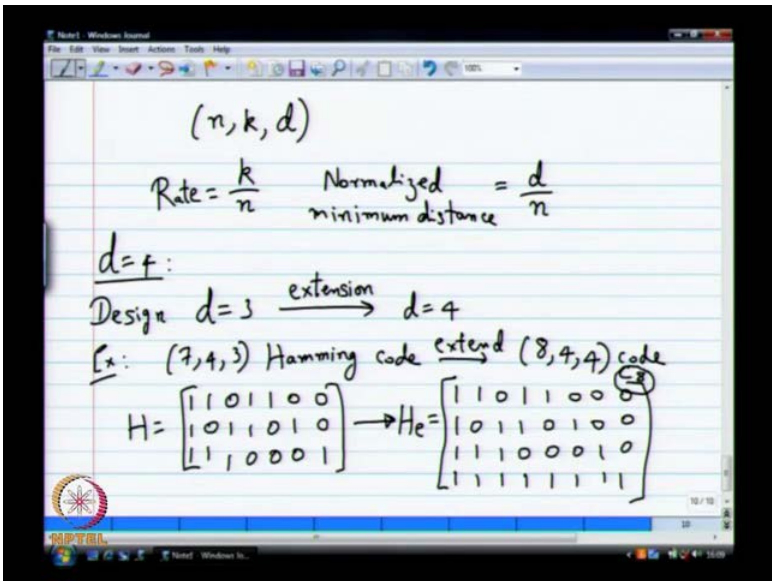
Coding Theory

Prof. Dr. Andrew Thangaraj
Department of Electronics and Communication Engineering
Indian institute of Technology, Madras

Lecture - 6
Bounds on Code Parameters

So, before we go to the bounds on the minimum distance in all there I promised I will show you something about d equals 4, I forgot about that is let us come back to that real quick.

(Refer Slide Time: 00:22)



(n, k, d)

$\text{Rate} = \frac{k}{n}$ Normalized minimum distance $= \frac{d}{n}$

$\underline{d = 4}$:

Design $d = 3 \xrightarrow{\text{extension}} d = 4$

Ex: $(7, 4, 3)$ Hamming code $\xrightarrow{\text{extend}}$ $(8, 4, 4)$ code $\boxed{=8}$

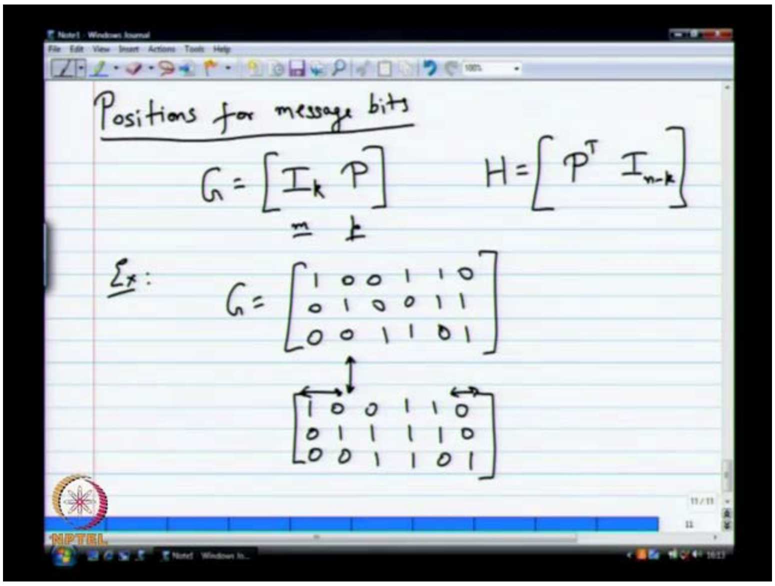
$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow H_e = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

So, you can see what modification will be most useful when you want to go to d equals 4. We know how to do d equals 3 how do you go from d equals 3 to d equals 4 extension and that turns out to be quite good in most values. So, basic idea for d equals 4 is you do d equals 3 design d equals 3. Then you do an noise extension to go to d equals 4. So, I illustrated only for the seven four hamming code just to see that it works. Then we will we will live at that simple example.

So, let us start with the seven four three hamming code see you have a parity check matrix, which is maybe I will write it in systematic form. So, that some of you might complain I said the seven four hamming code was had some other kind of parity check matrix, but that and this parity check matrix are quite equivalent. What does the only thing I have done swap columns. We do not mind the swapping columns, so all equivalent, so we do not mind that. So, this is the seven four hamming code what will happen if I extended I am going to get a 8 4 4 code. So, this is the very interesting code it is a very well study code. We may not see all those properties, but I will simply write down what the extension will look like.

So, idea being, an extension as I am adding an overall parity check and basic thing has to simply add one and then put a zeros here. So, that would be a parity check matrix of the extended version. So here you can clearly see in this code C 8 is what. So, here is an another quick comment, which I want to get read of the 4 v before we proceed further. So, this is got to do with positions of message bit positions for message bit. So, far will been looking at systematic form for the parity may generate a matrix, which is of this form I p.

(Refer Slide Time: 04:26)



Positions for message bits

$$G = \begin{bmatrix} I_k & P \end{bmatrix} \qquad H = \begin{bmatrix} P^T & I_{n-k} \end{bmatrix}$$

$$\underline{m} \quad \underline{k}$$

Ex:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\updownarrow$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

So, we always think of a code word as having a message first then having a parity. So, if you take an example may be if I take the previous example. I need a example I could have a g, which look like this lets say 1 0 0 0 1 0 0 0 one say 1 1 0 0 one 1 1 0 1, something any many example you can take. So, when you start at it you think of the messages occurring in the first three positions. The next three positions is a is a code word. So, there are various interpretations, which you have be care full about for intense.

If I say x or the second and third row suppose if I x are the second and third row what do I get I will definitely get an equivalent generate a matrix and how that look 0 1 1 1 1 0 am I, right? If I want that, so the last row is going to remain the same. Now, if you look at that it is look like I mean where is the I gone the last column of I is gone to the last position. So, these two guys are reaming the same then the last column is also become. So, these three columns together will complete the identity matrix. Now, I can say in this version the first second and sixth bit are my message bits and third fourth and fifth are parity bits. So, what I want to point out the it just brief point it is not a major thing, but the positions of the message bits can be anywhere on the in the code word. So, depending on row manipulations you can get it to be anywhere not necessarily in the first three.

First three is a convenient thing to keep for various implementation purposes and all that, but otherwise it does not really matter, can be anywhere. What is the condition on say some k columns suppose I pick any k columns of the generated matrix. When will those k columns be capable of being message bits, what does the condition linearly independent? Does what important?

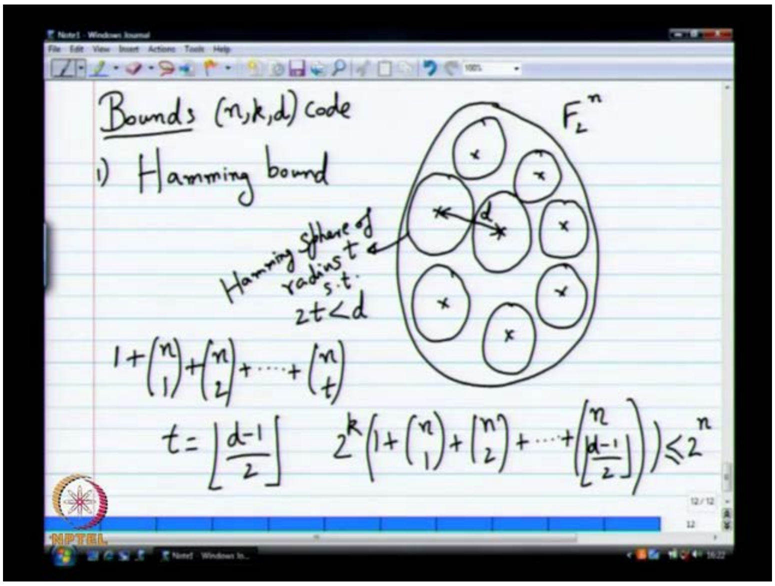
Of course, after row elimination I should be able to create a identity matrix there, which means originally they those k columns must have been linearly independent longest you have k columns of g, which are linearly independent, those can possibly be message bits positions. Now, if you go to the parity check matrix you will have a similar situation, but there it is little bit more settled, if you look at the parity check matrix.

So, usually think of a it as speed transpose I and minus k. So, here also I can do row manipulations. Then move may I where ever I want, but what does the position of I indicate in the parity check matrix, is it message bits or the parity bits? It is a parity bits that is the difference. So, you have, so once again I can pick any n minus k columns in h and ask when will those n minus k columns correspond to parity bits? When will that be possible? When those when they are linearly independent.

So, that is it is a small point I wanted to make, which it should be familiar with. So, you will see a lot of people trying to move their message bits around here and there. It does not matter where they are as long as those columns linearly independent, you can either assign them to message and generate a matrix or parity in the parity check matrix.

So, that is the idea, so this I P and P transpose I just a convenient form I can the columns of I can be interprets all over the matrix. So, let us just do way with that and then move over to bounds. So, I will do them reasonably quickly it is good to know a few bounds. So, that we have feel forward happens with this parameters n k and d.

(Refer Slide Time: 09:01)



Bounds (n,k,d) code

1) Hamming bound

Hamming sphere of radius t s.t. $2t<d$

$1+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{t}$

$t=\left\lfloor\frac{d-1}{2}\right\rfloor$ $\quad 2^k\left(1+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{\lfloor\frac{d-1}{2}\rfloor}\right)\leq 2^n$

So, the first bound will see, I think I understand your question. So, you seen in the given form of G you may not have the I occurring in a set of linear k bits linearly independent columns. I will have do some row manipulation to get I there. If you want those to exactly correspond to your message, but they will those k bits will be in one to one correspondence with your message. You can always go from one to the other they may not exactly match a message bits, but there will be an invertible matrix, which is multiply a message bits to give you that.

So, let us see the first bound, which is call the hamming bound, since the useful bound to have. So, here a picture is a very nice to illustrate what the hamming bound does. It is also call the sphere packing bound, you will see how why that is that is called. So, our bounce will be for n k d codes. So, I will start with an n k d code and try an establish some bound between relating k n and d, what do you mean by bounds some in equality? So, I will come with some any equality, which relates n k and d and that lambda being bound on one of the other.

So, let us say this is F 2 n and let us say I have marked out all my code words. Now, what I am going to do is I am going to draw spheres a circles around each of my code words. What does that circle include? It will have say radius what I will use radius sum P some small t radius. So, what do I mean by radius? Now, I have to be careful may distance is not some nucleant distance, it is could be hamming distance. So, let us say will pick one guy and draw a sphere or radius of circle a circle of radius p. It is say as sphere of radius t it is called the hamming sphere, because it is in hamming distance. So, my question first question is, suppose I do it for some radius t how many vectors will be inside this sphere?

So, you think a little bit about this, so what does the question I am asking? I am giving you an arbitrary vector. Then I am asking how many vectors are a distance at most t away from that code words. So, have add up everything, so what will happen is first I have the code word itself. Then I have set of all vectors, which are a distance one away from my code word, how many vectors will that be?

How do I think of this I have one vector. I want all vectors that defer from this code word and only one position. How many ways can I pick that position and choose one. So, I will get n choose one vectors, which are a distance one away from my code word chosen code word. Then I will have n choose two code words vectors, which are away in two positions so on I can add up and go all the way to and choose t.

So, that is the first idea the lot of ideas here what hamming distance that will come out in this bound case. So, carefully all this useful also is it clear hopeful you are able to visualizes what happening have one vector. Then I see all vectors, which differ in this in only one position that I can have a 1 of them. Then I choose a n chose two I pick two position and I flip those bits I will get another vector n chose 2 n chose 3. So, what that many vectors you have an hamming sphere of radius t around each code word. Now, I want to draw such spheres around all the code words and still not have them over lap. What does the maximum possible t that I can pick this is a question.

Suppose, let us say there is one code word here and there is a another code word here, which is exactly a distance d away may be this two form a minimum code word pair, there will be two such things, because I have an n k d code. So, there will be clearly two code words, such as separated by exactly d positions. Now, if draw another guy of radius t here when can I be absolutely sure that they will not overlap. If two t is less than d, if two t is less than d. Then I can be definitely sure. Now, you are using some nuclear dent n sides the whom why I am dealing with hamming distance, how do in all such an any quality whole in hamming distance? So, you have to check it is easy to prove the doubts. So, hamming distance will obey a triangle in equality kind of thing.

So, if you have two things a distance d a part you started one and you flip P you started the other and flip another t arbitrarily. This two will not be meeting if 2 t is less than d. So, that require some expressive proof you can write it down it is not very difficult. So, I can pick such that P such that two t is less than d. So, there is another way of writing this down, what does the largest possible t, which will have 2 t less than d you have to do flour of t by 2.

So, 4 of t by 2 is a little bit dangerous, why? d is even you will have a problem. So, what how can you avoid that d minus 1 by 2 is just absolutely safe. So, nothing will happen to you. So, you pick a t like d minus 1 by 2. So, distance you might do not want to pick P equals d minus 1 by 2, say you can you can flour this. So, then there will be no problem, so you will definitely get a valid.

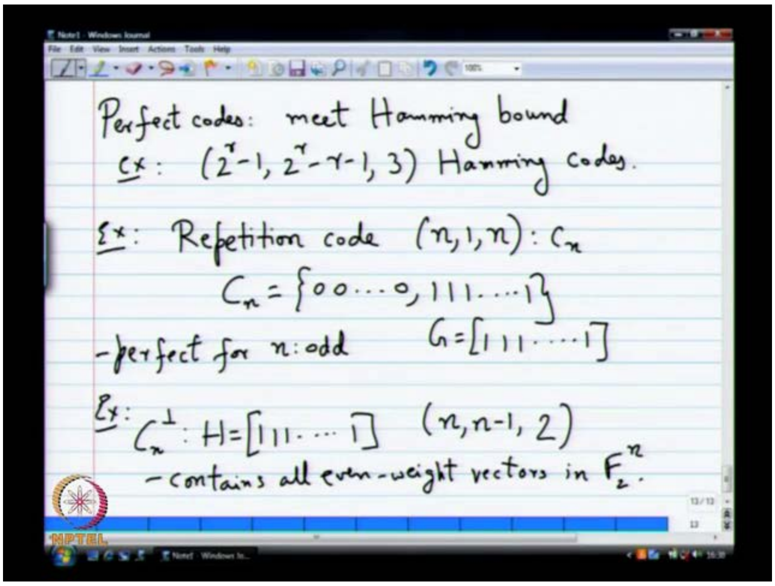
So, what does the next step, now I have drawn these spheres around each of this code words. I know none of these spheres overlap with each other, what do you mean by overlap? There is no vector in each of these spheres, which is common to two. There is no vector here, which is common to more than one. So, that is the idea, now I have a bound, how will I do a bound? Now, think a about it. So, if you now multiply this number of vectors in each sphere by two power k. Why do 1 multiplying by 2 power k I have 2 power k code words. So, I have 2 power k such spheres. Now, I am counting the total number of vectors inside the hamming spheres of radius t. So, that will be two per k times this. Now, I know, I did not over count, because none of this spheres have any overlap.

So, clearly this two per k times this should be less than that total number of possible vectors in the entire space, which is to power n. So, that is this sphere packing or hamming bound. So, 2 power k times 1 plus and choose one plus n choose 2 plus 1 till n choose d minus 1 by 2 floured. So, let me draw this down bit more clearly is less than or equal to power n. So, that is an equality, which relates n k and d and it is valid always. Any n k d code you have this has to be satisfied cannot valid this and k d binary code will satisfy this condition. So, codes that mean the hamming bound what do I mean by meeting the hamming bound? This inequality is actually a equality. That is what that is when something meets an equality.

So, if you meets this bound then those codes are called perfect. So, there is a reason why they are call perfect will see them later on, but those are called perfect. There is a very a surprising statement about perfect codes, where people have characterized all possible perfect codes. They are very few number and not too many of them very few codes. You can imagine it is a significant commentarial achievement characterize. All perfect codes not at all obvious. How you do it? First of all can you give me an example of a perfect code? Hamming code is a perfect code.

So, you can check that foe instead if you put k equals 3, d equals k equals 4, d equals 3 and n equals 7. We will see that will make this bound and hamming code meets this bound. All hamming codes meet the hamming bound imagine the name matches it is must be as be true. So, let us see perfect codes meet hamming bound there are examples the two power r minus 1 2 power r minus r minus 1 comma 3 hamming codes.

(Refer Slide Time: 19:28)



Perfect codes: meet Hamming bound

ex: $(2^r-1, 2^r-r-1, 3)$ Hamming codes.

Ex: Repetition code $(n,1,n)$: C_n

$C_n = \{00\cdots0, 111\cdots1\}$

-perfect for n: odd $\quad G=[1\,1\,1\cdots\cdots1]$

Ex: $C_n^\perp$: $H=[1\,1\,1\cdots\cdots1]$ $\quad (n, n-1, 2)$

-contains all even-weight vectors in F_2^n.

So, I will give you I will try and ask you question about another example. There is a another code, which is called the reputation code. I am not introduce this formally, but may be this is a good time to introduce it. The parameters are basically n comma 1 comma n. So, I will ask you a challenging question, I will give you the parameters of a code you have to now tell me what the code is hamming code? You know how to do it.

So, I am telling you the reputation code, it is parameters are n comma 1 comma n what will be the code?

See, if all n is the number of bits l is one which means there are only two code words. If you have two code words, one of them you already know, what is one code word is always all zero. There is only one other code word and what does the minimum distance n. So, what should the other code word be all ones that is why it is call the reputation code. So, this is the reputation code and this code basically if you call that C n is basically all zeros and all ones. You can see why it is call the reputation code. There is only one message bit, which is say this 0 or 1. If it is 0 you transmit all 0 you repeat 0 and times. That is why it is called reputation code.

If it is if message bit is 1you will pit one and times called the reputation code obviously, is this a perfect code? That is my question n and if the pen is even it is not perfect we correct. It could be correct saying it is not correct, could be correct. So, for n all let us perfect right people are clear and you can put it in to that formula. You will get the answer I think. So, for n all that is perfect, so that is some you can check. So, when is even they will be like this one overlap, which will kill you? No one, this one problem, which will not give you it will be almost perfect. It will not quite be perfect another example children write down is going to be the dual of the reputation code. This is a good test of whether you understood do well or not.

So, what will be the dual of reputation code? First of all what are the parameters for the dual of reputation code and remains the same. Then n minus 1 that is also is easy. Now, we have to characterize the dual. Tell me what will be the minimum distance? What is the easiest way to do? This not to difficult, see you have come up with the generator matrix for C n. What will that be for C and pop the dual? It will be the parity check matrix. So, that is the easiest way to go about doing this. If I have the generate a matrix what is the generated matrix for the C n? This is the only generator matrix for the reputation code. So, if I now think of C n pop what will be the parity check matrix?

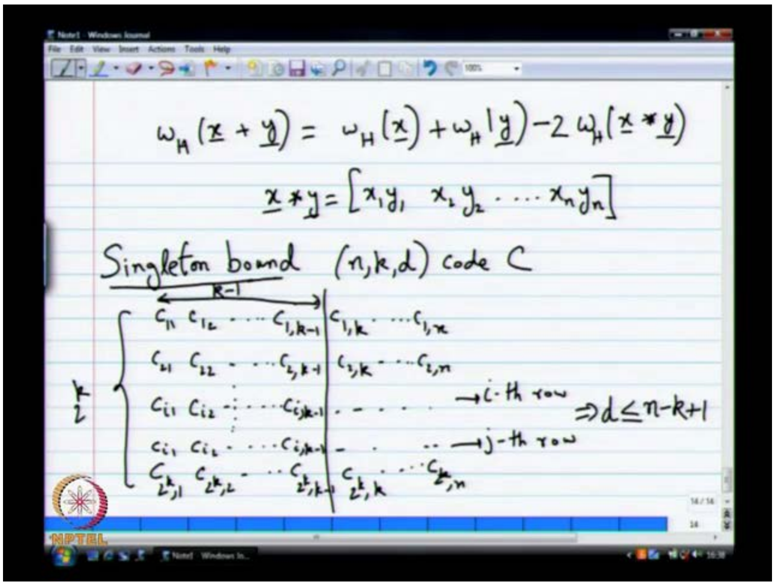
Now, answer the question what is the minimum distance cannot using the definition carefully when will you have minimum distance one. If there is an all zero column is there is an all zero column here, no when will you have minimum distance two. If you have a reputation, do you have a reputation here? Yes, so clearly minimum distance will be 2. This is C n pop there is a very easy way to characterize C n pop. How will you describe it in words? As to be, so it is what does call say the even parity code or the even weight code. The code contains all the even weight vectors in F 2 n C n pop contains all even weight vectors n F 2 n. I mean first of all it has to contain clearly the you all the even weight vectors only. How do I know that C n pop will contain only even weight vectors?

Some has to be 0 the parity check is all ones, nothing like n to bounded it has to contain only even with vectors. How do I know it will contain all of them? So, that is another we have think about this 2 per n minus 1 code words as to be exactly to per n minus 1 even weight vectors of length n. It has to include everything there is no problem we are looking at this. You know it as work out how will I generate a matrix for this? Look like very easy if the split as P n. So, I will come just here, so it will be just simply I 1. So, you do I n minus 1 and then all ones. So, when you encode this what do you have? n minus bits and the nth bit is what parity of all the n minus 1. So, that is why it is also called the even parity bit even parity code.

So, it is a very simple code to describe it as solve this inters cases. So, one thing in this would not be perfect is if you have even minimum distance, it would not be perfect is will be a problem with the spheres and intersecting. So, it would not go a code can be anywhere this is a good code to keep in mind. So, another curious thing, which if you do not know about binary vectors, it will be a little bit more interesting here. So, note this C and pop I know it is a linear sub space. It is a linear code clearly it is a linear sub space it contains all even weight vectors. So, clearly what should happen if I add two even weight vectors being the same set? It should all also be a even weight vector. So, that is an interesting relationship, which you can independently check for hamming distance.

So, you can show this in interesting relationship the hamming weight of x plus y. You can show will be equal to hamming weight of x plus hamming weight of y minus 2 times hamming weight of x star y. What does this x star y I have to tell you what x star y is x star y is basically x 1, y 1, x 2, y 2, x n, y n.

(Refer Slide Time: 26:56)



$$\omega_H(\underline{x}+\underline{y}) = \omega_H(\underline{x}) + \omega_H(\underline{y}) - 2\,\omega_H(\underline{x} * \underline{y})$$

$$\underline{x} * \underline{y} = [x_1 y_1 \quad x_2 y_2 \cdots x_n y_n]$$

Singleton bound (n,k,d) code C

$$\begin{array}{c|c} \xleftrightarrow{k-1} & \\ c_{11}\ c_{12} \cdots c_{1,k-1} & c_{1,k} \cdots c_{1,n} \\ c_{21}\ c_{22} \cdots c_{2,k-1} & c_{2,k} \cdots c_{2,n} \\ c_{i1}\ c_{i2} \cdots c_{i,k-1} & \cdots \quad \to i\text{-th row} \\ c_{j1}\ c_{j2} \cdots c_{j,k-1} & \cdots \quad \to j\text{-th row} \\ c_{2^k,1}\ c_{2^k,2} \cdots c_{2^k,k-1} & c_{2^k,k} \cdots c_{2^k,n} \end{array} \quad \Rightarrow d \le n-k+1$$

2^k

So, the two times is critical the way to prove it. As you write x on the top row write y below, when you add what will happen? Whenever, there is a overlap two ones are disappearing. So, this is x star y counts the number of overlaps between x and y in position. Where, you have a 1 when you add both them go away in the sum. So, the you have subtract two times weight of the overlap.

So, if x and y have both even weight what will happen to the weight of x plus y? It will again be even what does a x n y are both odd weight. Once again you will get even if one odd one is even what will you get? Odd set all this things you can nicely prove using this simple things. Anyway, so I dragnets a little bit from our bounds any way I thinks it is a good dig rations, so nice thing to know these things what the binary vector space. So, the next bound we are going to do is a, what is called the singleton bound. So, it is a very famous bound. So, here is how I am going to prove it. Once again I have an n k d code C what I am going to do is I am going to take all the code words of C, write them one below the other. So, just for convince system the proof can also imagine, but I will write it down just for just for the sake of completeness.

So, if C 1 1 C 1 2 soon tell it says C 1 C 1 k minus 1 C 1 k so on C 1 n. Then I write C 2 1 C 2 2 so on till C 2 k minus 1 C 2 k C 2 n. See, imagine that the first k positions are message bits. So, the that is another thing I am assuming here. So, all the way down to C what 2 power k 1 2 power k 2 C 2 power k, k minus 1 C 2 power k k all the way to C 2 power k. So, these are all the 2 power k code words that you have an that have an this is that you having this code C. So, once again I told you the first k positions are systematic assume as half message bits. So, the message bits positions are also called systematic positions. So, that is a ton that is used all the time think I do not know why, where it came from is called systematic, it is for some reason some reason.

So, the first k positions are message bits, what does it mean to say the first k positions are message bits? All the 2 power k possibilities will occur in the first k position. That is what it really means. If I listed down one below the other it means the first k positions all the 2 power k possibilities will occur. That is when it becomes a message bits, because the first k can be message in an everything, every possibility will occur that. Now, suppose I look at the first k minus 1 position. So, there is a good reason why I wrote k minus 1. Suppose, I only look at these guys, first k minus 1 positions, how many of them do? We have 2 power k, how many possibilities are there for k minus 1 positions 2 power k minus 1?

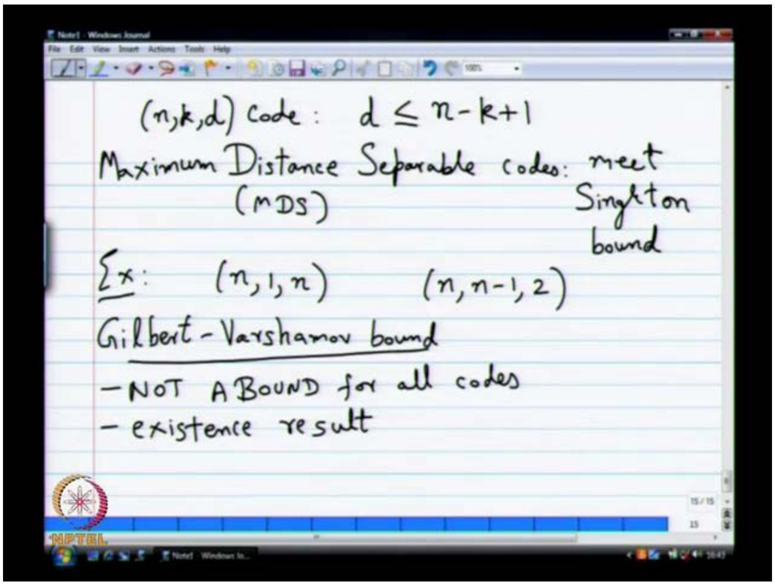
So, you have and you have 2 power k vectors. So, what should happen? There should be two rows back this two things, repeat at least one for possible reputation as to happen. So, may let us say the, I throw and the jth row. Repeat, so in the row and j th row, what will happen in the j th row? I will once again have C I 1 C I 2 C I k minus 1. So, here you will have a all kinds of stuff. So, I do not know what else is there. So, what can I say from the I th row and j th row I am going to say with one more step. You will get a relationship between d n and k. If you add this two what will happen last bit first k minus 1 bits will become 0. Another way of looking at it instead of adding in all the thing if you look at the distance between the code words the I th code word and j th code word. What can possibly be the distance n minus k minus 1, which is n minus k plus 1.

So, whenever I wrote down 2 per k code words there will be two such code words, which define, which can possibly differ only in the last n minus k plus 1 positions. So, clearly the minimum distance cannot be greater than that. So, minimum distance is less than or equal to n minus k plus 1. So, this is the this is what does known as the singleton bound single tend is a name of the person who came up with this bound. Also, there other ways of proving this, which u is generator matrix and parity check matrix. What could be one advantage of using this proofs that I write down here that is generator matrix. Also, always start comfort linear code, it is not necessary, they restrict linear you might want have non-linear codes.

So, the proof I wrote down will work even for non-linear codes., if the code is not linear as long as it as just a simple commentarial. What this is the called pigeon hole principle and commentarial, just the simple argument nothing beyond. So, if use this generator matrix parity check matrix are proof wholes only for the linear codes. So, that is all we would not say non-linear codes in this codes. So, do not worry about it, but never the less this is good proof to know.

Now, let us see if you can give me examples of codes that meet the singleton bound. So, codes the meet the singleton bound are called, once again let me remind you what the singleton bound is have an n k d code. I know definitely d is less than or equal to n minus k plus 1. Codes that meet the singleton bound are called maximum distance upper above or M D S M D S codes meet, singleton bound maximum distance separable is the name.

(Refer Slide Time: 34:45)



(n,k,d) Code : $d \leq n-k+1$

Maximum Distance Separable codes: meet Singlton bound
(MDS)

Ex: $(n,1,n)$ $(n,n-1,2)$

Gilbert - Varshamov bound

- NOT A BOUND for all codes
- existence result

So, I want some examples, so reputation code clearly is an M D S code, see once I write n one n at becomes unique weight distance only the reputation code, this is no other code that can work out. So, n 1 n and you will see the dual the n n minus 1 two code will also be an M D S code. In fact you can show this is general property, if C is M D S C pop will also be M D S.

So, we would not see C such proves today, but anyway it is a good thing. If you think you can prove it you can try to prove it, it is an interesting proof. Anyway, so if code and it is dual will both be M D S. So, what is another interesting is there is no other binary M D S code that is all we listed out all the binary M D S codes, no other binary code is M D S. If you go to others finite fields there are M D S codes, of course the celebrated, which all almond code is an M D S code.

So, we will see that eventually, but in the binary field there are no other M D S codes. So, for intense the 7 4 3 hamming code is kind of optimal in some where they cannot be a 7 4 4 code. That it is possible minimum distance, so that is I am not proved that u you can try a proof. I will encourage you to try it to prove that statement, why you said that binary codes? You can never have any other code.

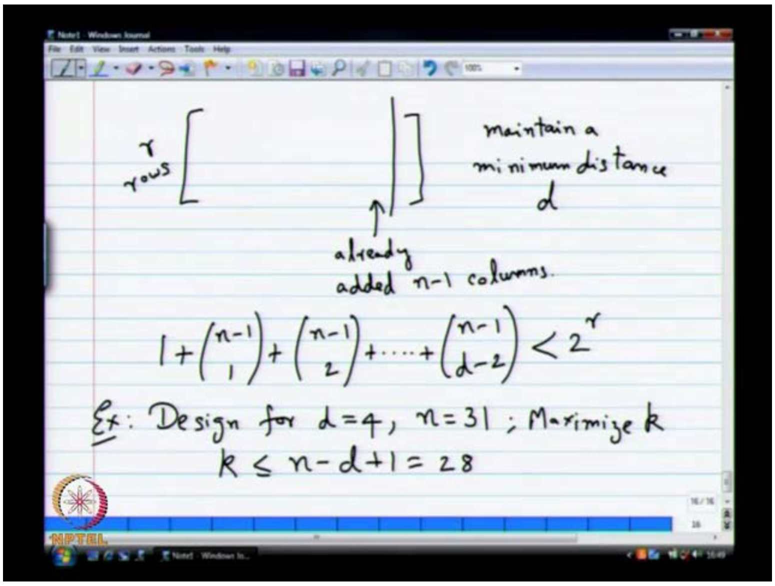
So, basically you have to show if k is greater than 2 greater than equal to 2 immediately you cannot have M D S property, that is the enough the dual. Like I said will also be a M D S codes n minus k will also occur. It is not a problem to show of k is greater than equal to 2. You cannot have this that is the idea, of course I mean n minus 1 is possible. So, a limited that n minus 2, that is the idea, so that is single tern bound.

So, the last ten minutes will see another bound, which we would not use to much, but that is also very little any interesting bound, which call the Gilbert Varshamov of bound. So, the previous two bounds that we saw are any qualities, which have to satisfied by all codes start with an n k d code. They have absolutely satisfy that this inequalities the Gilbert Varshamov bound is an actually an existence result.

So, it is not a bound for all codes there can be codes that violet. This in equality that I am going to write down, but you design existence result. What do we mean by existence result? So, if the you have an n k and d with satisfy this relationship. Then there exist at least one code with those parameters that is what it means. There is another way of looking at, which makes this code really this bound really very important. That will not go too far n to that.

Basically, it is an existence result the way you prove that does a little bit interesting. I will write it down real quick we do not have too much time. We would not spend too much time on this. The basic idea is to construct a parity check matrix with r rows columns by column. So, you keep constructing it column by column and you always maintain a minimum distance d, you want to keep maintaining a minimum distance d. You have r rows and you keep adding column by columns.

(Refer Slide Time: 38:54)



r rows

maintain a minimum distance d

already added n-1 columns.

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

Ex: Design for $d=4$, $n=31$; Maximize k

$k \leq n-d+1=28$

Suppose, you have added n minus 1 columns we have already added n minus 1 columns can of an inductive thing, but any ways let me just write it down real quick to get you the final answer. I will write down an equation or in any equality, which will tell me that I can add a n th column also. When can I add take n th column? Is the question, I will write down an inequality. It is quite easy to write down, but if that is satisfied it means I can add a n th column also. Then you keep it iterating on this any quality you will get this bound is that is the idea.

So, let me write it down, so you have n minus 1 columns that I have already added. When can I add the n th column? Is the question. See, the n th column cannot be the all zero vector, it cannot be the all zero vector. It cannot be any one of the previous columns that I added I write cannot be the sum of any two of the previous things that I added. So, I can go to it cannot be the sum of any d minus 2 columns, that I added can be the sum of d minus 1. Because, in that case I will have minimum distance d, it is not a problem. These are the total number of possibilities, which are excluded for the n th column, I can never add any of this case. In fact I might be over counting here, but does not matter. They just want a bound and this is good enough bounds. So, I do not care, so I am just adding it up this has to be strictly less than 2 power r.

If it is less than 2 power r, then I can definitely add the n th column. So, this is the condition for the Gilbert Varshmov of bound, so it is a good bound. I am just giving you an argument for why that bound comes from and it is very useful. So, codes that meet or are better than the Gilbert Varshmov bound usually call good, you can say that good, now are the great words for code does usually, suppose to be good will. It is not really that is not the only condition good is good, but in general if you meet the Gilbert Varshmov of bound. It means you are doing something, which is not very bad. It is a good thing, but of course you can they can be parameters, which beat this bound and still they will be code.

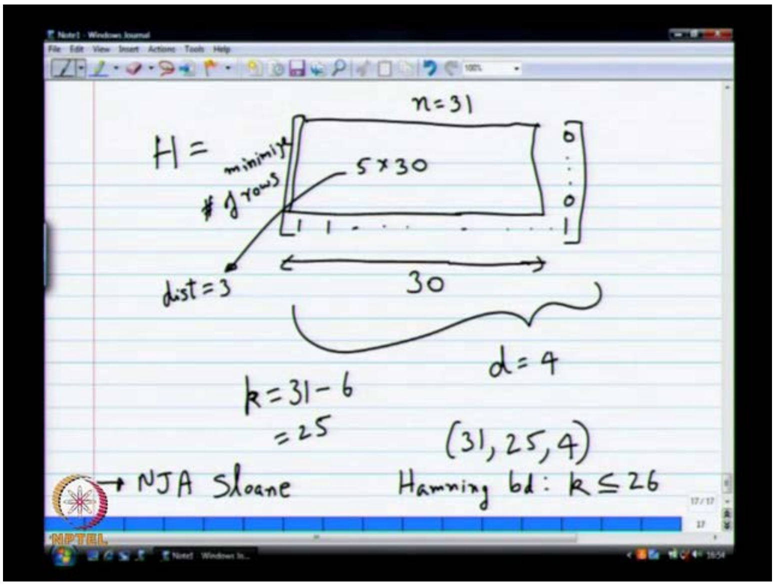
So, this is not a bound in that sense just showing you an existence result, there is this condition is satisfied you can have a bound and it is a good bound. Now, what we are going to do next is final contact why a simple little example, which will may be bring together some of this bounds and give sum feel forward happens. So, here is an example, for designing let’s say a design for it's a d equal to 4 n equals 31. Suppose, I tell you my block length I want this 31 the minimum distance say I want this 4. Let us say greater than or equal to 4 let us say greater than or equal to 4 you might would not this is pick 4. So, I want maximize k what does the largest possible k that I can have, which will give me for n equals 31 minimum distance of 4.

You can say greater than I equal to 4 for d, but then you would pick d equals 4. If I want a maximize k you want to pick the smallest possible d, because you know k goes up d will go down. So, as how it will work? So, it is a good picture to keep in mind is that sphere packing picture that I know. If k keeps on increasing the number stars is increasing, you should be able to only draw a smallest sphere. So, that is the simple picture you can keep in your head to convince yourself, why this the tension is there between k and d for a fixed n. So, take a crack at that is an interesting problem, what is the maximum possible k? That you can come up with page turn and knowledge, that you have do not blindly apply the bounds.

I want an actual parity check matrix bound would not give you give a parity check matrix. So, one bound if you use this singleton bound like, he is using you know k as to be less than or equal to n minus d plus 1,0 which works out to 28. You would not get any where closed to 28, believe me, will this you know already that I told you and informal result not a informal result very valid result, but I told you informally that no binary code will meet this singleton bound. So, you would not get that and you do not have to too many perfect code save that. So, even hamming bound will be violet rally you have to construct a parity check matrix. You have to construct an actual parity check matrix, which will give me k, you have do extension. The question is how many rows will you have for the parity check matrix. You have to minimize the number of rows for the parity check matrix, am I right?

So, suppose if you want to construct parity check matrix H you want to minimize the number of rows. Why do I want to minimize a number of rows? So, n is fixed at 31 I want a maximize k, which means I have to minimize a number of rows. So, I want to minimize number of rows, but I still want to have minimum of distance of 4.

(Refer Slide Time: 44:59)



So, remember how are we going to construct minimum distance 4. We are going to start with minimum distance three and then extend. So, in extending I will go from n to n plus 1. So, after extending I want 31 before I extending what should be my code length thirty. So, I have, now think of just the first 30 columns and put down some rows. Then finally, I will add one row for the extension. So, what does the minimum number of rows I should have for a length thirty code with minimum distance 3? That is the question, 5, right?

You have to have 5, 4 is not enough Why is 4 not enough? First 2 power 4 is only 16 clearly you are going to repeat something you cannot repeat. So, if you should not be repeating we need at least 5 rows is that for this you need 5 rows. So, for the 30 you need 5 rows. So, this guy will be 5 cross 30 will have distance 3, it will be equal to 3. That also you can be reasonably sure you cannot be greater than 3. It will be equal to 3 was a picking out 30 you come out to be 3. Then you extended extend one then you put all ones here then put all zeros here the overall guy will have a distance equal to 4. What is k now? So, I have be sure that this 5 rows that I put down will give me rank five. You can be sure that will also be true you putting. So, many vectors you can surely find 5 columns, which are full rank.

So, you we can think about it convince yourself that is true. So, it is not good if you tell to show that, so rank will be 5. So, clearly k works out to k works out to 31 minus 6, which is 25. So, we have a 31 comma 25 comma 4 code. Now, the question is how good is this code? Can I am got in 26, 27, I do not know. So, we do not know this things very well. In fact there are people who maintain extensive tables of this pass will codes. So, you can go to one of these tables. Just a table maintain by n G a s loan who works in a t n t billiards is a very famous coding theories. You can search for m he maintains a lot of tables. One of the tables he maintains is for the best possible code. So, you can go look it up in that table. So, you will know another thing to do is to plug in the bounds. See, how close you are to the meeting the hamming bound and the Gilbert Varshmov of bound.

If you have closed enough then you can be reasonably sure that you are ok. So, plug this is values in to the lets say the hamming bound and tell me what k you get there 26 for the hamming bound. What about Gilbert Varshmov bound? So, was claim here, which says hamming bound tells you k is less than or equal to 26, which already means you are doing a reasonably good job, you are not too of the charts. Then the Gilbert Varshmov bound, if it gives you k equals 26 for intense. Then it means you are not the best possible clearly, but if it gives you anything else you cannot be sure, that is the problem. So, you can check those things, you do not have to do it now. You can do it later plug in this values of n and d in those bounds. Find the find a bound for k compare with what do you have got and see if you are doing or not.

So, once again remember the interpretation for Gilbert Varshmov is different. If you get the larger k what does it mean in Gilbert Varshmov? It means if you get for intense in Gilbert Varshmov k equals 26 is possible. Then it means you can get something that strictly batter, but if you get for intense k equals 25 in Gilbert Varshmov in you cannot be sure. I know 25 is possible, it says 25 is possible, 26 is also possible you cannot rule it out, that is the problem said, so will stop here.