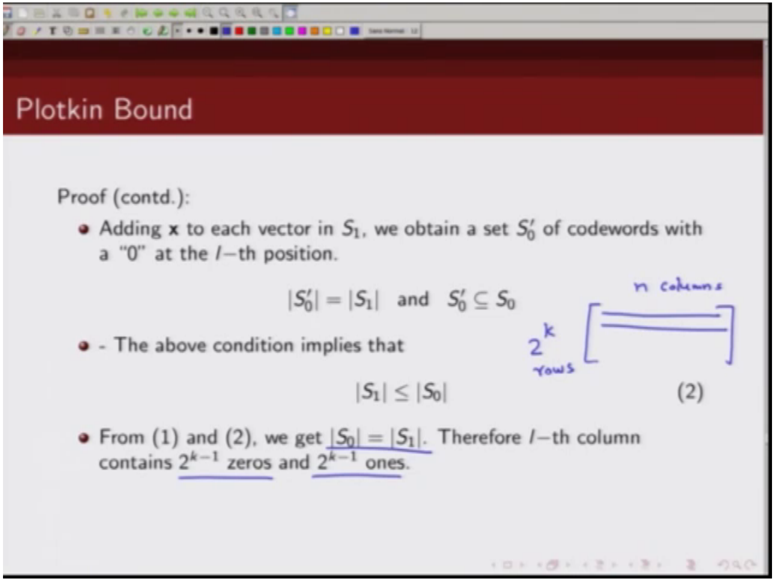
- - The above condition implies that

$$|S_1| \le |S_0| \qquad (2)$$

- From (1) and (2), we get $\underline{|S_0| = |S_1|}$. Therefore l−th column contains 2^{k-1} zeros and 2^{k-1} ones.

so each column will have 2 k minus 1 zeroes and 2 k minus 1 1's. And how many such columns are there? We are talking about code array. So we are talking about, so these are all codewords. These are all codewords. So each one of them are n bit. So we have total n columns and we have 2 raised to power k rows. So what we have shown is,

(Refer Slide Time 20:57)



Plotkin Bound

Proof (contd.):

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \text{ and } S_0' \subseteq S_0$$

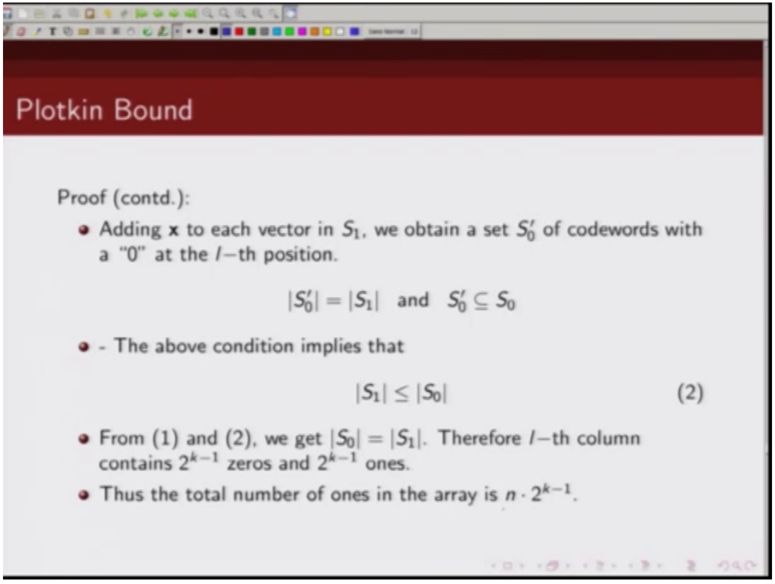
- - The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

- From (1) and (2), we get $\underline{|S_0| = |S_1|}$. Therefore l−th column contains $\underline{2^{k-1} \text{ zeros}}$ and $\underline{2^{k-1} \text{ ones}}$.

each row will have 2 k minus 1 zeroes and 2 k minus 1 1's and there are total n such columns. So total number of 1's is how much? n times 2 k minus 1. So what we

(Refer Slide Time 21:19)



Plotkin Bound

Proof (contd.):

- Adding **x** to each vector in S_1, we obtain a set S_0' of codewords with a "0" at the l−th position.

$$|S_0'| = |S_1| \text{ and } S_0' \subseteq S_0$$

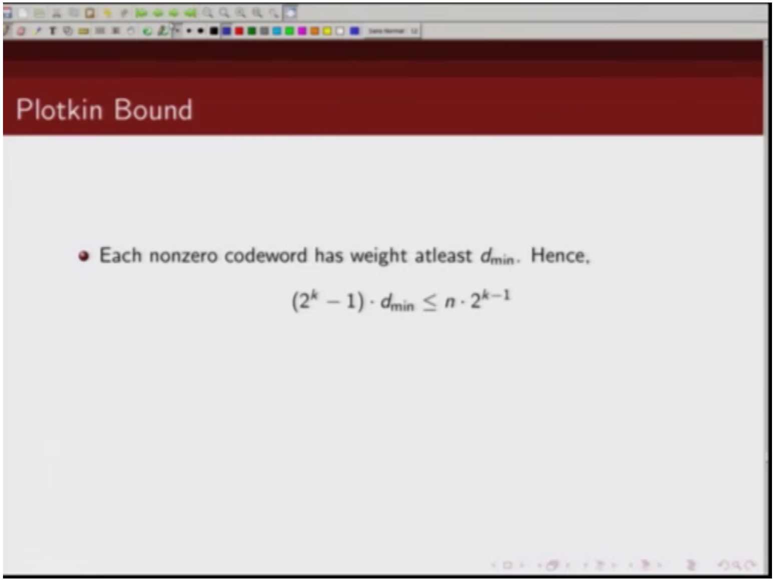
- - The above condition implies that

$$|S_1| \leq |S_0| \qquad (2)$$

- From (1) and (2), we get $|S_0| = |S_1|$. Therefore l−th column contains 2^{k-1} zeros and 2^{k-1} ones.
- Thus the total number of ones in the array is $n \cdot 2^{k-1}$.

have shown so far is number of 1's in this array is n into 2 raised to power k minus 1. Now

(Refer Slide Time 21:30)



Plotkin Bound

- Each nonzero codeword has weight atleast d_{min}. Hence,

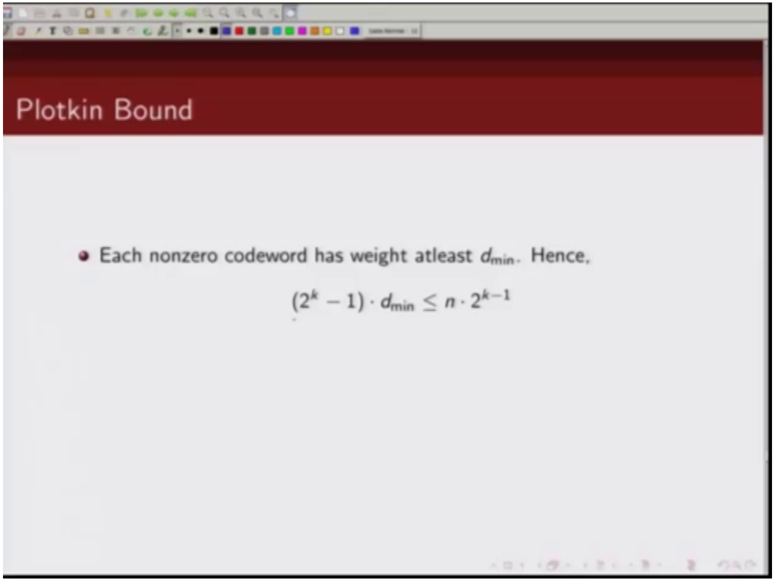
$$(2^k - 1) \cdot d_{\text{min}} \leq n \cdot 2^{k-1}$$

what was this array made of? This array consists of 2 k codewords, right? So out of those 2 k codewords, one of them will be all zero codeword because we are talking of linear codes. So remaining 2 k minus 1 codewords

(Refer Slide Time 21:47)

will have minimum distance, minimum weight, minimum distance is equal to minimum weight of the codeword. So the minimum weight of the codeword is at least d min right? So number of non zero

(Refer Slide Time 22:01)



Plotkin Bound

- Each nonzero codeword has weight atleast $d_{\min}$. Hence,

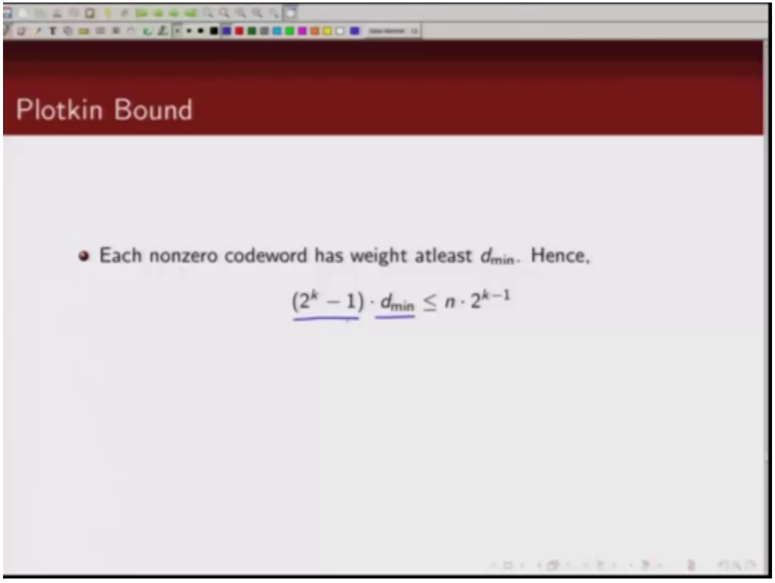
$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

codewords is given by this and each one of them will have weight at least equal to d min because d min is the minimum distance of the code. So the minimum weight of a non zero

(Refer Slide Time 22:14)

codeword must be at least d min. So 2 k minus 1 which is number of

(Refer Slide Time 22:20)



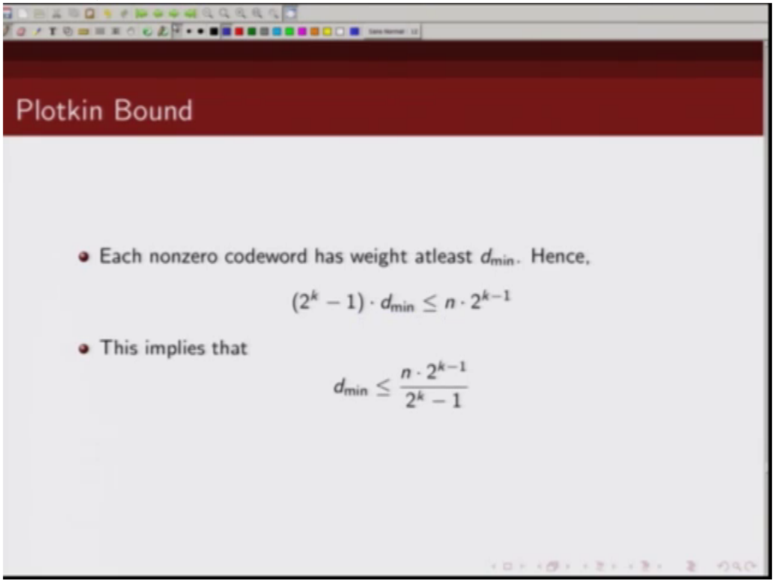
Plotkin Bound

- Each nonzero codeword has weight atleast $d_{\min}$. Hence,

$$\underline{(2^k - 1)} \cdot \underline{d_{\min}} \leq n \cdot 2^{k-1}$$

non zero codewords multiplied by weight of, minimum weight of a codeword; this number should be less than number of 1s in this array. And what is number of 1's in this array? That is given by n into 2 raised to power k minus 1, so from this we get this relationship

(Refer Slide Time 22:44)



Plotkin Bound

- Each nonzero codeword has weight atleast $d_{\min}$. Hence,

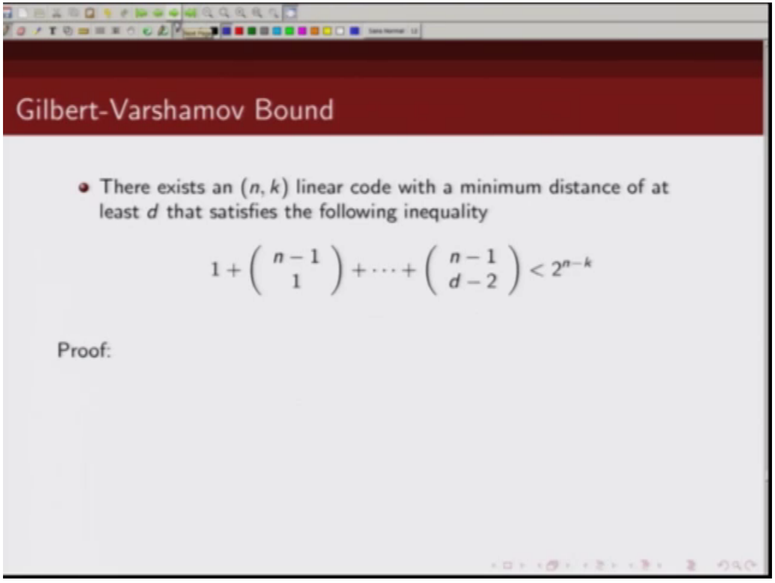
$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

- This implies that

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

that minimum distance of the code is upper bounded by this and this is known as Plotkin bound.

(Refer Slide Time 22:55)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

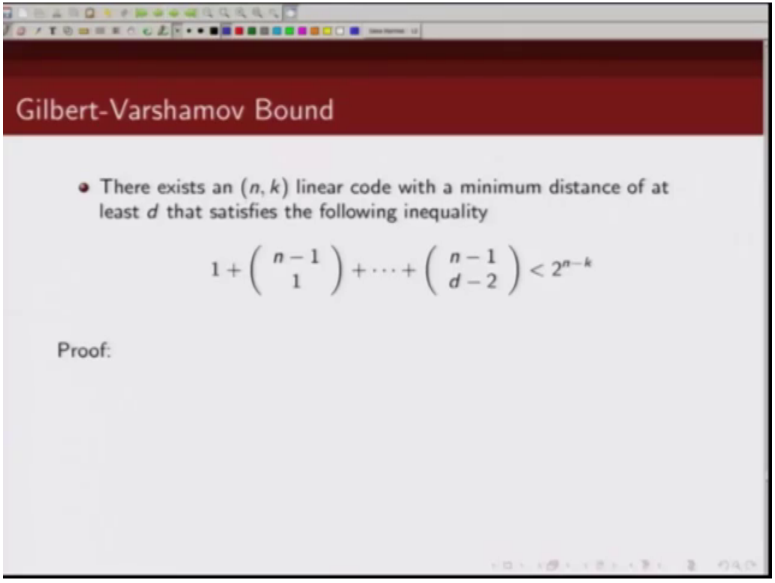
Proof:

Finally we conclude this lecture with Gilbert-Varshamov bound. So what does Gilbert-Varshamov bound says?

(Refer Slide Time 23:04)

If you have a

(Refer Slide Time 23:06)



Gilbert-Varshamov Bound

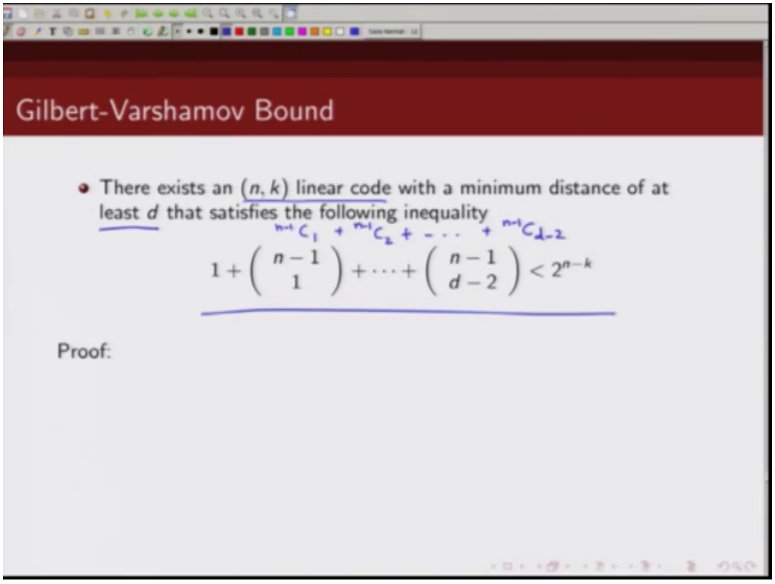
- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1+\binom{n-1}{1}+\cdots+\binom{n-1}{d-2}<2^{n-k}$$

Proof:

n k linear code whose minimum distance is at least d then following inequality must be satisfied and what is this inequality says 1 plus n minus 1 C 1 plus n minus 1 C 2 plus up to n minus 1 C d minus 2 should be less than

(Refer Slide Time 23:34)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

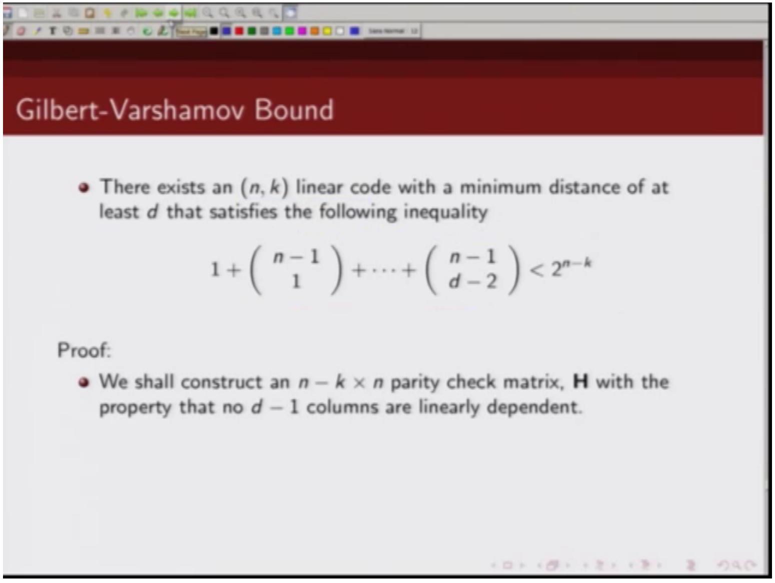
$${}^{n-1}C_1 + {}^{n-1}C_2 + \cdots + {}^{n-1}C_{d-2}$$

$$1+\binom{n-1}{1}+\cdots+\binom{n-1}{d-2}<2^{n-k}$$

Proof:

2 n minus k. So let us prove

(Refer Slide Time 23:40)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

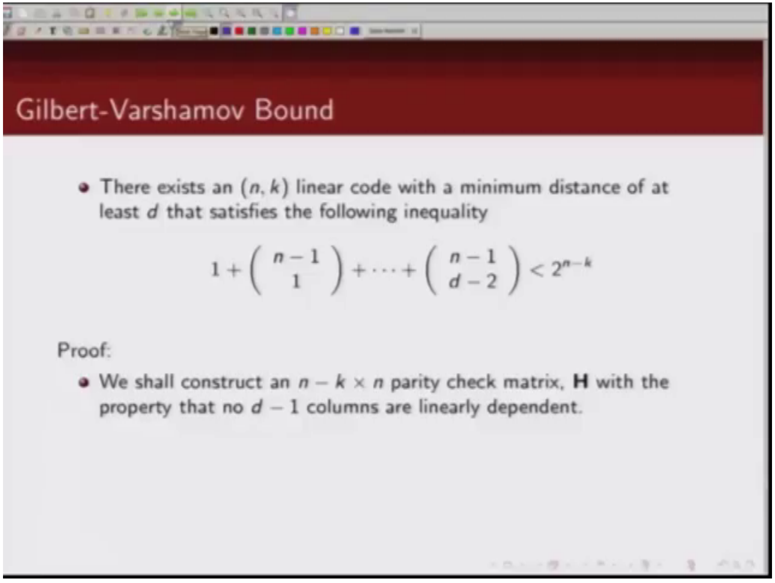
- We shall construct an $n - k \times n$ parity check matrix, $\mathbf{H}$ with the property that no $d - 1$ columns are linearly dependent.

this result. So we know how is minimum distance of the code related to the columns of the parity check matrix. Now if

(Refer Slide Time 23:55)

the minimum distance is at least d, then we know uh d minus 1 columns are linearly independent. So no combinations of d minus 1 columns of this parity check matrix will be linearly dependent, right? So let us construct

(Refer Slide Time 24:19)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

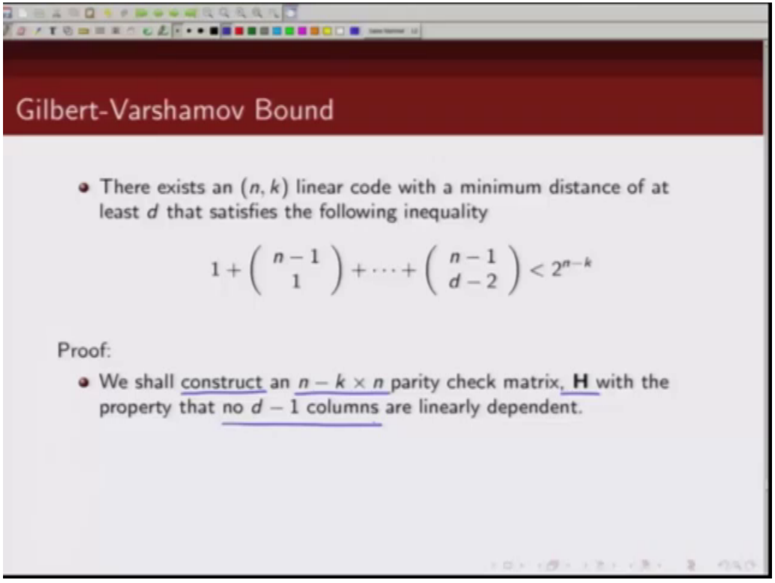
- We shall construct an $n - k \times n$ parity check matrix, $\mathbf{H}$ with the property that no $d - 1$ columns are linearly dependent.

a parity check matrix H which is an n minus k cross n matrix. Now we will try to construct this parity check matrix such that no d minus 1 columns are linearly dependent. Now if we can ensure that no d minus 1 columns are linearly dependent then we are ensuring that minimum distance is at least d.

(Refer Slide Time 24:46)

And that's what we want to show that if the minimum distance

(Refer Slide Time 24:50)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

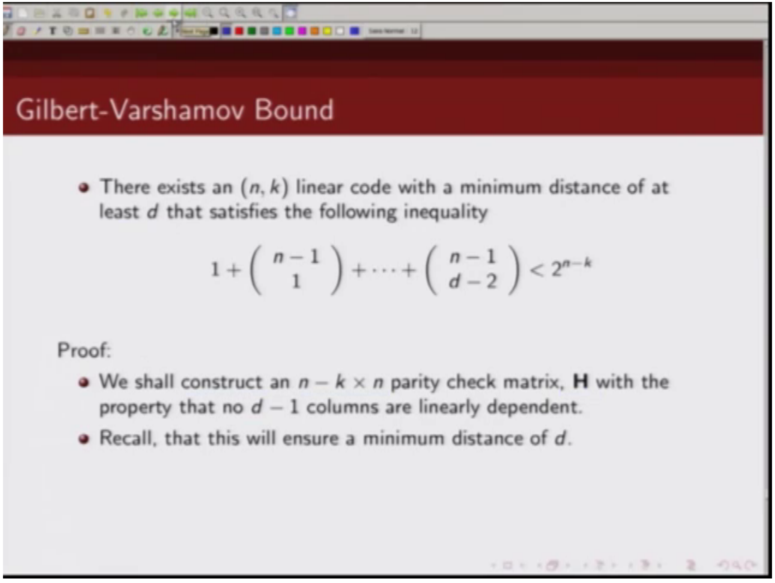
$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

- We shall $\underline{\text{construct}}$ an $\underline{n-k \times n}$ parity check matrix, $\underline{\mathbf{H}}$ with the property that $\underline{\text{no } d-1 \text{ columns}}$ are linearly dependent.

is at least d then this condition has to be satisfied. So

(Refer Slide Time 24:56)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

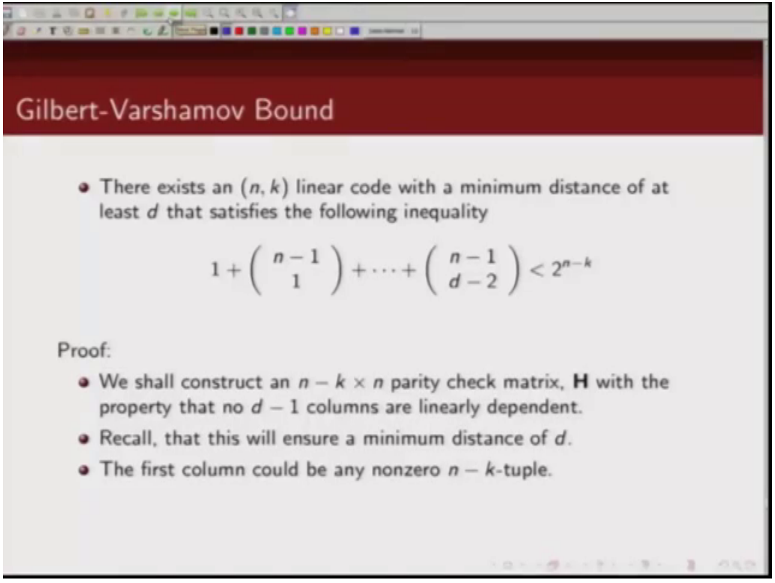
- We shall construct an $n-k \times n$ parity check matrix, $\mathbf{H}$ with the property that no $d-1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.

how do we construct this parity check matrix H such that no

(Refer Slide Time 25:02)

d minus 1 columns, any combinations of up to d minus 1 columns do not add up to zero.

(Refer Slide Time 25:11)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

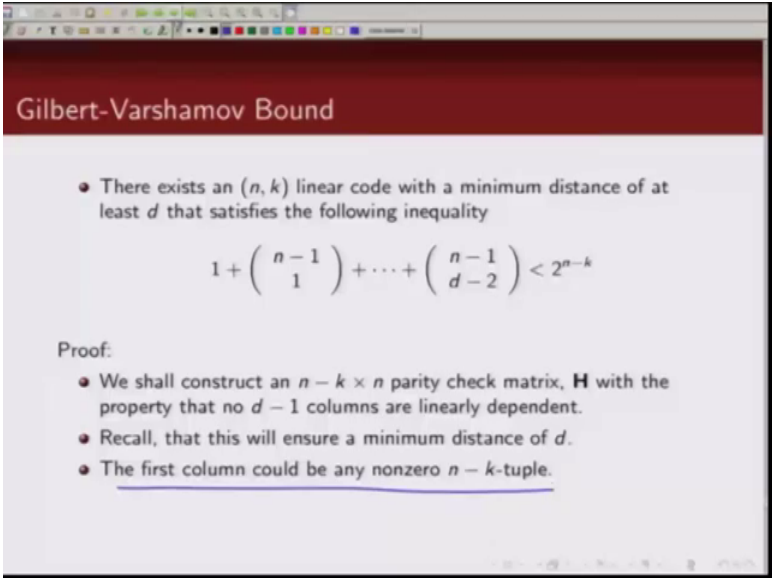
- We shall construct an $n - k \times n$ parity check matrix, $\mathbf{H}$ with the property that no $d - 1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.
- The first column could be any nonzero $n - k$-tuple.

So let's start with first column. Now first column of this parity check matrix would be any n minus 1, n minus k tuple, any non zero n minus k tuple, it could be one all zeros,

(Refer Slide Time 25:26)

zero one all zeroes, zero one all zeroes or whatever. It could be any non zero n minus k tuple.

(Refer Slide Time 25:32)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

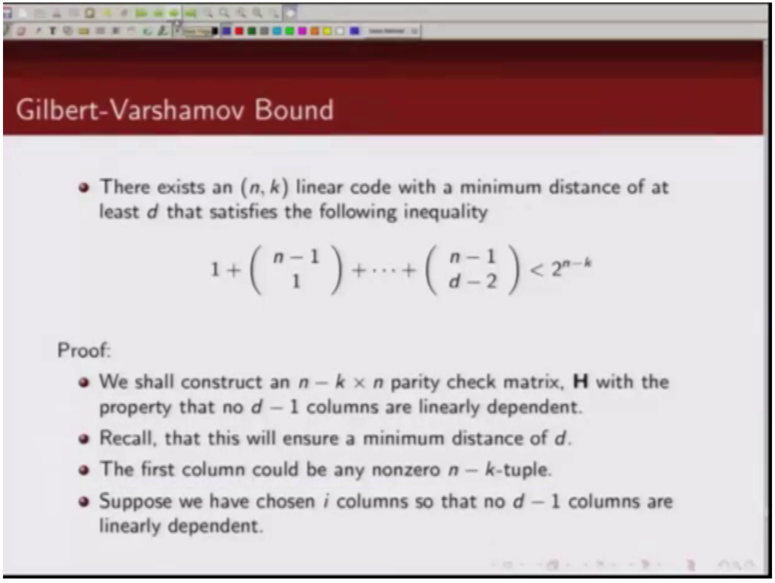
$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

- We shall construct an $n - k \times n$ parity check matrix, **H** with the property that no $d - 1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.
- The first column could be any nonzero $n - k$-tuple.

And let us

(Refer Slide Time 25:34)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

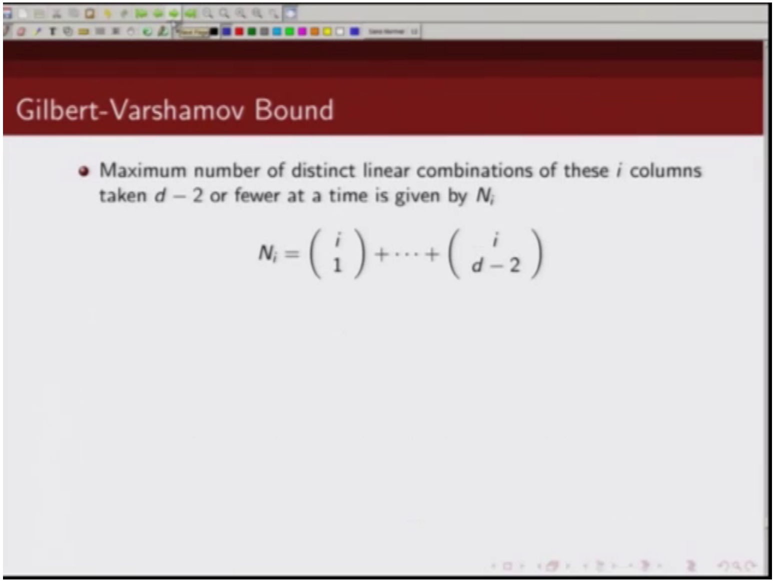
$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

- We shall construct an $n - k \times n$ parity check matrix, $\mathbf{H}$ with the property that no $d - 1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.
- The first column could be any nonzero $n - k$-tuple.
- Suppose we have chosen i columns so that no $d - 1$ columns are linearly dependent.

assume that we have chosen i columns of this parity check matrix such that no d minus 1 columns are linearly dependent. Now

(Refer Slide Time 25:46)



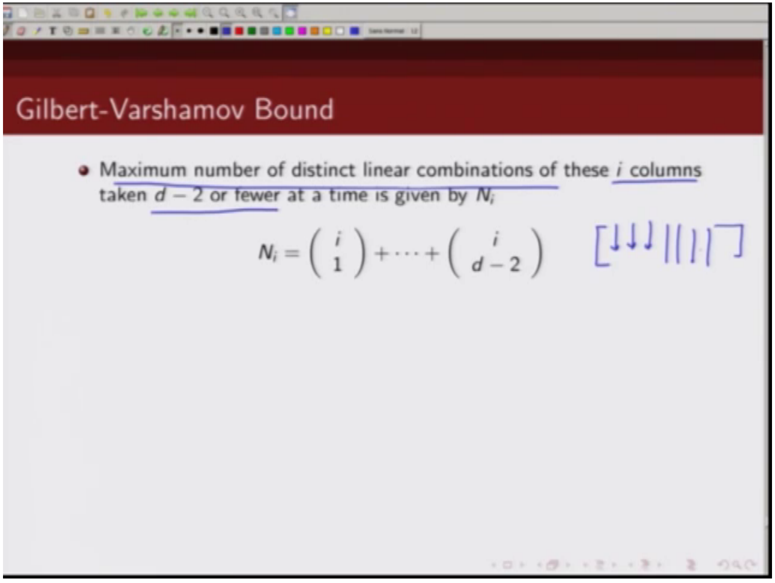
Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d - 2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

if you want to do that, basically we have to ensure that; so let's look at maximum number of distinct linear combinations of i columns taken d minus 2 or fewer at a time. So what we are doing is, so we have this H matrix, right and these are the columns of the H matrix basically. These are columns of this H matrix. Let's say we have constructed

(Refer Slide Time 26:16)



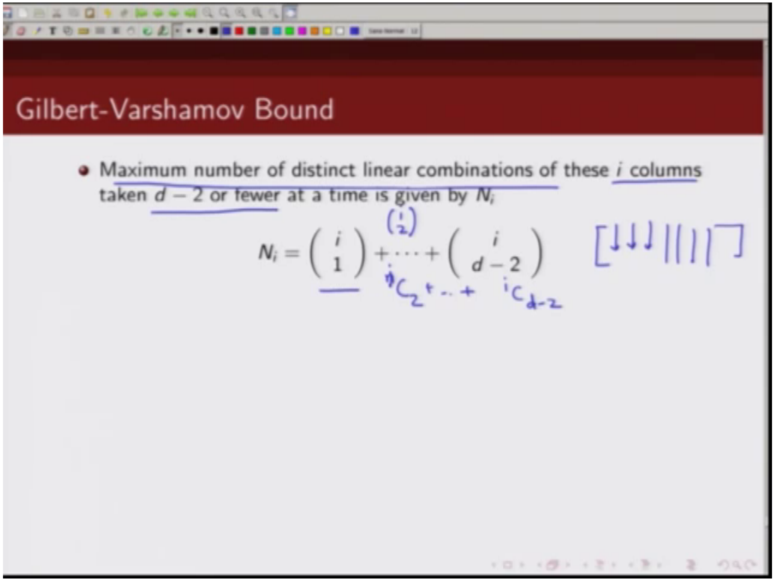
Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2} \qquad [\downarrow\downarrow\downarrow\,||||\,]$$

i columns. Now what we are doing is, we are taking linear combinations of one column, two columns, three columns, four columns up to d minus 2 columns, right. So we are taking linear combination of these i columns taken d minus 2 or fewer at a time and we are trying to find out how many such linear combination exist. So if I take i columns taken one at a time, this is the number that I get, i C 1. If i take i columns taken 2 at a time, I get n C, i C 2. Like that if I consider i columns taken d minus 2 at a time I get i C d minus 2.

(Refer Slide Time 27:07)



Gilbert-Varshamov Bound

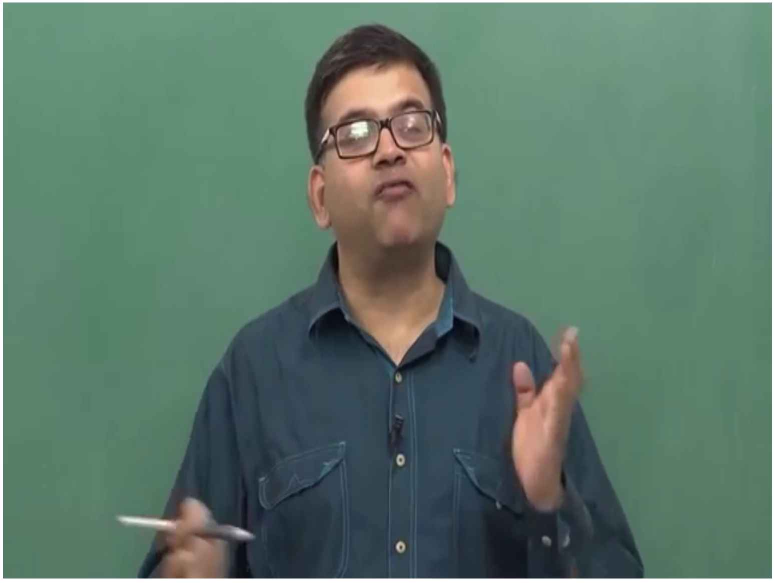
- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \underline{\binom{i}{1}} + \overset{\binom{i}{2}}{\cdots} + \binom{i}{d-2} \qquad [\downarrow\downarrow\downarrow\,||||\,]$$

$${}^{i}C_2 + \cdots + {}^{i}C_{d-2}$$

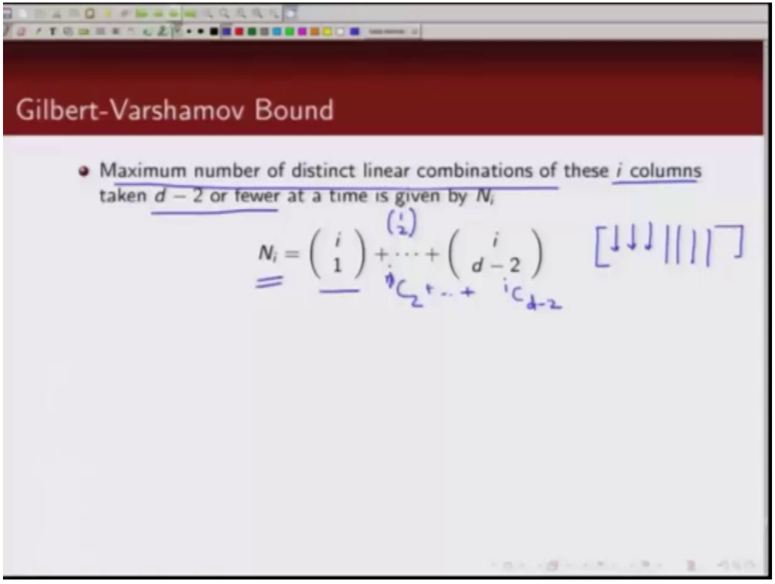
So n i will give me number of linear

(Refer Slide Time 27:12)



possible, linear combinations of these i columns taken one at a time, two at a time, three at a time up to d minus 2 at a time.

(Refer Slide Time 27:22)



Gilbert-Varshamov Bound

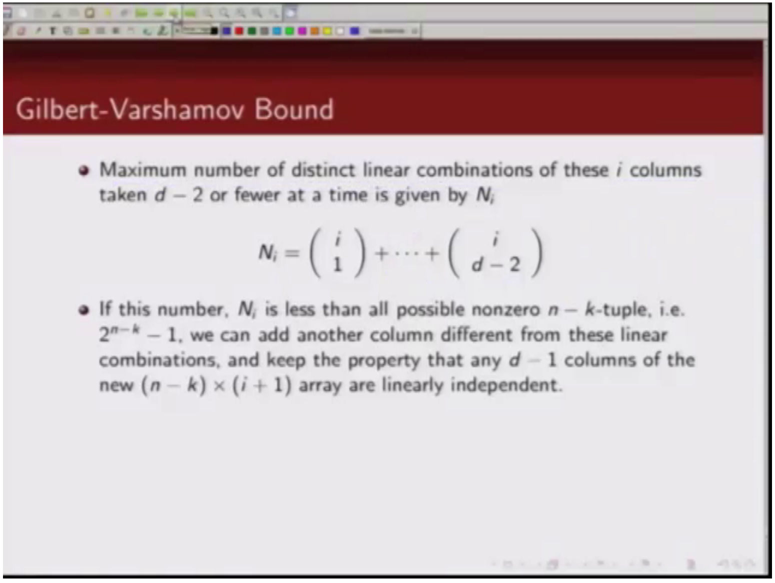
- Maximum number of distinct linear combinations of these *i* columns taken *d* − 2 or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \overset{\binom{i}{2}}{\cdots} + \binom{i}{d-2}$$

$$= {}^iC_2 + \dots + {}^iC_{d-2}$$

Now

(Refer Slide Time 27:23)



Gilbert-Varshamov Bound

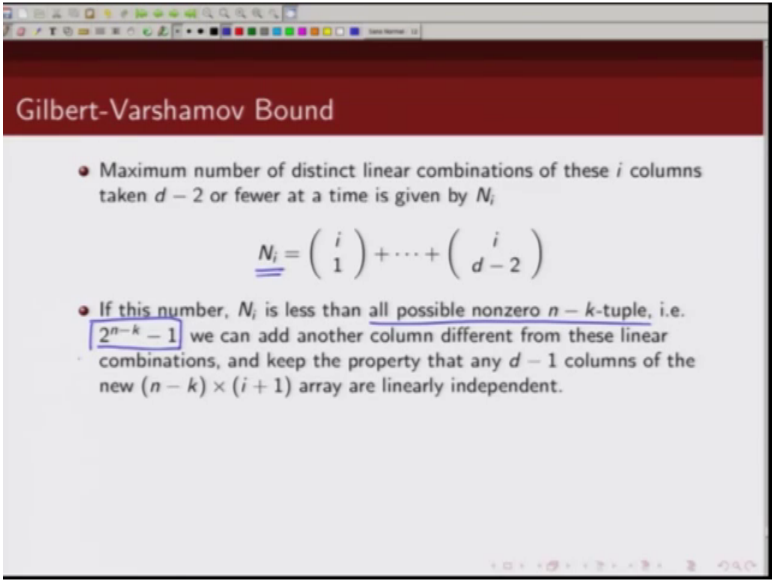
- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.

if this number n i which we just computed is less than all possible n minus k n tuple, non zero n minus k n tuple. Now how many non zero n minus k tuple we have? We have total 2 raised to power n minus k minus 1 because one of them will be all zero n tuple so these many number of n, uh non zero n tuples we have. Now if this number n i is less than this number what does it mean?

(Refer Slide Time 28:00)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$\underline{\underline{N_i}} = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

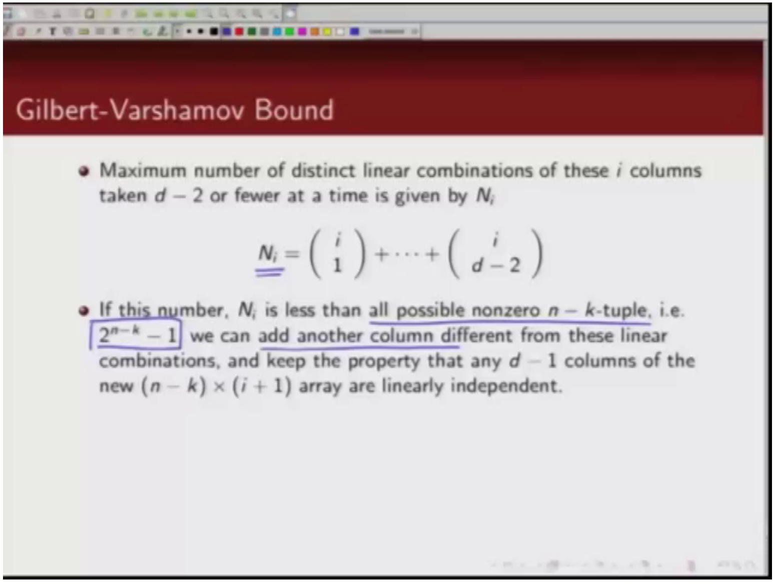
- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $\boxed{2^{n-k}-1}$ we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.

We can find another n tuple which would be linearly independent of any of these columns. So if this number n i is less than 2 raised to power n minus k minus 1, it means we can add another column which is different from any

(Refer Slide Time 28:26)

of the linear combinations of these i columns taken one at a time, two at a time, three at a time, d minus 2 at a time, right and this will still ensure that d minus 1 columns of this parity check matrix would not add up to zero. In other words they are linearly independent. So if we can ensure that this

(Refer Slide Time 28:53)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$\underline{\underline{N_i}} = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

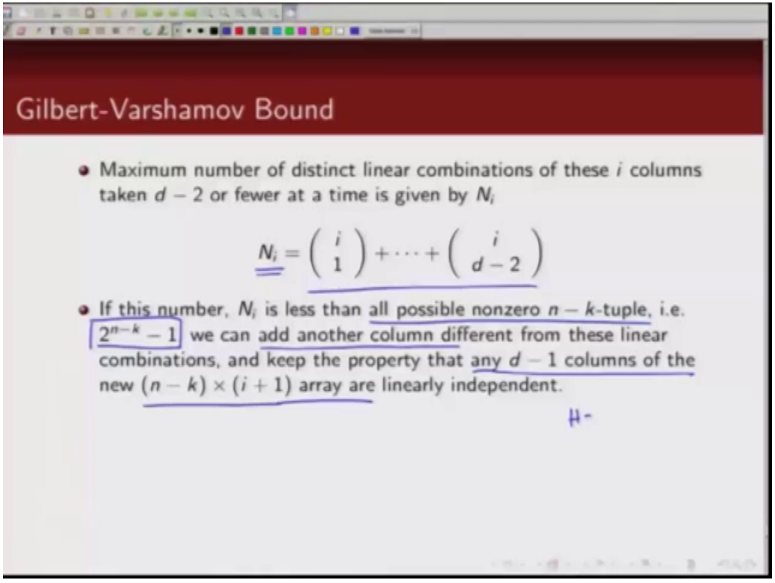
- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $\boxed{2^{n-k}-1}$ we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.

number n i is less than 2 n minus k minus 1, then we can add another n minus k column which would be different from any linear combinations which is basically counted here. And this will keep the property that any d minus 1 columns of this newly constructed H matrix where we are adding this column, this would not add up to zero. So this new matrix n minus k cross i plus 1 matrix will be linearly independent. Now we have to ensure that this property holds for i, all i's,

(Refer Slide Time 29:43)

i up to n because we have, when we are constructing the parity check matrix we have to construct

(Refer Slide Time 29:51)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

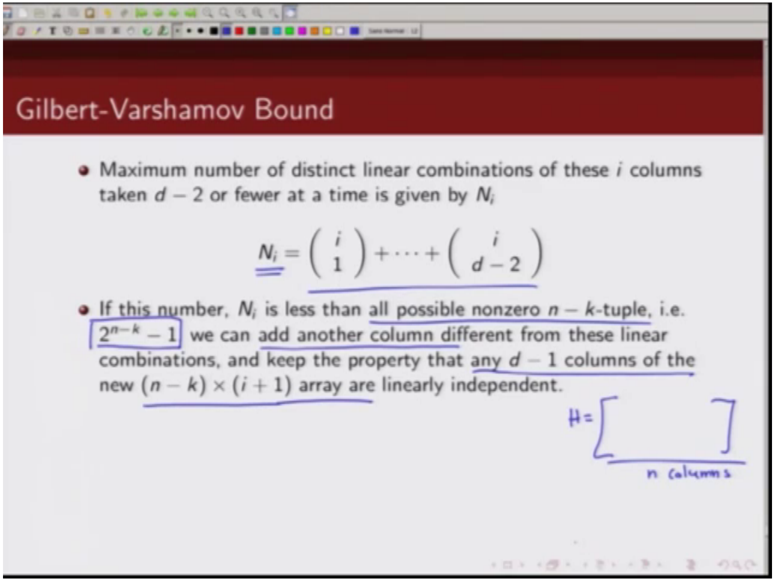
$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $\boxed{2^{n-k}-1}$ we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.

H-

total n columns, the number of columns are n, n columns. So this property I mentioned here, it should hold for i equal to n.

(Refer Slide Time 30:05)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

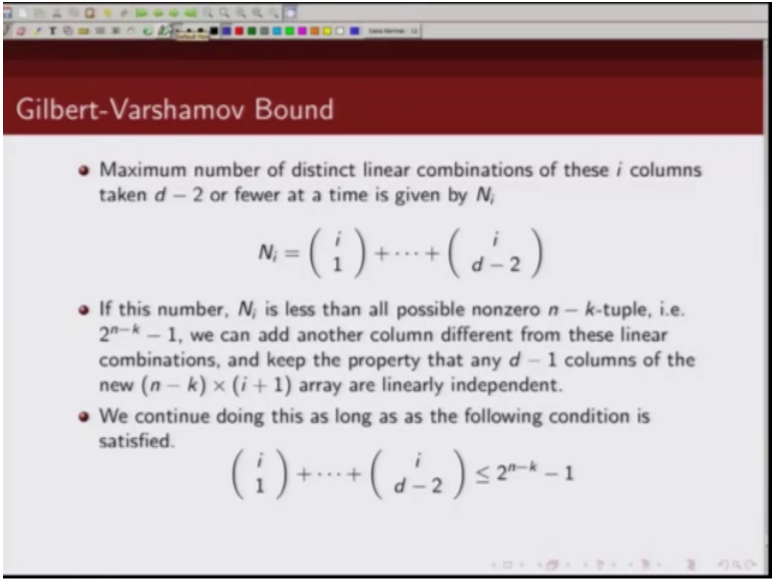
- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $\boxed{2^{n-k}-1}$ we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k)\times(i+1)$ array are linearly independent.

$H = [\quad]$

n columns

So we continue

(Refer Slide Time 30:08)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

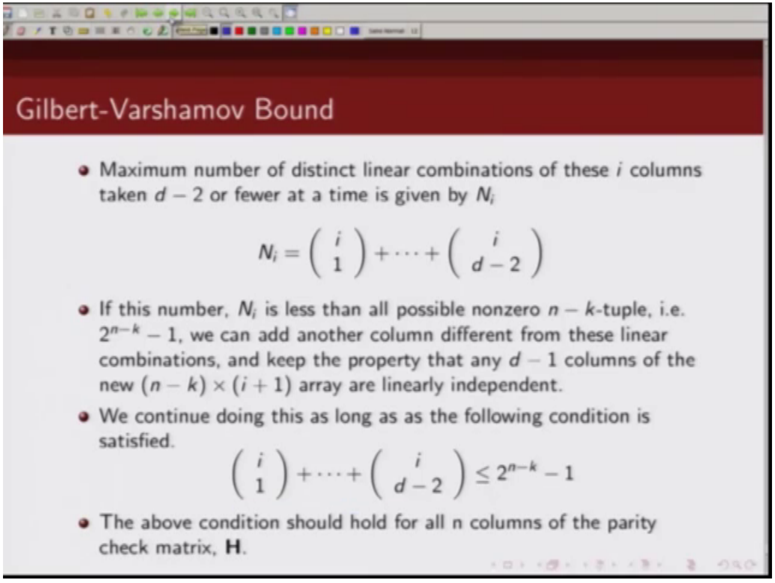
$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k)\times(i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

$$\binom{i}{1} + \cdots + \binom{i}{d-2} \leq 2^{n-k}-1$$

doing this. So what we want is this number n i should be less than 2 raised to power n minus k minus 1 and this should be true for, this should be true for

(Refer Slide Time 30:23)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

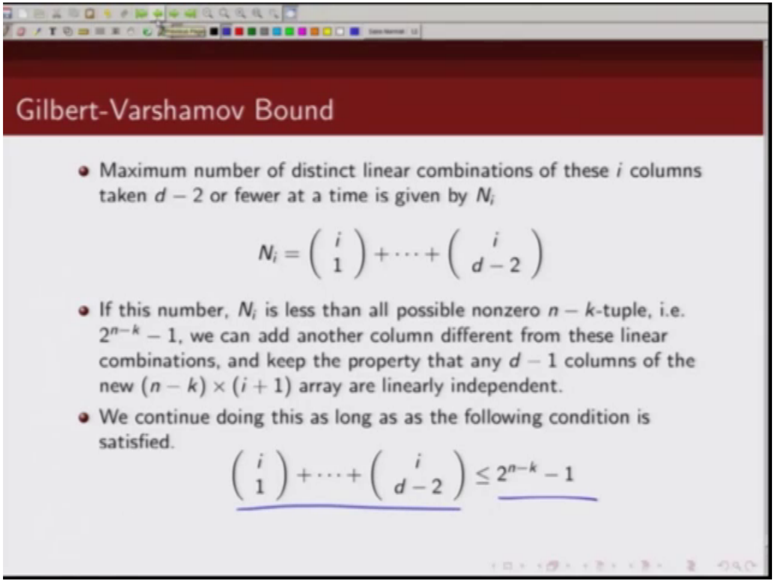
- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

$$\binom{i}{1} + \cdots + \binom{i}{d-2} \leq 2^{n-k}-1$$

- The above condition should hold for all n columns of the parity check matrix, **H**.

all the n columns of the parity check matrix. Hence this condition should be satisfied for all the n columns and that essentially proves our result that

(Refer Slide Time 30:36)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

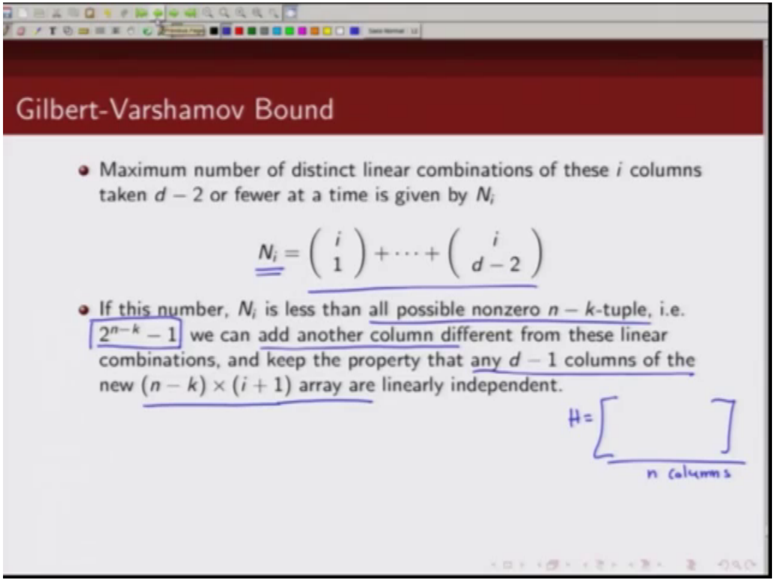
$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

$$\underline{\binom{i}{1} + \cdots + \binom{i}{d-2}} \leq \underline{2^{n-k}-1}$$

uh if we take,

(Refer Slide Time 30:40)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

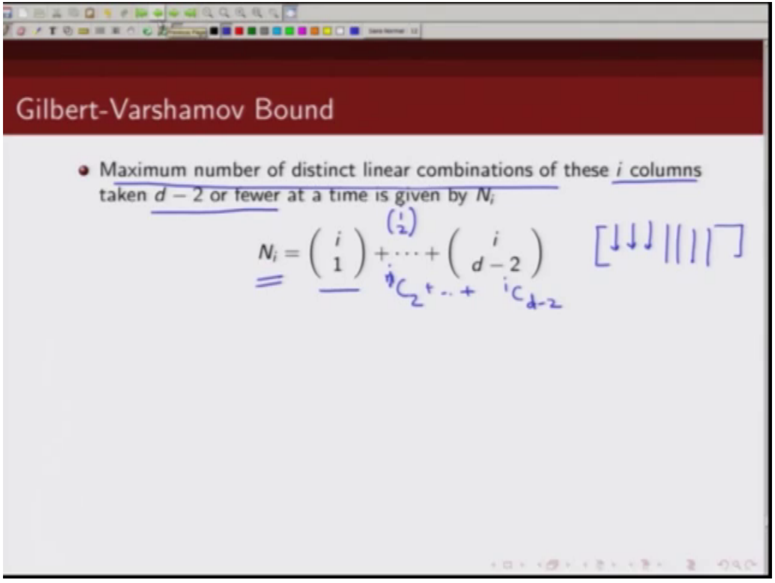
$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $\boxed{2^{n-k}-1}$ we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k)\times(i+1)$ array are linearly independent.

$$H = \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{n \text{ columns}}$$

so if you go back

(Refer Slide Time 30:42)



Gilbert-Varshamov Bound

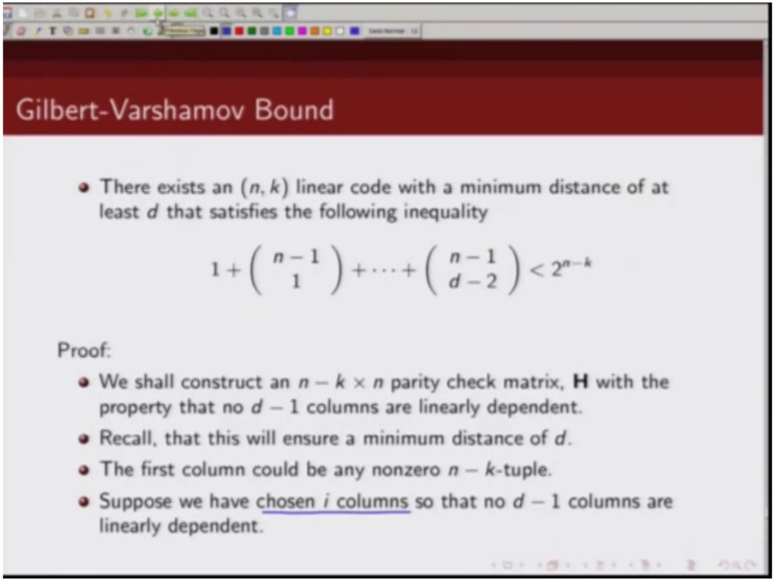
- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \overset{\binom{i}{2}}{\cdots} + \binom{i}{d-2}$$

$${}^iC_2 + .. + {}^iC_{d-2}$$

here, so if I

(Refer Slide Time 30:44)



Gilbert-Varshamov Bound

- There exists an (n, k) linear code with a minimum distance of at least d that satisfies the following inequality

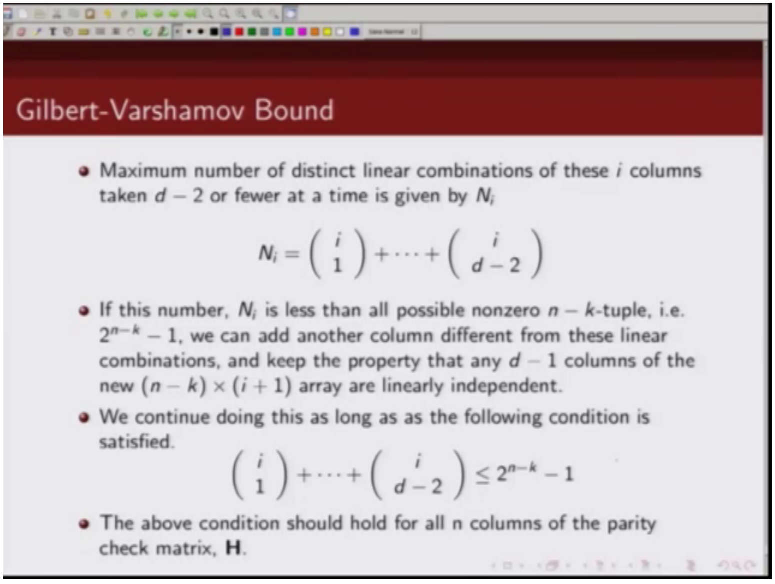
$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

- We shall construct an $n-k \times n$ parity check matrix, **H** with the property that no $d-1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.
- The first column could be any nonzero $n-k$-tuple.
- Suppose we have chosen i columns so that no $d-1$ columns are linearly dependent.

put i equal to n minus 1, right, if I put i equal to n minus 1, let's go back here. If I put i, this should hold for

(Refer Slide Time 30:54)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

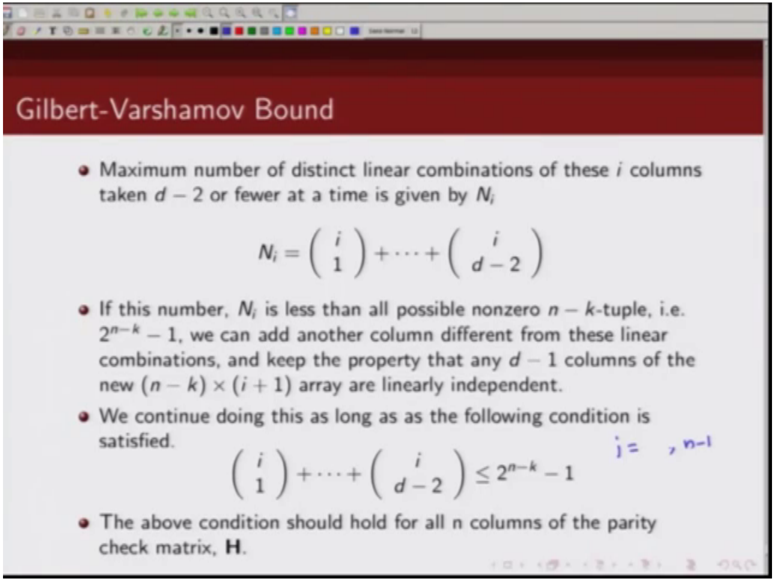
- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

$$\binom{i}{1} + \cdots + \binom{i}{d-2} \leq 2^{n-k} - 1$$

- The above condition should hold for all n columns of the parity check matrix, **H**.

i equal to 1, 2, 3, 4 up to n minus 1. This condition should hold for

(Refer Slide Time 31:02)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k) \times (i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

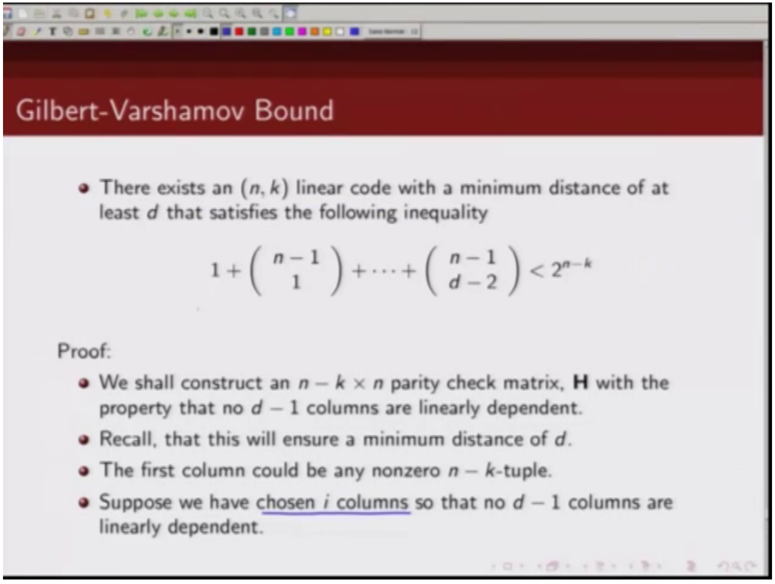
$$\binom{i}{1} + \cdots + \binom{i}{d-2} \leq 2^{n-k}-1$$

i = , n-1

- The above condition should hold for all n columns of the parity check matrix, **H**.

n minus 1. Then I can add another column which will be the nth column and still I would not have d minus 1 columns of this parity check matrix adding up to zero. So this should hold for all i up to n minus 1. So when we put i equal to n minus 1, so what we get is n minus 1 C 1, n minus 1 C 2 should be less than equal to 2 n minus k minus 1 and 1 we can write as, so 1 we can bring it this side and hence we will get our desired expression

(Refer Slide Time 31:42)



Gilbert-Varshamov Bound

- There exists an (n,k) linear code with a minimum distance of at least d that satisfies the following inequality

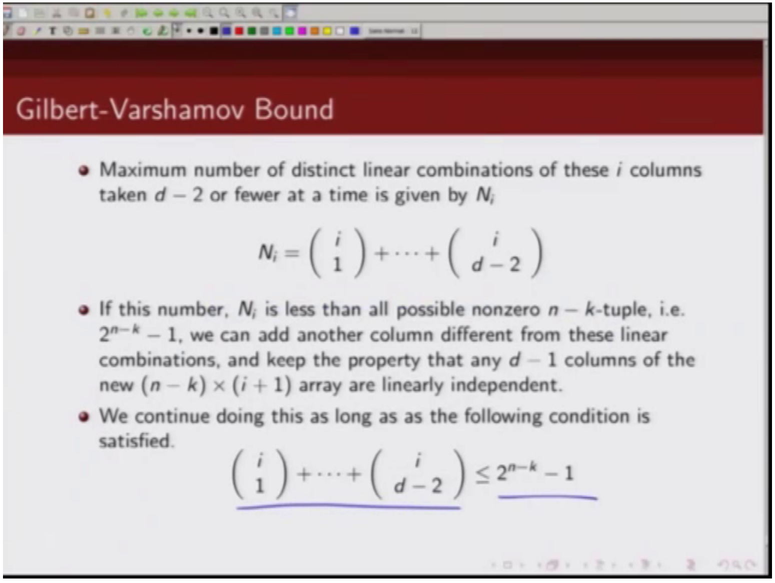
$$1 + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}$$

Proof:

- We shall construct an $n-k \times n$ parity check matrix, **H** with the property that no $d-1$ columns are linearly dependent.
- Recall, that this will ensure a minimum distance of d.
- The first column could be any nonzero $n-k$-tuple.
- Suppose we have chosen i columns so that no $d-1$ columns are linearly dependent.

which is this, Ok. This we just want it,

(Refer Slide Time 31:48)



Gilbert-Varshamov Bound

- Maximum number of distinct linear combinations of these i columns taken $d-2$ or fewer at a time is given by N_i

$$N_i = \binom{i}{1} + \cdots + \binom{i}{d-2}$$

- If this number, N_i is less than all possible nonzero $n-k$-tuple, i.e. $2^{n-k}-1$, we can add another column different from these linear combinations, and keep the property that any $d-1$ columns of the new $(n-k)\times(i+1)$ array are linearly independent.
- We continue doing this as long as as the following condition is satisfied.

$$\underline{\binom{i}{1} + \cdots + \binom{i}{d-2}} \leq \underline{2^{n-k}-1}$$

we want this linear combination to be less than 2 n minus k; then only we have additional, we still have a n minus k tuple which we can add as an additional column of this H matrix. So with this we conclude our lecture

(Refer Slide Time 32:06)

on bounds on the size of the code, thank you.