Modern Digital Communication Techniques
Prof. Suvra Sekhar Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Lecture – 10
Source Coding (Contd.)

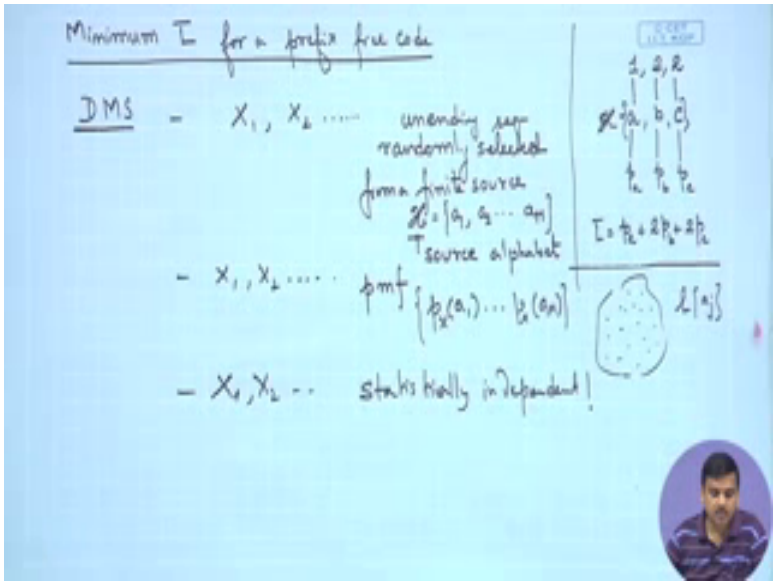
Welcome to the lectures in modern digital communication techniques. So, what we have seen in the previous lecture is the Kraft inequality which tells us that what are the code word lengths which can be used to construct a prefix free code. If someone tells you that I want to use a code word lengths of 1 1 and 2 then you can clearly establish that it is not possible to construct a prefix free code because you are going to get 2 to the power of minus 1 plus 2 to the power of minus 1 plus 2 to the power of minus 2. So, that is more than 1 it does not satisfy the Kraft inequality we cannot construct prefix free code and that the example that we had for this particular numbers are the code which are like 0, the second code was 1, and the third code was 1 0 which could be an send to a b and c and where we found that it is not possible to construct prefix free codes with these codes. And what we can now establish is that without verifying with the prefix free property goals or not by simply not satisfying the Kraft inequality it can tell us that it is not possible to construct the prefix free code.

So, now once we have done that what we have find is that could be of our interest is what is the minimum length on these code words that one could find now this is very important because all we are trying to do is to find the minimum number of bits that could be used in the source so, that when this bits go out or when the symbols go out into the channel we take the least amount of bandwidth or least amount of capacity to transmit this information.

Now if you could ask or if you would ask that why would you ask this question it takes a minimum bandwidth we would like to convey the information with as little resource as possible. Now this discussion is important because one of the important things that digital communication engineers do is that they want to reduce the resource require. If you think in terms of resource especially for wireless communications resource is very very limited because bandwidth is limited power is limited and along with many other resources as well. So, if you talk of limited bandwidth you would like to occupy as little

bandwidth as possible for one particular source, the reason is you can connect many sources to many destination so that you utilise the bandwidth as efficiently as possible and will see the term towards the end of this particular course where you will use the terminology spectral efficiency. So, this is why we are doing this for first part of the communications course on source coding.

(Refer Slide Time: 03:10)



So, moving on further we would like to find the minimum L bar so that means, we have to talk about L bar. L bar would be the average length of code words now what we have seen is that there code word length could be 1 it could be 2 and 2. So, if these are three code word lengths if someone ask us that what is the average length of the code word. So, we will go back and we will see that one the length one was for b as for a and b had length of 2, and c had a length of 2.

Now, if a has a certain probability p of a, b has a certain probability p of b, and c has a certain probability p of c then one could say that it is p of a plus twice p of b because two bits plus 2 p of c is the average length of this particular source which has these symbols right. So that means, we are talking about the average, now the question that one can ask is that well if you assign particular source a set of code words then or code word lengths then one could assign or one could ask the question what is the minimum of all possible averages. So, why we have this question because if I have a source which is very which has a large number of symbols in it right, and these symbols are assigned to certain

lengths right. So, when we say that L bar, L bar is the average length for particular mapping what we trying to what we are interested in is amongst all possible mapping of lengths to code word lengths this lengths to code words, which particular combination gives us the minimum or rather what is the minimum value then one go ahead and find the mapping which gives this minimum value. So, with the intension that we would like to reduce the number of bits per source symbol.

So, if we are talking about the minimum L bar for a prefix free code, now when we talk about minimum L bar for a prefix free code we would begin with a discreet memory less source. So, when we say a discreet memory less source you can go into the definition of discreet memory less source, but what we have done is almost defined a discreet memory less source with the examples that we have taken. So, we can say that it generates an unending sequence of x 1, x 2 and so on. So, these you can say there is an unending sequence right it is a source so that means, it keeps on a continuously generating this symbols. So, these are the symbols which are randomly selected, which are randomly selected from a finite source let us say x, which contains a 1, a 2, a m right. So, this is the source alphabet right. So, these are the symbols. So, x 1 could be this x 2 could be this and so on and so forth and each source output; that means, this x 1 x 2 dot dot dot all these are selected with some probability mass function and probability mass function would be p of x because this is this this alphabet p of x of a 1, up to p of x of a m right.

So, there is a certain probability assigned to these particular symbols; when we say that assigned it is not actually something that we assigned to a source, it is just source inherent nature. So, if we take this particular lecture for example, and we look at all the things that we write and we try to plot the histogram of the letters in it, and normalise that histogram then we can possibly find the probability of different letters. One could also find probability of the different words that we are using so that means, if I look at the source as one which generates this letters for this particular lecture that we are looking at we are interested in, we will find a certain set of probabilities like.

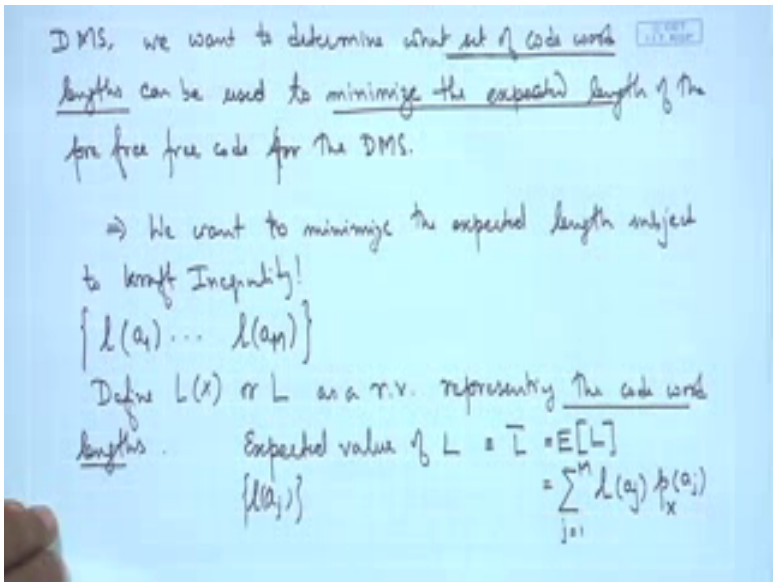
If you are looking at some other lecture and of a different subject then there is high chance that you would encounter a different probability the reason being in this particular lecture we are concerned with the particular subject and we are using some words more often than if it was different subject and the other object or the other reason could be that we are using some particular letters more often for example, I have used a

is quite often right and I have used x is quite often. So, if it was not this particular lecture and some other lecture and people could have been use the letters let us say z, and they could have use gamma they could have use alpha. So, they could have been higher probability of those than these letters that we are looking at.

So, it is an inherent nature of the source right. So, that is of course, what we are not assigning, but it is present and each of these x the symbols they are statistically independent. This is one of the most important things which we did not see it till now we are almost seen the first two postulates the third one tells us that when one of the symbol comes out it is independent of what has happened before and what has towards to going to happen afterwards. For example, if we take the example of tossing of these dye or throwing of the dye what we find is that the probability if it is a fair dye of course, when we are playing ludo I mean it does not work out, that it often happens for us this 6 does not land up more often compare to one of our partners. Now it is very difficult to explain that, but if you take the all possible throws what we find is that in that particular game that the probability of getting this 6, of probability of getting a 3, and probability of getting a 1 over several trails turns out to be almost equal.

Right and; that means, there of equal probability and the next equal probability does not necessarily see that it is independent what we are trying to say here is that once I throw it if I get a 6 there is no relationship with what is going to happen in the next one or vice versa if I have thrown a 6 or if I get a 6 it is no relationship with that whether 6 is happened before that or whether one is happened before that. So, statistically independent would in other words what does it means is that there is no memory in it right. So, we have talked about discreet and we have talked about memory less ness.

(Refer Slide Time: 11:16)



So that means, source what we have already taken, but now we add the term that there is no memory in this particular source right.

So, we are taking discreet memory less source and we want to determine what set of code word lengths can be used to minimize the expected lengths of the prefix free code for the discreet memory less source. So, we are talking about this set of code word lengths; that means, l of a, l of b up to l of let us say z or l of a 1 l of a 2, l of a 3 up to l of a of m which can be used to minimize the expected length of it right.

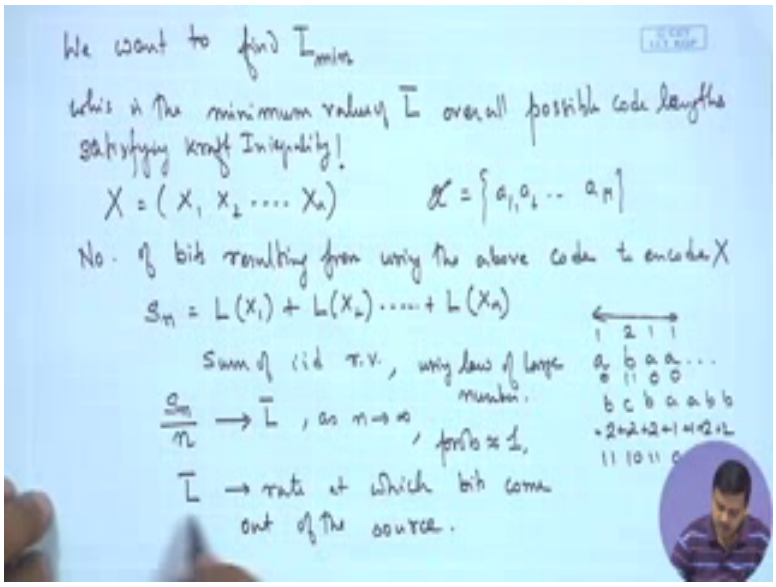
So, in other words what we want to say is that, we want to minimize the expected length subject to Kraft inequality why do we bringing this second statement the reason of bringing in the second statement is that we are talking about prefix free code so; that means, we have a discreet memory less source and we are saying that the codes will be prefix free. So, even if these are prefix free we have to minimize the length. So, the moments we say prefix free code we can immediately jump on to this Kraft inequality. So, to begin with this we will say that let there be this lengths l of a 1 up to l of a m which we have used before and of course, these lengths satisfy the Kraft inequality. So, it means these lengths already satisfy Kraft inequality because it is a prefix free code.

And then we would define L of x or simply L as a random variable representing the code word lengths for randomly selected symbols; that means, this could be this this this this or anything. So, this would take a value if I have let us say 3 5 6 and 7 as four possible

lengths then I could say that capital l chooses any of these values of lengths right. So, this is basically the random variable representing the code word length selected from this set. So, now, we write the expected value of L for a given code word what we could write this as L bar. So, this is what we have been writing is equal to E of L now this is the notation we are going to use e means expectation and we have already done this calculation in the previous note that we were writing this is simply sum of j 1 to m because there are m symbols length of the j th code word multiplied by the probability of the j th code word following this particular random sequence.

So, this we have already calculated in our previous example here we have already done this calculation. So, that is what we are writing more formally over here. So, this is. So, now, what we are trying to say is that we would like to minimize this L bar because you could assign different lengths to different ages right or you could assign different lengths to different probabilities. So, would like to find this match between these two. So, that over all possible lengths sets; that means, if l of a j over all possible lengths sets, which particular length sets would minimize this value would give me this smallest value of this. So, that is we can write.

(Refer Slide Time: 16:04)



So, we say that we want to find L bar min we would like to find L bar min, which is the minimum value of L bar over all possible code lengths satisfying Kraft inequality. So, to proceed with this what we do is we take long sequence X. So, let be a let x be a long

sequence. So, what we mean is that I take a sequence of symbols and I take them x 1, x 2 x 3, x 4 up to x n. So, this is what I mean by a capital X.

And these are of course; selected from these alphabets and they have these different lengths. Now the number of bits resulting from using the above code word lengths or the above code to encode this long block or to encode x let us say in short. So, we could say that S n is the sum of this bits which is length of x 1. So, I have this long sequence plus the length of x 2 up to plus length of x n look at this what I have done is I have taken x 1 x 2, x 3, x n up to x n as a long sequence. So, x n will have a length L of x 1 because these are random variable. So, that is why I have taken L of x 1, L of x 2 will be the number of bits required to encode x 2 and l of x n will be the number of bits required to represent x n for example, if this is a this is b then again a and a, a and a I would have one going by our previous example here I will take length of 1 plus 2 then I will take one plus one plus one so; that means, if I have a sequence a b a a dot dot dot or if I had a sequence let us say b c b a a b b in this case the length of 1 2 1 1 would be taken in this case 2 2 2 1 12 2 would be taken.

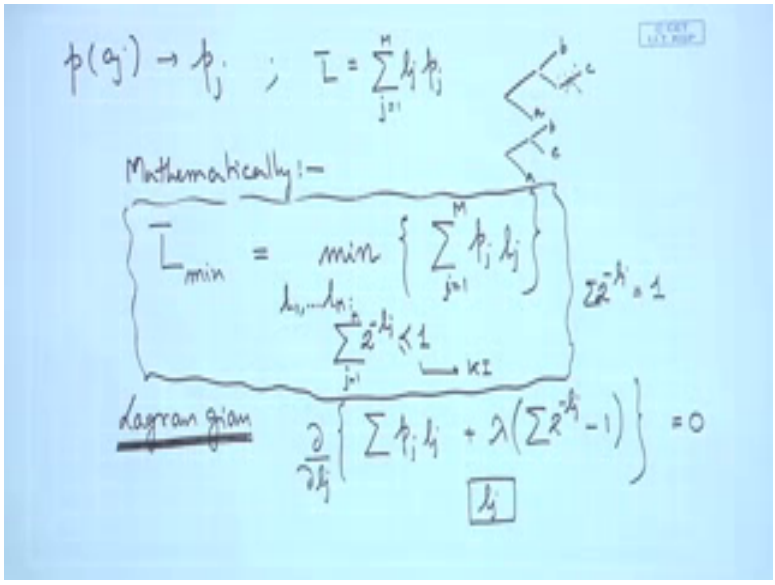
This sum would be the number of bits because b would be in this case 1 1 and c would be a 1 0 a 1 1 is 0 0 1111 where is in this case this is 0 this is 1 1 0 and 0. So, here is an 1 2 3 4 5 bits in 4 symbols, here if I take 1 2 3 4 symbols I have 1 2 3 4 5 6 7 bits. So, this sum is what is being represented here. So, if, now, these l s that you see over here these are IID random variables because these x s are IID random variables. So, therefore, definitely lengths are also IID random variables. So, if these are random variables these S n we can say it is a sum of IID random variables right if it is true in that case we can say that s n upon n this ratio would tend to the expected value of n if n is very large. So, using law of large numbers we could say this as n tends to infinity and not only that we can say that this approaches this with this nearly probability of this happening is almost close to one.

Now, if you are used to the asymptotic equipartition, then also this result would come true; that means, if I am taking a very very long sequence then the and I am and I am adding the up these numbers the lengths of it and dividing it by n what you end up with is that this number tends towards the expected value of that particular sum or the individual numbers and that happens with probability almost equal to one and this happens only when n is very large. So, if I take 4 of them I cannot ensure that is going to

happen, but if you going to have it as very very large sequence let us say I take 100 or 1000 or 10000 such symbols in one sequence, and I do this then I can find that this number would be almost equal to the average value; that means, the sample mean is almost equal to the un sampled mean. So, you can do it like that this is law of large numbers approximately and if n is very large, this is almost with the very high probability.

Now, using this you can say that L bar that what you have is basically the rate at which the bits come out of the source right, since because this is the average number of bits.

(Refer Slide Time: 22:30)



Now once we identify that this is the average number of bits that come out of the source then what we can continue on is let p of a j will be written as p j that is the probability of getting this particular symbol or probability of getting this particular length. So, we can say that L bar is equal to sum over j equals to 1 to m we have already said this in a different way that we write it like this l j p j. What we have written before is that this is equal to l of a j times p of a j. So, we have shorten this and said l j times p j.

Now what we have is that this problem of minimizing mathematically you can state that L bar min; that means, the minimum value of L bar is equal to the minimum value of this sum j equals to 1 to M let us say p of j times l j or l j times p j does not matter. So, basically you are trying to minimize l bar, but you have the constraint that you have this

values which are given l 1 to l m subject to be Kraft inequality; that means, 2 to the power of minus l j j equals to 1 to m should be less than or equal to 1.

So; that means, overall possible lengths subject to this condition I would like to minimize this. Now this condition that we have over here is because of the Kraft inequality and what we have said is that this Kraft inequality comes in because we have this prefix free condition. So, we said that overall possible code word lengths which satisfy the prefix free condition we would like to minimise the code word length. In order to proceed with this we are going to use the Lagrangian method.

So, you can use the Lagrangian technique for this. So, of course, we have a particular relaxation now if you look at this particular problem these lengths that are there these are integer lengths. So, it is not possible to do a Lagrangian with integer lengths because you have to take this derivative. So, the first relaxation that we make is we will assume let this lengths to be real values and if Is lengths sorry are real values whatever number we get out of this could be a good approximation could be what possible lengths that we can choose from which you are going to minimize.

Now once we get these real approximations then we can find the nearest integer which is the best match to this and could be are possible solutions. So, the cost function we could write it down as sum over p j, l j plus we use Lagrangian multiplier and we have 2 to the power of minus l j is has this particular constraint minus 1.

Now why we take this of course, because we are seen that a full tree cannot be minimized; that means, if the tree is not full what is it mean the tree is not full means you can shorten the tree see. If you can shorten the tree; that means, the lengths becomes less see if the length becomes less; that means, the previous condition it is less than or equal to one, but when you are reduce the branch if you remember that we have we had this tree and we said that this length is extra you need you do not need this this length. So, instead you can draw tree right. So, the moment we compare these two trees what we find is that they were three code words, in this a b and c there are again the same code words in this, but c has a length which is smaller than this and c is also a prefix free code; that means, this tree is a prefix free tree this is the prefix free tree, but since this is full.

What it means is that you will have a length which is lesser than this that means, the average of this or L bar of this is going to be less than this otherwise we are not having the best possible length combination. So, therefore, instead of taking less than or equal to one we will use the constraint two to the power of minus l j sum over is equal to one because this is going to minimize it.

So, with this as our problem you could take a derivative of this with respect to l j now since I have to take a derivative that is why we said initially that let our constraint be relaxed and we can say that let these real numbers once we solve for it then we can find a better number integer number which is matching to this. So, once you take the derivative and said this equal to 0 and use this constraint that we have used over here we can find l j s which could be useful; the set of lengths which are going to minimize and solve this particular problem right.

So, we stop this particular lecture over here and we will carry one in the next lecture with this particular derivation and we will see that at what particular lengths find that we find would be helping us to minimize the expected length of the code words so, that we can use as minimum number of bits as possible to encode a particular source.

Thank you.