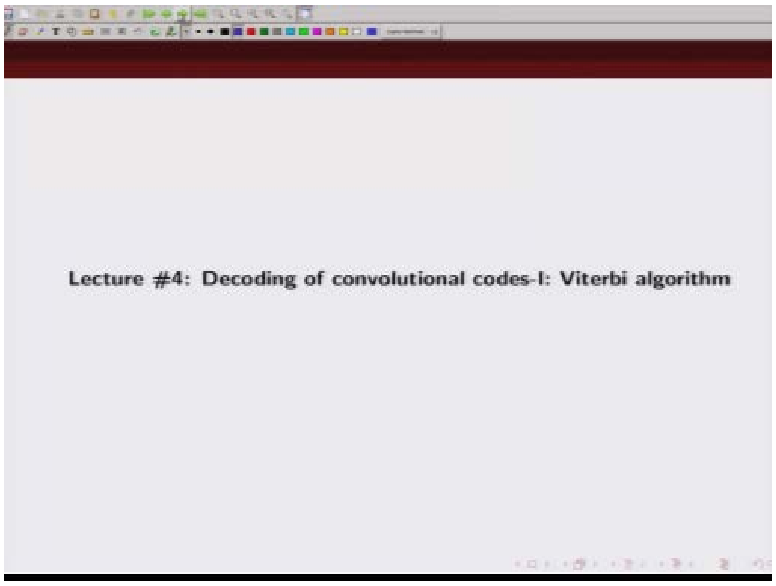
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Convolutional Codes

Lecture-4
Decoding of Convolutional Codes-I: Viterbi Algorithm

by
Prof. Adrish Banerjee
Dept. Electrical Engineering, IIT Kanpur

Welcome to the course on error control coding, an introduction to convolutional codes.

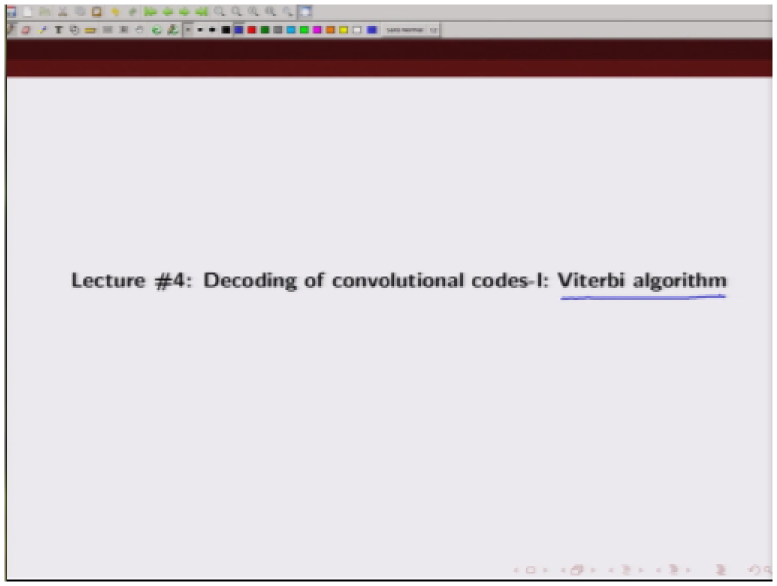
(Refer Slide Time: 00:21)



Lecture #4: Decoding of convolutional codes-I: Viterbi algorithm

Today we are going to discuss how to decode convolutional code.

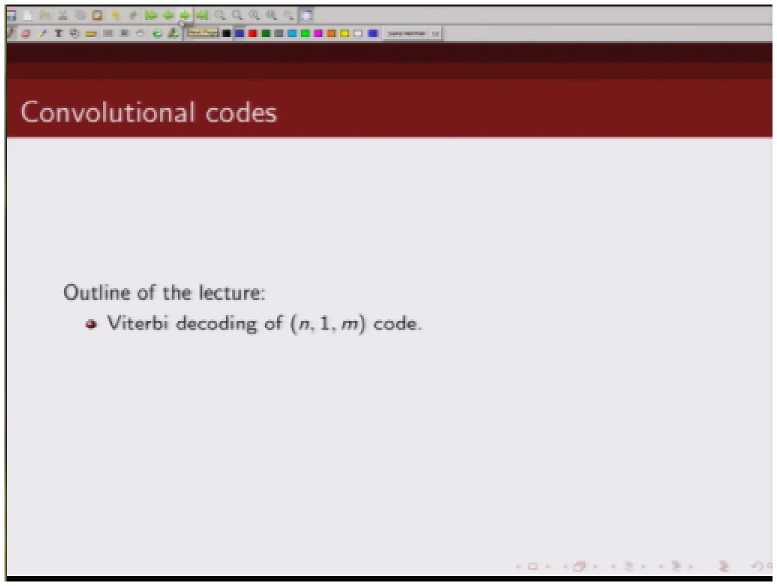
(Refer Slide Time: 00:26)



Lecture #4: Decoding of convolutional codes-I: Viterbi algorithm

And in this regard we are going to talk about a maximum likelihood decoding algorithm known as Viterbi algorithm for decoding of convolutional codes.

(Refer Slide Time: 00:39)



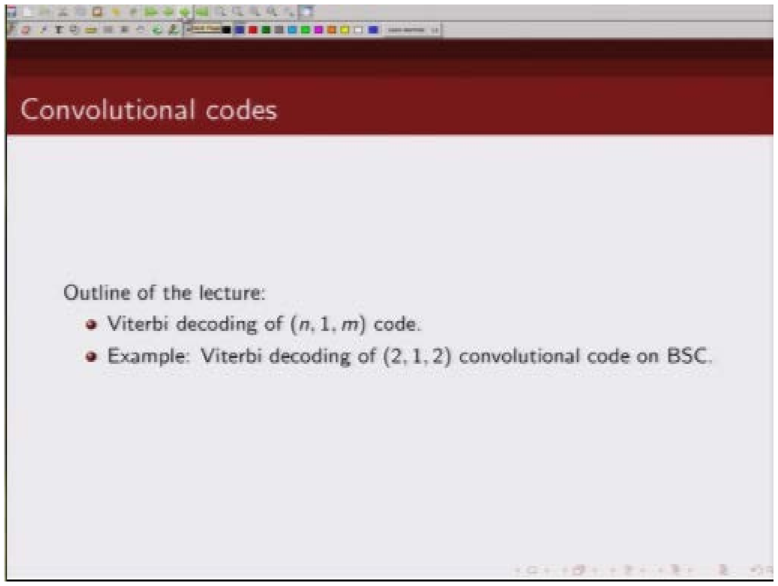
Convolutional codes

Outline of the lecture:

- Viterbi decoding of $(n, 1, m)$ code.

So we are going to illustrate Viterbi decoding of a rate 1/n convolutional code whose memory order is m.

(Refer Slide Time: 00:51)



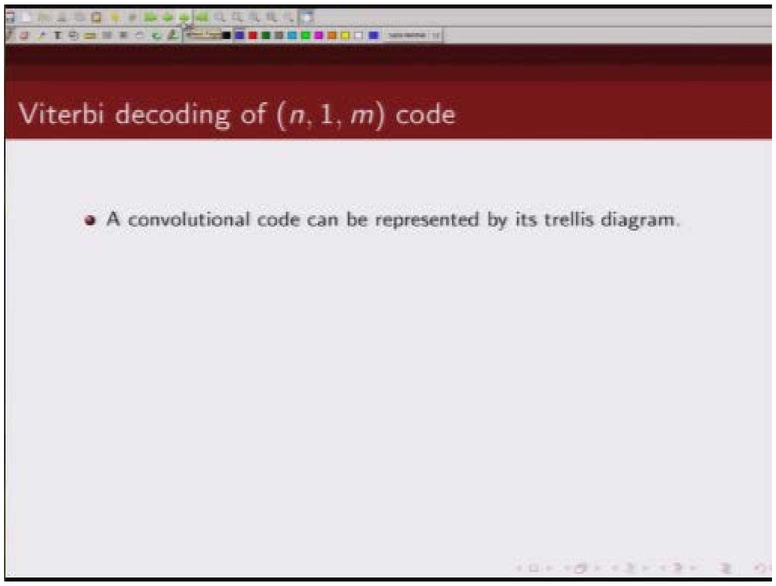
Convolutional codes

Outline of the lecture:

- Viterbi decoding of $(n, 1, m)$ code.
- Example: Viterbi decoding of $(2, 1, 2)$ convolutional code on BSC.

And we are going to illustrate this decoding using a simple example of rate ½ convolutional code whose memory order is 2. In other words it has four states and we are going to assume that our transmission medium channel is a binary symmetric channel.

(Refer Slide Time: 01:19)

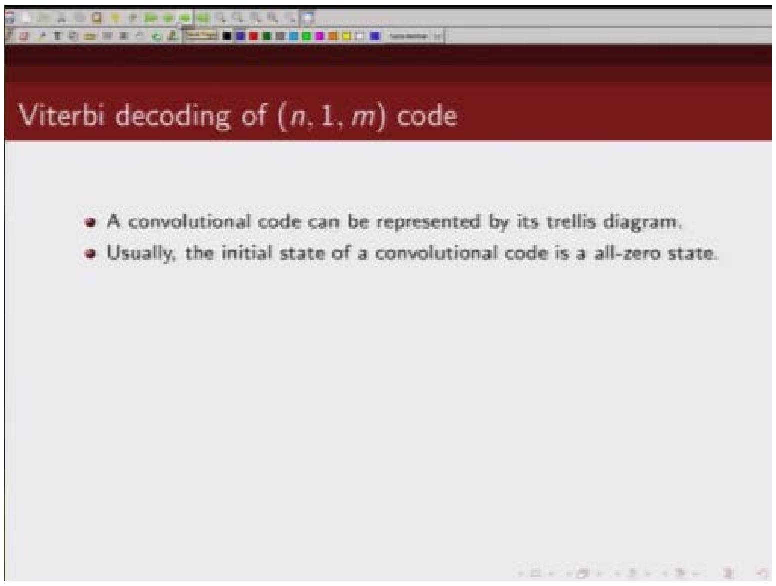


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.

So as we know a convolutional code can be represented by its corresponding state diagram or trellis diagram.

(Refer Slide Time: 01:30)

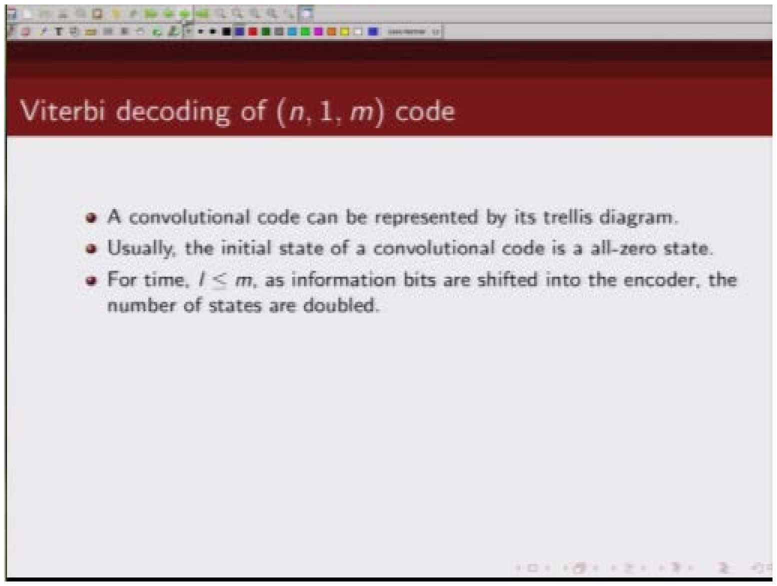


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.

And we are going to make use of the trellis representation of a convolutional code to decode it. So initially the convolutional code is assumed to be in all-zero state.

(Refer Slide Time: 01:46)

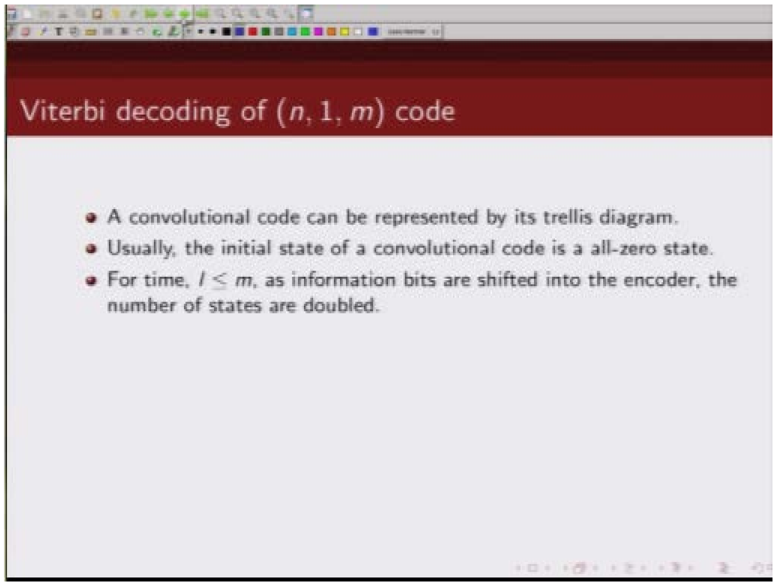


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.
- For time, $l \leq m$, as information bits are shifted into the encoder, the number of states are doubled.

So for time l, which is less than the memory order we notice that the information bits are getting shifted into the encoder one bit at a time. And the number of states get doubled, so initially it is in all-zero state then once one bit get in, there are two possibilities where we can be in.

(Refer Slide Time: 02:17)

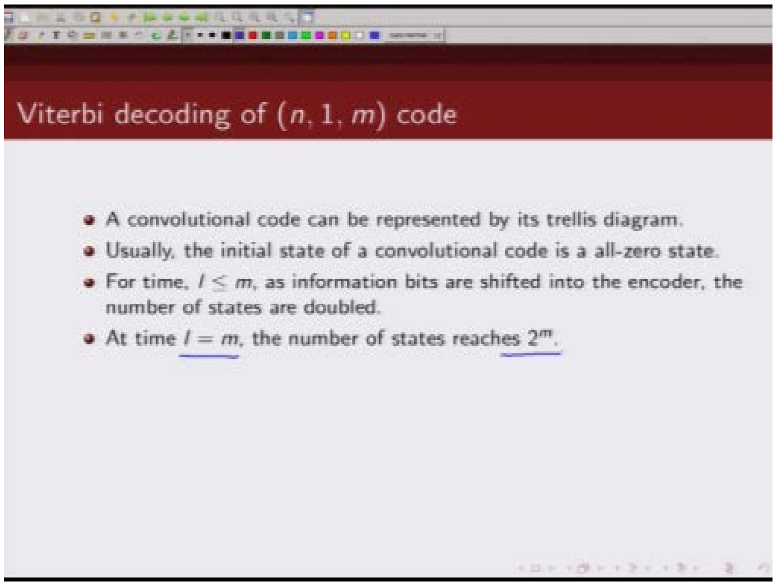


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.
- For time, $l \leq m$, as information bits are shifted into the encoder, the number of states are doubled.

And so as long as l is less than memory order we see it each time instance the number of possibilities of number of states gets doubled.

(Refer Slide Time: 02:28)



Viterbi decoding of $(n, 1, m)$ code

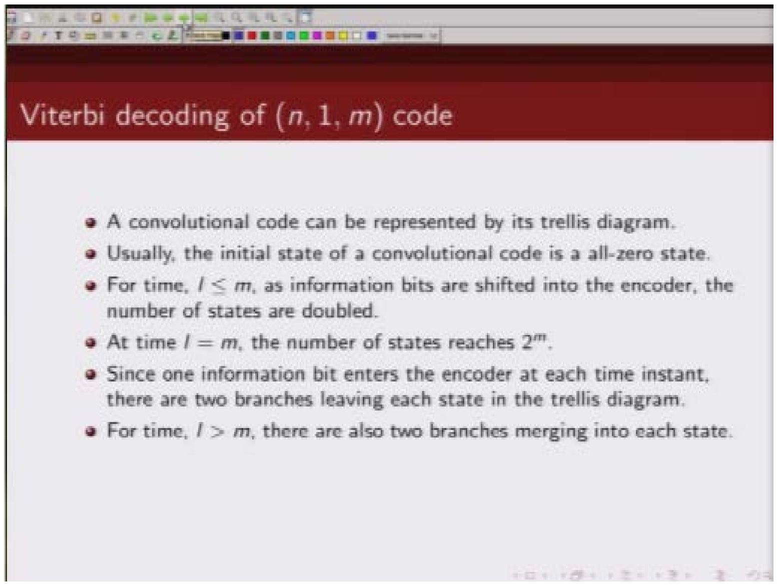
- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.
- For time, $l \leq m$, as information bits are shifted into the encoder, the number of states are doubled.
- At time $\underline{l = m}$, the number of states reaches $\underline{2^m}$.

Until we reach time l = m when we reach all possible states which is basically 2^m, because our convolutional code has a memory of m, so total it has 2^m states.

(Refer Slide Time: 02:50)

So since we are considering a rate 1/n code so k is 1, so each time instance we have one information bit that enters the encoder at each time. So there are two possible branches leaving at each state, one corresponding to information bit zero, other corresponding to information bit 1. So in our trellis diagram we will see that there are two branches leaving each state.

(Refer Slide Time: 03:30)

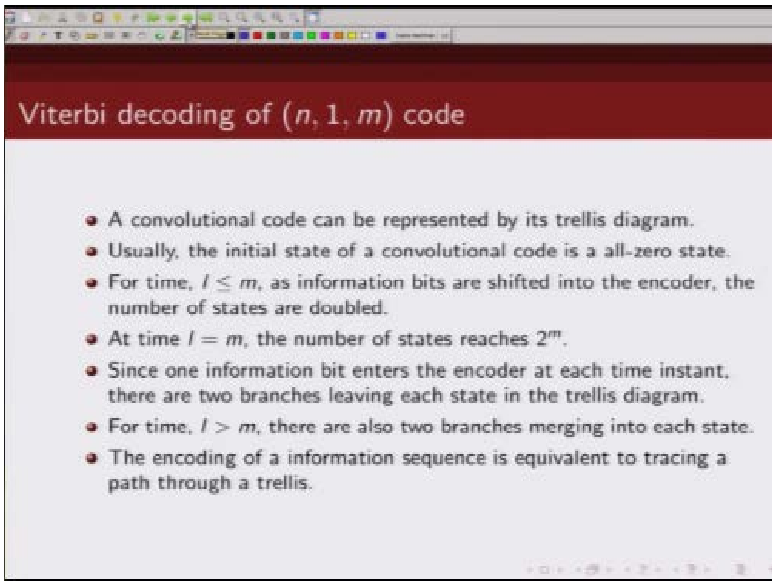


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.
- For time, $l \leq m$, as information bits are shifted into the encoder, the number of states are doubled.
- At time $l = m$, the number of states reaches 2^m.
- Since one information bit enters the encoder at each time instant, there are two branches leaving each state in the trellis diagram.
- For time, $l > m$, there are also two branches merging into each state.

Now what happens if when the time instance is more than the memory order, we will see that two branches are merging into each state.

(Refer Slide Time: 03:42)

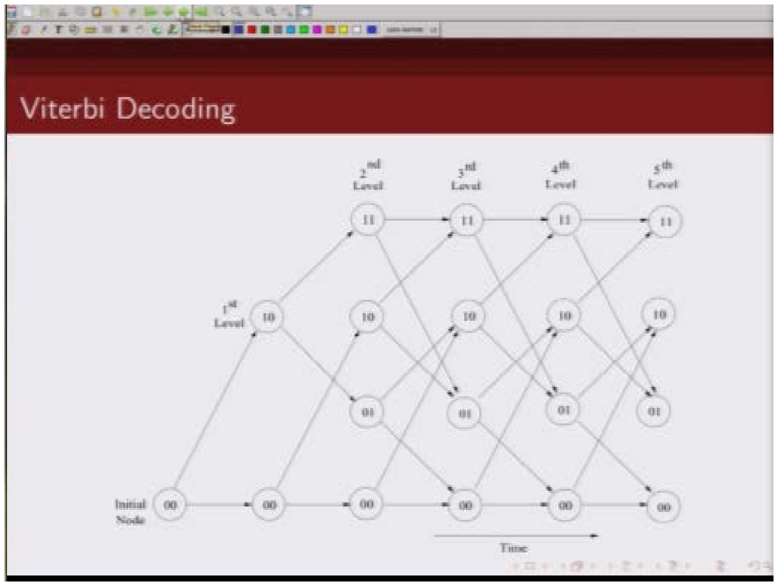


Viterbi decoding of $(n, 1, m)$ code

- A convolutional code can be represented by its trellis diagram.
- Usually, the initial state of a convolutional code is a all-zero state.
- For time, $l \leq m$, as information bits are shifted into the encoder, the number of states are doubled.
- At time $l = m$, the number of states reaches 2^m.
- Since one information bit enters the encoder at each time instant, there are two branches leaving each state in the trellis diagram.
- For time, $l > m$, there are also two branches merging into each state.
- The encoding of a information sequence is equivalent to tracing a path through a trellis.

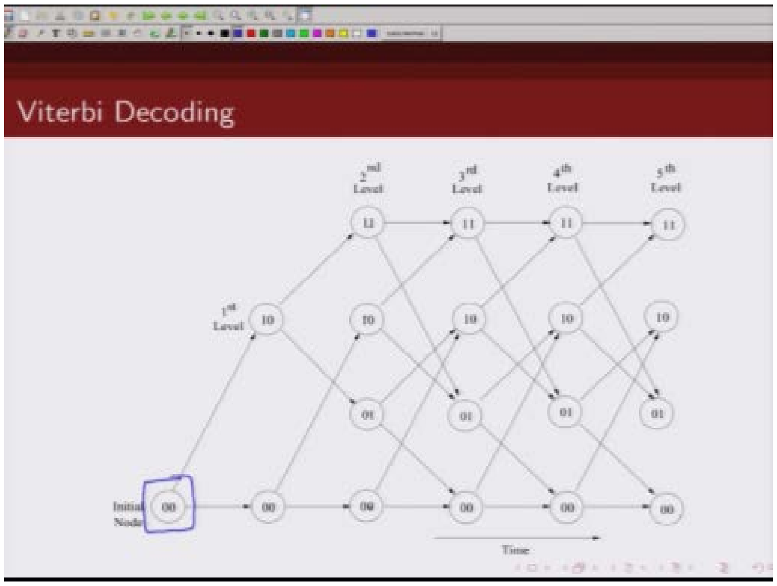
And encoding of information sequence can be considered as if we are tracing a path through this trellis. So any path through this trellis is a valid code word.

(Refer Slide Time: 04:00)


Viterbi Decoding

This is just an example of a trellis of a convolutional code. So initially as I said, we start off with all-zero state.

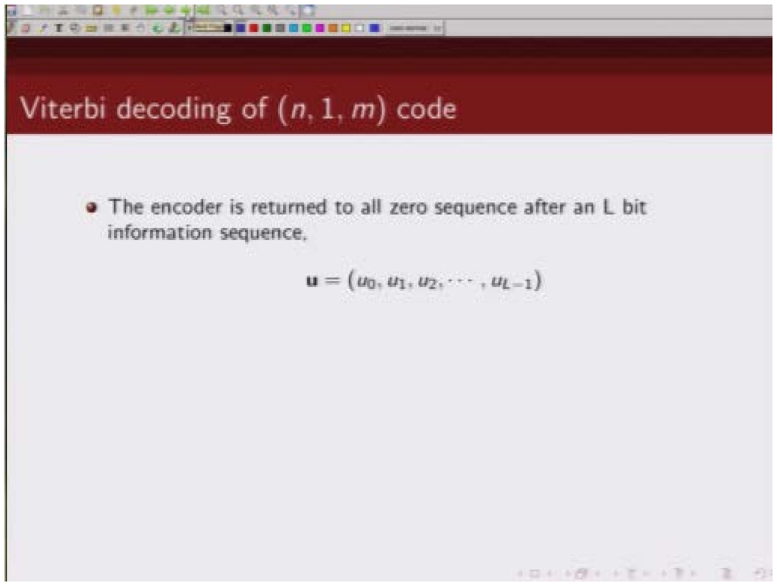
(Refer Slide Time: 04:11)


Viterbi Decoding

So initially the encoder is assumed to be in all-zero state. So until – so in this particular example the number of states is four, you can see 00, 01, 10, 11 these are the four states, I am denoting them by 00, 01, 10 and 11, so until you reach time t = 2 which is the memory order of this code. We will see that number of states are getting doubled; initially there was only one state all-zero state then you had two states and then you had four states.

And you can see from each state there are two branches leaving that particular state. And once your time becomes more than m, then we see that at each state there are two branches which are merging okay. I am just denoting, so this is on the X axis is my time, I am just denoting them by level, first level, second level, third level, fourth level which is just how the time is progressing.

(Refer Slide Time: 05:19)



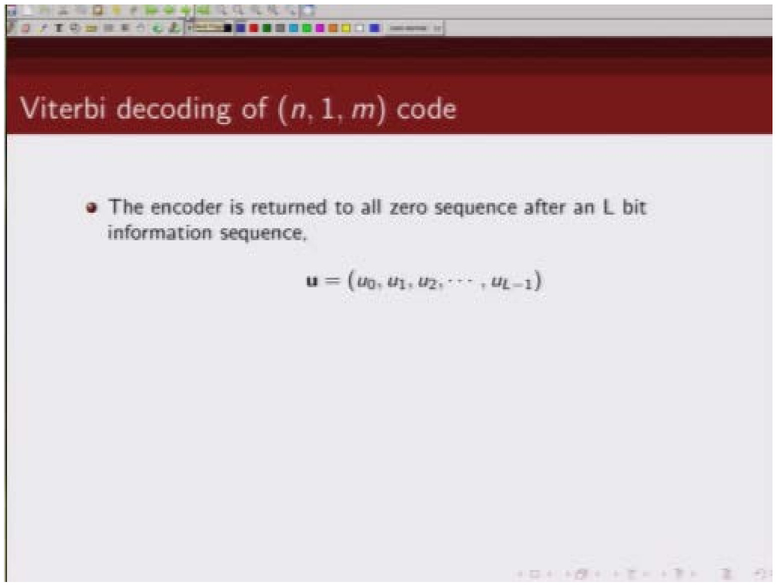
Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence.

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

Now usually after we encode the information sequence we bring the – we terminate the convolutional code and what do we mean by terminate the convolutional code, we bring the convolutional code back into all-zero state.

(Refer Slide Time: 05:38)



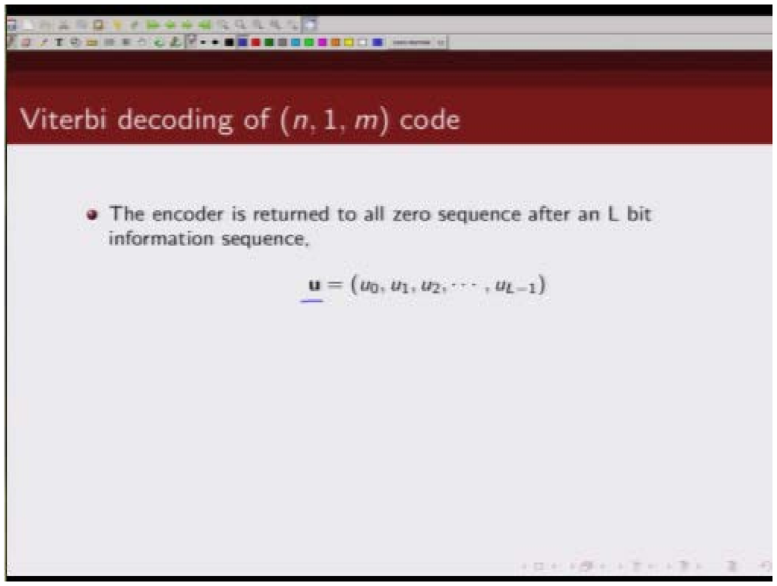
Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence.

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

So the encoder is typically returned to all-zero state after we have encoded our l bit sequence.

(Refer Slide Time: 05:52)



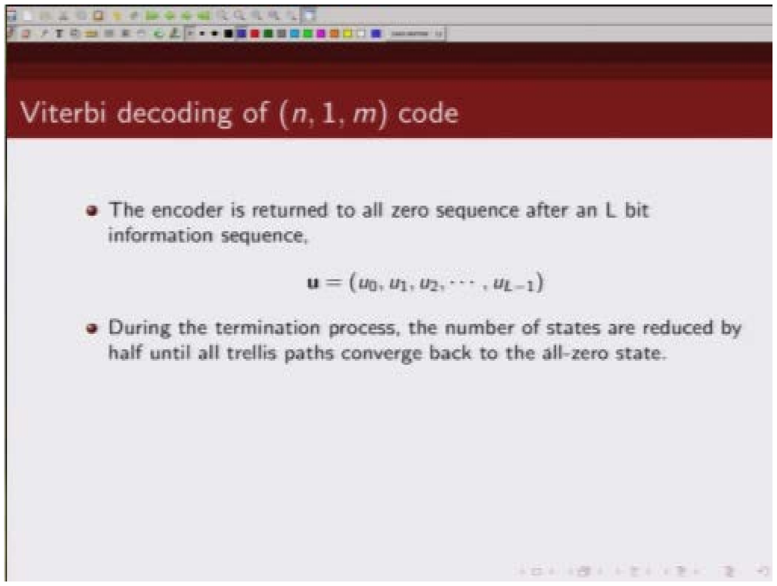
Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence,

$$\underline{\mathbf{u}} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

So if you have an information sequence which is l bit denoted by u, so our information sequence is u0, u1, u2, ul-1 where these are basically zeros and ones.

(Refer Slide Time: 05:59)



Viterbi decoding of $(n, 1, m)$ code

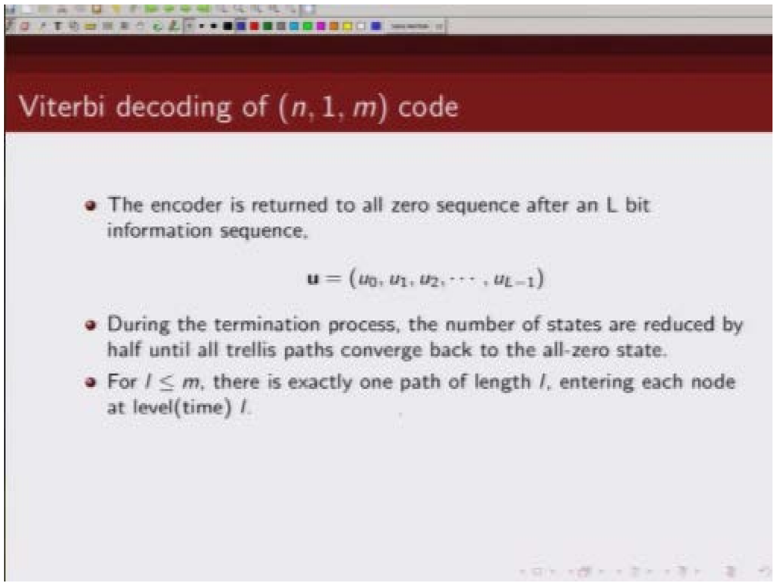
- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.

Now during the termination process, because we are trying to bring it back to all-zero state what happens is number of states get reduced by half until all the trellis path will converge to all-zero state.

(Refer Slide Time: 06:19)



Viterbi decoding of $(n, 1, m)$ code

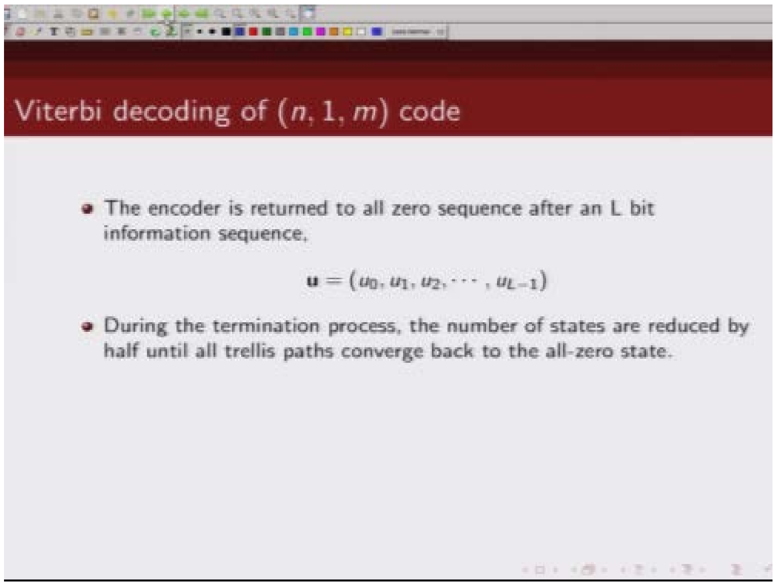
- The encoder is returned to all zero sequence after an L bit information sequence.

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.
- For $l \leq m$, there is exactly one path of length l, entering each node at level(time) l.

So as we said for time less than memory order there is only one path of length l entering each state, each node.

(Refer Slide Time: 06:30)



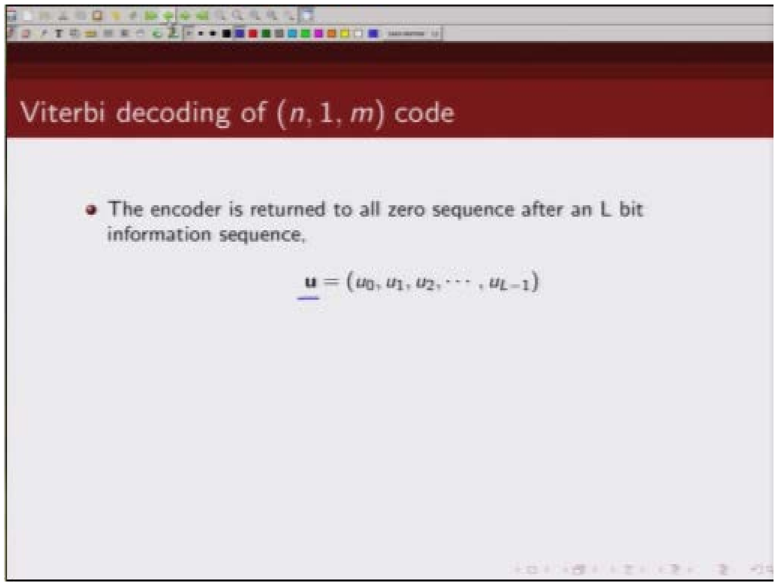
Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence.

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.

(Refer Slide Time: 06:31)

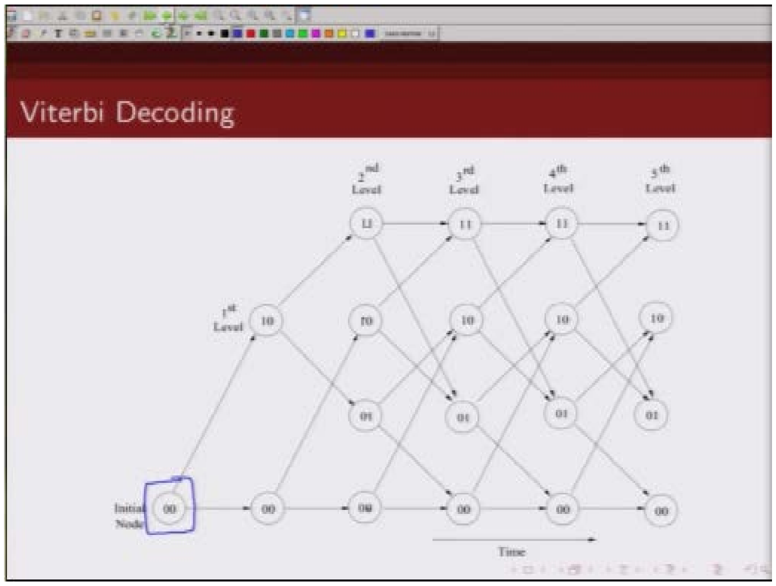


Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence.

$$\underline{\mathbf{u}} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

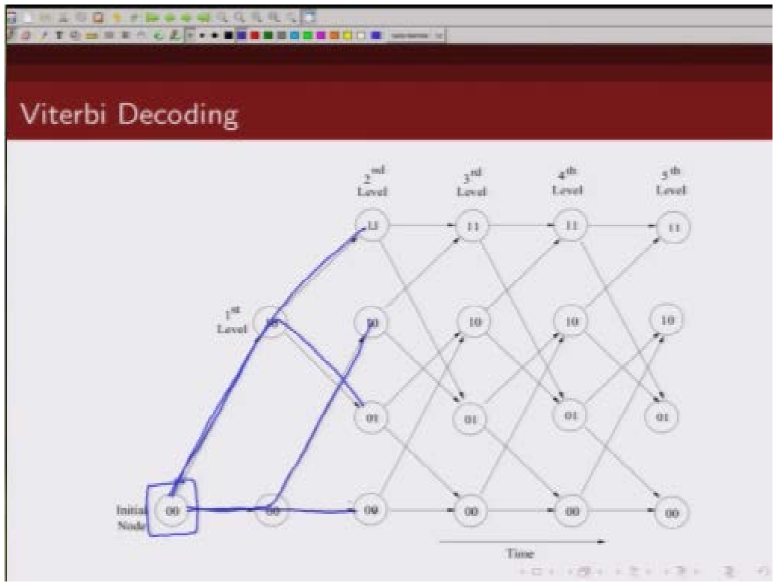
(Refer Slide Time: 06:31)



Viterbi Decoding

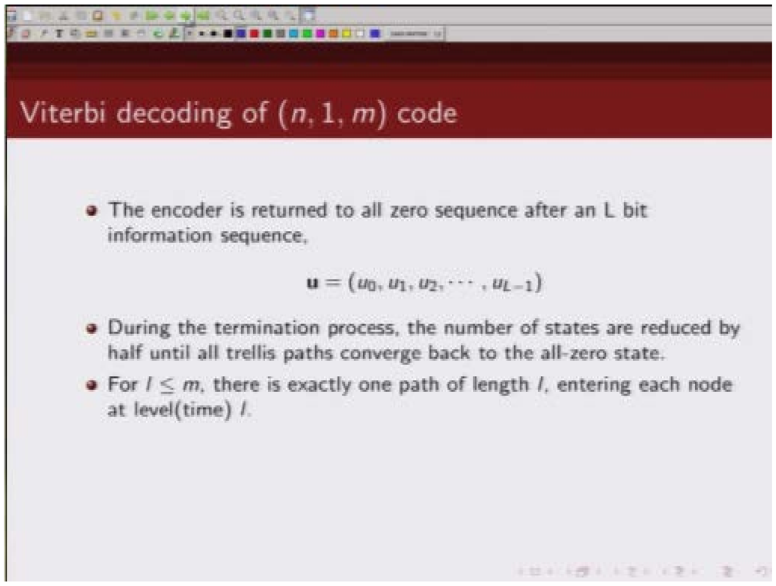
You can see, go back to this diagram up to this path, there is only one.

(Refer Slide Time: 06:41)


Viterbi Decoding

Let us say I want to reach 11 from 00 there is only one path through this. I want to reach here there is only one path, I want to reach this state, there is only one path. I want to reach this state, there is only one path.

(Refer Slide Time: 06:58)



Viterbi decoding of $(n, 1, m)$ code

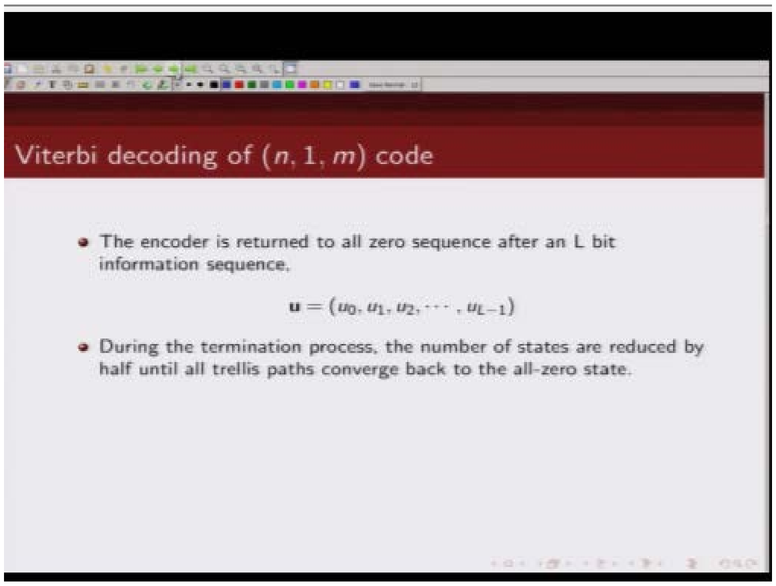
- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.
- For $l \leq m$, there is exactly one path of length l, entering each node at level(time) l.

So until time is less than equal to memory order there is exactly one path of length l, entering each node at each time.

(Refer Slide Time: 06:59)



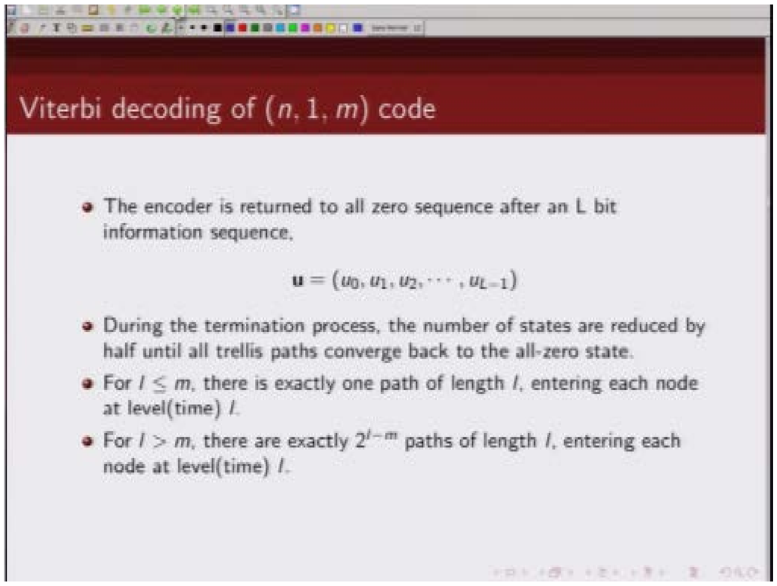
Viterbi decoding of $(n, 1, m)$ code

- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.

(Refer Slide Time: 07:09)



Viterbi decoding of $(n, 1, m)$ code

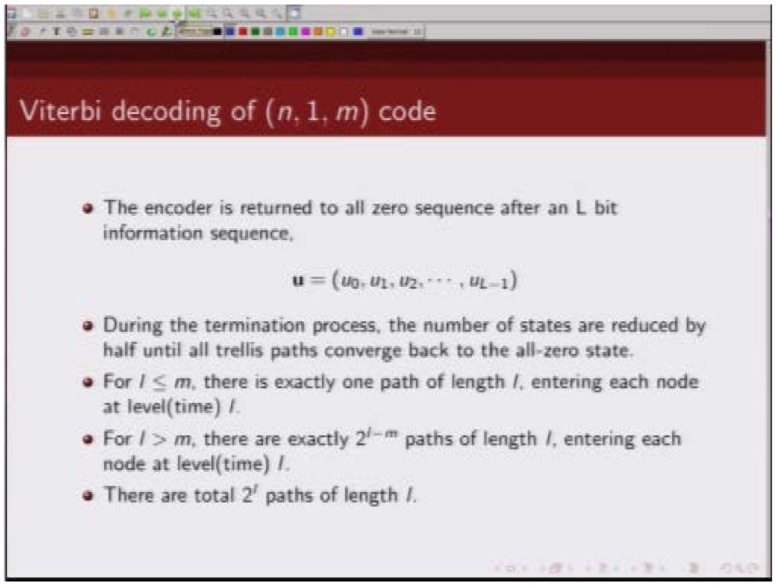
- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.
- For $l \leq m$, there is exactly one path of length l, entering each node at level(time) l.
- For $l > m$, there are exactly 2^{l-m} paths of length l, entering each node at level(time) l.

And for l greater than m there are 2^{l-m} paths of length l entering each state at each time instance.

(Refer Slide Time: 07:25)



Viterbi decoding of $(n, 1, m)$ code

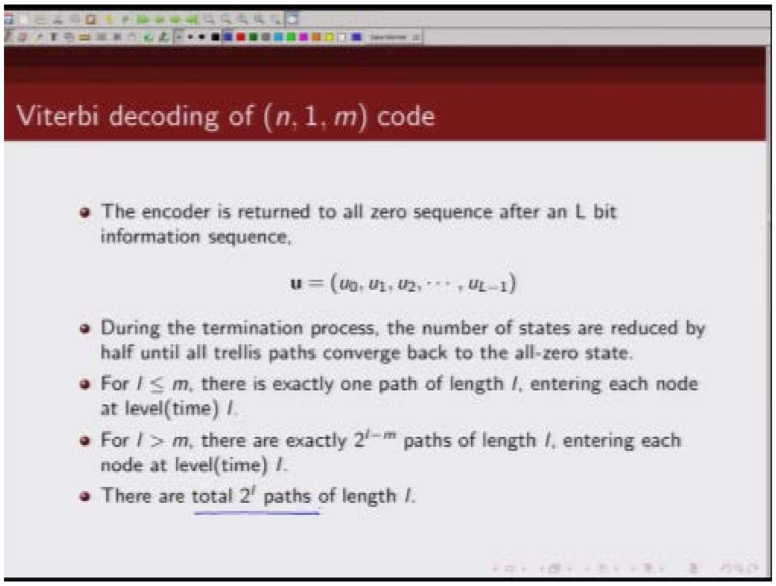
- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.
- For $l \leq m$, there is exactly one path of length l, entering each node at level(time) l.
- For $l > m$, there are exactly 2^{l-m} paths of length l, entering each node at level(time) l.
- There are total 2^l paths of length l.

And total there are.

(Refer Slide Time: 07:28)



Viterbi decoding of $(n, 1, m)$ code

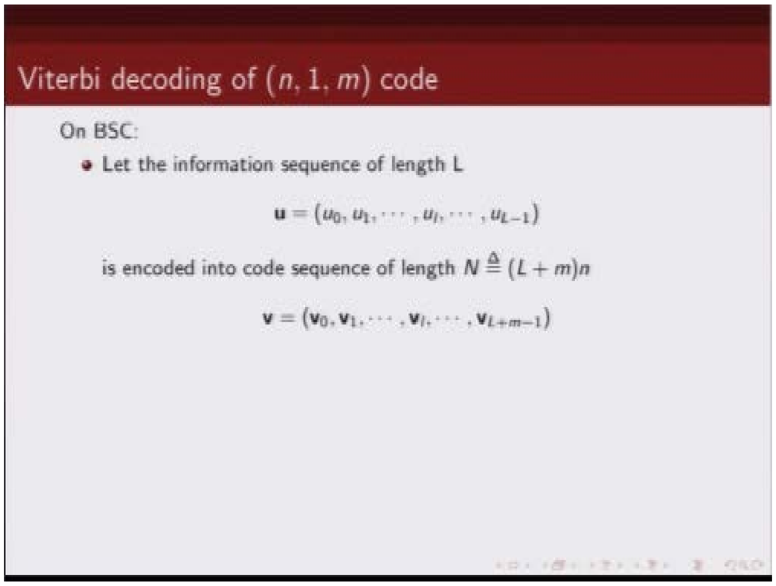
- The encoder is returned to all zero sequence after an L bit information sequence,

$$\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{L-1})$$

- During the termination process, the number of states are reduced by half until all trellis paths converge back to the all-zero state.
- For $l \leq m$, there is exactly one path of length l, entering each node at level(time) l.
- For $l > m$, there are exactly 2^{l-m} paths of length l, entering each node at level(time) l.
- There are $\underline{\text{total } 2^l}$ paths of length l.

2^l paths of length l

(Refer Slide Time: 07:33)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- Let the information sequence of length L

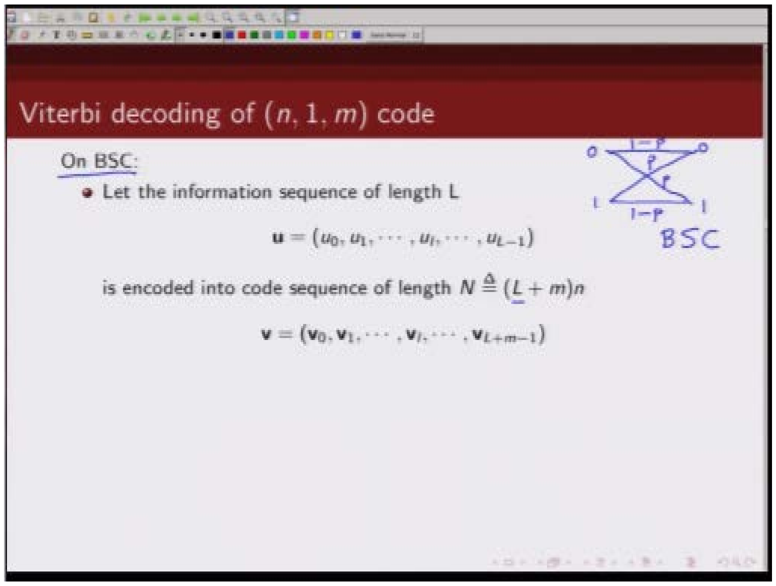
$$\mathbf{u} = (u_0, u_1, \cdots, u_l, \cdots, u_{L-1})$$

is encoded into code sequence of length $N \triangleq (L+m)n$

$$\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_l, \cdots, \mathbf{v}_{L+m-1})$$

In this trellis diagram, so let us consider how we are going to decode this convolutional code which can be represented by its trills diagram, so we are considering

(Refer Slide Time: 07:45)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- Let the information sequence of length L

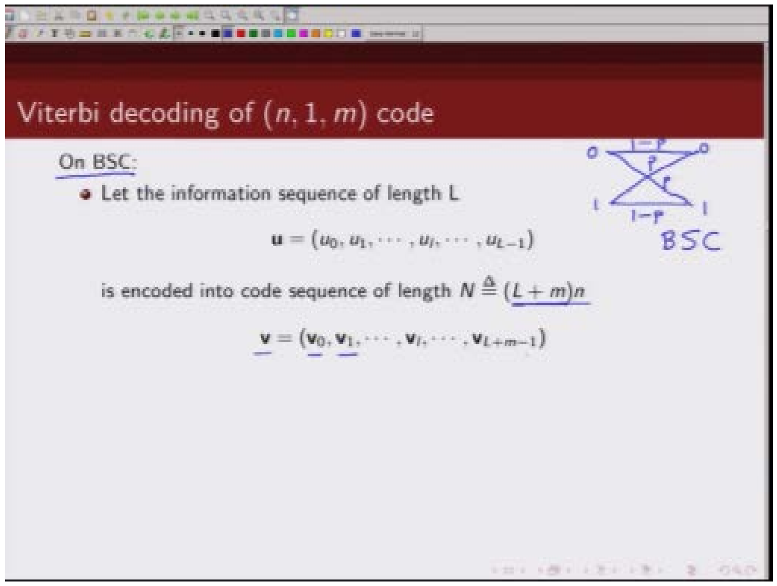
$$\mathbf{u} = (u_0, u_1, \cdots, u_l, \cdots, u_{L-1})$$

is encoded into code sequence of length $N \triangleq (\underline{L} + m)n$

$$\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_l, \cdots, \mathbf{v}_{L+m-1})$$

A binary systemic channel, now recall what is binary symmetric channel, so we have 2 inputs 0’s and 1 similarly we have 2 output 0 and 1 so there is a 1-P probability of getting the bits correctly and then there is a cross over probability P of bits getting flipped so this is our binary symmetric channel. So let us consider that we have our information sequence of length l denoted by u and it is been encoded into a sequence coded length n which is coded using a rate 1/n code now l corresponds to number of information bits, now remember we are terminating the convolutional encoders.

(Refer Slide Time: 08:45)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- Let the information sequence of length L

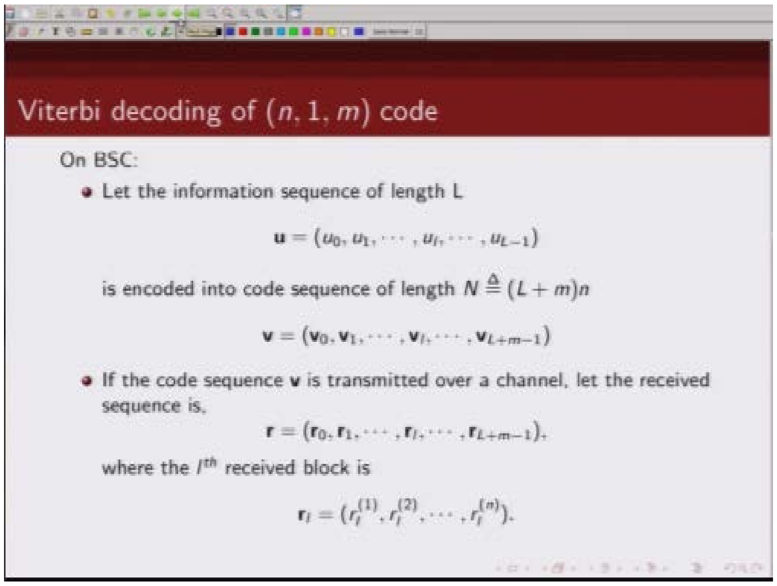
$$\mathbf{u} = (u_0, u_1, \cdots, u_l, \cdots, u_{L-1})$$

is encoded into code sequence of length $N \triangleq \underline{(L+m)n}$

$$\underline{\mathbf{v}} = (\underline{\mathbf{v}_0}, \underline{\mathbf{v}_1}, \cdots, \mathbf{v}_l, \cdots, \mathbf{v}_{L+m-1})$$

So we require m number of bits to terminate it to bring it back to all 0 state so total basically number of bits including the tail bits or termination bits v L + m and since it is the rate 1/n code, the code word, code sequence length will be given by (L + m) n so our code, code sequence is of length L+ m denoted by $v_{0,}$ $v_{1,}$ $v_{2,}$ v_{L+m-n} and where each of them $v_{0,}$ $v_{1,}$ $v_{2,}$ v_{L} are basically n bits.

(Refer Slide Time: 09:38)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- Let the information sequence of length L

$$\mathbf{u} = (u_0, u_1, \cdots, u_l, \cdots, u_{L-1})$$

is encoded into code sequence of length $N \triangleq (L+m)n$

$$\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_l, \cdots, \mathbf{v}_{L+m-1})$$

- If the code sequence $\mathbf{v}$ is transmitted over a channel, let the received sequence is,

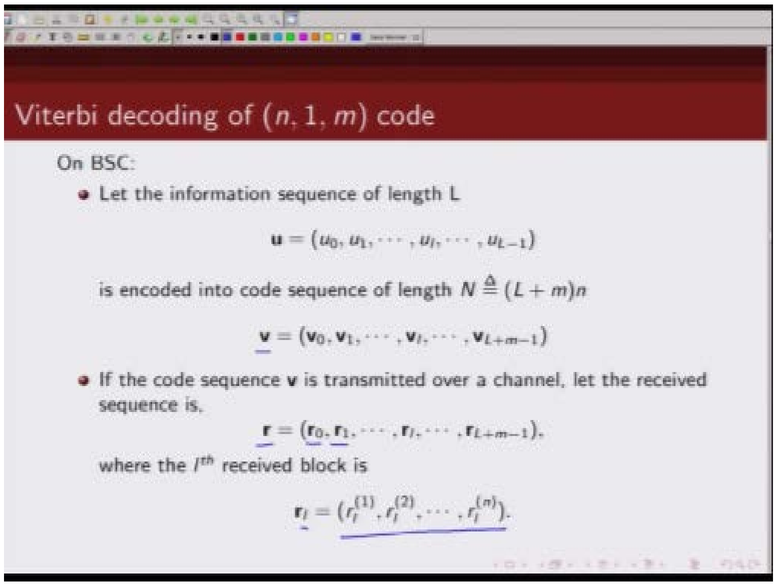
$$\mathbf{r} = (\mathbf{r}_0, \mathbf{r}_1, \cdots, \mathbf{r}_l, \cdots, \mathbf{r}_{L+m-1}),$$

where the l^{th} received block is

$$\mathbf{r}_l = (r_l^{(1)}, r_l^{(2)}, \cdots, r_l^{(n)}).$$

So if you transmit

(Refer Slide Time: 09:40)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- Let the information sequence of length L

$$\mathbf{u} = (u_0, u_1, \cdots, u_l, \cdots, u_{L-1})$$

is encoded into code sequence of length $N \triangleq (L+m)n$

$$\underline{\mathbf{v}} = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_l, \cdots, \mathbf{v}_{L+m-1})$$

- If the code sequence $\mathbf{v}$ is transmitted over a channel, let the received sequence is,

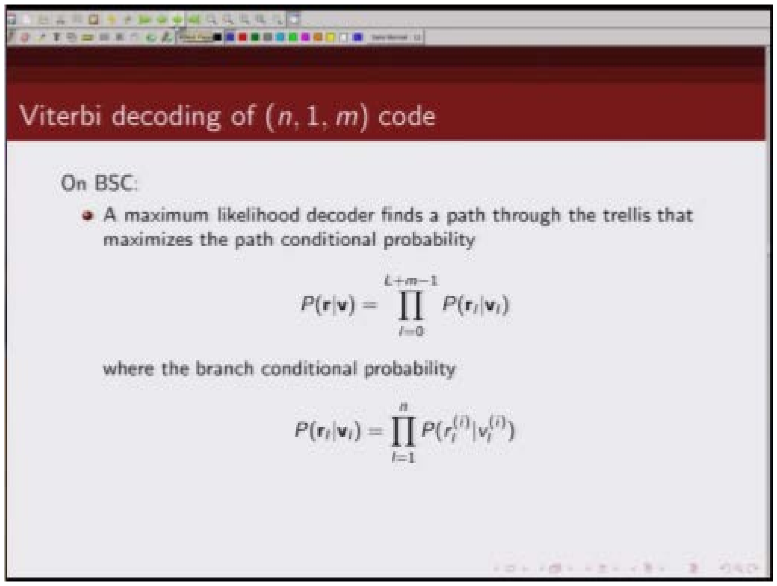
$$\underline{\mathbf{r}} = (\underline{\mathbf{r}_0}, \underline{\mathbf{r}_1}, \cdots, \mathbf{r}_l, \cdots, \mathbf{r}_{L+m-1}),$$

where the l^{th} received block is

$$\underline{\mathbf{r}_l} = \underline{(r_l^{(1)}, r_l^{(2)}, \cdots, r_l^{(n)})}.$$

This code sequence over this binary symmetric channel what we receive is denoted by receive sequence is denoted by r, so we receive $r_{0,}$ $r_{1,}$ $r_{2,}$ r_{l+m-1} where each of these $r_{i's}$ are consisting of n bits okay, so each of these $r_{i's}$ are basically this n bit vector. This is corresponding to the transmitted code sequence so this r is the receive code sequence corresponding to the code word or the code sequence

(Refer Slide Time: 10:19)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- A maximum likelihood decoder finds a path through the trellis that maximizes the path conditional probability

$$P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{L+m-1} P(\mathbf{r}_l|\mathbf{v}_l)$$

where the branch conditional probability

$$P(\mathbf{r}_l|\mathbf{v}_l) = \prod_{l=1}^{n} P(r_l^{(i)}|v_l^{(i)})$$

That we have transmitted. Now we know that a maximum likelihood decoder will try to maximize the likelihood function so a maximum likelihood decoder is going to find a path through this trellis that will maximize the path conditional probability so it will maximize the likelihood

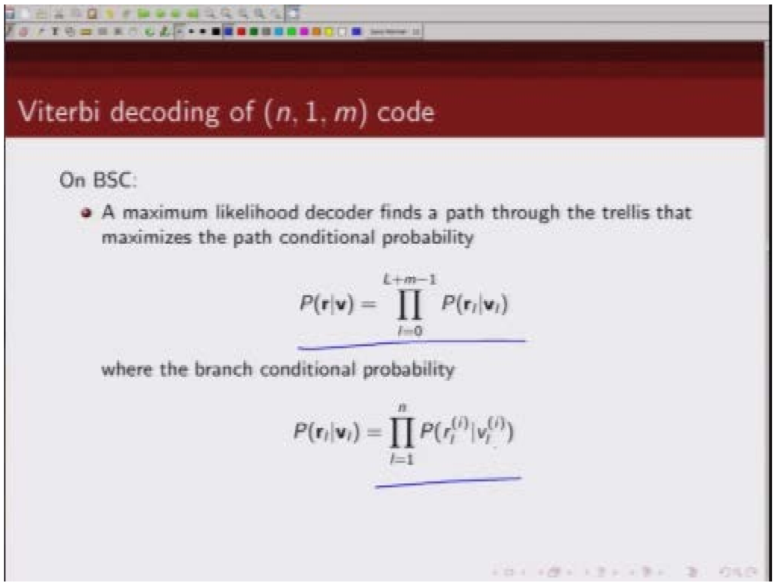
(Refer Slide Time: 10:41)

Function now we can define the $P(r \mid v)$ now what is this r?

(Refer Slide Time: 10:41)

r consist of these L+M elements and each of these $r_{i's}$ are further n bit vector

(Refer Slide Time: 11:05)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- A maximum likelihood decoder finds a path through the trellis that maximizes the path conditional probability

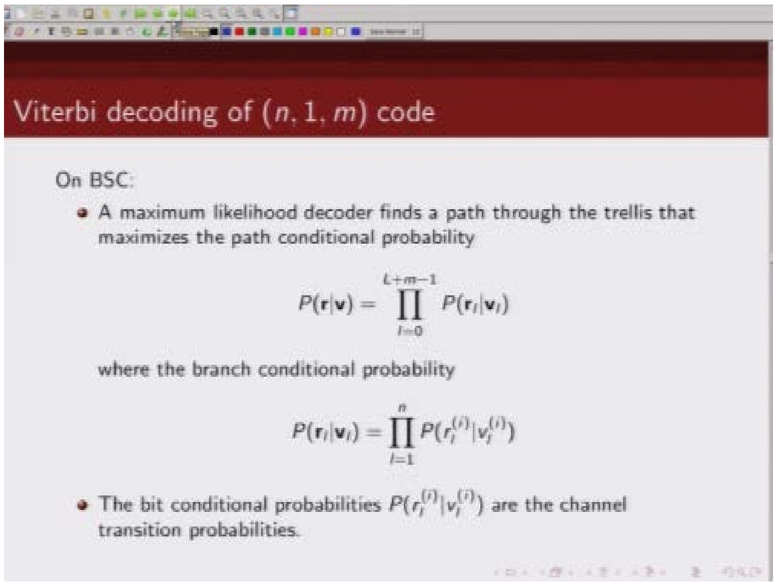
$$\underline{P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{L+m-1} P(\mathbf{r}_l|\mathbf{v}_l)}$$

where the branch conditional probability

$$P(\mathbf{r}_l|\mathbf{v}_l) = \underline{\prod_{l=1}^{n} P(r_l^{(i)}|v_l^{(i)})}$$

So we, since it is a memory less channel we can write this P(r | v) in terms of these $r_{i's}$ and $v_{i's}$ like this and this can be written in terms of their bit likelihood function, so we can write this in terms of $r_{i's}$ and $v_{i's}$ and this can be written further in terms of the bit matrix which is given by this.

(Refer Slide Time: 11:35)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- A maximum likelihood decoder finds a path through the trellis that maximizes the path conditional probability

$$P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{L+m-1} P(\mathbf{r}_l|\mathbf{v}_l)$$

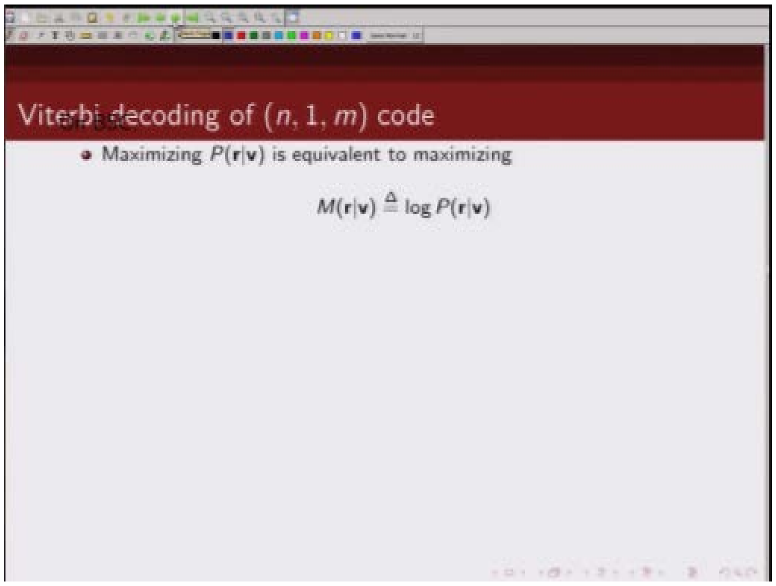
where the branch conditional probability

$$P(\mathbf{r}_l|\mathbf{v}_l) = \prod_{l=1}^{n} P(r_l^{(i)}|v_l^{(i)})$$

- The bit conditional probabilities $P(r_l^{(i)}|v_l^{(i)})$ are the channel transition probabilities.

And what are these probabilities, these probabilities are nothing but these are corresponding to the channel transition probabilities.

(Refer Slide Time: 11:47)



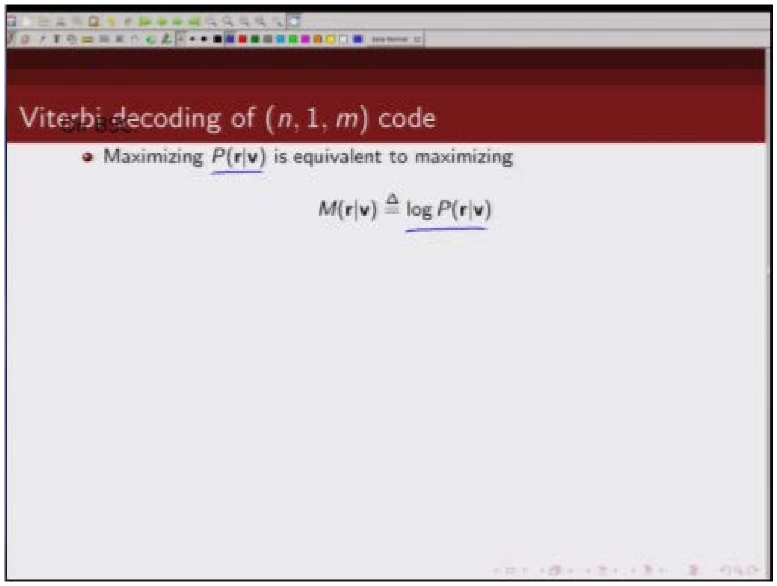
Viterbi decoding of $(n, 1, m)$ code

- Maximizing $P(\mathbf{r}|\mathbf{v})$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \log P(\mathbf{r}|\mathbf{v})$$

Now maximizing

(Refer Slide Time: 11:52)



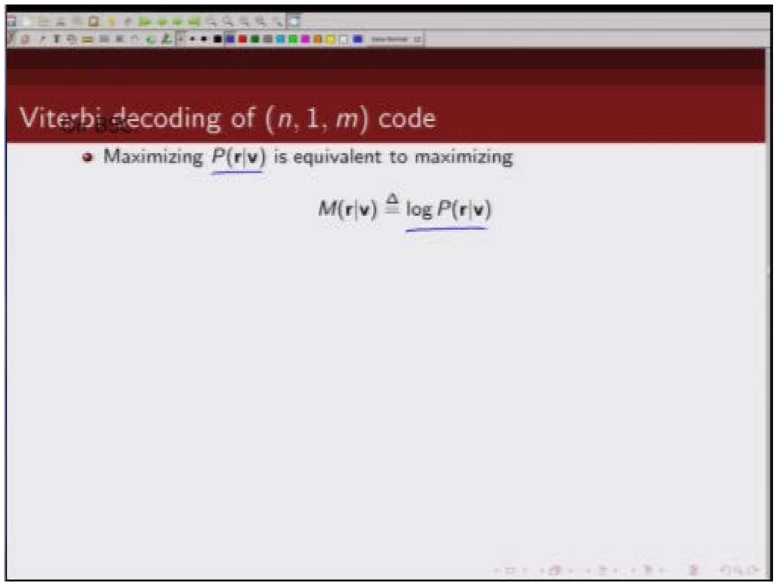
Viterbi decoding of $(n, 1, m)$ code

- Maximizing $\underline{P(\mathbf{r}|\mathbf{v})}$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

This likelihood function is equivalent to maximizing the log of the likelihood function, why because log is a non decreasing function so

(Refer Slide Time: 11:52)



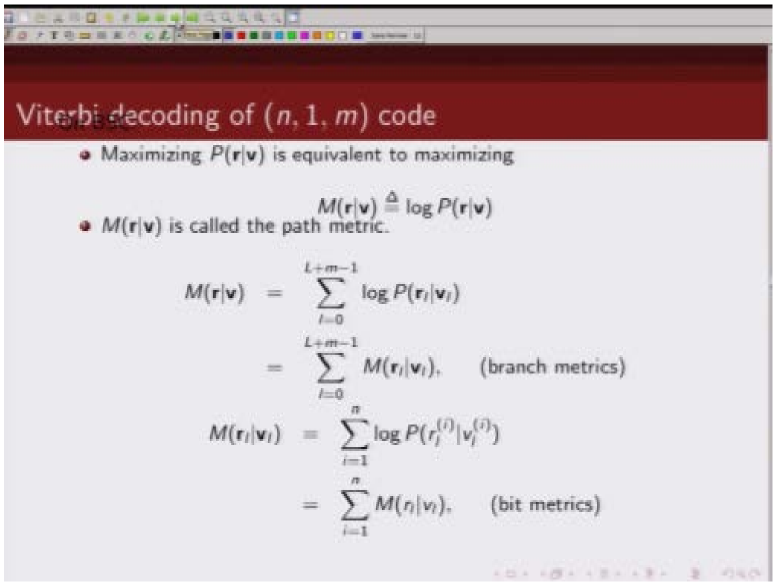
Viterbi decoding of $(n, 1, m)$ code

- Maximizing $\underline{P(\mathbf{r}|\mathbf{v})}$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

We can equivalently write maximizing this as our objective as maximizing the log of P(r | v)

(Refer Slide Time: 12:20)



Viterbi decoding of $(n, 1, m)$ code

- Maximizing $P(\mathbf{r}|\mathbf{v})$ is equivalent to maximizing

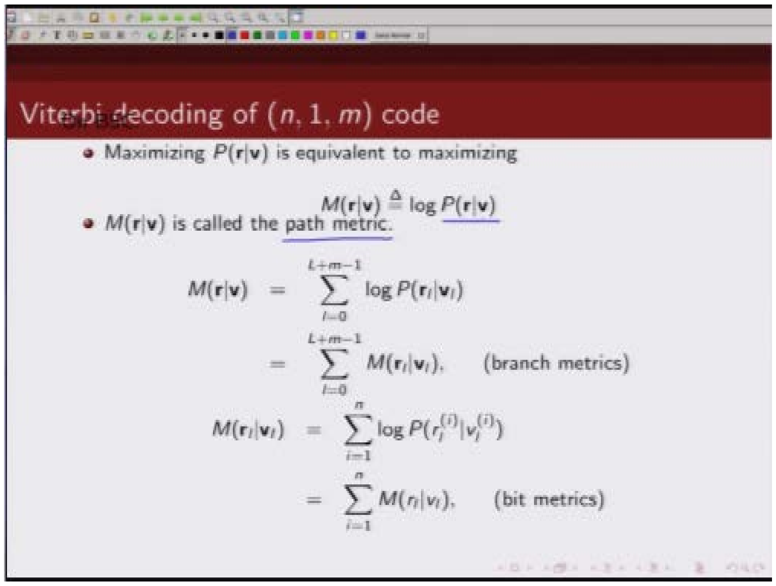
$$M(\mathbf{r}|\mathbf{v}) \triangleq \log P(\mathbf{r}|\mathbf{v})$$

- $M(\mathbf{r}|\mathbf{v})$ is called the path metric.

$$\begin{aligned} M(\mathbf{r}|\mathbf{v}) &= \sum_{l=0}^{L+m-1} \log P(\mathbf{r}_l|\mathbf{v}_l) \\ &= \sum_{l=0}^{L+m-1} M(\mathbf{r}_l|\mathbf{v}_l), \quad \text{(branch metrics)} \\ M(\mathbf{r}_l|\mathbf{v}_l) &= \sum_{i=1}^{n} \log P(r_l^{(i)}|v_l^{(i)}) \\ &= \sum_{i=1}^{n} M(r_l|v_l), \quad \text{(bit metrics)} \end{aligned}$$

And this is, we call this as

(Refer Slide Time: 12:25)



Viterbi decoding of $(n, 1, m)$ code

- Maximizing $P(\mathbf{r}|\mathbf{v})$ is equivalent to maximizing

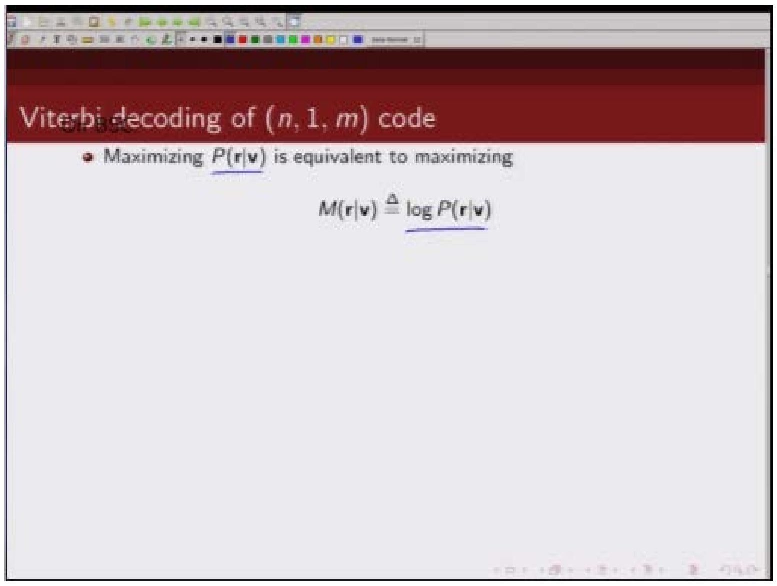
$$M(\mathbf{r}|\mathbf{v}) \triangleq \log P(\mathbf{r}|\mathbf{v})$$

- $M(\mathbf{r}|\mathbf{v})$ is called the path metric.

$$\begin{aligned} M(\mathbf{r}|\mathbf{v}) &= \sum_{l=0}^{L+m-1} \log P(\mathbf{r}_l|\mathbf{v}_l) \\ &= \sum_{l=0}^{L+m-1} M(\mathbf{r}_l|\mathbf{v}_l), \quad \text{(branch metrics)} \\ M(\mathbf{r}_l|\mathbf{v}_l) &= \sum_{i=1}^{n} \log P(r_l^{(i)}|v_l^{(i)}) \\ &= \sum_{i=1}^{n} M(r_l|v_l), \quad \text{(bit metrics)} \end{aligned}$$

Path metric, now how do we compute the path metric now go back and look into the expression of P(r | v)

(Refer Slide Time: 12:35)

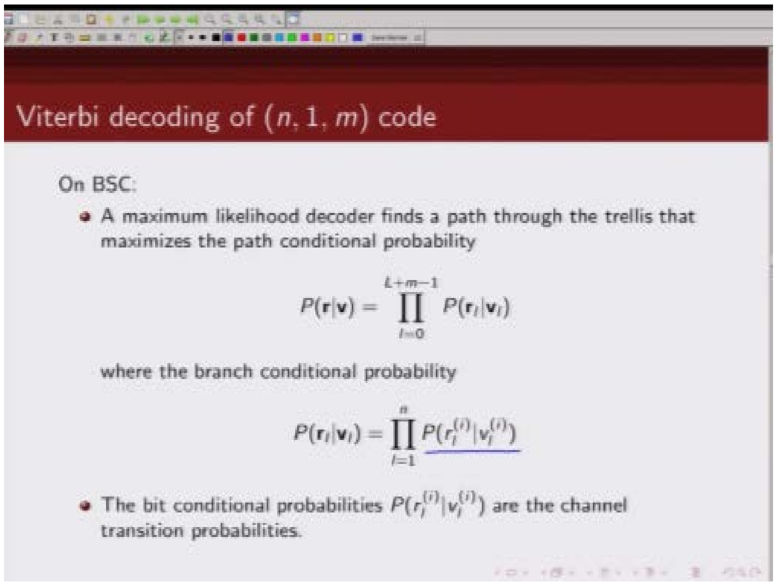


Viterbi decoding of $(n, 1, m)$ code

- Maximizing $\underline{P(\mathbf{r}|\mathbf{v})}$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

(Refer Slide Time: 12:36)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- A maximum likelihood decoder finds a path through the trellis that maximizes the path conditional probability

$$P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{L+m-1} P(\mathbf{r}_l|\mathbf{v}_l)$$

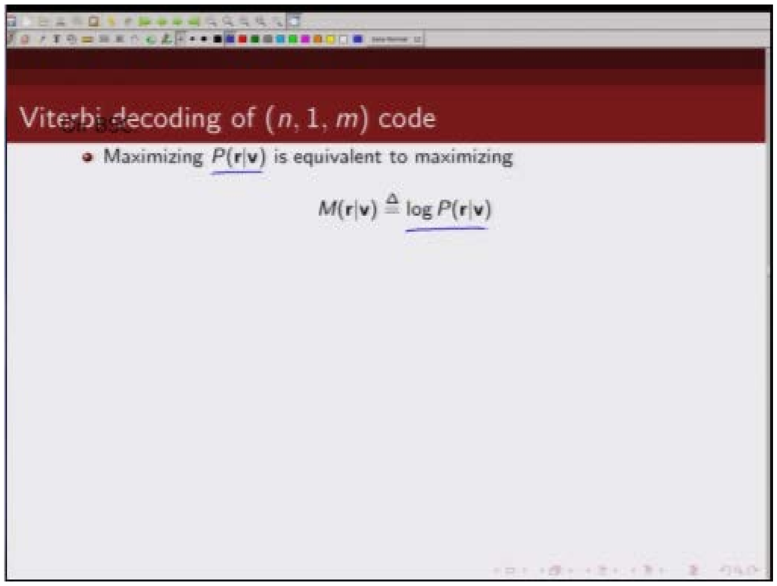
where the branch conditional probability

$$P(\mathbf{r}_l|\mathbf{v}_l) = \prod_{i=1}^{n} \underline{P(r_l^{(i)}|v_l^{(i)})}$$

- The bit conditional probabilities $P(r_l^{(i)}|v_l^{(i)})$ are the channel transition probabilities.

So if you look at a P (r | v) is given by this, so if we take a log of this, this product term becomes Σ.

(Refer Slide Time: 12:47)

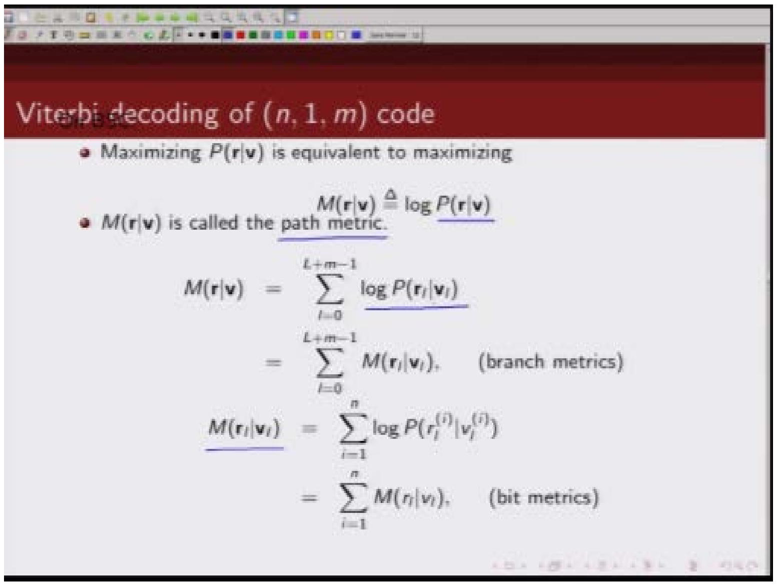


Viterbi decoding of $(n, 1, m)$ code

- Maximizing $\underline{P(\mathbf{r}|\mathbf{v})}$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

(Refer Slide Time: 12:47)



Viterbi decoding of $(n, 1, m)$ code

- Maximizing $P(\mathbf{r}|\mathbf{v})$ is equivalent to maximizing

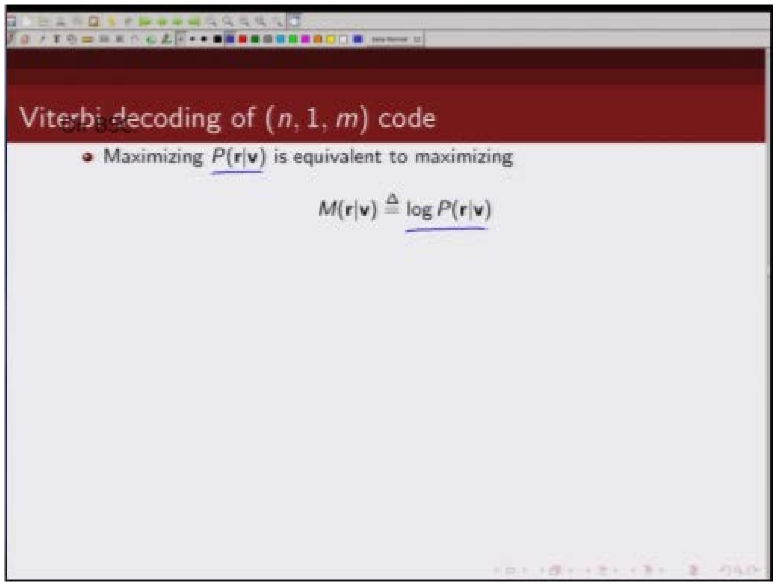
$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

- $M(\mathbf{r}|\mathbf{v})$ is called the <u>path metric</u>.

$$\begin{aligned} M(\mathbf{r}|\mathbf{v}) &= \sum_{l=0}^{L+m-1} \underline{\log P(\mathbf{r}_l|\mathbf{v}_l)} \\ &= \sum_{l=0}^{L+m-1} M(\mathbf{r}_l|\mathbf{v}_l), \quad \text{(branch metrics)} \\ \underline{M(\mathbf{r}_l|\mathbf{v}_l)} &= \sum_{i=1}^{n} \log P(r_l^{(i)}|v_l^{(i)}) \\ &= \sum_{i=1}^{n} M(r_l|v_l), \quad \text{(bit metrics)} \end{aligned}$$

So we can write basically them as so we can write our path metric as Σ from 0 to L+m-1, log P $(r_l | v_l)$ where L goes from 0 to L + m-1, and we are calling this as our branch metric, so this is our branch metric further the branch metric can be written in terms of bit metric. Now we know what is the $P(r_l | v_l)$

(Refer Slide Time: 13:29)



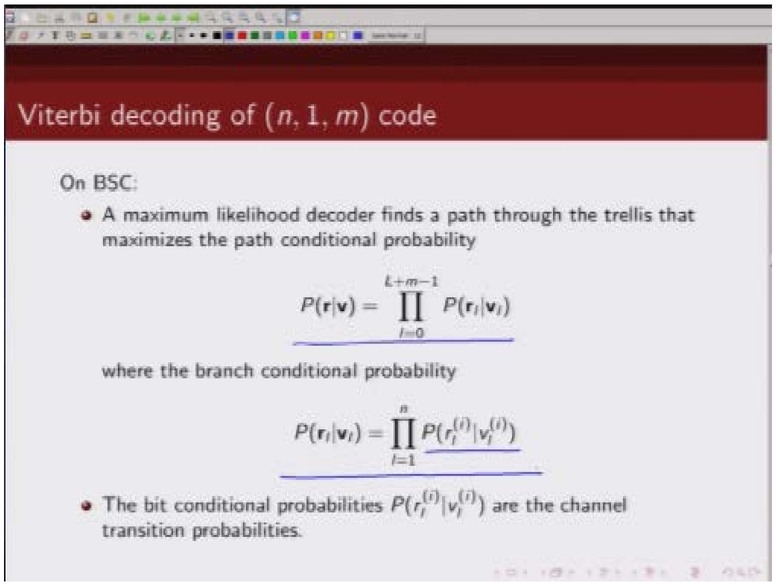
Viterbi decoding of $(n, 1, m)$ code

- Maximizing $\underline{P(\mathbf{r}|\mathbf{v})}$ is equivalent to maximizing

$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

This is given by

(Refer Slide Time: 13:29)



Viterbi decoding of $(n, 1, m)$ code

On BSC:

- A maximum likelihood decoder finds a path through the trellis that maximizes the path conditional probability

$$\underline{P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{L+m-1} P(\mathbf{r}_l|\mathbf{v}_l)}$$

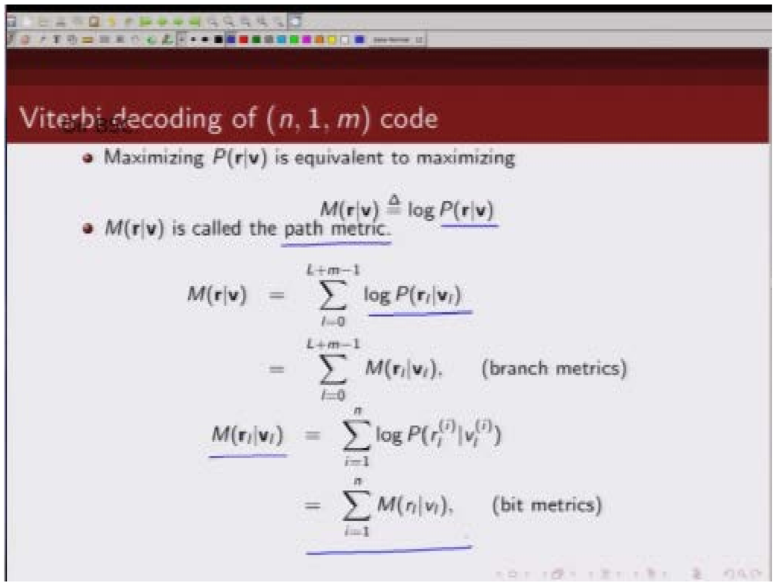
where the branch conditional probability

$$\underline{P(\mathbf{r}_l|\mathbf{v}_l) = \prod_{i=1}^{n} \underline{P(r_l^{(i)}|v_l^{(i)})}}$$

- The bit conditional probabilities $P(r_l^{(i)}|v_l^{(i)})$ are the channel transition probabilities.

This expression, so if we take a log of this again this product term will become Σ so what we would get is

(Refer Slide Time: 13:43)



Viterbi decoding of $(n, 1, m)$ code

- Maximizing $P(\mathbf{r}|\mathbf{v})$ is equivalent to maximizing

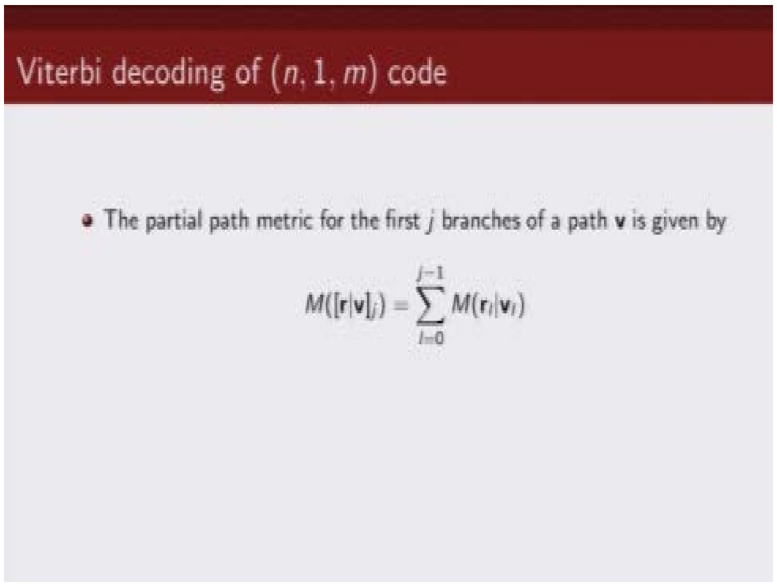
$$M(\mathbf{r}|\mathbf{v}) \triangleq \underline{\log P(\mathbf{r}|\mathbf{v})}$$

- $M(\mathbf{r}|\mathbf{v})$ is called the $\underline{\text{path metric.}}$

$$\begin{aligned} M(\mathbf{r}|\mathbf{v}) &= \sum_{l=0}^{L+m-1} \underline{\log P(\mathbf{r}_l|\mathbf{v}_l)} \\ &= \sum_{l=0}^{L+m-1} M(\mathbf{r}_l|\mathbf{v}_l), \quad \text{(branch metrics)} \\ \underline{M(\mathbf{r}_l|\mathbf{v}_l)} &= \sum_{i=1}^{n} \log P(r_l^{(i)}|v_l^{(i)}) \\ &= \underline{\sum_{i=1}^{n} M(r_l|v_l),} \quad \text{(bit metrics)} \end{aligned}$$

This, so you can see we can conveniently write path metric as sum of branch metric and branch metric we can write as sum of this bit metric

(Refer Slide Time: 13:58)



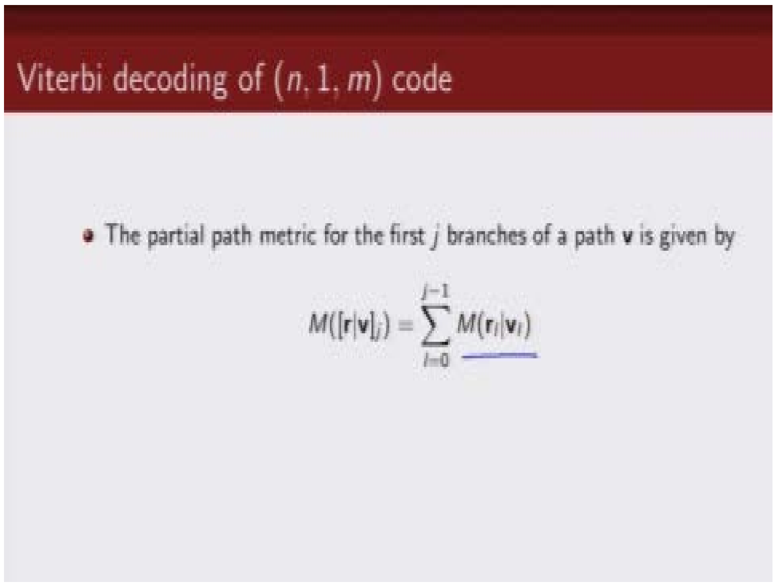
Viterbi decoding of $(n, 1, m)$ code

- The partial path metric for the first j branches of a path $\mathbf{v}$ is given by

$$M([\mathbf{r}|\mathbf{v}]_j) = \sum_{l=0}^{j-1} M(\mathbf{r}_l|\mathbf{v}_l)$$

Similarly we can define a partial path metric for the up to j^{th} branch of a path v

(Refer Slide Time: 14:09)



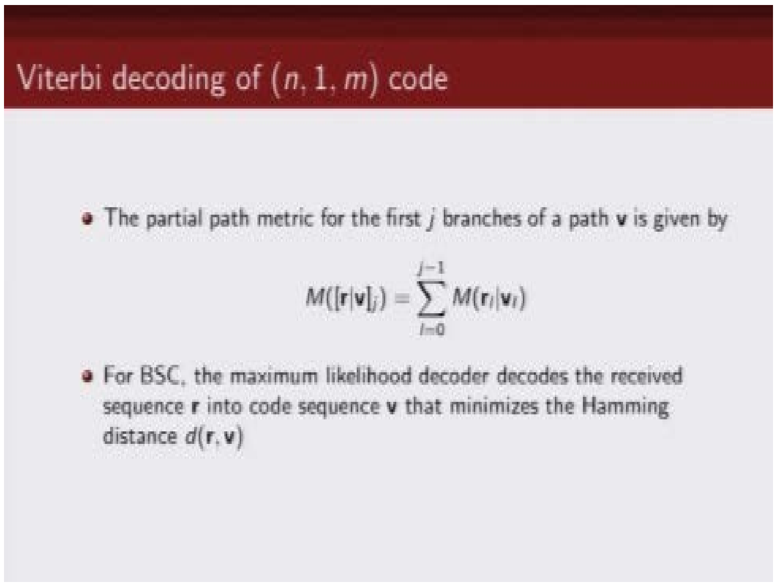
Viterbi decoding of $(n, 1, m)$ code

- The partial path metric for the first j branches of a path $\mathbf{v}$ is given by

$$M([\mathbf{r}|\mathbf{v}]_j) = \sum_{l=0}^{j-1} \underline{M(\mathbf{r}_l|\mathbf{v}_l)}$$

As Σ of path metrics up to j^{th} branch, so this is path metric computed from 0 to j-1 so this is what we are defining as partial path metric.

(Refer Slide Time: 14:24)



Viterbi decoding of $(n, 1, m)$ code

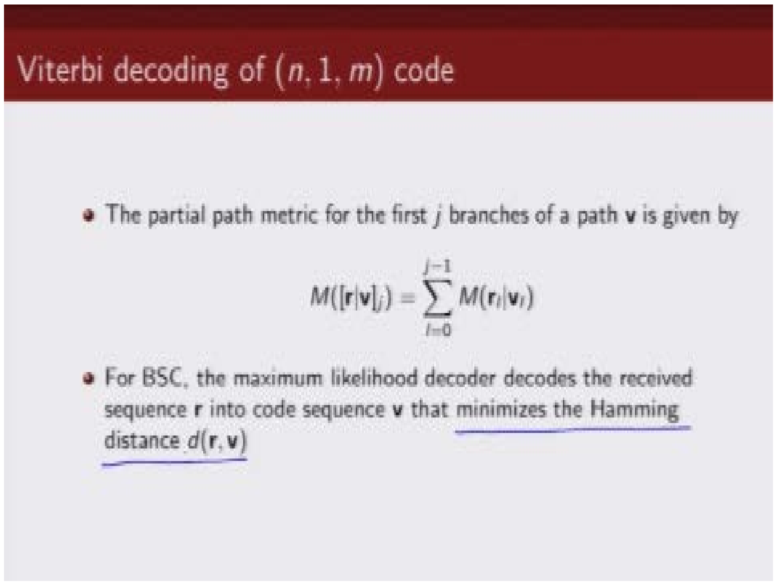
- The partial path metric for the first j branches of a path $\mathbf{v}$ is given by

$$M([\mathbf{r}|\mathbf{v}]_j) = \sum_{l=0}^{j-1} M(\mathbf{r}_l|\mathbf{v}_l)$$

- For BSC, the maximum likelihood decoder decodes the received sequence $\mathbf{r}$ into code sequence $\mathbf{v}$ that minimizes the Hamming distance $d(\mathbf{r}, \mathbf{v})$

Now what is the maximum likelihood decoding rule for a binary symmetric channel? For a binary symmetric channel the maximum likelihood decoder is one that decodes received sequence r into code sequence v.

(Refer Slide Time: 14:43)



Viterbi decoding of $(n, 1, m)$ code

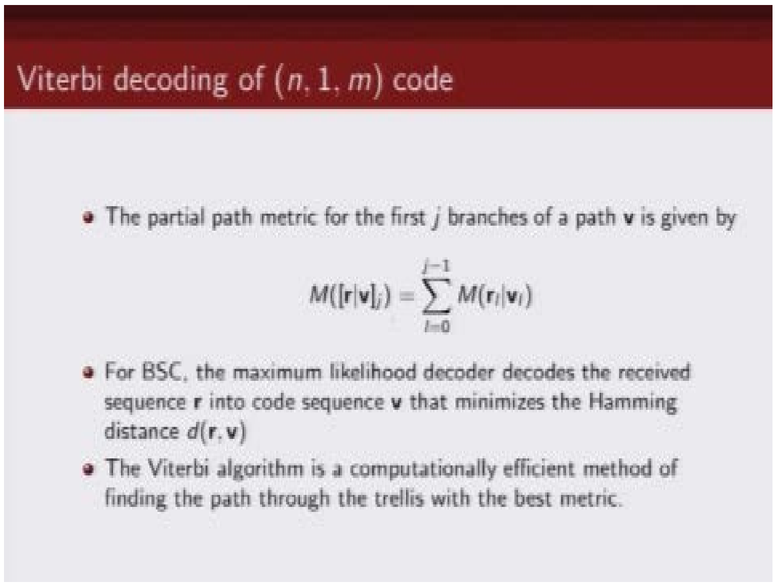
- The partial path metric for the first j branches of a path $\mathbf{v}$ is given by

$$M([\mathbf{r}|\mathbf{v}]_j) = \sum_{l=0}^{j-1} M(\mathbf{r}_l|\mathbf{v}_l)$$

- For BSC, the maximum likelihood decoder decodes the received sequence $\mathbf{r}$ into code sequence $\mathbf{v}$ that minimizes the Hamming distance $d(\mathbf{r}, \mathbf{v})$

Such that the hamming distance between the receive code sequence and the transmitted code sequence is minimized, this we have seen in the initial lecture on introduction to convolutional code, that maximum likelihood decoder for a binary symmetric channel will try to minimize the hamming distance between the receive code sequence and the transmitted code sequence.

(Refer Slide Time: 15:13)



Viterbi decoding of $(n, 1, m)$ code

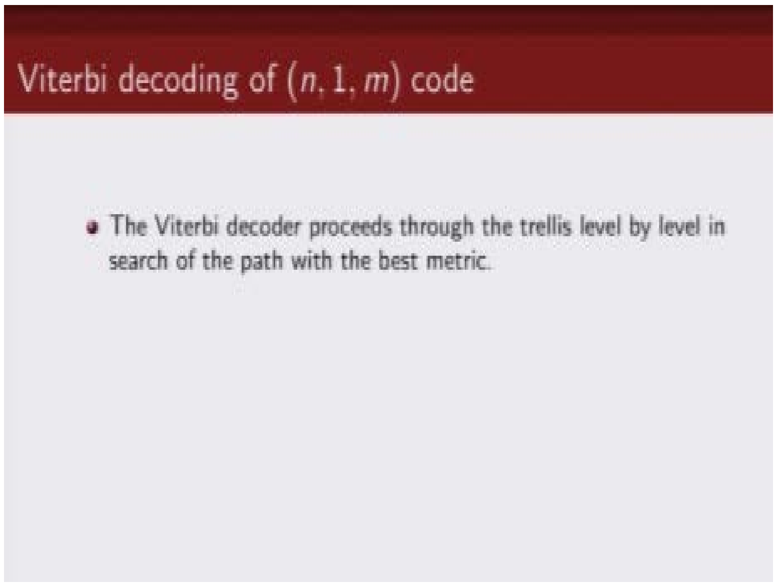
- The partial path metric for the first j branches of a path $\mathbf{v}$ is given by

$$M([\mathbf{r}|\mathbf{v}]_j) = \sum_{l=0}^{j-1} M(\mathbf{r}_l|\mathbf{v}_l)$$

- For BSC, the maximum likelihood decoder decodes the received sequence $\mathbf{r}$ into code sequence $\mathbf{v}$ that minimizes the Hamming distance $d(\mathbf{r}, \mathbf{v})$
- The Viterbi algorithm is a computationally efficient method of finding the path through the trellis with the best metric.

And Viterbi algorithm provides a computationally efficient method of basically just doing that and it tells us a way to efficiently find a path through this trellis which will maximize the likelihood function, in other words it will choose a path that will basically maximize the path metric.

(Refer Slide Time: 15:40)

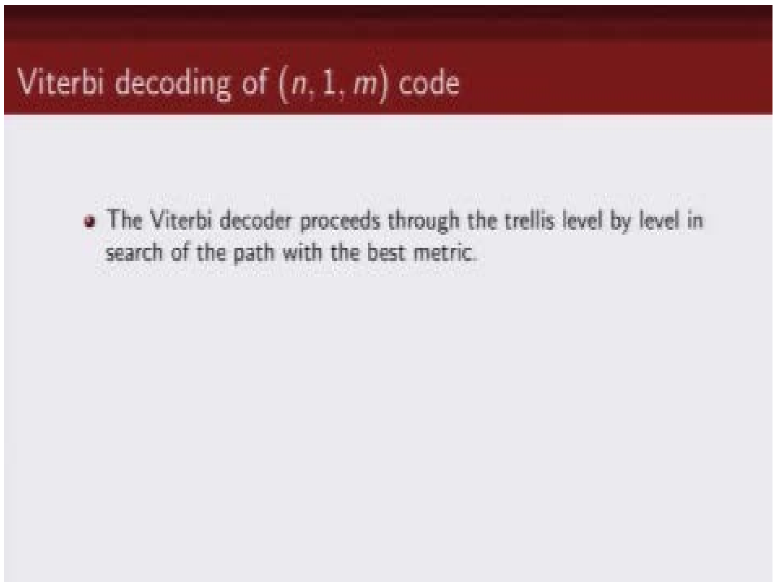


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.

Will choose a path which will maximize the path metric, so how does viterbi decoder works? So it proceeds through the trellis one time instant at a time in search of the best partial path metric up to that particular time instance, so you start with time t = 1 at that time instance you will try to compute the best path metric at each of the nodes, at each of the stage and this you repeat for next time instance, you do the same thing, you try to find out what is the best path metric for that particular time at that particular state.

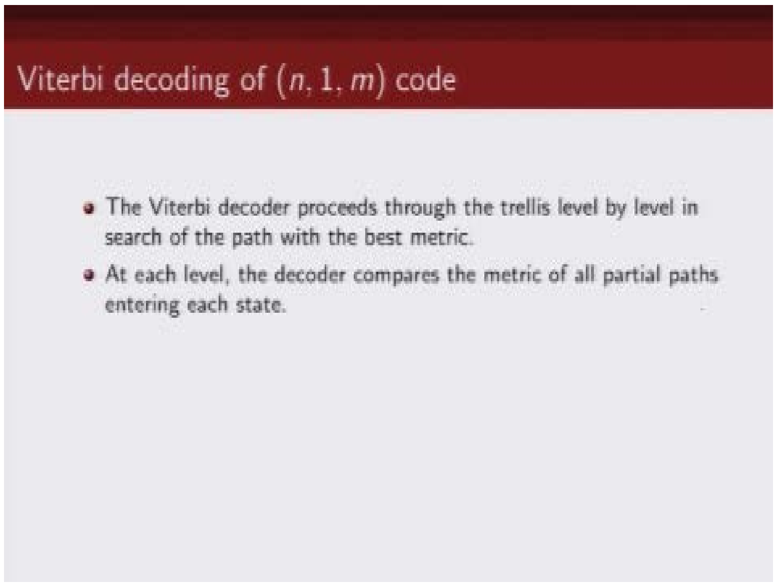
(Refer Slide Time: 16:25)



Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.

(Refer Slide Time: 16:27)

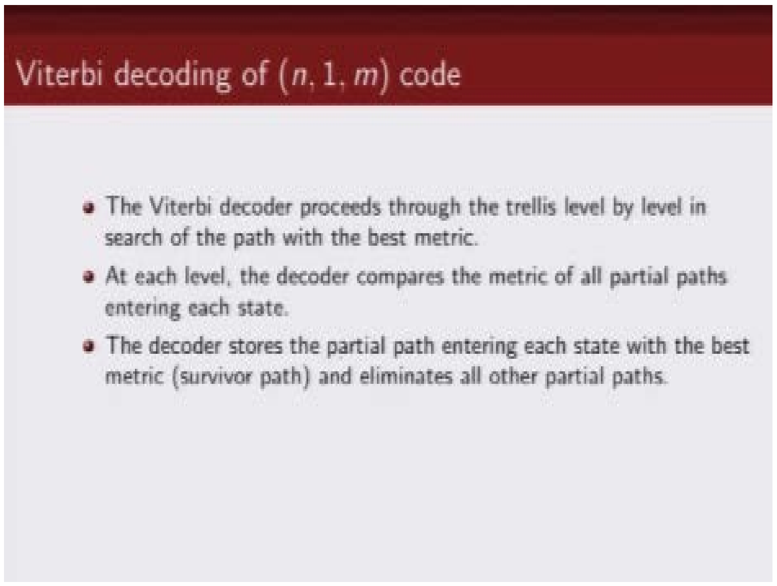


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.

So at each level the decoder is going to compute and compare the matrix of all partial path entering that particular state and it is going to choose the one which has the best partial path metric at that particular time instance.

(Refer Slide Time: 16:45)

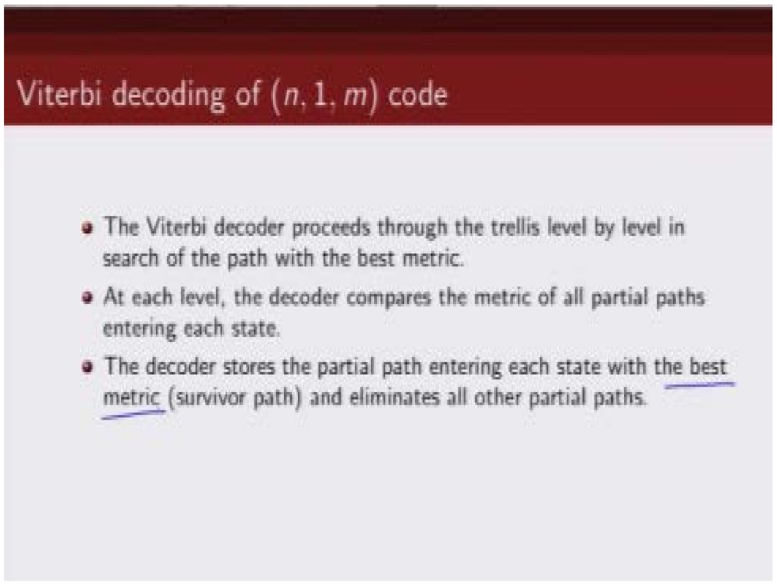


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.

So the decoder stores the partial path entering each state with the best metric.

(Refer Slide Time: 16:56)

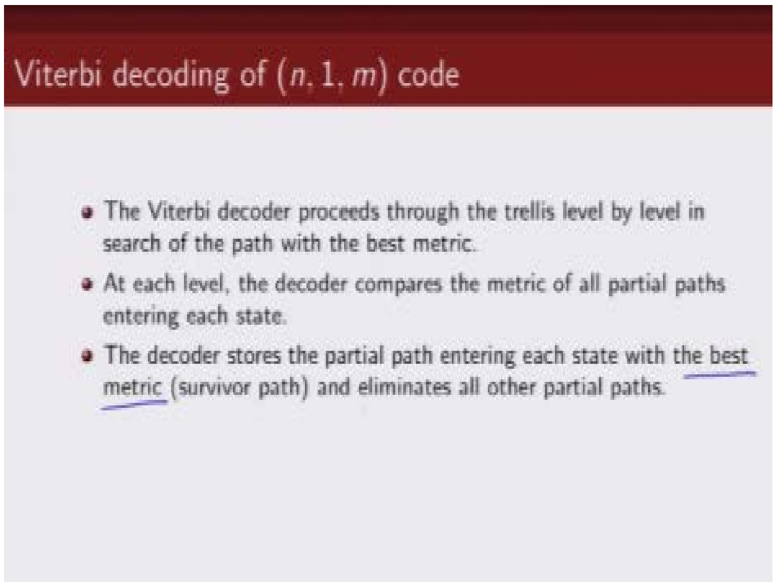


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.

Note that there are two branches which are converging for time t > m, there are two branches which are converging at each state. So from those two states from those two branches it is going to pick up the one branch which gives us the best path metric.

(Refer Slide Time: 17:18)

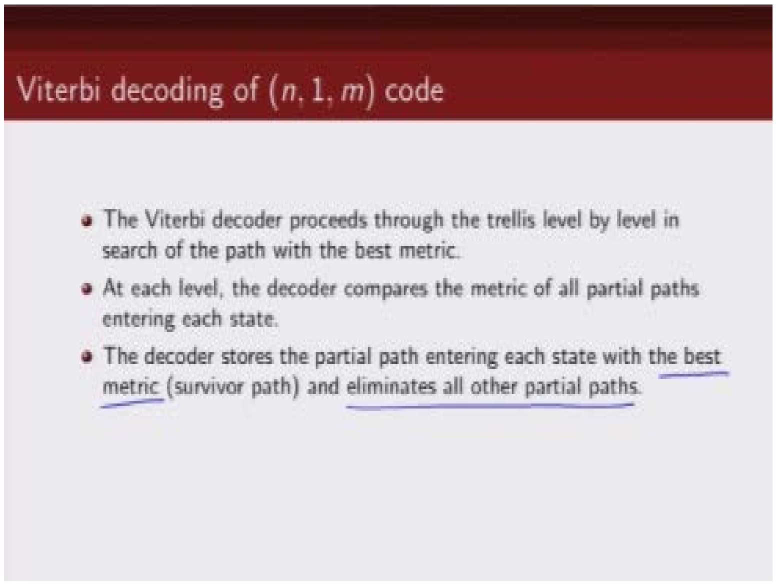


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.

So it will store that best path metric.

(Refer Slide Time: 17:22)

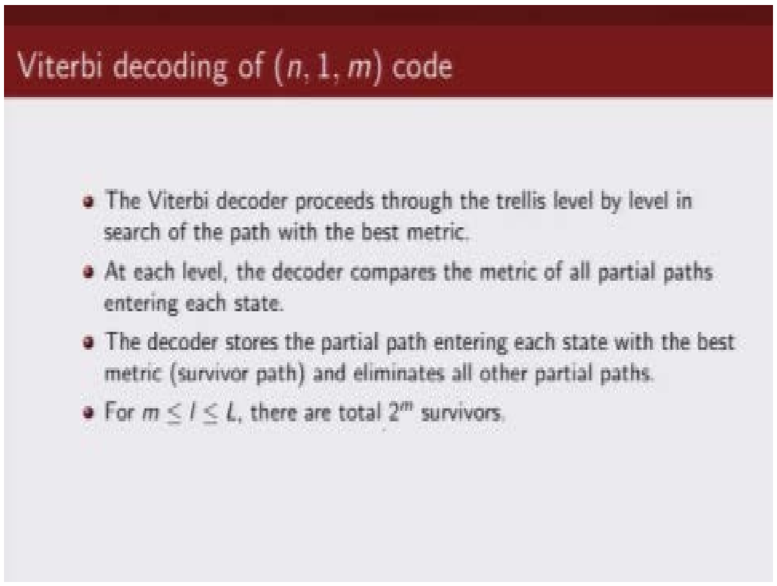


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.

And remove all other path metrics which do not gives us the best path metric.

(Refer Slide Time: 17:30)

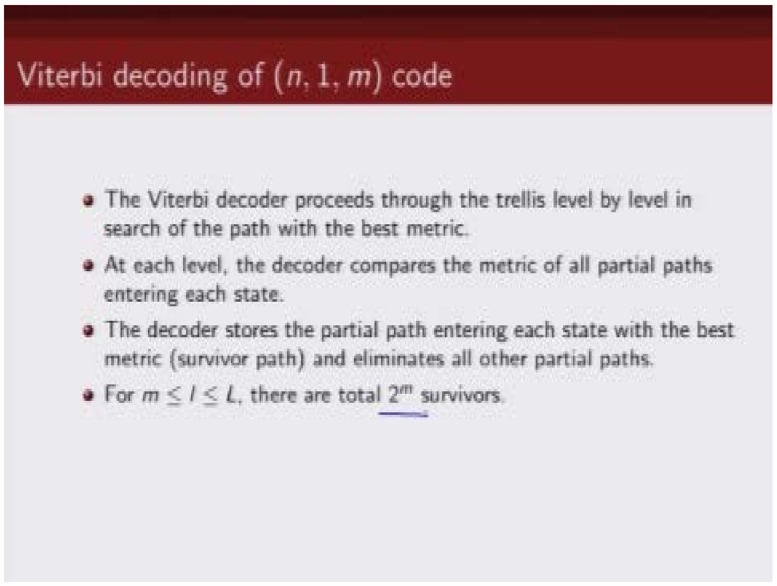


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.
- For $m \leq l \leq L$, there are total 2^m survivors.

So for time instance l between m and l there will be total 2^m survivors.

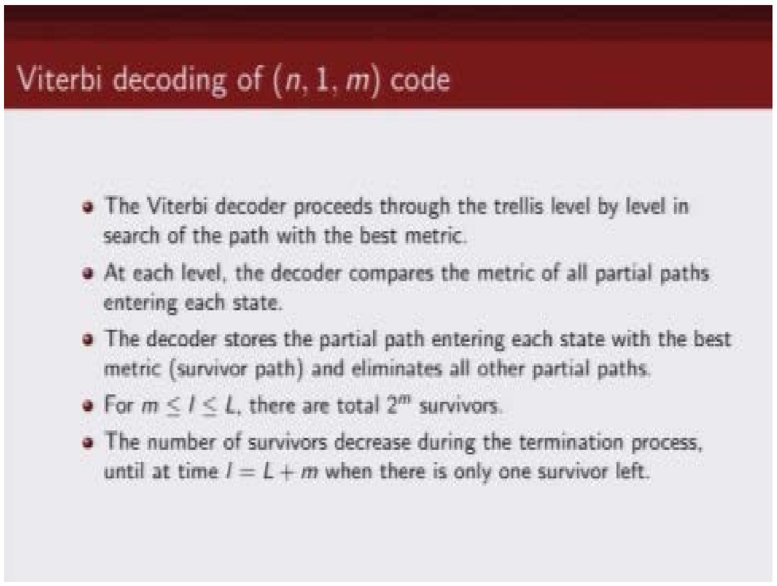
(Refer Slide Time: 17:37)



Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.
- For $m \leq l \leq L$, there are total $\underline{2^m}$ survivors.

(Refer Slide Time: 17:39)

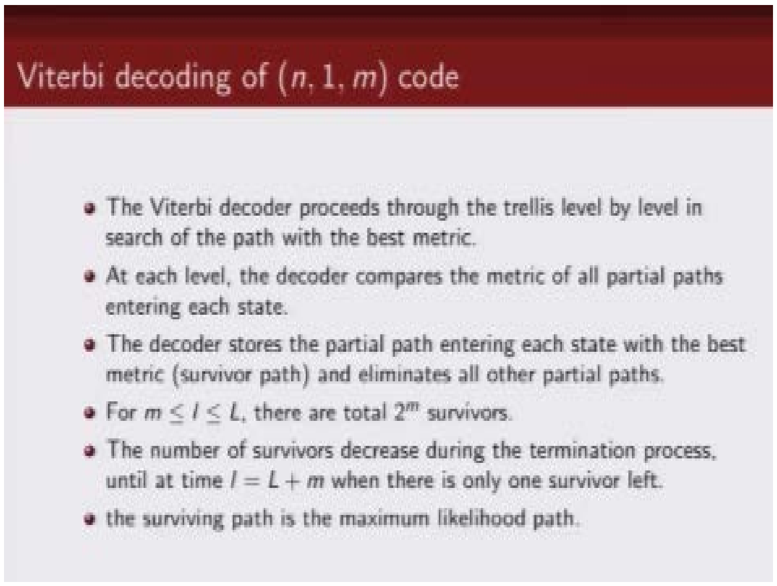


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.
- For $m \leq l \leq L$, there are total 2^m survivors.
- The number of survivors decrease during the termination process, until at time $l = L + m$ when there is only one survivor left.

And during this termination process the number of survivors will diminish until we reach the final state which all zero state where we will be left with only one survival and then we backtrack to get back our decoded path or decoded code word.

(Refer Slide Time: 17:58)

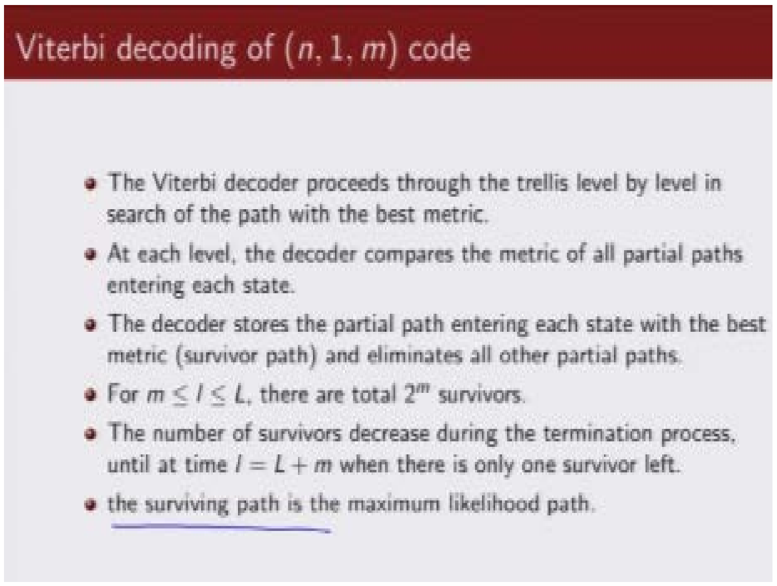


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.
- For $m \leq l \leq L$, there are total 2^m survivors.
- The number of survivors decrease during the termination process, until at time $l = L + m$ when there is only one survivor left.
- the surviving path is the maximum likelihood path.

So the, when we reach the final time instance which is final state all zero state we will be left with only one path and that is the surviving path is our maximum likelihood decoded path.

(Refer Slide Time: 18:08)

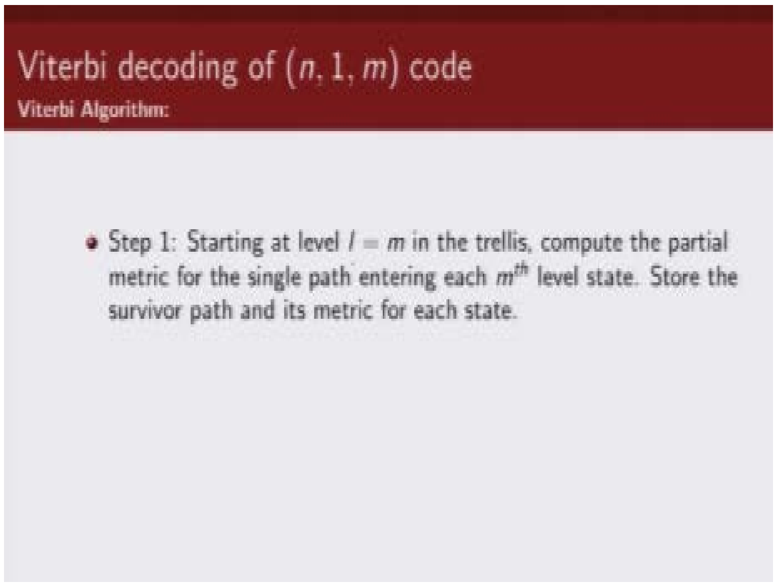


Viterbi decoding of $(n, 1, m)$ code

- The Viterbi decoder proceeds through the trellis level by level in search of the path with the best metric.
- At each level, the decoder compares the metric of all partial paths entering each state.
- The decoder stores the partial path entering each state with the best metric (survivor path) and eliminates all other partial paths.
- For $m \leq l \leq L$, there are total 2^m survivors.
- The number of survivors decrease during the termination process, until at time $l = L + m$ when there is only one survivor left.
- the surviving path is the maximum likelihood path.

And then we back track to get back what were the code bits and information bits corresponding to that particular path.

(Refer Slide Time: 18:19)



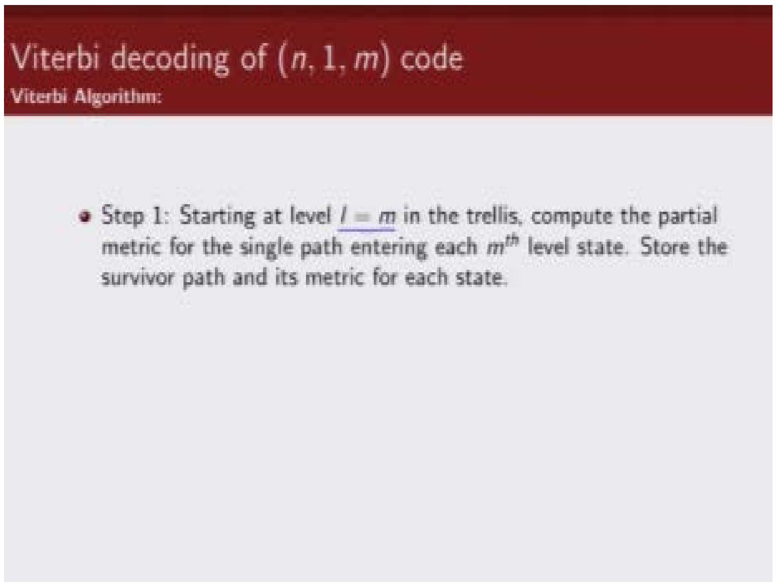
Viterbi decoding of $(n, 1, m)$ code

Viterbi Algorithm:

- Step 1: Starting at level $l = m$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.

So how does, so to summarize basically how does this viterbi algorithm works, so let us say starting at time t or l up to m.

(Refer Slide Time: 18:34)



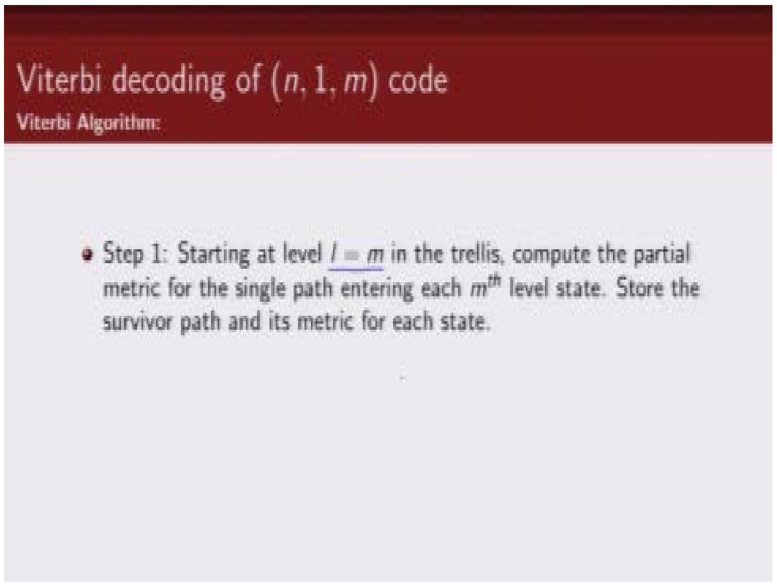
Viterbi decoding of $(n, 1, m)$ code

Viterbi Algorithm:

- Step 1: Starting at level $\underline{l = m}$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.

Note up to, up to the memory order there is only one path from the initial state or zero state to any other state.

(Refer Slide Time: 18:45)



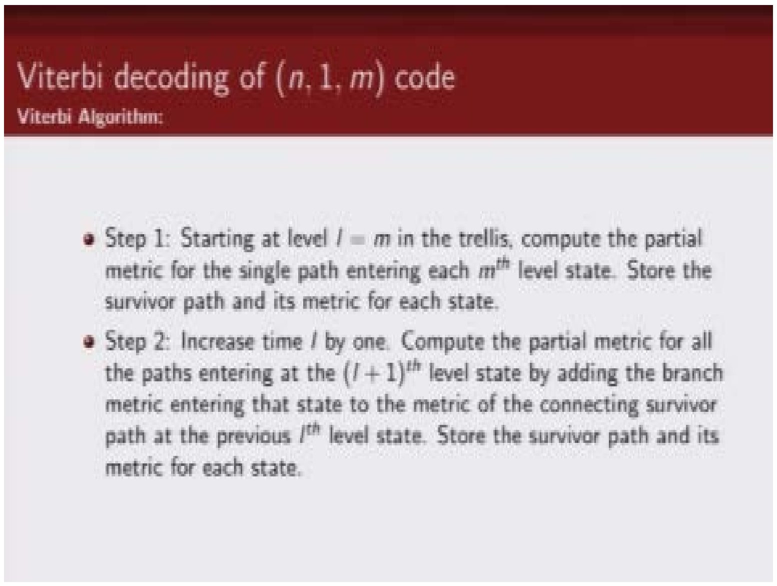
Viterbi decoding of $(n, 1, m)$ code

Viterbi Algorithm:

- Step 1: Starting at level $l = m$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.

So starting at time up to l = m in the trellis we compute the partial path metric but the single path from all zero state to any particular m^{th} state. And we store the survivor path as well as the metric value for that particular state.

(Refer Slide Time: 19:05)



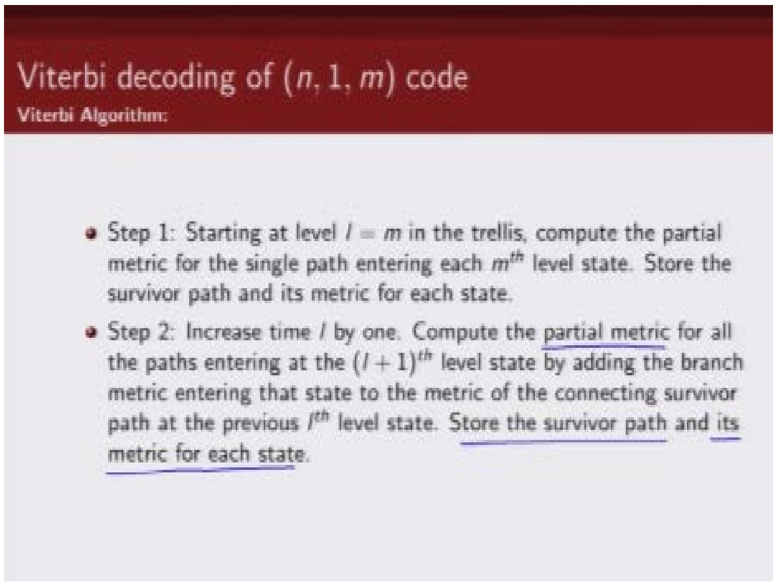
Viterbi decoding of $(n, 1, m)$ code

Viterbi Algorithm:

- Step 1: Starting at level $l = m$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.
- Step 2: Increase time l by one. Compute the partial metric for all the paths entering at the $(l+1)^{th}$ level state by adding the branch metric entering that state to the metric of the connecting survivor path at the previous l^{th} level state. Store the survivor path and its metric for each state.

Next we increase the time by one instance.

(Refer Slide Time: 19:12)



Viterbi decoding of $(n, 1, m)$ code

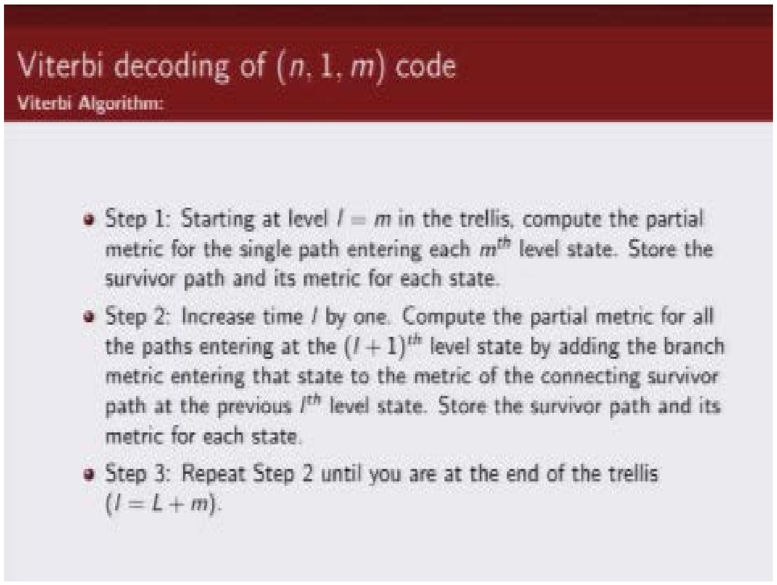
Viterbi Algorithm:

- Step 1: Starting at level $l = m$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.
- Step 2: Increase time l by one. Compute the partial metric for all the paths entering at the $(l + 1)^{th}$ level state by adding the branch metric entering that state to the metric of the connecting survivor path at the previous l^{th} level state. Store the survivor path and its metric for each state.

And now we compute the partial path metric for all the paths for at that time instance which are entering that particular state and we compare the partial path metric and choose the one which gives us the best path metric and we ignore other paths, so at next time instance we are going to store the survival path which is the one which has the best metric and we will also store what its metric value is and how do you compute the partial metric at time l+1 given that you know the path metric at time l.

So what we do is we add to the path metric of time l, we add the branch metric value to whatever is the partial path metric at that particular state and that is how we compute the path metric at time l+1, now this will be clear when we illustrate this with another with an example immediately after this.

(Refer Slide Time: 20:24)



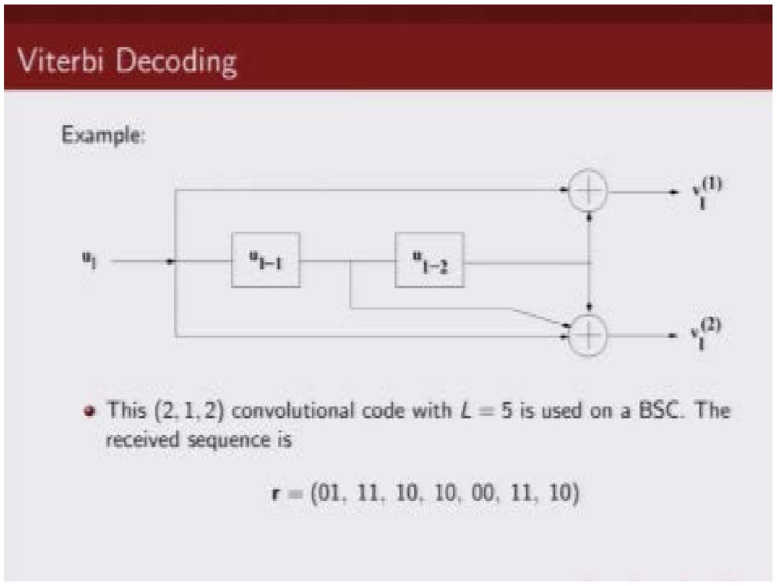
Viterbi decoding of $(n, 1, m)$ code

Viterbi Algorithm:

- Step 1: Starting at level $l = m$ in the trellis, compute the partial metric for the single path entering each m^{th} level state. Store the survivor path and its metric for each state.
- Step 2: Increase time l by one. Compute the partial metric for all the paths entering at the $(l+1)^{th}$ level state by adding the branch metric entering that state to the metric of the connecting survivor path at the previous l^{th} level state. Store the survivor path and its metric for each state.
- Step 3: Repeat Step 2 until you are at the end of the trellis $(l = L + m)$.

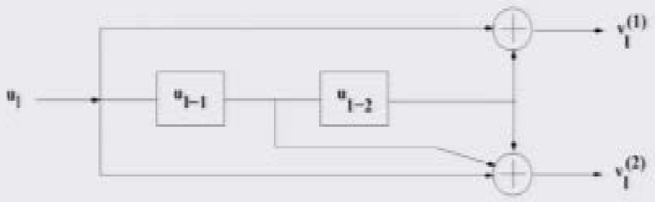
And we repeat this process until we reach the end of the trellis diagram.

(Refer Slide Time: 20:31)



Viterbi Decoding

Example:

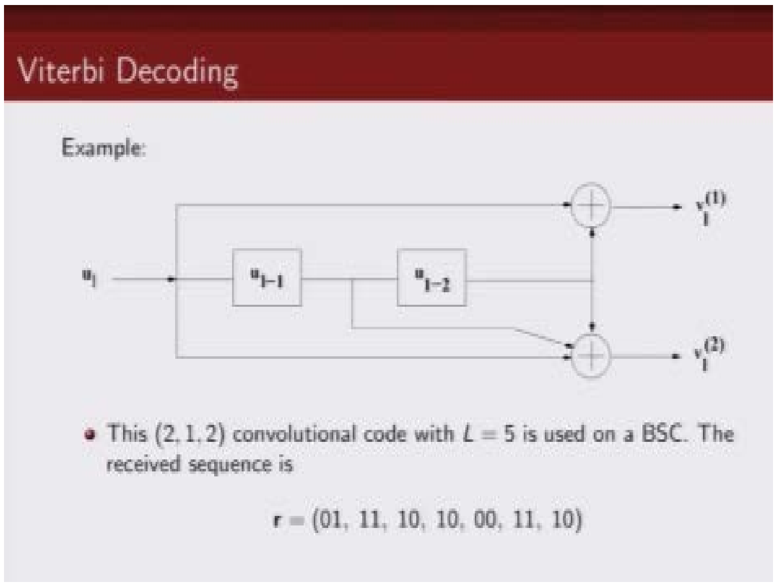


- This $(2,1,2)$ convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

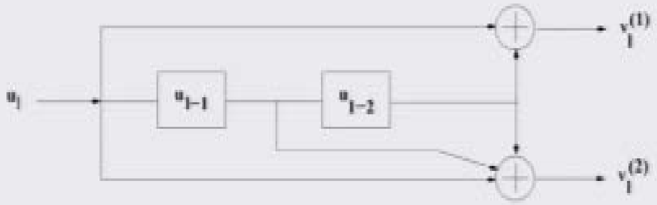
So let us come to an example, to show how we compute the partial path metric, how do we choose the best surviving path, how do we eliminate other paths which do not give us the best path metric, okay.

(Refer Slide Time: 20:51)



Viterbi Decoding

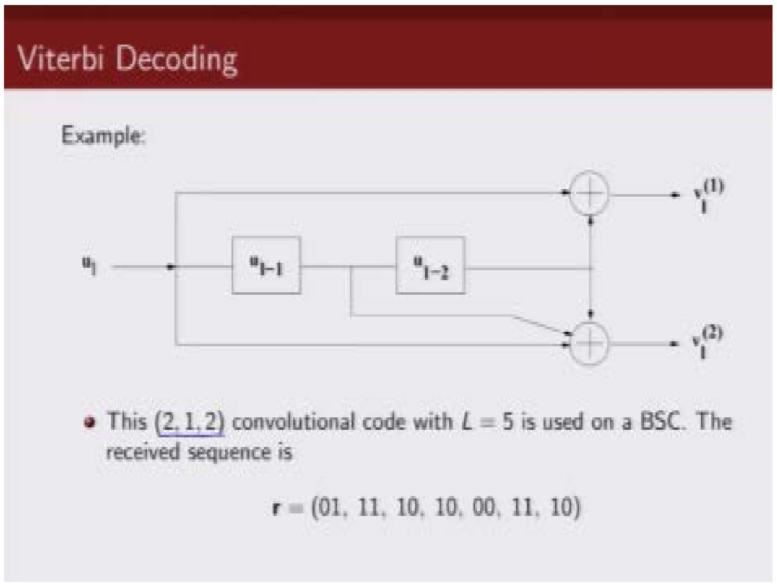
Example:



- This (2, 1, 2) convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

(Refer Slide Time: 20:55)



Viterbi Decoding

Example:

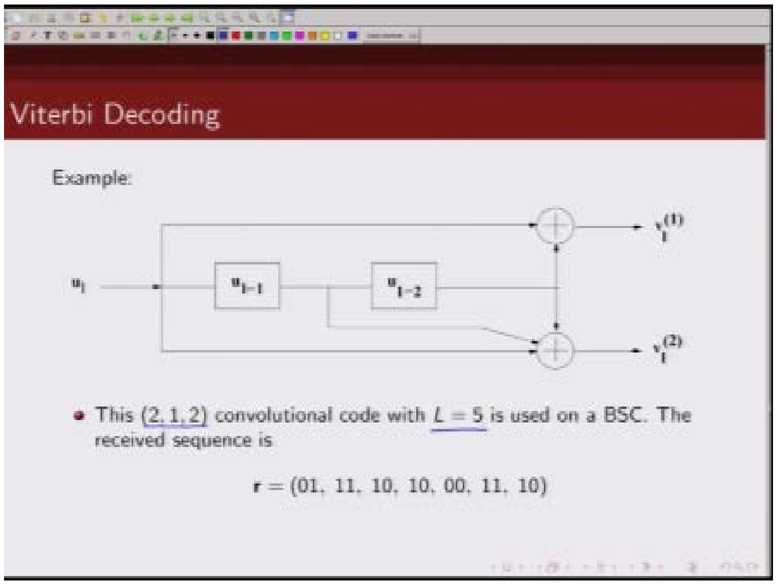
u_l
u_{l-1}
u_{l-2}
$v_l^{(1)}$
$v_l^{(2)}$

- This (2, 1, 2) convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

So let us take an example of a rate one by two code which has memory two this is the encoder of a rate one by two code.

(Refer Slide Time: 20:51)


Viterbi Decoding

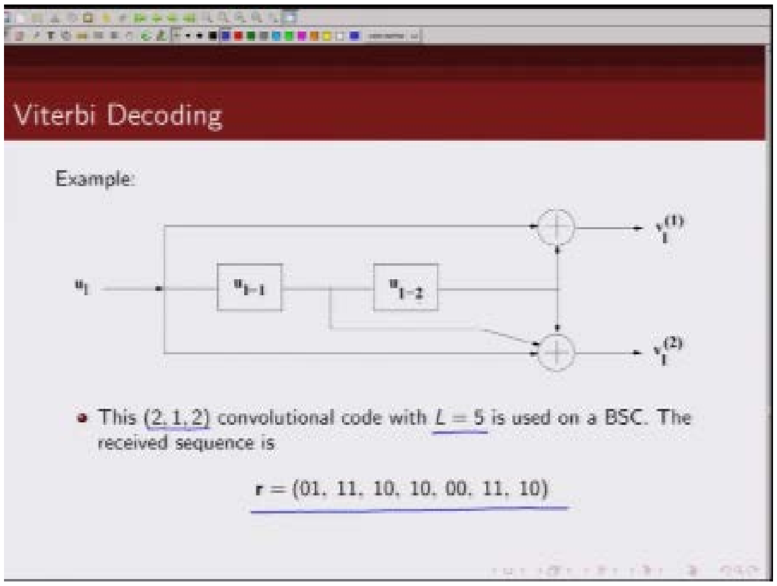
Example:

- This (2, 1, 2) convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

We are given that number of information bits were 5, now since its memory is 2 we require two bits to bring it back to all zero state so total basically

(Refer Slide Time: 21:18)



Viterbi Decoding

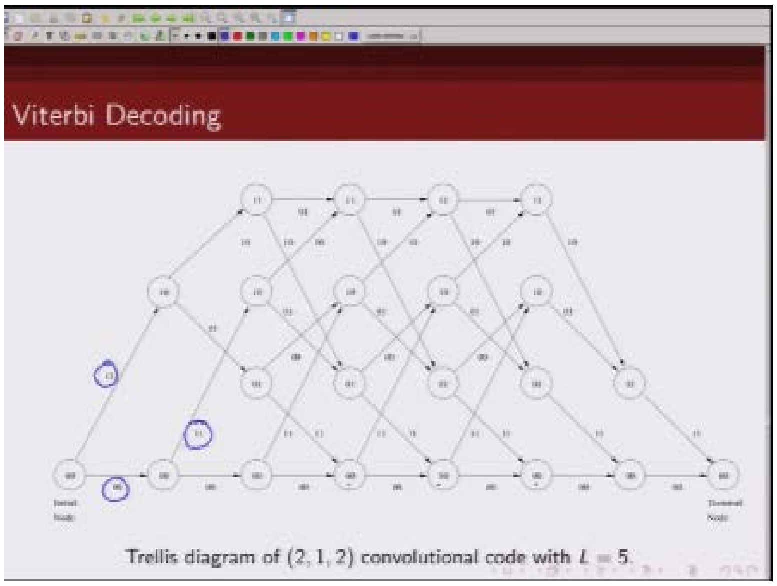
Example:

- This (2, 1, 2) convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

We are transmitting 5 information bits plus 2 tail bits. So total 7 bits and since it is a rate ½ code so 7 bits will result in 14 coded bits. So this is basically the receive sequence corresponding to those 14 bits that we transmitted. This is the receive sequence over the binary symmetric channel. And the question that we have to answer is given this receive sequence we need to find out what is the transmitted code word or what is the transmitted information sequence. So how do we proceed, the first step is we will draw the state diagram of this convolutional encoder, and we will draw the trellis diagram of this convolutional encoder.

(Refer Slide Time: 22:10)

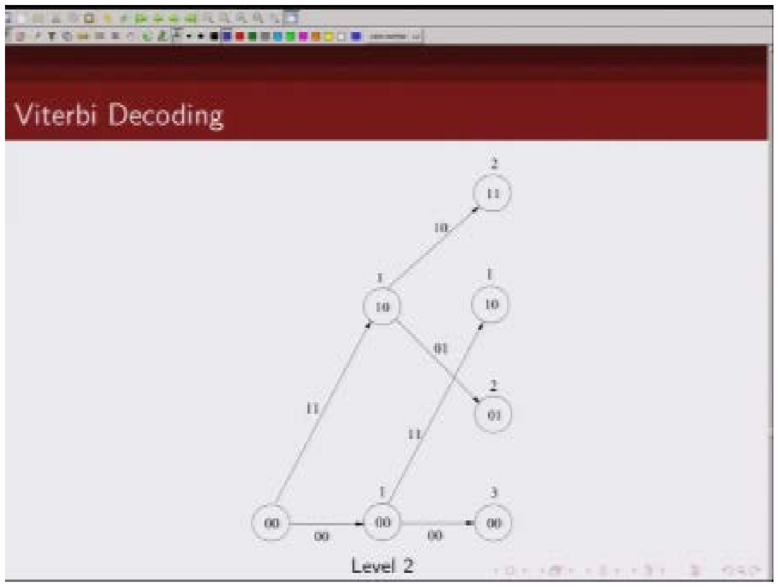


Viterbi Decoding

Trellis diagram of (2, 1, 2) convolutional code with $L = 5$.

So what I have here is I have drawn the trellis diagram of this rate ½ convolutional encoder. Please note, initially we are starting with all zero state so its time instance 1, 2, 3, 4, 5. These were our information bits, now what happened after this? We are terminating the convolutional code to all zero state. So number of states reduces by half at each time instance so from here from 4 it becomes 2 and from here it becomes ones, one state. So you can see basically we are transmitting 5 information bits and 2 tail bits and what I have labeled here is I have, I have at each branch I have labeled what is the code word correspond to this transition.

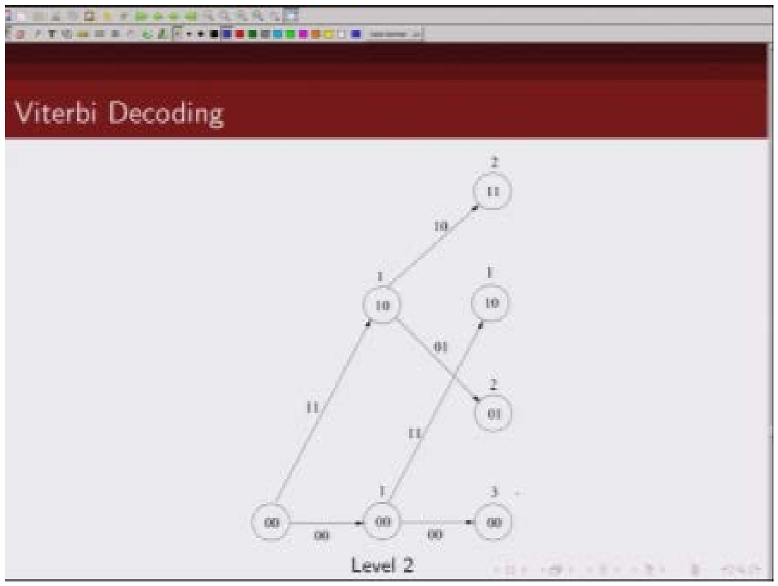
(Refer Slide Time: 23:09)



Viterbi Decoding

So what is the first step, we will first try to compute the partial path metric up to time t=2. Why time t up to, t up to 2 because the memory order of this convolutional code was 2.

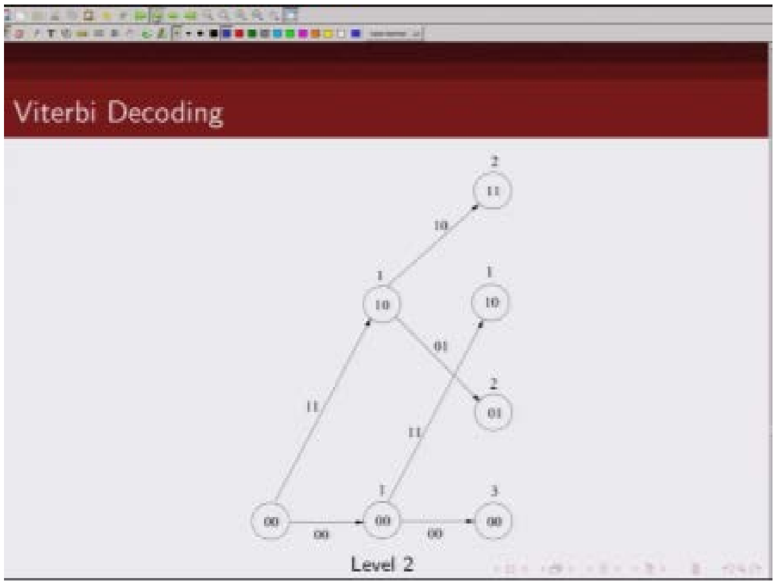
(Refer Slide Time: 23:25)



Viterbi Decoding

So from all zero state up to time instance is 2, there is a single path from all zero state to any particular state. So for example, if you want to reach state 01, we can reach it from first we will go to 00 state to 10 state, and from 10 we will go to 01 state okay. Now how do we compute these path metric, so let us go back and see.

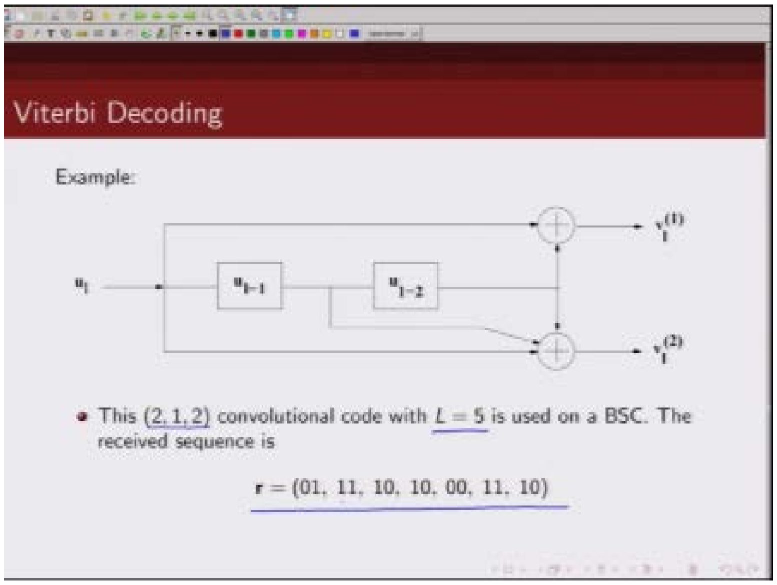
(Refer Slide Time: 23:53)



Viterbi Decoding

Level 2

(Refer Slide Time: 23:54)



Viterbi Decoding

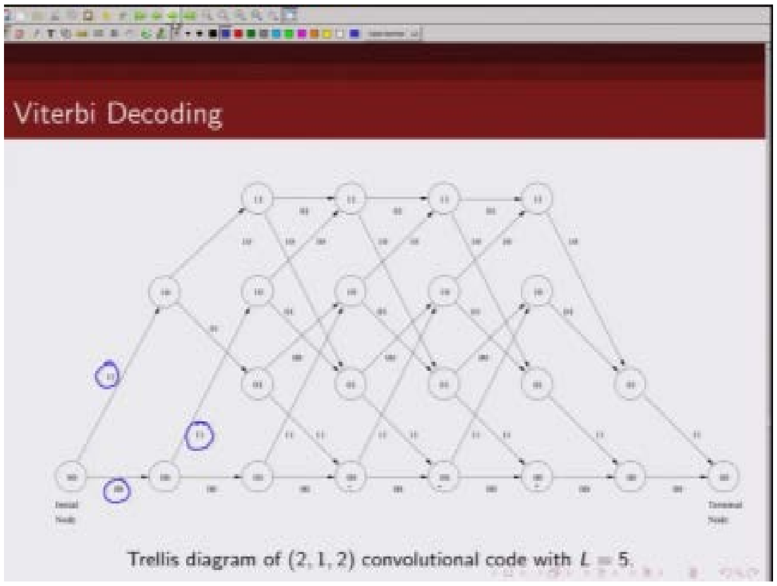
Example:

- This $(2, 1, 2)$ convolutional code with $L = 5$ is used on a BSC. The received sequence is

$$\mathbf{r} = (01,\ 11,\ 10,\ 10,\ 00,\ 11,\ 10)$$

What our first to receive sequence word, it was 01 and 11.

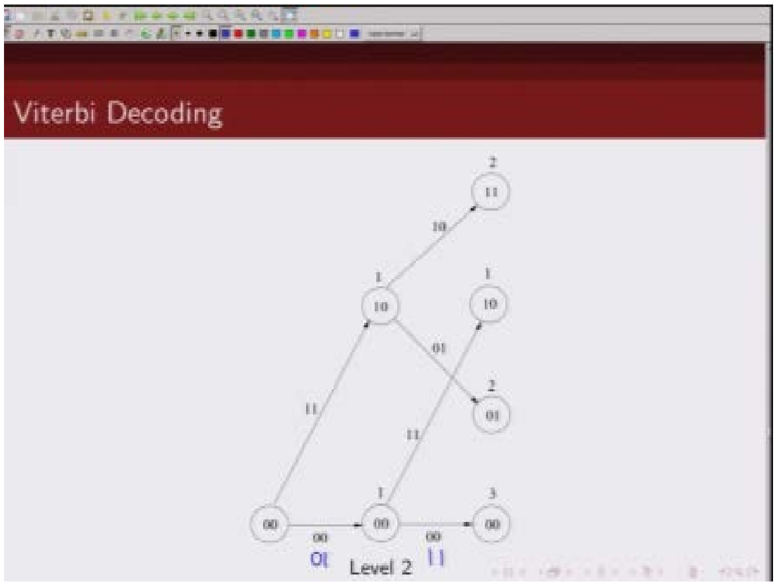
(Refer Slide Time: 24:02)



Viterbi Decoding

Trellis diagram of $(2, 1, 2)$ convolutional code with $L = 5$.

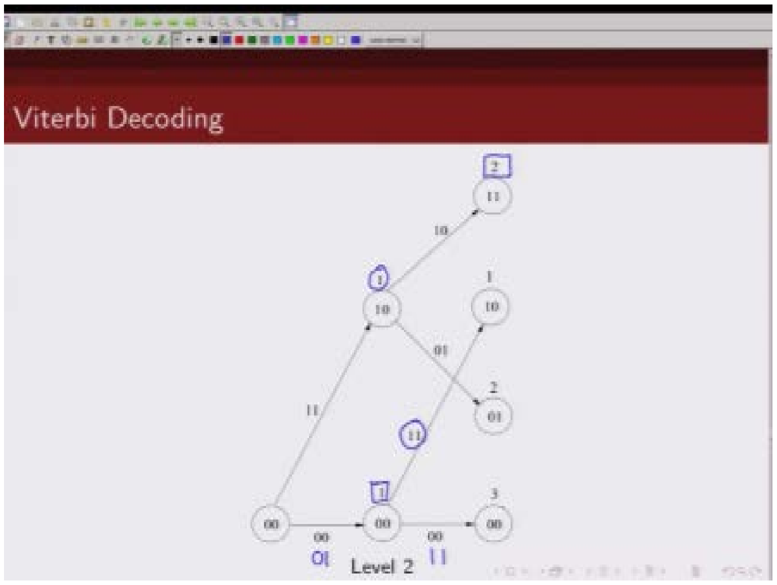
(Refer Slide Time: 24:02)



Viterbi Decoding

So we had received here 01 and 11. Now remember for a binary symmetric channel the maximum like you decoder is trying to find the hamming distance between the receive sequence and the code sequence and it will try to choose a path which will minimize the hamming distance between the r receive sequence and code word v.

(Refer Slide Time: 24:34)

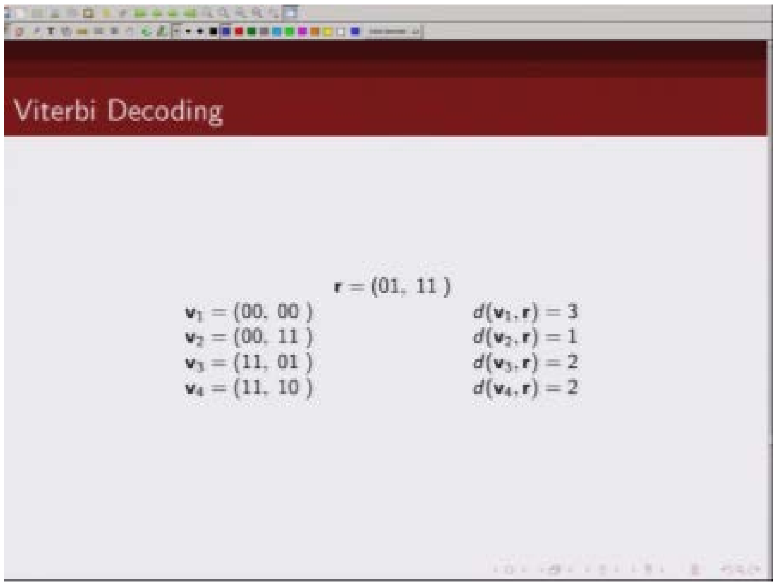

Viterbi Decoding

So let us look at this path from 00 to 00, what is what should have been the code word 00 we had received 01, so what this a path metric corresponding to this that is 1. So this is the path metric. Now look at here, code sequence is 11 this is 01 so path metric is differing in hamming distance 1 so that is my path metric here. Now what is the path metric here, this is equal to the path metric, partial path metric, at this time plus branch metric corresponding to this.

So this is 10 and what we received was 11, so what is the hamming distance between 11 and 10 that is 1. So 1 plus 1 so 2, so this will have a path metric of 2. Similarly this, what is the path metric at this instance so note the branch metric is for this branch is 0, because code sequence is 11 and receive sequence is 11, so hamming distance is 0. 1 what was the partial path metric the partial path metric here was 1 plus 1 and the branch metric here is 0, so 1 plus 0 that is 1.

Similarly we can find out what is the partial path metric here, this is going to be partial path metric here plus the branch metric of this. And what is the branch metric of this, this is 01 and receive sequence is 11 so it is differing in the first bit location. So 1 plus 1, that is 2. And similarly here the partial path metric is 1, the branch metric in this case is 2, so 1 plus 2 so partial path metric here is 3.

(Refer Slide Time: 26:33)



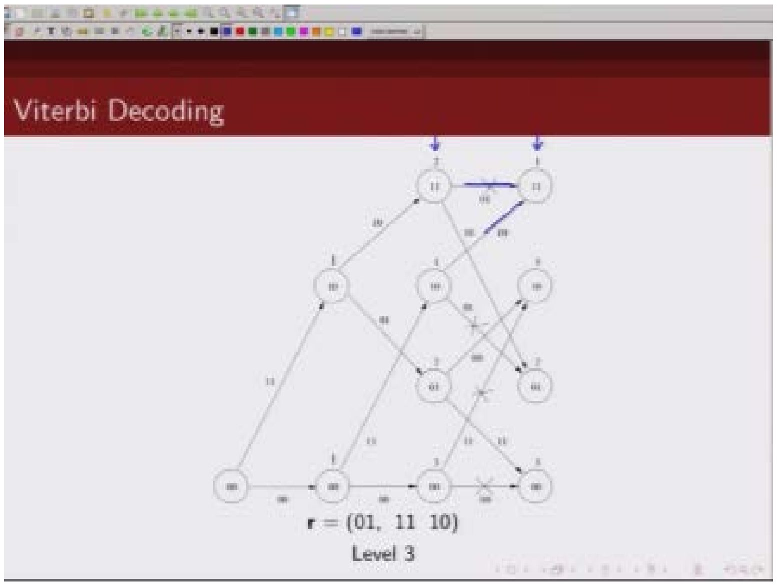
Viterbi Decoding

$\mathbf{r} = (01,\ 11)$

$\mathbf{v}_1 = (00,\ 00)$	$d(\mathbf{v}_1, \mathbf{r}) = 3$
$\mathbf{v}_2 = (00,\ 11)$	$d(\mathbf{v}_2, \mathbf{r}) = 1$
$\mathbf{v}_3 = (11,\ 01)$	$d(\mathbf{v}_3, \mathbf{r}) = 2$
$\mathbf{v}_4 = (11,\ 10)$	$d(\mathbf{v}_4, \mathbf{r}) = 2$

And this is what I have also illustrated here our receive sequence of 01 and basically these are the hamming distance between code sequence and the receive sequence.

(Refer Slide Time: 26:45)



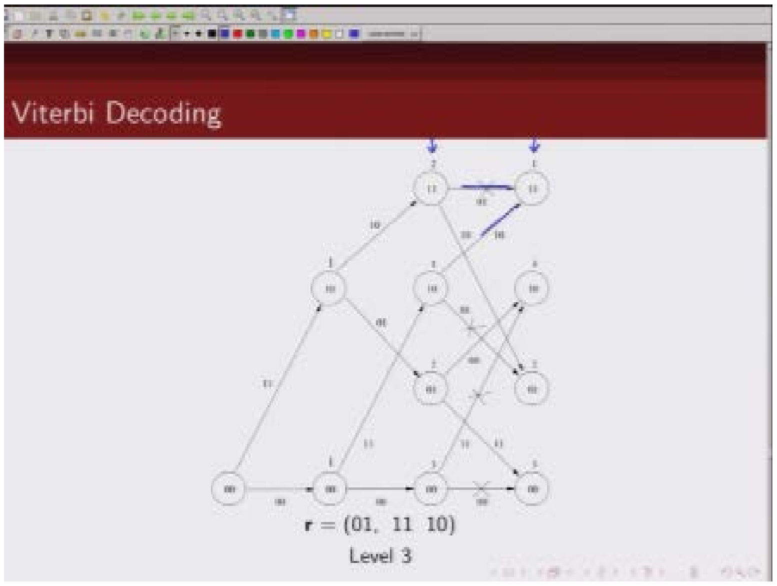
Viterbi Decoding

r = (01, 11 10)

Level 3

Now once we know what are the path metric here at this time instance. The next step what do we do, we need to find the path metric here. Now how do you find the path metric here, now note here we have two branches which are entering the state, one is this, other is this right. So how do we compute the partial path metric here, this we compute by adding to the partial path metric of this state plus the branch metric of this branch or the path metric at this state plus the branch metric corresponding to this transition. So we are going to choose which one of these two is a better gives us a best, better path metric and then we are going to choose that path as our surviving path.

(Refer Slide Time: 27:54)

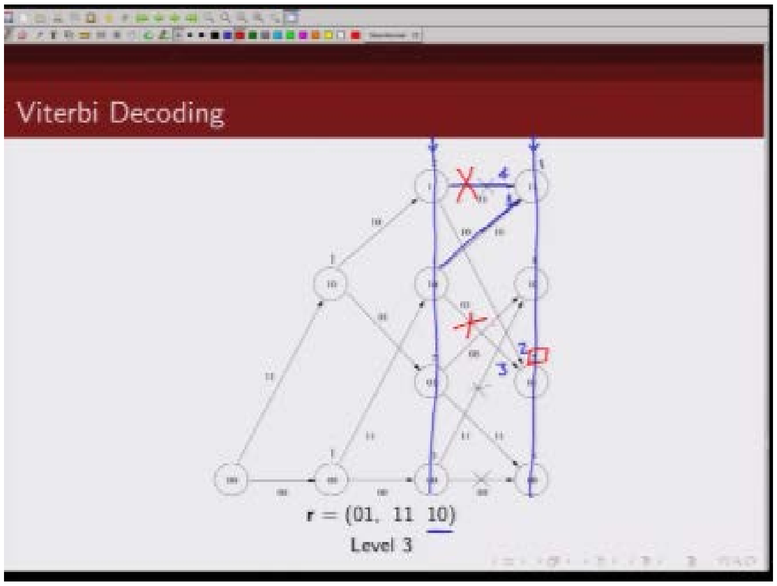

Viterbi Decoding

$\mathbf{r} = (01,\ 11\ \ 10)$

Level 3

And we are going to delete the path which does not give us a best metric and we will also store what is the best path metric corresponding to that particular transition.

(Refer Slide Time 28:00)


Viterbi Decoding

r = (01, 11 10)

Level 3

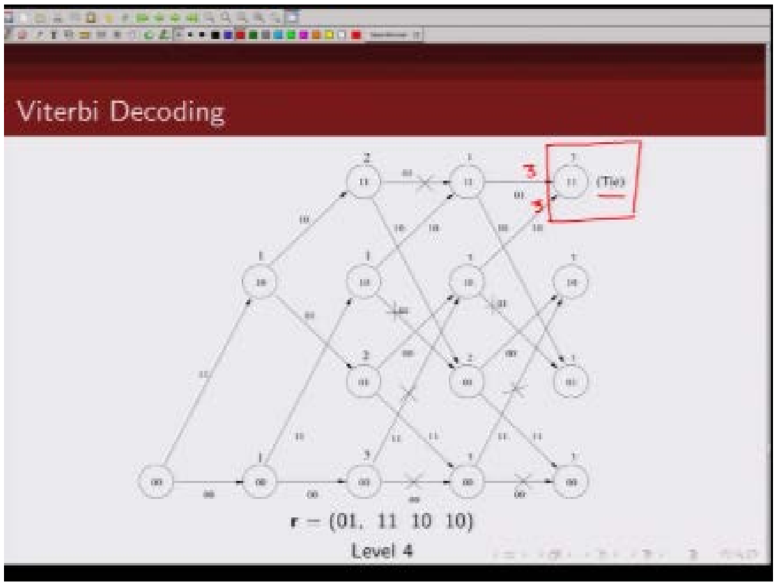
So we are looking at this time instance, this trellis section, we are looking here so what was a receive sequence here it was one zero, so what is the branch metric corresponding to this transition, this transition, this is 01 this is one zero so hamming distance in this case is 2, and what was the hamming distance here, path metric was 2, so 2+2 this would give me partial path metric of 4. Now let's compute the path metric of this.

Here the partial path metric was one and this was one zero, one zero and one zero the hamming distance is zero, so corresponding to this path metric is one. Now which one is a better path metric, one or a 4? This one is a better path metric, why because hamming distance here is less it is one whereas for corresponding to this path it is 4, so what we are going to do we are going to delete this path and we are going to keep this path, not only we are going to keep this path we are also going to keep.

The corresponding path metric which is one, similarly we will do the same step for other nodes, let us take this example. This node zero one so one path metric we will get by adding to the path metric of this partial path metric of this is which is one and zero one and one zero which is hamming distance of 2, so 1+2 this path is giving me a partial path metric of 3. What about this path, this is 2 and this is one zero, one zero, one zero having distances zero, so 2+0 that will give me partial path metric of 2.

Now clearly this is not the best path metric so I will keep this surviving branch and corresponding path metric which is basically given by 2. So at each node I am going to find out partial path metric corresponding to all the branches that terminate at that particular state and I am going to keep the same, I am going to keep the one which gives me the best path metric, I am going to keep track of what is the path metric value and also keep track of the surviving path, so this step will be repeated.

(Refer Slide Time 30:52)



Viterbi Decoding

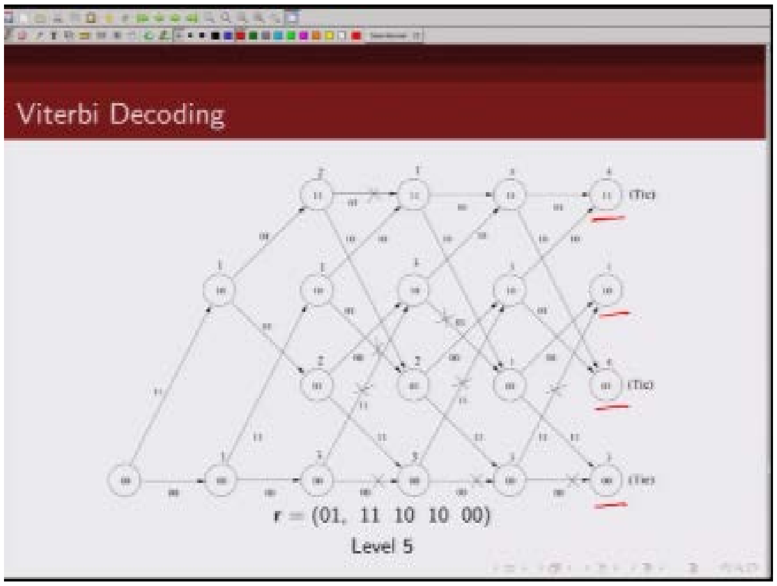
r = (01, 11 10 10)

Level 4

The same thing I do here, now let us take an example here of this particular node, now how do we compute the path metric here, the partial path metric here. This is the partial path metric here will be either partial path metric here plus this branch metric which is one zero and one zero, so this is branch metric here is zero so 3+0 is 3. Now look, let us look at this the partial path metric here is one and this is zero one, so zero one and one zero so this hamming distance is 2, 1+2 is 3 so note here both the paths here

Have the same path metric so there is a tie, so what do I do when there is a tie, so in this case both the paths are equally likely so I can just pick any one of them, so this process is going to be repeated so I will continue with this process.

(Refer Slide Time 32:01)


Viterbi Decoding

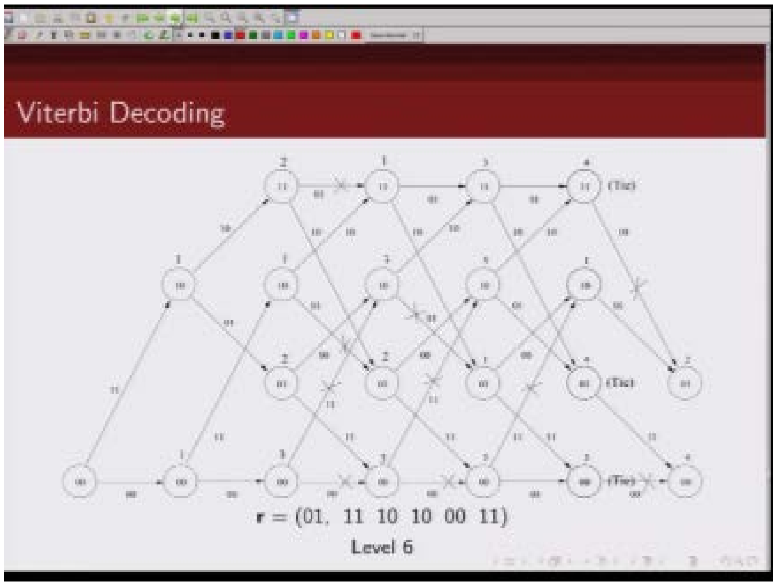
r = (01, 11 10 10 00)

Level 5

So again I repeat at each time instance what I am doing is I am trying to compute the partial path metric at each of these states, and how do I compute the partial path metric at each of these state? This is the partial path metric at previous time instance + the branch metric corresponding to the transition which will bring you to that particular state and when you have multiple branches converging at particular state you need to pick the branch which will give you the best path metric.

That will be your surviving path and you need to keep track of the metric value as well. So we continue this step

(Refer Slide Time 32:51)



Viterbi Decoding

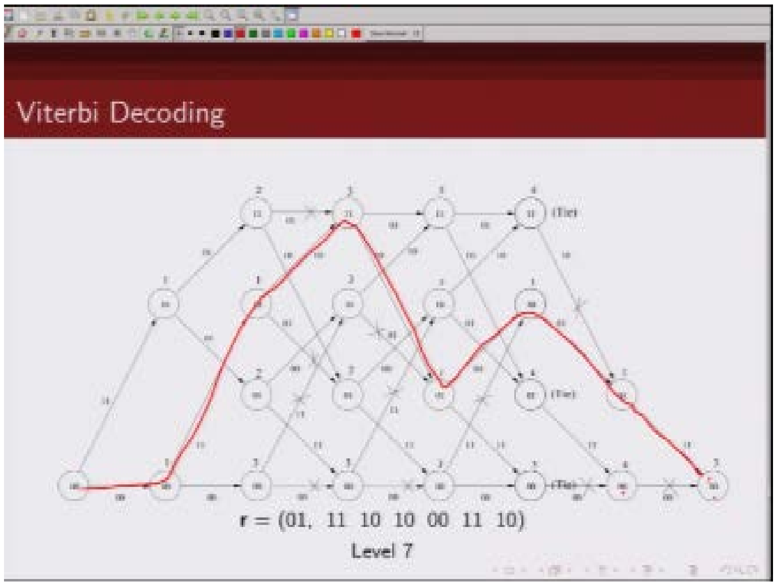
$\mathbf{r} = (01,\ 11\ 10\ 10\ 00\ 11)$

Level 6

Until we come to the end, so as we know after 5 bits because tail bits are coming, so number of states are getting reduce so from 4 the number of states became 2 here, and then you have number of states basically one. So once you come to this point then there will be only one surviving path okay. So once you have only one surviving path then what you need to do is you just need to backtrack to find out what is the transmitted code word, so you can see here when we reached this all zero final state.

There is only one surviving path and that path if we track back

(Refer Slide Time 33:39)



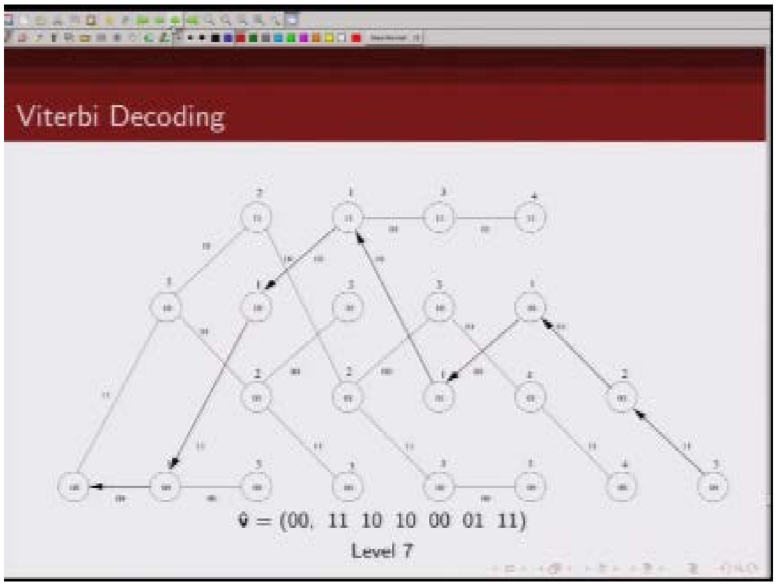
Viterbi Decoding

r = (01, 11 10 10 00 11 10)

Level 7

It is this path, this path, this path, this path, this path, this path, so you can see there is one survival path basically that you will get and this is my decoded code sequence. So corresponding to this receive sequence my decoded code sequence is zero, zero, one, one, one zero, one zero, zero, zero, zero one, one one, okay and this is

(Refer Slide Time 34:13)



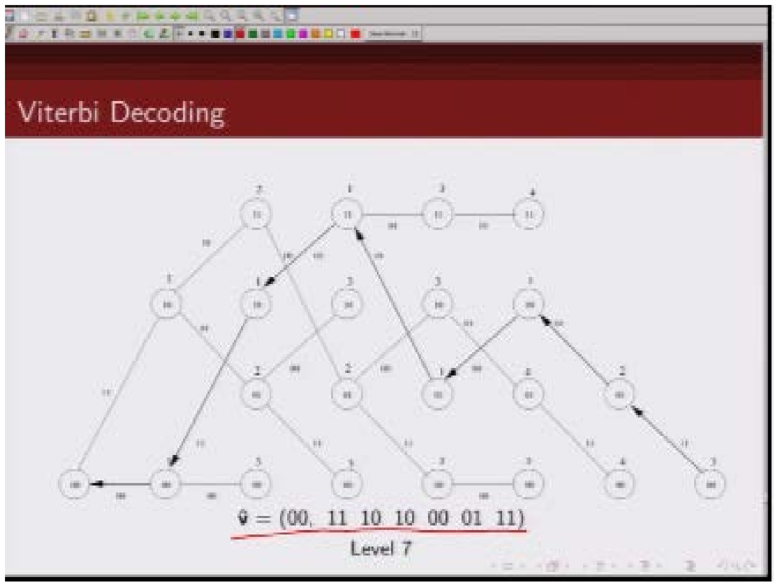
Viterbi Decoding

$\hat{v} = (00,\ 11\ 10\ 10\ 00\ 01\ 11)$

Level 7

Basically shown here so this is my back drag path and this is my estimated code sequence.

(Refer Slide Time 34:24)



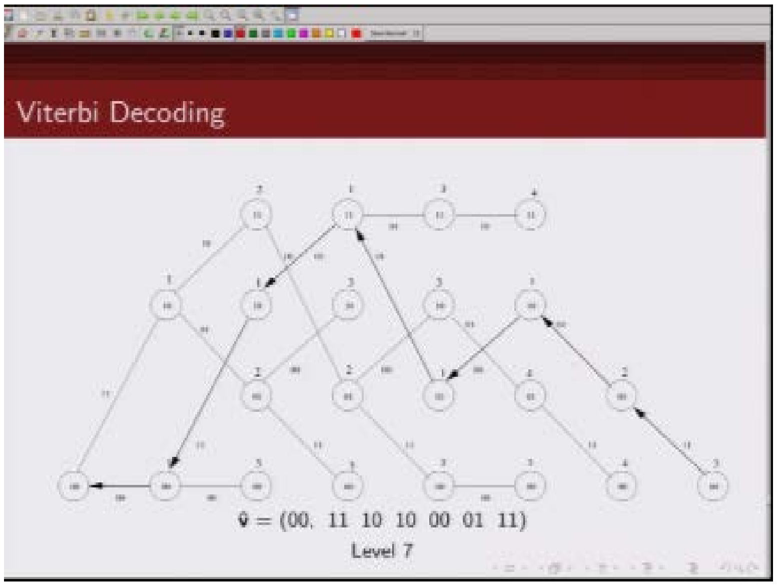
Viterbi Decoding

$\hat{\mathbf{v}} = (00,\ 11\ 10\ 10\ 00\ 01\ 11)$

Level 7

So this is how basically

(Refer Slide Time 34:26)



Viterbi Decoding

$\hat{v} = (00,\ 11\ 10\ 10\ 00\ 01\ 11)$

Level 7

We decode a convolutional code using Viterbi algorithm, thank you.

Acknowledgement

Ministry of Human Resource & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapobrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

Sanjay Mishra

Shubham Rawat

Shikha Gupta

K. K. Mishra

Aradhana Singh

Sweta

Ashutosh Gairola

Dilip Katiyar

Sharwan

Hari Ram

Bhadra Rao

Puneet Kumar Bajpai

Lalty Dutta

Ajay Kanaujia

Shivendra Kumar Tiwari

an IIT Kanpur Production