Problems: Convolutional Codes

Prof. Sundeep Rangan

1. *Convolutional Encoder.* Consider a rate 1/2 convolutional encoder with polynomials:

$$c_1[t] = b[t] + b[t-1] + b[t-3],$$
$$c_2[t] = b[t] + b[t-1] + b[t-2].$$

   (a) What is the constraint length K?

   (b) What are the generator polynomials, g_1 and g_2, in binary and octal?

   (c) Suppose we wish to encode $\boldsymbol{b} = [1, 0, 1, 1]$. How many tail bits do you add?

   (d) Write the output $c_1[t]$ and $c_2[t]$ for the input bits in part (c).

   (e) For this input, what is rate of the code including the tail bits?

Solution:

(a) $K = 4$ since the outputs depend on the last four samples.

(b) The polynomials in binary and octal are:

$$g_1 = [1, 1, 0, 1]_b = [1, 5]$$
$$g_2 = [1, 1, 1, 0]_b = [1, 6]$$

(c) You add $K - 1 = 3$ tail bits. So, the input would be:

$$\tilde{b} = [1, 0, 1, 1, 0, 0, 0].$$

(d) Running the convolution the two outputs are:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

(e) There are 7 outputs in each stream for a total of 14 output bits. Since there are 4 input bits, the rate is $R = 4/14 = 2/7$.

2. *FSM representation of convolutional encoders.* Consider a convolutional encoder

$$c_1[t] = b[t] + b[t-1],$$
$$c_2[t] = b[t] + b[t-2].$$

We will use the state $x[t] = (b[t-1], b[t-2])$.

$x[t]$	$x[t+1]$		$c[t] = (c_1[t], c_2[t])$	
	$b[t]=0$	$b[t]=1$	$b[t]=0$	$b[t]=1$
(0,0)				
(0,1)				
(1,0)				
(1,1)				

Table 1: Problem 2: State transition and output table to be completed.

$x[t]$	$x[t+1]$		$c[t] = (c_1[t], c_2[t])$	
	$b[t]=0$	$b[t]=1$	$b[t]=0$	$b[t]=1$
(0,0)	(0,0)	(1,0)	(0,0)	(1,1)
(0,1)	(0,0)	(1,0)	(1,1)	(1,0)
(1,0)	(0,1)	(1,1)	(1,0)	(0,1)
(1,1)	(0,1)	(1,1)	(1,1)	(0,0)

Table 2: Problem 2 Solution: State transition and output table

(a) Complete Table 1 to indicate the next state $x[t+1]$ and output $c[t]$ for each current state $x[t]$ and input $b[t]$.

(b) Given the table in part (a), draw a state diagram:

- Draw one node for each state.
- Draw arrows indicating the transitions. Use a different line type (e.g., solid and dashed) for transitions for $b[t] = 0$ and $b[t] = 1$.
- Draw the output bits $c[t]$ above each transition.

Solution:

(a) See Table 2.

(b) See Fig. 1.

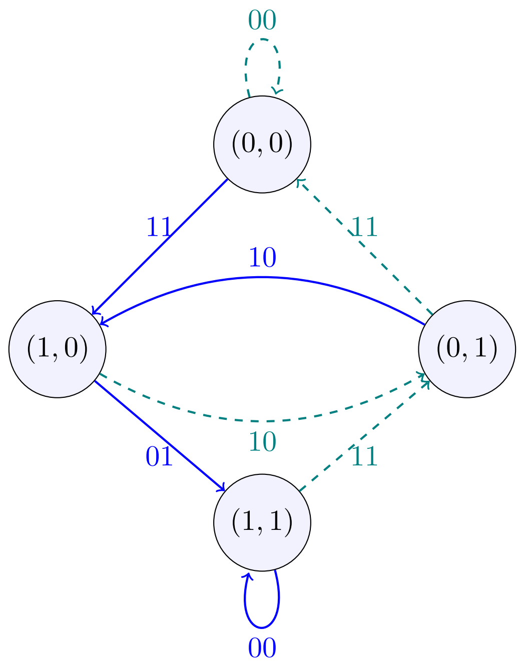

Figure 1: Problem 2 solution. Representation of the convolutional encoder as a state machine. The solid blue lines are the transitions for $b[t] = 1$, and the dashed teal lines are the transitions for $b[t] = 0$. The labels on the lines are the output bits $c[t] = (c_1[t], c_2[t])$

3. *Viterbi decoding.* Consider a encoder described by the FSM in Fig. 2. This FSM is not from a real convolutional encoder – it is completely made up to make the problem simple. At each time step, the FSM takes a binary input $b[t] \in \{0, 1\}$. There are three states, $x[t] \in \{0, 1, 2\}$. There are two outputs $c[t] = (c_1[t], c_2[t])$. The initial state is $x[0] = 0$.

   (a) Suppose that the information bits are

   $$\boldsymbol{b} = (b[0], b[1], b[2]) = (1, 0, 1).$$

   What is the state sequence $x[t]$ and output sequence $c[t]$?

   (b) Draw the trellis diagram for the states $x[t]$, $t = 0, 1, 2, 3$. On each branch of the trellis:

   - Use a different line type (e.g., solid and dashed) for transitions for $b[t] = 0$ and $b[t] = 1$.
   - Draw the output bits $c[t]$ above each transition.

   (c) Now suppose that the input bits $(b[0], b[1], b[2])$ are not known. To estimate the bits, we maximize the value function:

   $$J(c) = \sum_{i=1}^{6} c_i L_i,$$

   for the LLRs:

   $$L = (L_1, \ldots, L_6) = (-1.5, 1, 0.3, -2, 1.8, 0.5).$$

   On the trellis diagram from part (b), draw the branch metrics above each branch.

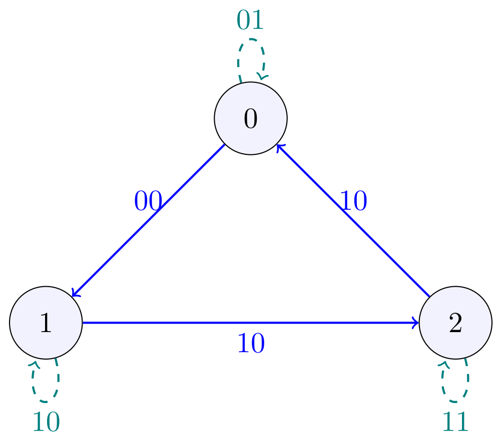

Figure 2: Problem 3. FSM representation of an encoder with three states $x[t] \in \{0, 1, 2\}$. The solid blue lines are the transitions for $b[t] = 1$, and the dashed teal lines are the transitions for $b[t] = 0$.

(d) Use the Viterbi algorithm to compute the partial value function at each node. Write the values in the node. Find the sequence $\boldsymbol{b} = (b[0], b[1], b[2])$ that results in the highest value.

Solution:

(a) Following the FSM in Fig. 2, we get

$$\boldsymbol{x} = (x[0], x[1], x[2], x[3]) = (0, 1, 1, 2).$$

The output sequence is:

$$\boldsymbol{c} = (c[0], c[1], c[2]) = (00, 10, 10).$$

(b) The trellis diagram for $t = 0, 1, 2, 3$ is shown in Fig. 3.

(c) The branch metrics are shown on the branches in Fig. 3.

(d) The value function is shown in each node in Fig. 3. We see that at the final node, the highest value function is at $x[3] = 2$. Tracing back from this node the optimal sequence is:

$$x[3] = 2 \Rightarrow x[2] = 2 \Rightarrow x[1] = 1 \Rightarrow x[0] = 0.$$

The sequence that results in the highest value is:

$$\boldsymbol{b} = (1, 1, 0).$$

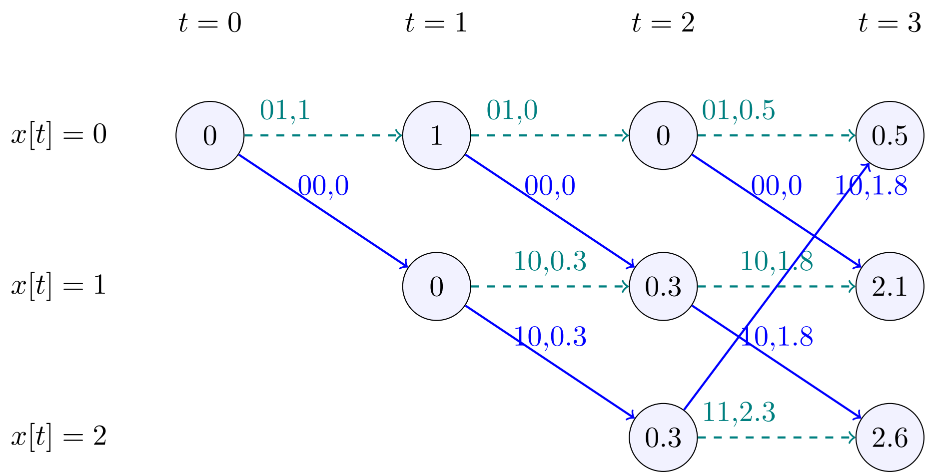

Figure 3: Problem 3 solution. Trellis diagram where the solid blue lines are the transitions for $b[t] = 1$, and the dashed teal lines are the transitions for $b[t] = 0$. The branches are labeled with (output bits, branch metric). The nodes are labeled with the partial value function from the Viterbi algorithm.