

Decoding of LDPC codes using the Sum-Product algorithm for the AWGN channel with BPSK modulation

Telecommunications Laboratory

by Alex Balatsoukas-Stimming

Technical University of Crete

January 22nd, 2009

1 Example definition

2 Decoding

- Initial Step
- Horizontal Step
- Vertical Step
- Termination

3 Simulation Results

Example definition

- We will use the AWGN channel model, meaning that the received symbol is:

$$y = x + n, \quad n \sim \mathcal{N}(0, N_o)$$

- We will use a BPSK modulation with the mapping $\{0, 1\} \rightarrow \{-1, 1\}$ for the transmission of the coded bits.
- For simplicity's sake, our code's parity-check matrix will be in systematic form.
- We will focus on the decoding of one of the codeword's bits using the SPA. All other bits are decoded accordingly.

Example definition

- Consider the following parity-check matrix:

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = [\mathbf{P} \mid \mathbf{I}]$$

Note that the above parity-check matrix will result in a factor graph with no 4-cycles.

- Given the \mathbf{H} matrix, we can construct the code's generator matrix:

$$\mathbf{G} = [\mathbf{I} \mid \mathbf{P}^T] = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Example definition

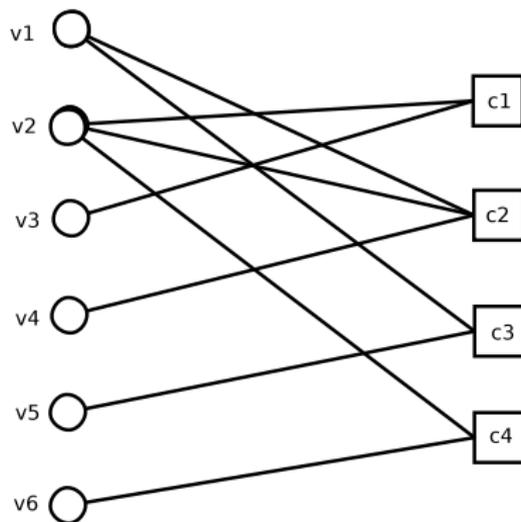


Figure: Factor graph corresponding to the above parity-check matrix.

- No 4-cycles are present, as already stated.

Example definition

- Suppose that we want to send the information bit vector $\mathbf{u} = [1 \ 0]$.
- The corresponding codeword is:

$$\mathbf{c} = \mathbf{uG} = [1 \ 0] \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} = [1 \ 0 \ 0 \ 1 \ 1 \ 0]$$

- Which in turn corresponds to the BPSK mapping:

$$\mathbf{c}_{\text{BPSK}} = \{ 1, -1, -1, 1, 1, -1 \}$$

Decoding - Initial Step

- Suppose that at the other end, the received codeword is:

$$\mathbf{y} = \{ 0.7, -1.2, 0.3, 1.1, -0.1, -0.8 \}$$

- The first step of the SPA is to compute the probabilities $P_i = p(c_i = 1|y_i)$ and $(1 - P_i) = p(c_i = 0|y_i)$ at the variable nodes and send them to the check nodes.
- For the AWGN case we know that $p(c_i = b|y_i) = (1 + e^{-\frac{2by_i}{N_0}})^{-1}$
- If we focus on the 5th codeword bit and assuming $N_0 = 0.1$, we will have that:

$$p(c_i = 1|y_i = -0.1) = (1 + e^{-\frac{2y_i}{N_0}})^{-1} = (1 + e^2)^{-1} = 0.12$$

$$p(c_i = 0|y_i = -0.1) = 1 - p(c_i = 1|y_i) = 0.88$$

- So, the message from variable node 5 to the check nodes that are connected to it (in this case, only check node 3) will be:

$$q_{53}(0, 1) = \{0.8800, 0.1200\}$$

- We will now focus on check node 3 who also gets a message from variable node 1 which is:

$$q_{13}(0, 1) = \{0.0009, 0.9991\}$$

Decoding - Initial Step

- The initial messages are computed as follows:

```
...
% Calculate  $P(x_i = b \mid y = y_i)$  for the first step
P1 = ones(size(rx))./(1 + exp(-2*rx./(NO/2)));
P0 = 1 - P1;

% Create table with initial messages from variable node i
% to check node j wherever connections are present
% (meaning  $H(j,i) = 1$ )
qij0 = H.*repmat(P0', M, 1);
qij1 = H.*repmat(P1', M, 1);
...
```

Decoding - Horizontal Step

- Check node 3 now has to calculate, for each variable node, the probability that the 3rd check equation is satisfied, given the messages that it received from the variable nodes connected to it except the one the message is headed to.
- We will need the following result. The probability that a vector of bits contains an even number of 1's is:

$$\frac{1}{2} + \frac{1}{2} \prod_i (1 - 2p_i)$$

where p_i is the probability that the bit at position i is equal to 1.

Decoding - Horizontal Step

- Obviously, p_i is equal to the messages $q_{i3}(1)$ received from each variable node.
- So, for the message r_{31} headed to variable node 1 we will have:

$$r_{31}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \neq 1} (1 - 2q_{i3}(1)) = \frac{1}{2} + \frac{1}{2}(1 - 2q_{53}(1)) = 0.88$$

which means that:

$$r_{31}(1) = 1 - r_{31}(0) = 0.12$$

- Accordingly, for the message r_{35} headed to variable node 5 we will have:

$$r_{35}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \neq 1} (1 - 2q_{i3}(1)) = \frac{1}{2} + \frac{1}{2} (1 - 2q_{13}(1)) = 0.0009$$

which means that:

$$r_{35}(1) = 1 - r_{35}(0) = 0.9991$$

Decoding - Horizontal Step

- We will also need to calculate the following messages in order to calculate the next message from variable node 1 to check node 3:

$$q_{12}(0, 1) = \{0.0009, 0.9991\}$$

$$q_{22}(0, 1) = \{0.9999, 0.0001\}$$

$$q_{42}(0, 1) = \{0.0001, 0.9999\}$$

- The above result in:

$$\begin{aligned} r_{21}(0) &= \frac{1}{2} + \frac{1}{2} \prod_{i' \neq 1} (1 - 2q_{i'2}(1)) \\ &= \frac{1}{2} + \frac{1}{2} (1 - 2q_{22}(1))(1 - 2q_{42}(1)) \\ &= 0.5 + 0.0001 = 0.5001 \end{aligned}$$

and

$$r_{21}(1) = 1 - r_{21}(0) = 0.4999$$

- The variable node now has enough information to calculate the probability that its corresponding bit of the codeword has a certain value:

$$Q_i(1) = K_i P_i \prod_j r_{ji}(1)$$

$$Q_i(0) = K_i (1 - P_i) \prod_j r_{ji}(0)$$

the constant K_i is chosen to ensure that $Q_i(0) + Q_i(1) = 1$.

- For the variable node 5 we have:

$$Q_5(1) = P_5 \cdot r_{35}(1) = 0.12 \cdot 0.9991 = 0.1199$$

$$Q_5(0) = (1 - P_5)r_{35}(0) = 0.82 \cdot 0.0009 = 0.0007$$

- For the constant K_5 we have:

$$K_5 = (Q_5(0) + Q_5(1))^{-1} = 8.29$$

which gives us:

$$Q_5(1) = 0.9940$$

$$Q_5(0) = 0.0060$$

- Since $Q_5(1) > Q_5(0)$, we decide that the 5th bit of the codeword is equal to 1 for this iteration.
- We see that the decision for this bit is now correct, in contrast with the decision we would have made if the variable node had not communicated with check node 3.

- The next step consists of calculating the messages from variable nodes to check nodes, given the channel observations and the incoming messages from the check nodes using the equation:

$$q_{ij}(0) = K_{ij}(1 - P_i) \prod_{j' \neq j} r_{j'i}$$

- The constant K_{ij} is chosen to ensure that $q_{ij}(0) + q_{ij}(1) = 1$.

- Using the above equation we get that:

$$q_{13}(0) = (1 - P_1)r_{21}(0) = 0.0009 \cdot 0.4999 = 0.0005$$

$$q_{13}(1) = P_1r_{21} = 0.9991 \cdot 0.4999 = 0.4995$$

- So, $K_{13} = (q_{13}(0) + q_{13}(1))^{-1} = 2$ which gives us:

$$q_{13}(0) = 0.0010$$

$$q_{13}(1) = 0.9990$$

- And so on.

- The above procedure is continued until the decided bits form a valid codeword, or until a maximum number of iterations is reached, whichever occurs first.

```
for k = 1:maxiter
    ...
    % If current decision is a codeword, stop decoding
    if( mod(H*vHat', 2) == zeros(M,1) )
        fprintf('Codeword retrieved!\n')
        return
    end
end
```

Simulation Results

