EECE 580B Modern Coding Theory | Homework assignment #7
WWW: http://dde.binghamton.edu/filler/mct/hw/7 | Due date: November 19, 2:50pm

Bitwise ML decoder over tree graphs

In this assignment, you will learn how the Bitwise ML decoder works on codes whose Tanner graph does not contain cycles. This is very important step towards the decoder used for decoding LDPC codes over arbitrary memoryless channel (BSC, Gaussian, …). If the Tanner graph of the code does not contain any cycle, we call the graph a tree. The reason for having the "tree assumption" is that, in this case, we can do the calculations for the decoder efficiently by evaluating the expression in the correct order.

In practice, codes usually have loops in their Tanner graphs. One of the advantages of LDPC codes of large length is that these loops are usually long. This means that when we use only a small neighborhood of the variable node for its decoding, this small neighborhood will not contain any cycle and we can use our approach. In practice, we use the Belief-propagation algorithm even when the graph contains cycles.

Let $y = (y_1, \dots, y_n)$ be a received vector and let $x = (x_1, \dots, x_n) \in C$ be the codeword we sent. In this assignment, assume we use the BSC(0.1) for all the transmission, thus $P(y_i = 0|x_i = 0) = P(y_i = 1|x_i = 1) = 0.9$ and $P(y_i = 1|x_i = 0) = P(y_i = 0|x_i = 1) = 0.1$. Because the BSC channel is memoryless, $P(y|x) = \prod_{i=1}^{n} P(y_i|x_i)$.

The central problem here is to evaluate the likelihood of individual bits, (see slides from the last lecture)

$$\hat{x}_i = \arg\max_{x_i \in \{0,1\}} P(x_i|y) = \arg\max_{x_i \in \{0,1\}} \sum_{x_j, j \neq i} P(y|x)\, \mathbb{I}_{\{x \in C\}}, \qquad (1)$$

where $\mathbb{I}_{\{x \in C\}}$ is the code membership function, $\mathbb{I}_{\{x \in C\}} = 1$ if $x \in C$ and $\mathbb{I}_{\{x \in C\}} = 0$ otherwise.

Problem 1: (Tanner graph without cycle)
Let C be a linear code represented by the following parity-check matrix

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

1. Draw a Tanner graph of this code.
2. Write how $\mathbb{I}_{\{x \in C\}}$ can be factorized into a product representing individual parity-check equations. See the slides from the last lecture for examples and notation.
3. Write down a specific version of equation (1) for $\hat{x}_1$ of this problem. Simplify this equation by breaking the big sum into smaller sums by using a distributive law ($ab + ac = a(b + c)$) as much as possible.
4. Write down a specific version of equation (1) for $\hat{x}_3$ of this problem. Simplify this equation by breaking the big sum into smaller sums by using a distributive law ($ab + ac = a(b + c)$) as much as possible.
5. Assume $y = (1\ 1\ 0\ 1\ 1\ 0\ 0\ 0)$ was received. Evaluate the expressions from Part 3 and 4 of this problem. While evaluating this expression, interpret the partial results as messages (pair of numbers) in the Tanner graph. Draw these messages and their values into the Tanner graph.

Show all the work. I need to see all the calculations and numbers.

Problem 2: (Tanner graph WITH cycle)

Let C be a linear code represented by the following parity-check matrix

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

1. Draw a Tanner graph of this code.
2. Write how $\mathbb{I}_{\{x \in C\}}$ can be factorized into a product representing individual parity-check equations. See the slides from the last lecture for examples and notation.
3. Write down a specific version of equation (1) for $\hat{x}_3$ of this problem. Simplify this equation by breaking the big sum into smaller sums by using a distributive law ($ab + ac = a(b + c)$) as much as possible.
4. Assume $y = (1\ 1\ 0\ 1\ 1\ 0\ 0\ 0)$ was received. Evaluate the expression from Part 3 of this problem. While evaluating this expression, are you still able to interpret all the partial results as messages (pair of numbers) in the Tanner graph?

Show all the work. I need to see all the calculations and numbers.

Problem 3: (Belief-propagation & Problem 1)

Let C be the linear code used in Problem 1. Use two iterations of the Belief-propagation algorithm derived on the lecture to calculate all the values of $\hat{x}_i$. Start with the initialization step, use both update rules two times and finish with the final step. Do you obtain the same results as in Problem 1?
Show all the work.