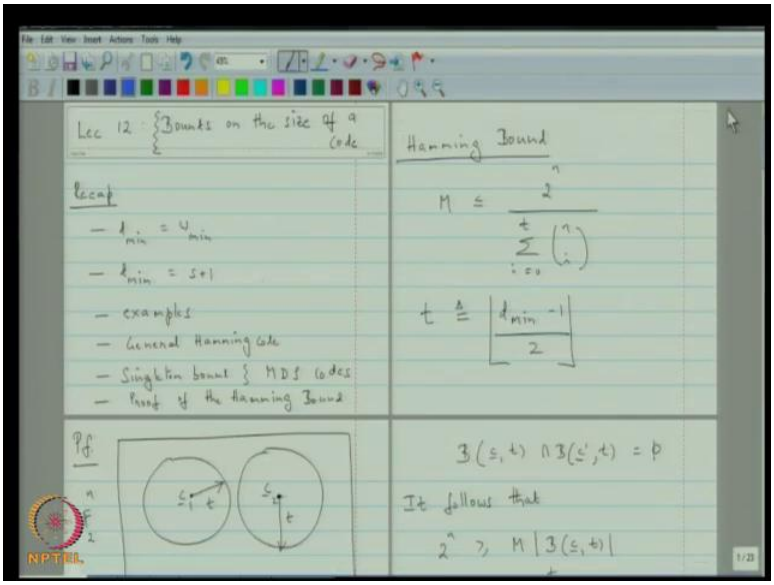
**Error Correcting Codes**
**Dr. P. Vijay Kumar**
**Department of Electrical Communication and Engineering**
**Indian Institute of Science, Bangalore**

**Lecture No. # 13**
**Asymptotic Bounds**

(Refer Slide Time: 00:20)

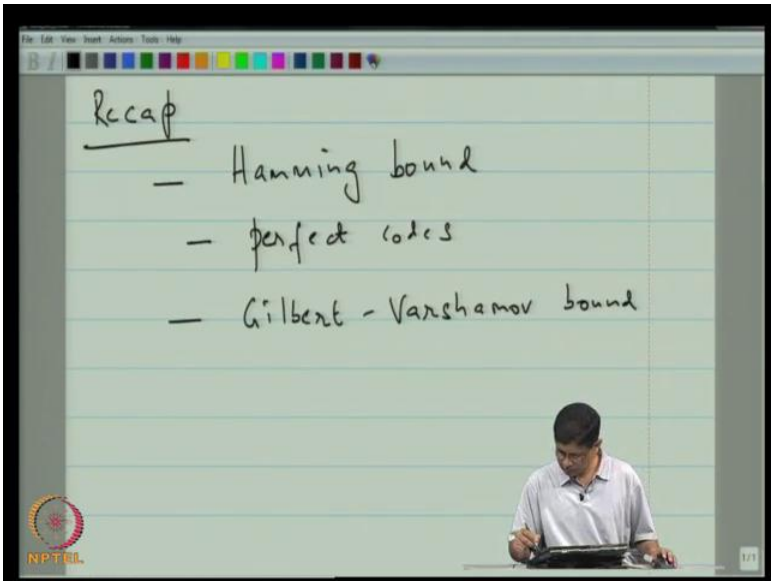



Good afternoon, welcome back, this is our thirteenth lecture, and I am going to title this, asymptotic bounds. And I think the reason for that will be clear from the last lecture, so perhaps we should just quickly look at last lecture, and we zoom out a little bit. In the last lecture, we started talking about... Well, we completed our discussion on the hamming bound, we prove the hamming bound, we also introduce the Gilbert Varshanov bound, which is a low bound; in between we said that codes that achieve the hamming bound with the quality, unknown is perfect code, so we talked about perfect codes. We also introduced the golay code as another example of a perfect code, and then we prove the Gilbert varshanov bound. And then I was telling you strain to motivate you, because our next thing that we are going to do is you are going to take the hamming bound and the Gilbert varshanov bound.

And then we are going to let n 10 to infinity and I was trying to explain to you y z that n ten into infinities of interest, because it gives you an approach towards attaining reliable communication and which in some sense is the ultimate aim of an error correcting code. An appointed out that

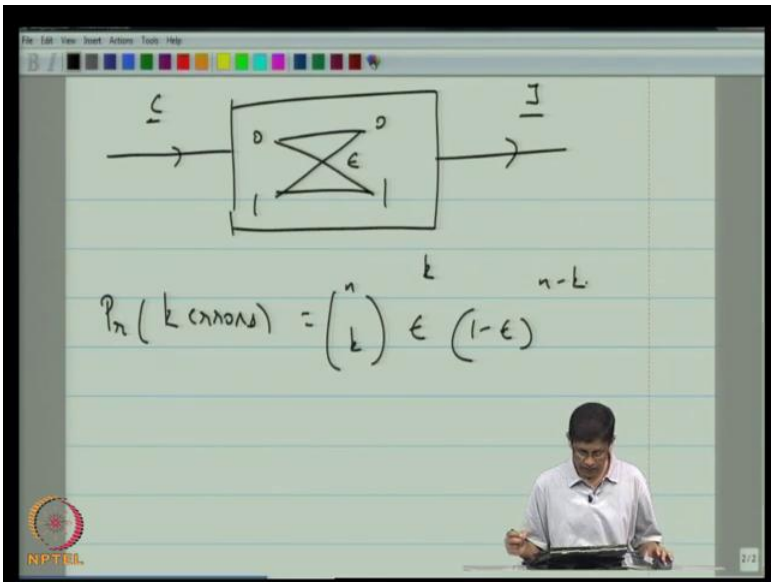
our approach is going to use long codes and also its going to use the bound distance decoding, explain what boundary distance decoder decoding meant and then we were in the mixed of describing of explaining, how we would achieve reliable communication? Let started out by looking at the binomial distribution etcetera.

(Refer Slide Time: 03:03)



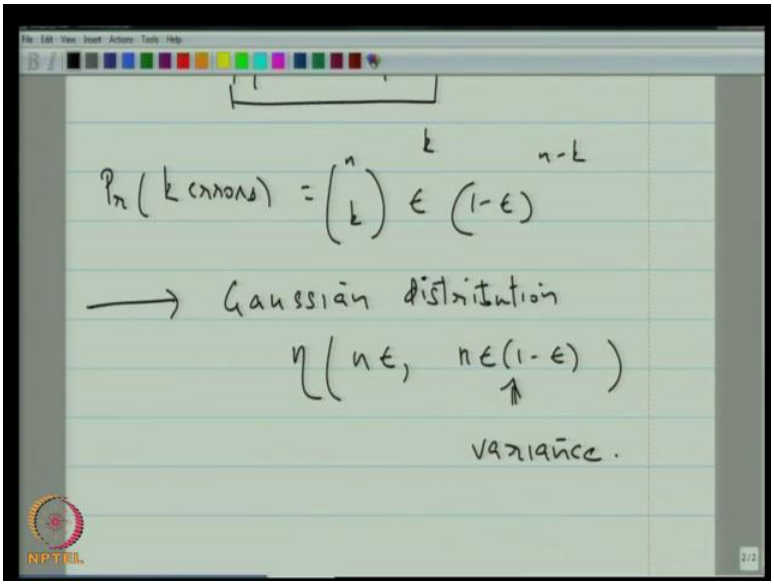

So, will pick up from there very quick recap, we looked at hamming bound looked at perfect codes. This included the golay code then we looked at the Gilbert varshanov bound and we will in the mixed of discussing an approach to achieving reliable communication.

(Refer Slide Time: 04:18)



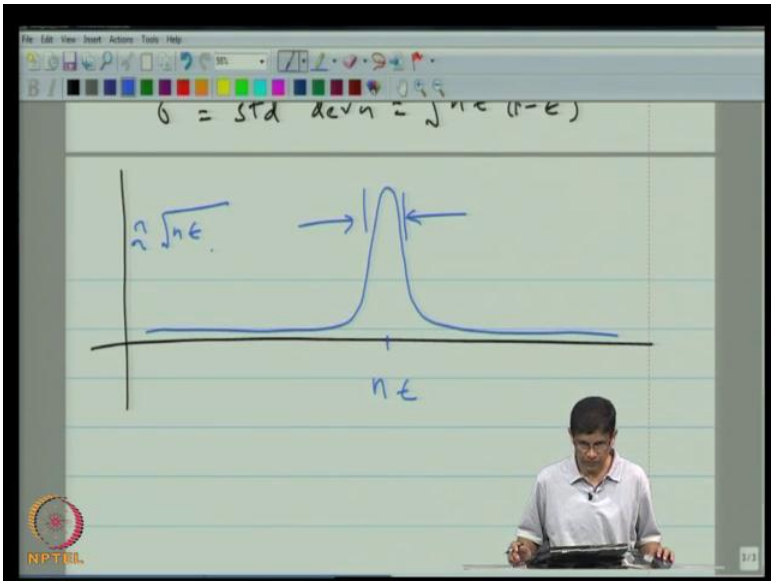

So, what it pointed out last time was that if you feed a code word to binary symmetric channel, this cross over probabilities epsilon, then the probability of k errors was n, choose k epsilon to the k one minus epsilon to the n minus k. But if n is very large, then this distribution tense to the Gaussian distribution.

(Refer Slide Time: 04:58)



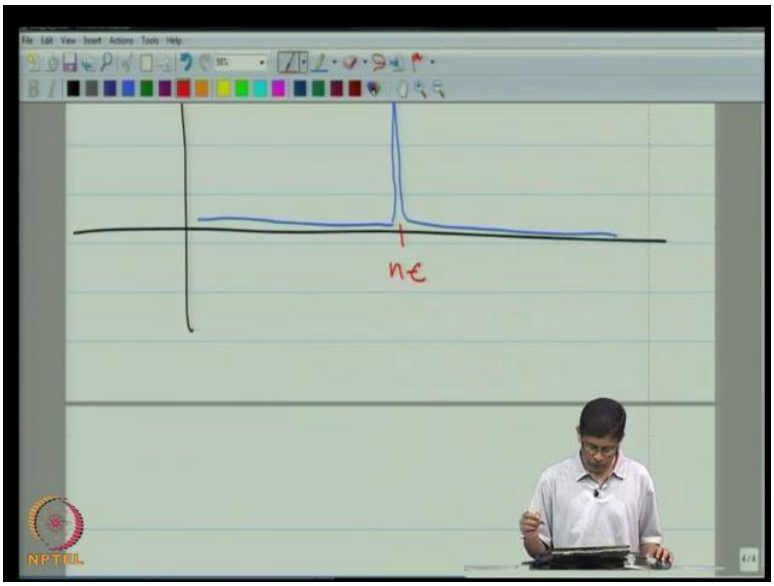
This distribution is normal with mean n times and with the variance n epsilon one minus epsilon, this is the variance now, and associated the variance is the standard deviation, which is the square root of n epsilon one minus epsilon.

(Refer Slide Time: 05:56)



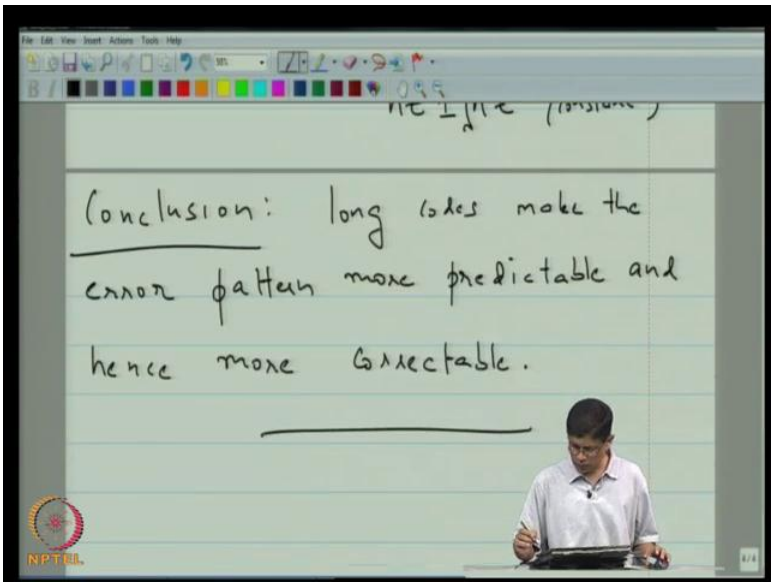

The picture is like this that for large, n the mean number of errors is going to be approximately n epsilon; the standard deviation is going to be a square root of n epsilon into one minus epsilon. The distribution is going to look like this, so it is going to peak... It will peak to other little bit better, so It is going to peak around in epsilon, then the standard deviation is going to be on the order of somewhere around here, we might say that the standard deviation is on the order of let say root n epsilon one minus epsilon. So, the vector the p is on the order of root n epsilon. So, one minus epsilon is the approximately, one so it is like root n epsilon.

(Refer Slide Time: 07:26)



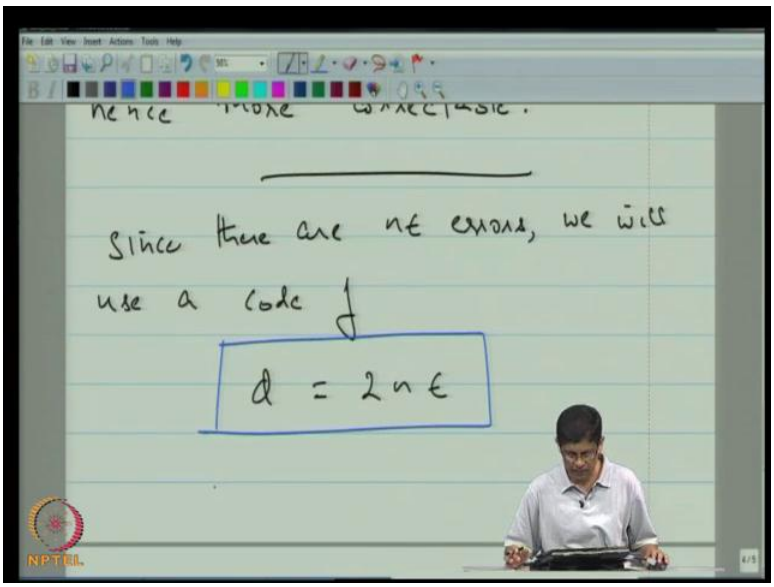

So, there is the thinning of these distribution has n gets large n large and you might imagine that for very, very, very large n. It is going to look vary pointed in between it is going to looks at something like this, to the extent that you can say well the errors the number of errors is almost exactly n epsilon. You can actually, So, say that you conclusion is that, you could say that the number of errors is within plus, minus some constant times root n epsilon. And will preteen that is negligible, so that we can say that the number of errors is almost exactly, equal to n epsilon and this is sometimes called sheer hardening.

(Refer Slide Time: 08:41)




The conclusion is that long codes make the error pattern more predictable, and hence more correctable and part of our approach was using long codes.

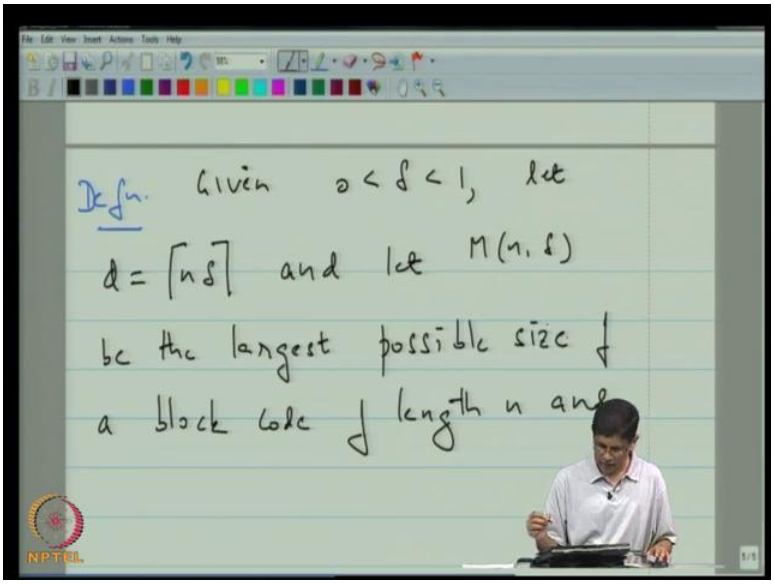
(Refer Slide Time: 09:19)




The second pair of approach was since there are n epsilons errors we will use a code of minimum distance d equal to two n epsilon. On tactics you may need to make it 2 n epsilon plus some small quantity, but will ignore that now, with this we will have achieved reliable communication.

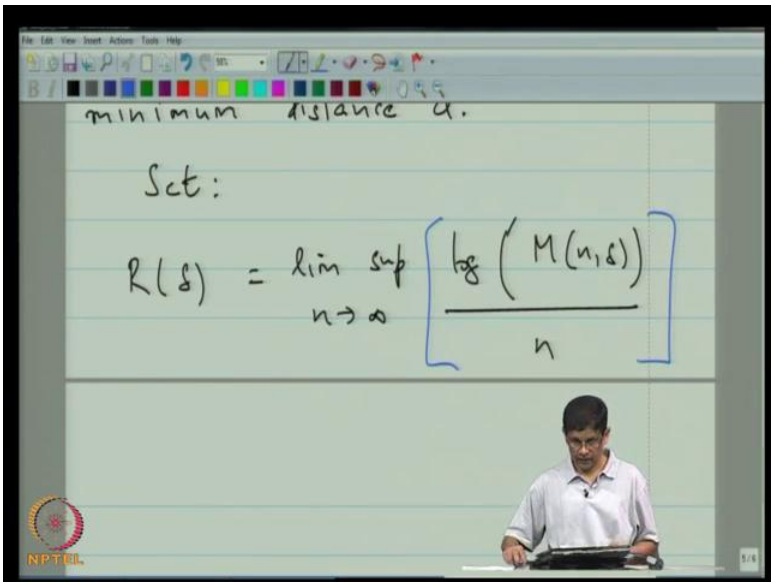
So, you might say us in the problem than been solved not quit, because what channels tells is that given the channel thus the quantity called channel capacity, which governs the maximum rate of information transfer. The questions still remains are you achieving the tray tuff information transfer? Let is set that this side for the moment and now, will get back to our hamming and Gilbert varshanov bounds and look at them from the point of view their values, when n is very large.

(Refer Slide Time: 10:58)



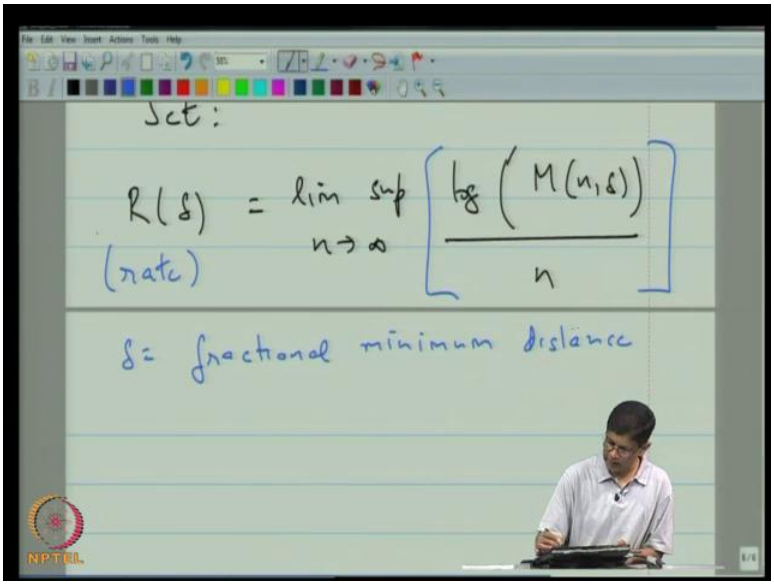

So, will begin with the definition giving delta between 0 and 1, let d equals n delta and I take the celling function that is the smallest integer greater than or equal to n delta and let us M n delta be the largest possible size of block code of length n and minimum distanced d.

(Refer Slide Time: 12:45)



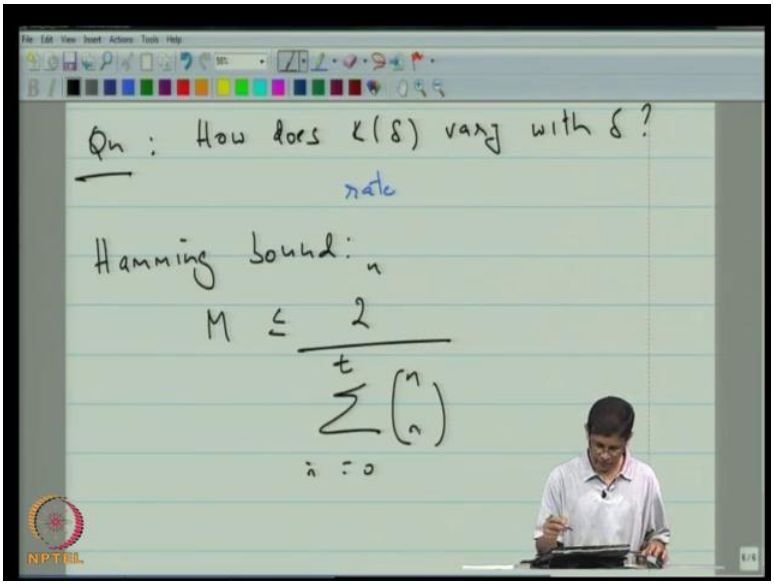

Then define since, this already definition we were not repeat myself. This is the set R delta to be the lim sup has n tense to infinity of log of M n delta divided by n. We talking will lim sup of this now. Do not get thrown away too much by the fact that you have this lim sup essentially. You are trying to take the limit only thing is that is the technicality, that sometimes the limit may not exist, but the lim sup is always guaranty to exist and what the lim sup really says is that but may be this sequence does not ten to a limit but there are sub sequences within the sequence that ten to differently method then we ages going to look at the supreme sum of all those sub sequential limits and that is guaranty to exist so this is define to be R delta.

(Refer Slide Time: 13: 43)



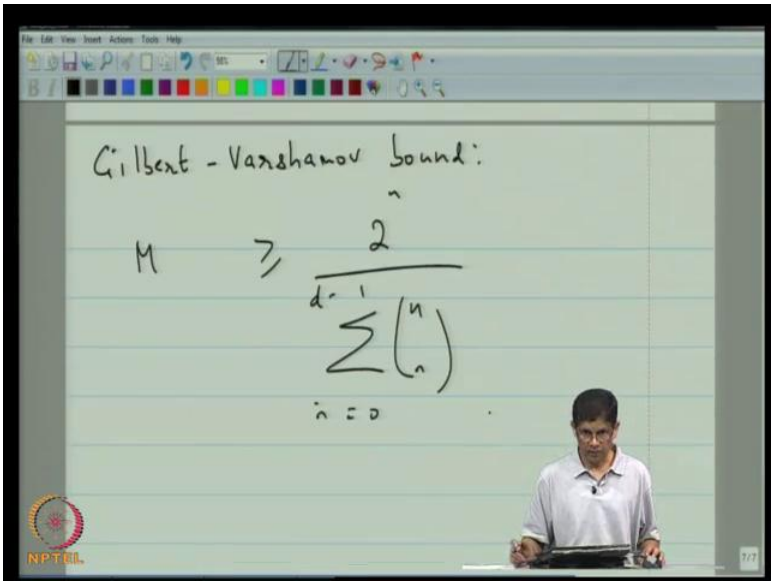

Now, what the way you should look at, this is R delta is really a talking about the rate of a code. This is this represents the rate and you think of delta s been the fractional minimum distance and that is why you set remember that we set that will set d to the n delta. Why you think of a delta is the fractional minimum distance? In this setting, what is of interest is how R delta varies with delta.

(Refer Slide Time: 14:36)



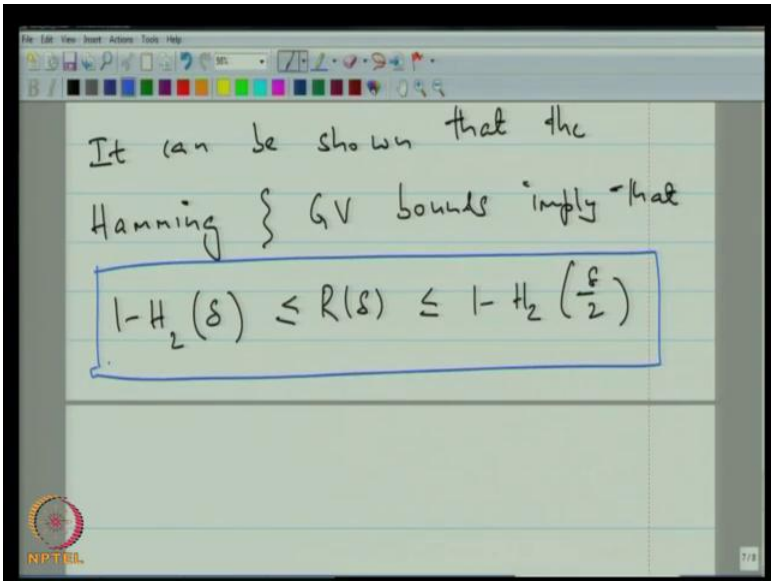
Question, how does our delta vary with delta? Or in other words, how does the rate of the code vary with fractional minimum distance? The hamming bound the hamming bound tell is tell is that M is less than or equal to 2, the n divided by the sum n choose i.

(Refer Slide Time: 15:33)



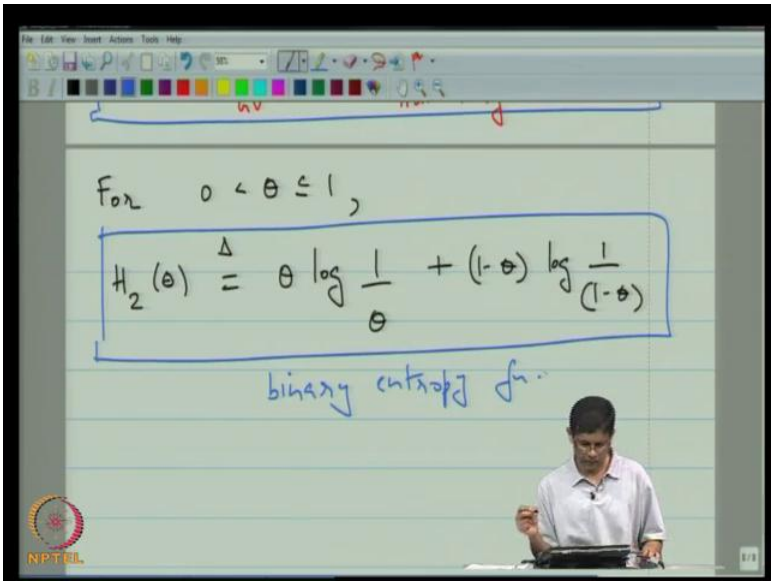

On the other hand, the Gilbert varshanov on bound says that is m greater than or equal to 2, the n now, the way the way of return the hamming in the Gilbert varshanov bounds. They are written for any value of n. Not just for n tell into infinity, but in terms that you can simplify this expression for the case, when n tense to infinity and then for large n.

(Refer Slide Time: 16:26)



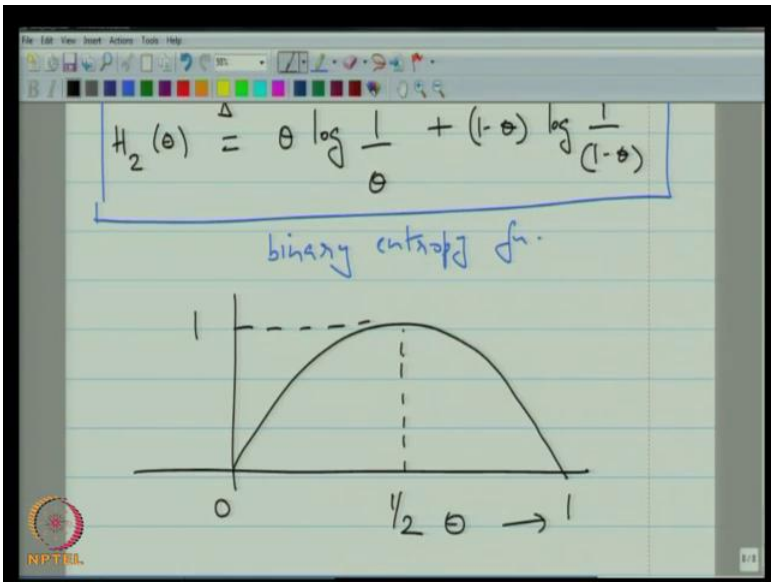

It can be shown it can be shown that the hamming and Gilbert varshanov bounds imply, that R delta is greater than or equal to 1 minus H 2 of delta by two corrections, you have small correction this should be delta and is less than 1 minus H 2 of delta by 2. This part this part comes from Gilbert varshanov where is, this part comes from the hamming bound. Now, how does thus I will just tell you quickly in words? How you get this bound? Basically, this bound comes from telling approximation for n factorial you can actually, bound n factorial in terms of this age function.

(Refer Slide Time: 18:17)




What is this age function? So for me for delta lying between 0, and 1 H 2 of delta or if that is confusing backs. I just call it each to of theta, let is defined to be theta log one upon theta, again I get side do want this to be equal to 0 plus 1 minus theta log one upon one minus theta and for those, who taken information theory? This is nothing but the binary entropy function.

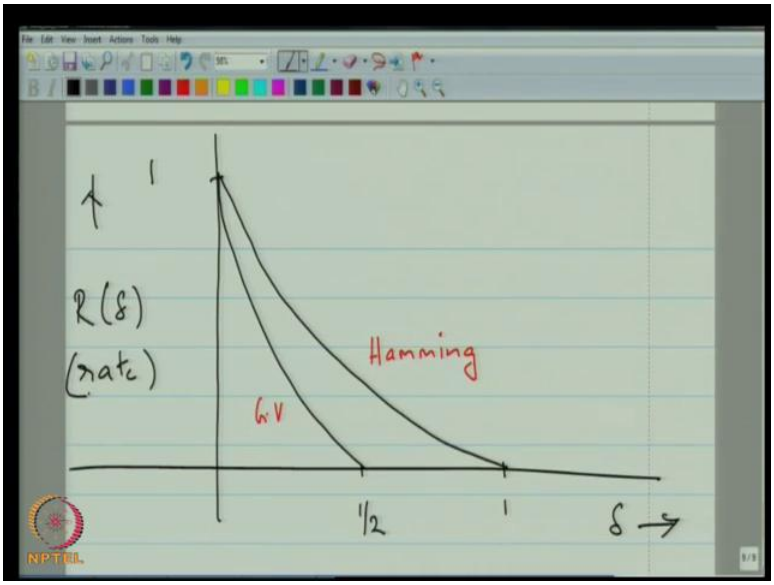
(Refer Slide Time: 19:31)




This function if when plotted looks like, theta as a function of theta it goes up it goes up, and then comes down it attains a maximum value of 1 a theta equal to one half and is 0 at either

extreme. Here what you do is you interpret a value at 0 you interpret this quantity as 0 as 0 times infinity is still 0 and in that sense and similarly 0, here, perhaps let me, just with that common put this back in and this is picture here.

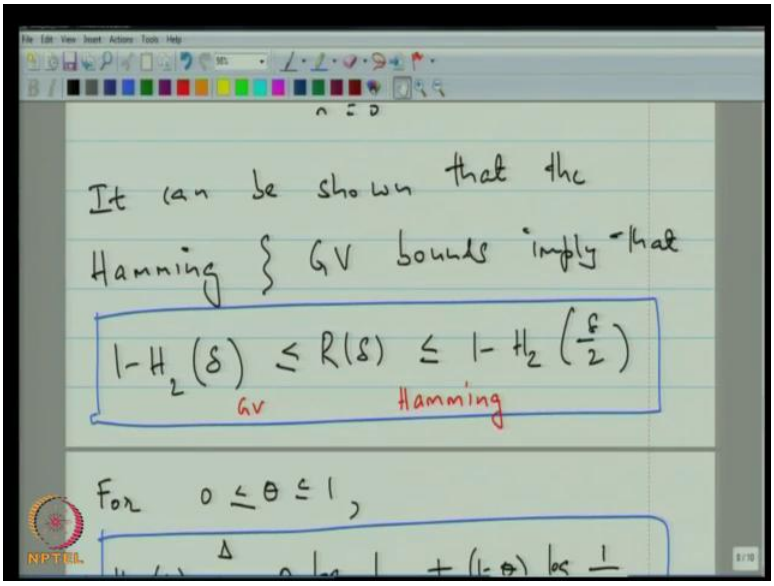
(Refer Slide Time: 21:00)




If, you use this and plot you will absorb the following you will absorb that, the he what we have plotting here, on this axis s delta and on this axis R delta and hamming and Gilbert varshanov bounds. Let packing draw this correctly, they give you this bound, so that the hamming bound the Gilbert varshanov bound is this bound. The hamming and this low bound is the Gilbert varshanov bound and you should interpret this. As stated earlier as the rate of the code. I think of rate it in the case of linear code the rate would simply, be the dimension divided by the length of the code and delta.

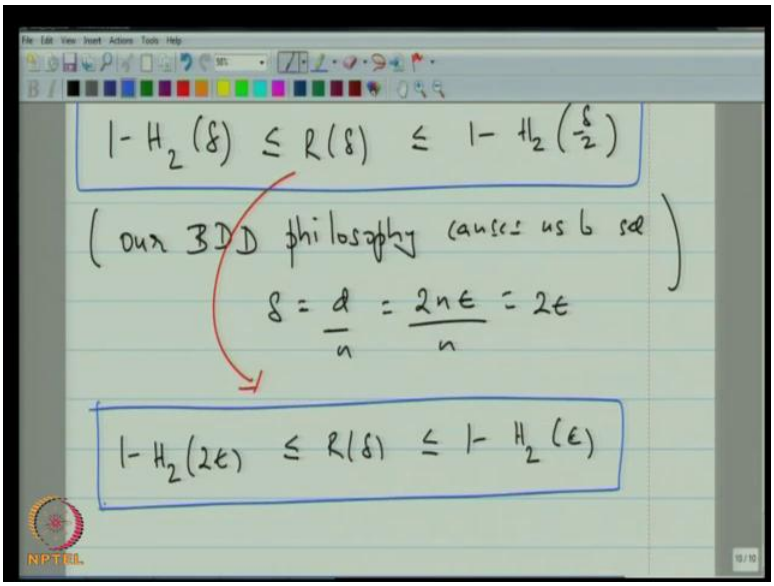
Since, you are actually plotting something which is like the relate to versus minimum distance expect that your normalizing, then by the block length of the code instead of the dimension. You have the rate of the code and instead of distance minimum distance of the code you have the fractional minimum distance, and what this is telling you is that if you are looking for the best long codes then the best long codes, on this asymptotic plot like somewhere between the Gilbert varshanov and in the hamming bound, so in some sense we potentially lie somewhere in this region.

(Refer Slide Time: 23:15)




Now, we will compact to this plot in just a second, let take a look at this expression that you just written it tells us that the maximum rate, at which you can communicate is between, these two on the other hand, the limit put that down just for clarity. I am going to repeat that expression.

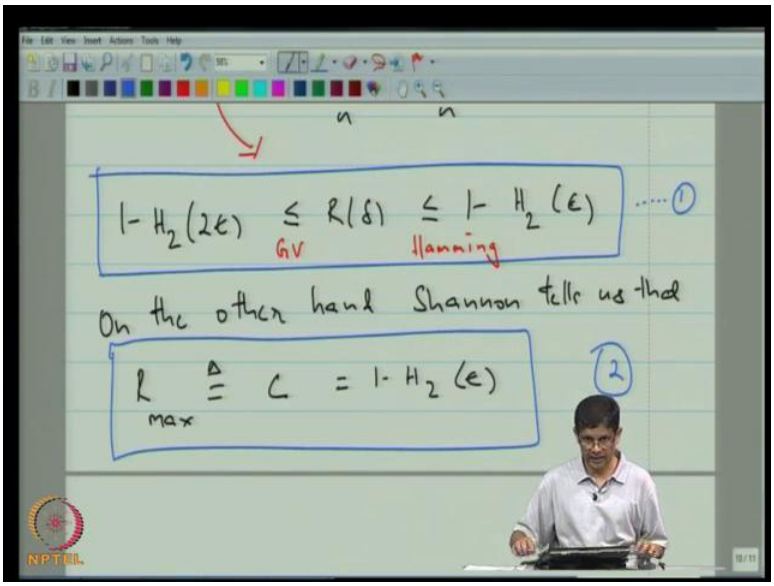
(Refer Slide Time: 23:33)




We know that R delta is less than one minus H 2 of delta by 2 and greater that are equal to 1 minus H 2 of delta however, if you combine this with our philosophy of saying that whenever, we use code for the binary symmetric channel will take the minimum distance to be twice n

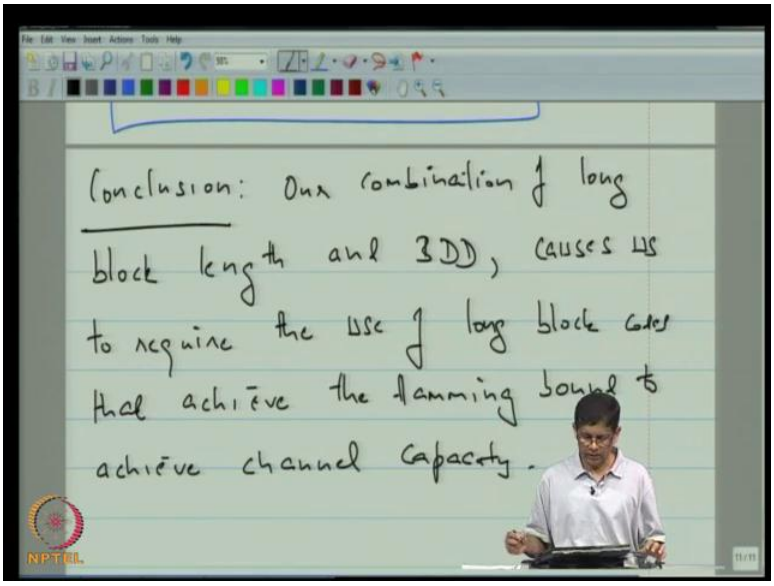
times epsilon. So, our bounded distance decoding philosophy causes as to set delta. This is d by n to be 2 n epsilon by n, which is 2 epsilon. You can interpret this as saying because of this you can actually interpret this as saying, where as saying that are delta is greater than or equal to 1 minus H 2 of two epsilon, n less than 1 minus. 1 minus H 2 of epsilon and once again this comes from the Gilbert varshanov bound and this comes from the hamming bound.

(Refer Slide Time: 25:56)



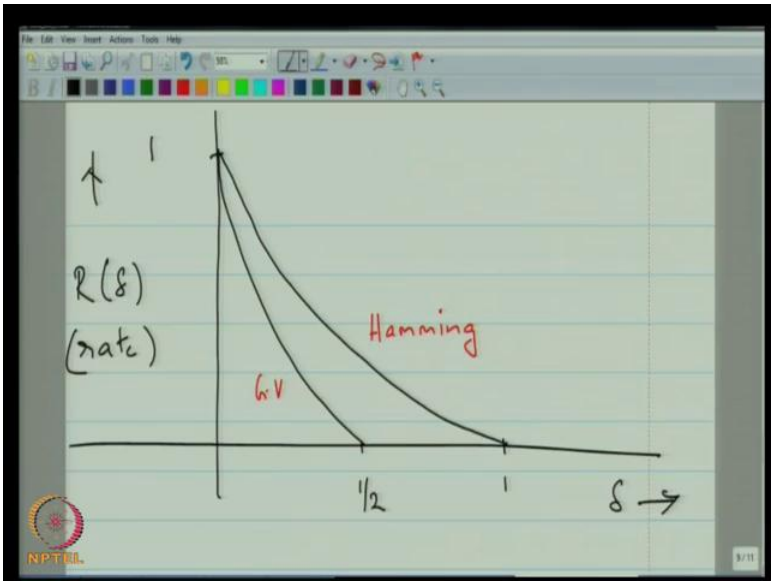

On the other hand on the other hand Shannon tells us that our capacity, so the binary symmetric channel has a certain capacity to transmit information reliably. Shannon tell us that rate or is equal to one minus H 2 of epsilon. Shannon tells is that this is the capacity of the channel in fact instead of R. Let me write or rewrite that little bit differently. I will rewrite that is R max, which is better finishing to C is equal to this. What Shannon has told as now, if you compare this expression? Let say that, I call this 1 and 2 if you compare these will see that in order to operate the channel like capacity. What you need to do this, you can see that the capacity formula in some sense agrees with the hamming bound. It is just that one way to approach to use a code that you choose the hamming bound in the quality. And use a long code philosophy; there in terms are that these problems that there are no long codes there you chief the hamming bound hamming bound in the quality.

(Refer Slide Time: 27:30)



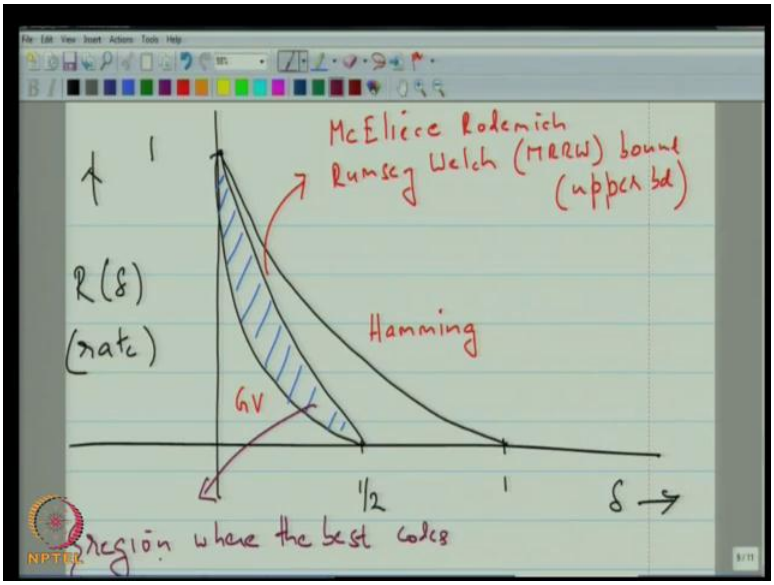

The first conclusion is that combination of long block length and bounded distance decoding causes us to require, the use of long block codes that achieve the hamming bound to achieve channel capacity. This is just our approach, but in the bad news is that in fact is, I just mention there are no long block codes that achieve hamming bound and an in fact what is even more true is that although we only discussed, two bounds.

(Refer Slide Time: 29:07)



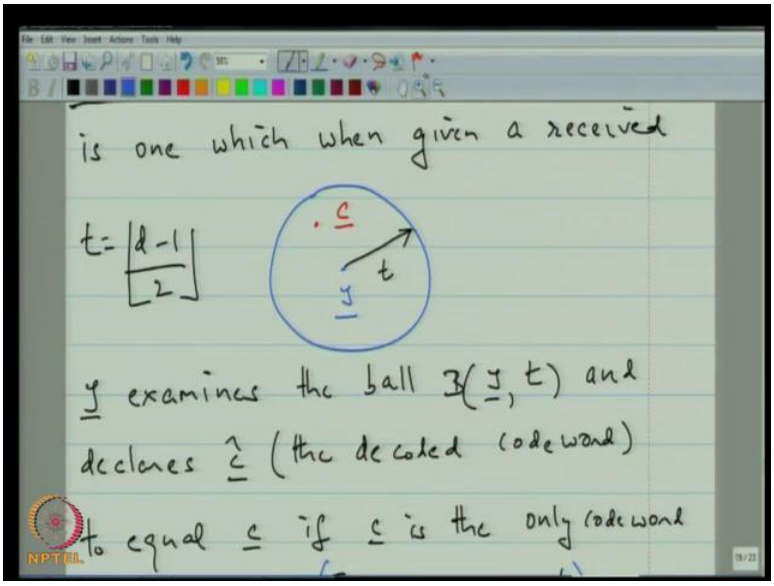
There is another bound we should discovered, sometime in the met to late seventies around seventy seven by four people from the us and which actually results in an upper bound, which has the following which is of the following type in, how packing get this write.

(Refer Slide Time: 29:28)



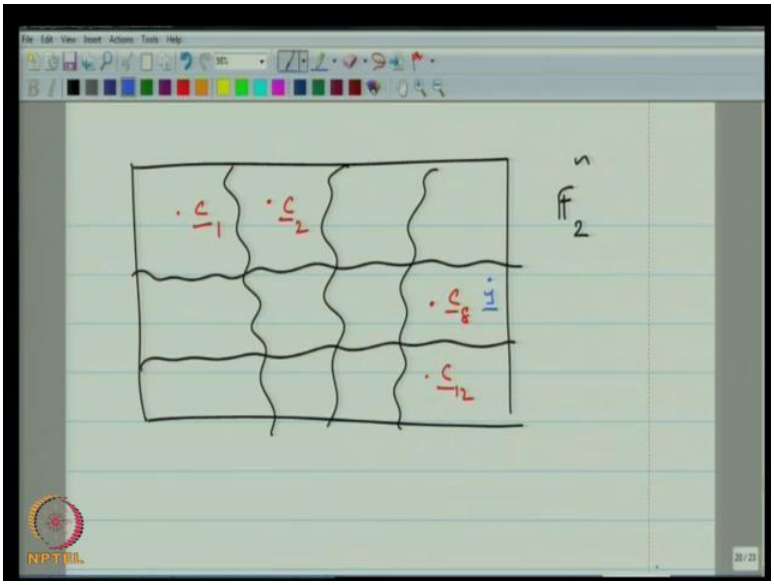

It is a bound that is goes like this. What that tells us? I think, I can improve this figure a little bit. Perhaps I should do that. Let get it of this, I will draw this more pronouns like this, so that is a much better depiction. This is the thing here is the Gilbert varshanov and new bound, which have just introduced here is called the MC Elicce Rodemich Rumsey Welch are the MRRW bound and this is also an upper bound, which says that the best codes are to be found in this region, this is the region by the best codes is to be found. I will just quick remember that green does not show up very well, let us, you say, that this is the region. Where the best codes lie so our philosophy of using the long code in conjunction with bounded distance decode in decoding is fails to achieve channel capacity, and you might say what when wrong now when wrong is precisely.

(Refer Slide Time: 31:41)




We actually discussed this in our earlier lecture; let me just quickly go back to that the bounded distance decoding philosophy. One in which you do the following you take you take the received vector then you look in a ball of radius, you will look in a ball of radius t.

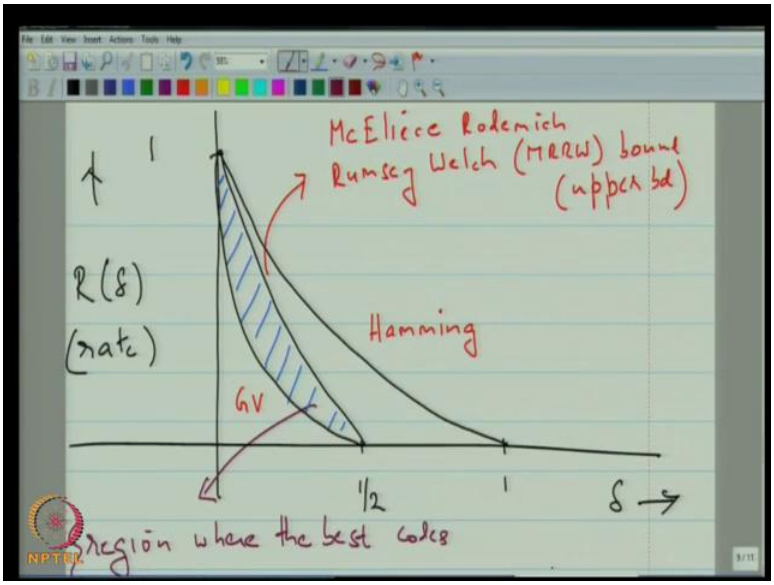
(Refer Slide Time: 32:18)




However, you can do better and we are going to look at that by doing this kind of a thing, where you actually decode to the nearest code word without regard. Whether it is in a ball of radius t or anything like that, this is actually a better method decoding. This method, here and are achieving

capacity that, I just outline feels simply, because the where insisting on doing boundary distance decoding. Since, that is only thing when you to do at that stage.

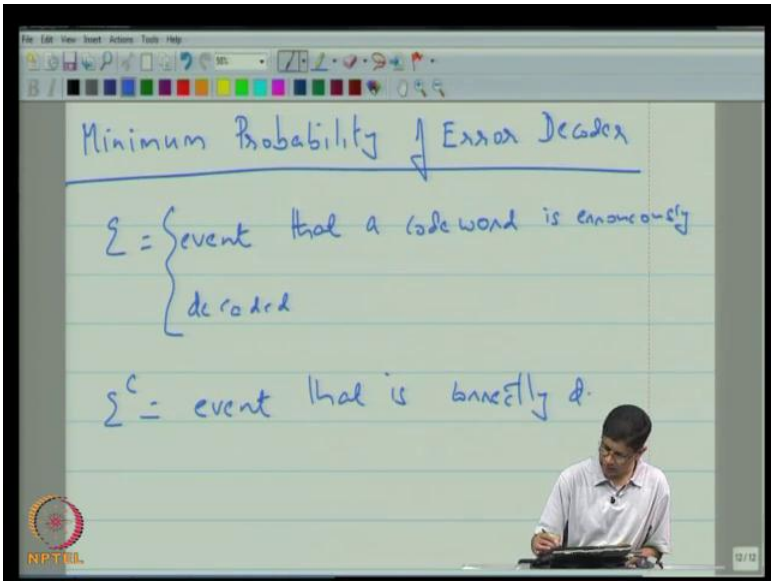
(Refer Slide Time: 33:01)




Those wines up the discussion on asymptotic bounds perhaps, it is worth of summarizing in summary. We can say that the best codes we have determine the region by the best codes lie and this region corresponds two pair one being the fractional minimum distance, which is the reflection of the error correcting capability of the code and the second thing is the rate. Now, you might say is that all the test your code and the answer is know because both rate and a fractional minimum distance are calculated based on the dimension on the code.

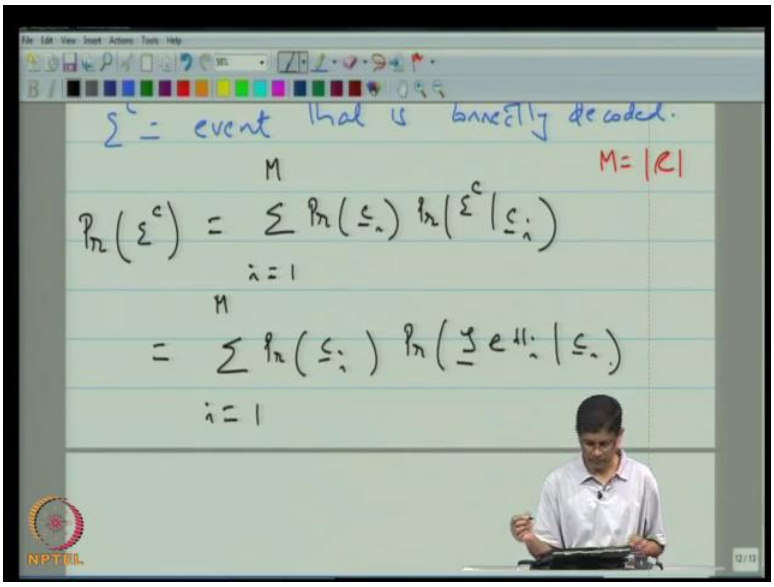
Which is the reflection of the code size that certainly important and the minimum distance of the code by the minimum distance of the code only tells you, what is the minimum separation between a pair of code words? You can imagine that you would get more complete information, if in you the complete distance distribution, of the code in other words you knew the distances is between every pair of code words. So, that is in other limitation of this view point, but anyway it is nice to have this picture, which tells you there, your limits are from the point of you refer minimum distance and that rate. A completes this part of the discussion. Next, what I would like to actually look at is other means of decoding a code. In fact will ask the question well supposing I am interested in trying to do the best possible? And that I will interpret that is saying the one that next, least number of errors.

(Refer Slide Time: 34:42)




So, will called that the minimum the minimum probability of error decoder, so the minimum probability of error decoder and we like an expression for it. Let be the event that a code word is erroneously, decoded that then e complement would be the event that it is correctly decoded, and we wanted to maximize the probability of correct decisions.

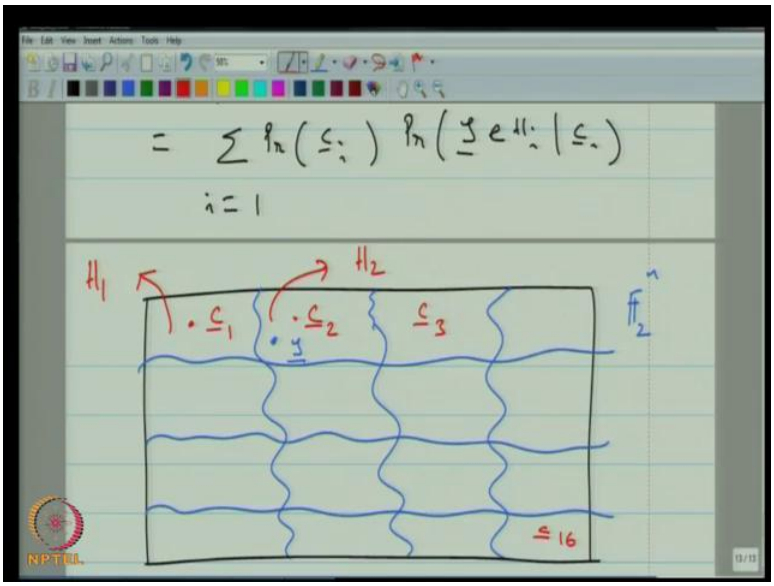
(Refer Slide Time: 36:07)




Let us look at what that would be, so that probability of e complement. So, the probability I can actually a condition this upon the probability that a certain code word is transmitted and then I

am using the total probability theorem to calculate the probability of this event, here M is as always the number of code words. So, M is the size of the code, which this expression can also be written as the sum i is equal to 1 to n probability of c i times the probability, that y belongs to h i given c i and this many some explanation, but hopefully not too much.

(Refer Slide Time: 37:20)



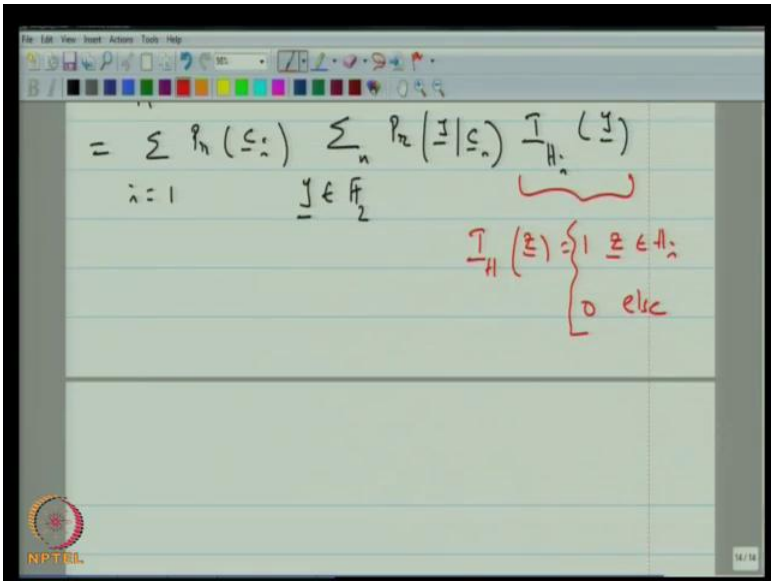
The pictures like this, any decoder any decoder, whatever is going to partition the set of all vectors into regions where, which we associate with different code words. And whenever a particular received vector lays in a region associated with the code word, then what you do is you declare that particular code word to have been transmitted. So, for example, here if this y happens to belong to this block, where c 2 is located then you actually declare that c 2 was the transmitted code word. You might say wait a minute wait a minute you talk about a bounded distance decoding and you did not tell as that was the case that is true because what bounded distance decoding does is there it looks with in a ball of radius t surrounding the received vector and if, there is no code word within it gives up so that means there are certain situation in which it does nothing.

When we talk about now obviously if instead of doing nothing in futures guess some code word you be doing better because the some small chance that you might be occasionally correct. The bounded distance decoded is a practical convenience, but from a theoretical point of view you can always find another algorithm which does the same thing as bounded distance decoding.

Expect there in should giving up when these nothing inside the ball it does something else. To from that point of view every algorithm basically impose down to associating making an association between the set of received vectors on one hand and the set of code words and this is just an abstract depiction. When I have actually, showed you these boxes, I do not necessarily mean that these vectors are contiguous to each other. This just an abstract depiction there is a certain subset of vectors.

Whenever I write each i here what I would mean is precisely this region of a here. I am calling this region here as H 1, H2, and so on. Now, this of course as nothing to do the binary entropy function these are just region something like the regions. So, the probability of a correct decision then the i th code word is transmitted is of course the probability that the received vector lies in the region, where you will declare the i th code word to have been transmit.

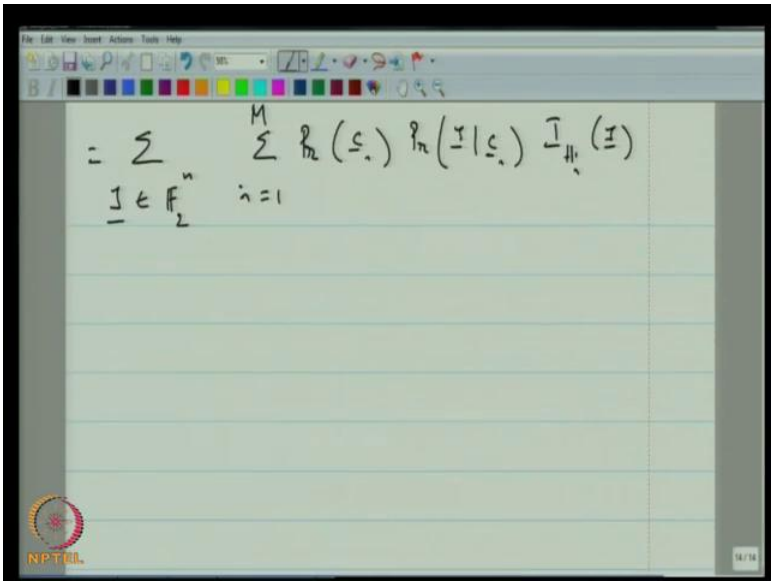
(Refer Slide Time: 40:30)



To proceeding further you can actually write this, as the sometimes the sum over all received vectors. It gives I should remind you that as always y is tense for the received vector. I am going to rewrite this last expression in a different way, am going to say that the probability of receiving y given c i times the indicator function of H i of y and what this function is given by i H of z is one z belong to H i and is zero. Else what this does is in just make sure that you associate with P of y giving c i only those vectors y which belongs to each i. To continuing on this now, I can interchange summations bring the summation on y outside and i can write the sum i is equal to

one to n probability of c i times the probability of y given c i and times I, each i of y now when you look at this expression here then if it keep in mind that as for as building a decoder goes the only thing was really in your hands.

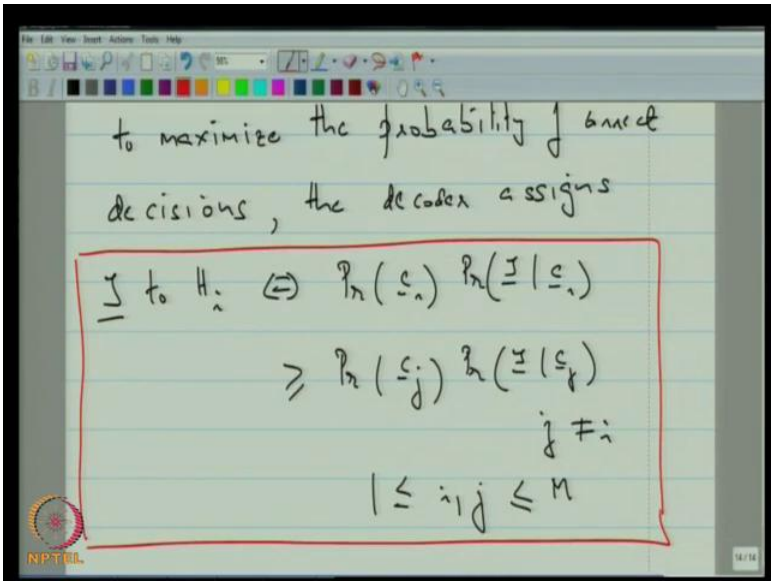
(Refer Slide Time: 43:18)



See the probability of selecting a code words s it is in your hands, but most of the time that is determinant by the transmitter end. The decoder majors have to live with whatever that, transmitter desires to do, the decoder does not have control over that the second term.

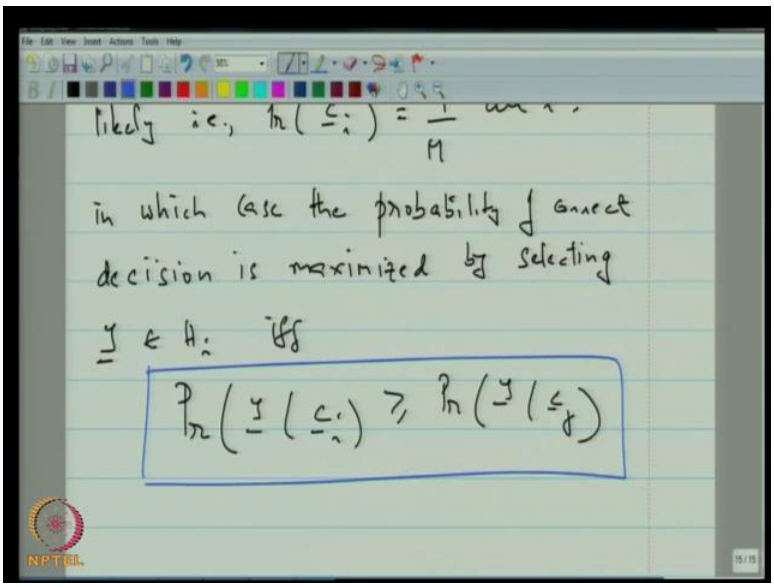
The probability of y given c I is an again something that is not in the hands of the decoder. Because that is something that happens across the channel the only thing that the decoder can really control is this i each i of y, because it selects the decoder selects the regions each i. That means the contribution, if you allocate y to each i then that put tick at the continuation of that particular y will become precisely this, quantity and you are interested in maximizing the probability of correct decision. You would like to maximize this so you would like to make an allocation of y to each i if this is the largest.

(Refer Slide Time: 44:26)



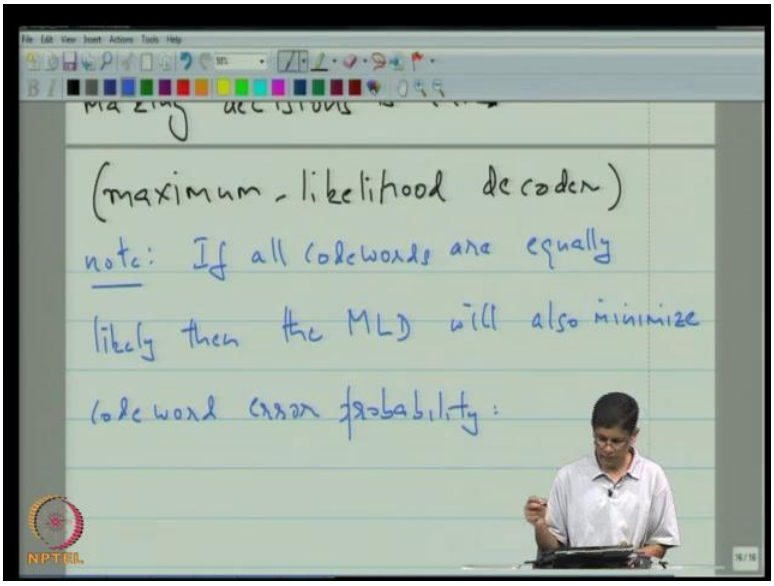

We just summarize, by saying to maximize the probability of correct decisions the decoder assigns y to each i, if and only if the probability of c i times the probability of y given c i is greater than or equal to the probability of c j times. The probability of y given c j for j not equal to i here both i and j. This is the decision rule that is adopted by a receiver that is interested in trying to maximize the probability of making a correct decision. Now, typically as I mentioned earlier.

(Refer Slide Time: 46:09)



Now, it is typically the case, all code words are equally likely, i e that probability of c i is equal to one upon m for all i. In which case when you trying to determine between these two the probability of c i now, goes out to the picture, because it is the constant, so you decide in which case the probability of correct decision is maximized by selecting y and h i. If, that probability of y given c i is the greater than or equal to the probability of y given c j. Now, a decoder which makes decisions based on this rule is called a maximum likely hood decoder. A decoder that uses this rule for making decisions is called a maximum likelihood decoder; just a quick comment a maximum likelihood decoder will employ this rule whether or not the code words are equally likely. But what we shown here, is that if the code words are equally likely than the maximum likelihood code word decoder minimizes the error probability.

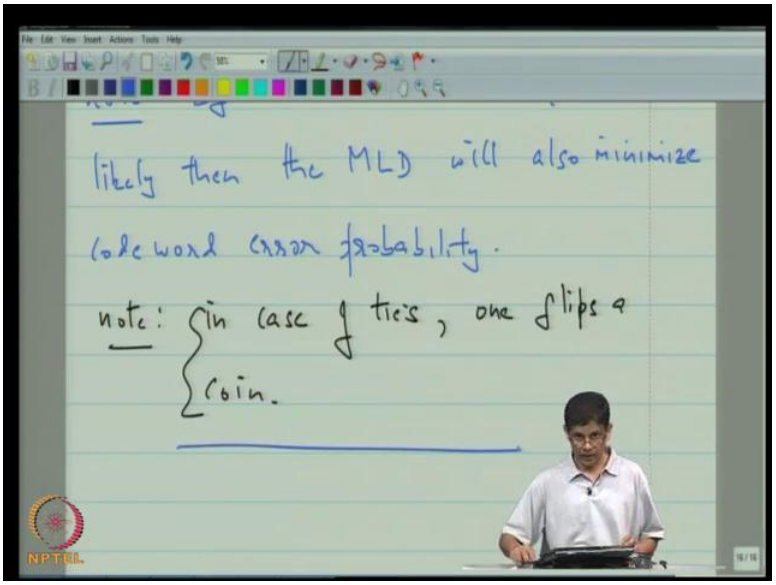
(Refer Slide Time: 49:26)




Note, if all code words are equally likely than the maximum likelihood decoder will also minimize code word error probability. This is another comment which is that behave this greater than or equal to though. This might cause some confusion, when you say well wait a minute, what if two code words the probabilities are the same then what I do and answer is simple just a looped coin choose between them in some arbitrary fashion. Only thing is that your decoder always has to be deterministic in the sense that given a y m as consistently, decode to certain code word. You cannot a flow to say out I am going to decode today to y one, tomorrow to y two for the same receive writer, but when you have designing your decoder and you are face to the

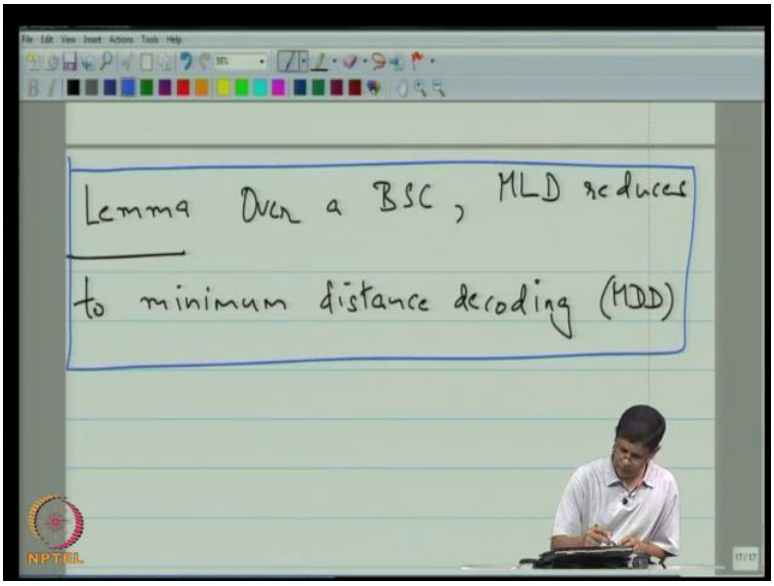
choice because the probabilities are equal there you just flip a coin. Once it is once you flip the coin then you made a decision are taking well, thus is clarification.

(Refer Slide Time: 51:04)



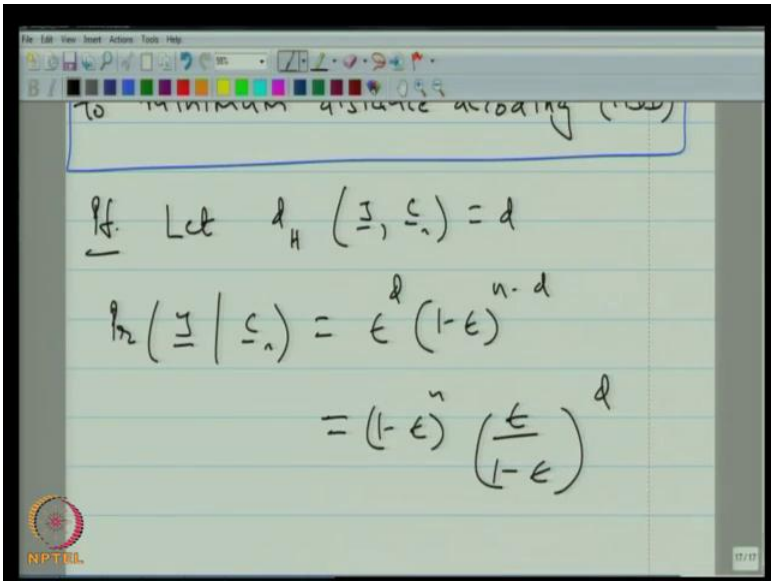

The most people just say that explain, a way this situation by saying in case of ties in case of ties on simply, one flips a coin and from that the meaning should be clear. Now, another question is, that mind me when you are talking about decoding and you are describing these regions? Such something about hamming distance is said that, you might want to choose the regions in such a way that those vectors there are in close in hamming distance would have some would almost likely being this regions in other words. We have to make the connection between maximum likelihood decoding, which after all thus the best that you can hope to do and the code word thus, equally likely and what you intuit ably want to do is decode to the nearest code word. I will just state that is a lemma.

(Refer Slide Time: 52:11)



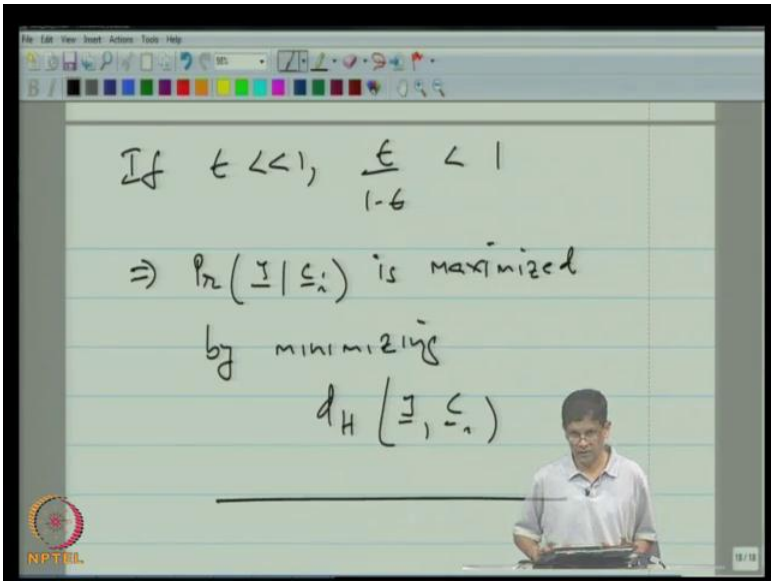

I am not show we will have enough time to prove it today, but let me a just state it lemma over a binary symmetric channel, maximum likely decoding reduces to minimum distance decoding, which will write as MDD?

(Refer Slide Time: 52:53)




So, the proof, let the hamming distance between a y and a c i be d then the probability of y given c i is epsilon to the d 1 minus epsilon to the n minus d, which you can write as 1 minus epsilon to the n divided by 1 minus epsilon to the d.

(Refer Slide Time: 53:53)




Now, if epsilon is much smaller than 1 is epsilon upon 1 minus epsilon will be greater than will be less than 1, which will imply that the probability of y is maximized by the minimizing. The hamming distance between y and c i and that is the connection. You will pick up on this am I repeat myself on this little bit next time, and then will actually expand this into a complete decoding algorithm, which is very simple and which does exactly this minimizes the probability of error, and in that sense which is an optimal decoder. Let stop here, I will recap and continue from this point onwards in the next lecture.