E2 205 Error-Control Coding
Lecture 22

Unnikrishnan N

October 28, 2019

1 GDL: The Conclusion

Explaining the term Belief Propagation

Assume only objective function $\beta_4(X_4)$ is of interest.

Here x_1, x_2, x_3, x_4 are constrained by parity check matrix H.

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

$$\begin{aligned} p(x_4|\mathbf{y}) &= \Pr(x_4|y_1 y_2) && (1) \\ &\propto \Pr(x_4 y_1 y_2) && (2) \\ &= \sum_{x_1,x_2} \Pr(x_1 x_2 x_4 y_1 y_2) && (3) \\ &\propto \sum_{x_1,x_2} \Pr(x_1 x_2 x_4) \Pr(y_1 y_2 | x_1 x_2 x_4) \chi_{124}(x_1 x_2 x_4) && (4) \\ &= \sum_{x_1,x_2} \chi_{124}(x_1 x_2 x_4) \Pr(y_1|x_1) \Pr(y_2|x_2) && (5) \end{aligned}$$

where χ is indicator function.

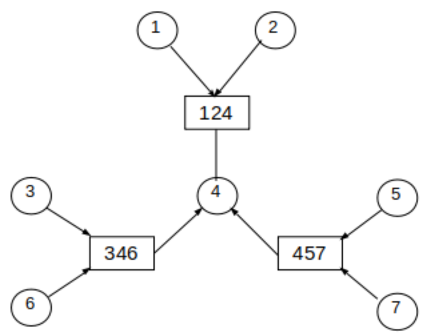
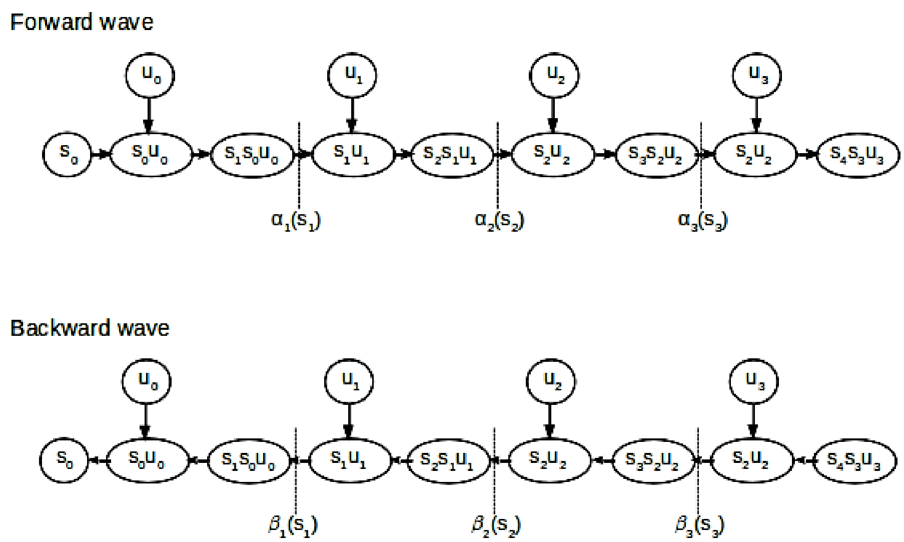


FIGURE 1: Junction tree representation of [7,4,2] code

1.1 Message Trellis

The message trellis can be a useful tool to determine the message passing schedule in cases where it is desired to compute more than a single objective function. All nodes will receive everyone's message after three stages.

		stage1	stage2	stage3
1	1	✓	✓	✓
	2			✓
	4			✓
	124			✓
2	1			✓
	2	✓	✓	✓
	4			✓
	124			✓
4	1			✓
	2			✓
	4	✓	✓	✓
	124			✓
124	1		✓	✓
	2		✓	✓
	4		✓	✓
	124	✓	✓	✓



Forward and Backward Schedules

$$\zeta_{ij}(X_{S_i \cap S_j}) = \sum_{X_{S_i \setminus S_j}} \alpha_i(X_{S_i}) \prod_{l \in N_i \quad l \neq j} \zeta_{li}(X_{S_i \cap S_j}) \tag{6}$$

$S_i \in \{00, 10, 01, 11\}$, N_i is nbd of i.

Forward Schedule

$$\alpha_3(s_3) = \sum_{s_2, u_2} \alpha_2(s_2) \Pr(s_3|s_2 u_2) \Pr(u_2) \Pr(y_2|s_2 u_2) \tag{7}$$

$$\propto \sum_{s_2} \alpha_2(s_2) \Pr(y_2|s_2 u_2) \tag{8}$$

$$\begin{bmatrix} \alpha_3(00) \\ \alpha_3(10) \\ \alpha_3(01) \\ \alpha_3(11) \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} 00 & 10 & 01 & 11 \end{array} \\ \begin{bmatrix} * & 0 & * & 0 \\ * & 0 & * & 0 \\ 0 & * & 0 & * \\ 0 & * & 0 & * \end{bmatrix} \end{array} \begin{bmatrix} \alpha_2(00) \\ \alpha_2(10) \\ \alpha_2(01) \\ \alpha_2(11) \end{bmatrix}$$

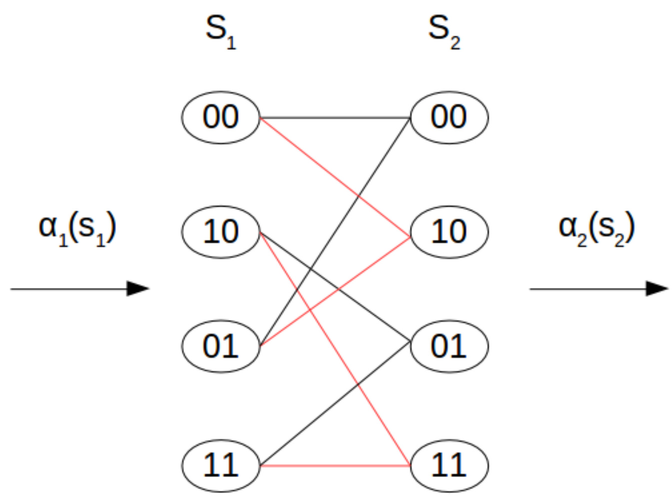


FIGURE 2: Trellis diagram of 1×2 convolutional code with $G(D) = [1 + D + D^2 \quad 1 + D^2]$

Backward Schedule

$$\beta_2(s_2) = \sum_{s_3, u_2} \beta_3(s_3) \Pr(s_3|s_2 u_2) \Pr(y_2|s_2 u_2) \Pr(u_2) \tag{9}$$

Once forward and backward phases are completed, objective functions of $u_0, u_1, u_2, \ldots$ can be evaluated (BCJR algorithm).
BCJR algorithm is in sum product form and viterbi algorithm is in max product form.

1.2 Complexity of GDL

Complexity associated with equation 6 is

$$q^{|S_i \cap S_j|}(q^{|S_i \setminus S_j|}(d_i - 1) + (q^{|S_i \setminus S_j|} - 1)) = q^{|S_i|}d_i - q^{|S_i \cap S_j|} \tag{10}$$

where $q^{|S_i \cap S_j|} q^{|S_i \setminus S_j|} = q^{|S_i|}$
Claim: Complexity of the single GDL solution to the MPF problem is given by

$$\sum_e \Psi(e) \tag{11}$$

where $\Psi(e) = q^{|S_i|} + q^{|S_j|} - q^{|S_i \cap S_j|}$ and
e is the edge in junction tree directed towards final destination point S_n.
It turns out that the complexity involved in computing the objective function at all nodes is bounded above by 4 times the complexity of single vertex complexity.

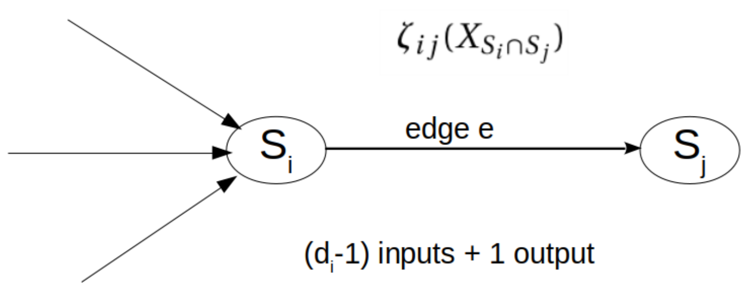


FIGURE 3: Node S_i with d_i degree

1.3 LDPC(Low Density Parity Check) Codes

Gallagar's thesis (1961)
Rediscovered on 1996 by M Sipser and DA Spielman and on 1999 by G McKay
Ref: The capacity of low-density parity-check codes under message-passing decoding, T.J. Richardson, R.L. Urbanke, IEEE Transactions on Information Theory (Volume: 47,Issue: 2, Feb 2001).

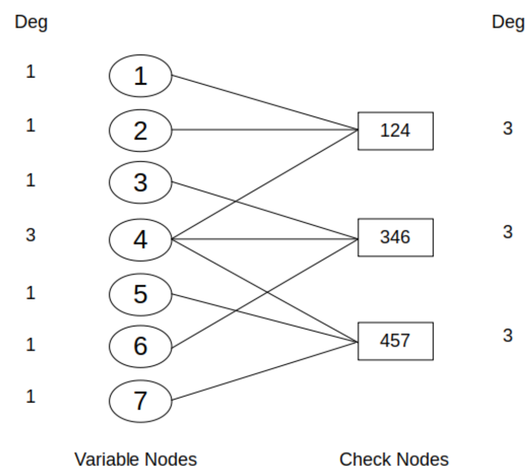


FIGURE 4: Bipartite graph for [7,4,2] code –Tanner graph of [7,4,2] code

Let $\mathscr{C}$ be a $[n,k]$ code with k,n large and rate $R=\frac{k}{n}$ with parity check matrix of size $n-k\times n$. The number of entries in this p.c matrix is $n(n-k)=n^2(1-R)=O(n^2)$. Thus a random parity check matrix would have $O(n^2)$ non zero entries. In the case of LDPC codes however the number of entries $=O(n)$, hence the name low density parity check codes.