Factor Graphs and Belief Propagation, Examples

Erwin Nindl, Daniel Walch
{nindl,d.walch}@student.tugraz.at

8.11.2010

1 Sum-Product Message Update Rules

1.1 Branching Point

$$X + Y - Z = 0 \tag{1}$$
$$f_{=}(x, y, z) \triangleq \delta(z - x) \cdot \delta(z - y) \tag{2}$$
$$\mu_Z(z) = \int_{x'} \int_{x''} \delta(z - x') \cdot \delta(z - x'') \cdot \mu_X(x')\mu_Y(x'')dx'dx'' \tag{3}$$
$$= \mu_X(z)\mu_Y(z) \tag{4}$$

For a binary code ($X, Y, Z \in \mathbb{F}_2$) this yields to following solution:

$$\mu_Z(z) = \sum_{x',x'' \in \{0,1\}} \delta(z - x') \cdot \delta(z - x'') \cdot \mu_X(x')\mu_Y(x'')dx'dx'' \tag{5}$$

$$\begin{aligned} \mu_Z(z) =& \delta(0) \cdot \mu_X(0) \cdot \mu_Y(0) && x' = 0, x'' = 0, z = 0 \\ &+ \delta(1) \cdot \mu_X(1) \cdot \mu_Y(1) && x' = 1, x'' = 1, z = 1 \end{aligned} \tag{6}$$

$$\begin{bmatrix} \mu_Z(0) \\ \mu_Z(1) \end{bmatrix} = \begin{bmatrix} \mu_X(0) \cdot \mu_Y(0) \\ \mu_X(1) \cdot \mu_Y(1) \end{bmatrix} \tag{7}$$

1.2 Even Parity Check

$X, Y, Z \in \mathbb{F}_2$:

$$f_{\oplus}(x, y, z) \triangleq \delta\left(x \oplus y \oplus z\right) \tag{8}$$
$$\mu_Z(z) = \sum_{x',x'' \in \{0,1\}} \delta\left(x' \oplus x'' \oplus z\right) \cdot \mu_X(x')\mu_Y(x'')dx'dx'' \tag{9}$$

$$
\begin{aligned}
\mu_Z(z) =&\delta(0)\cdot\mu_X(0)\cdot\mu_Y(0) && x'=0,x''=0,z=0 \\
&+\delta(1)\cdot\mu_X(1)\cdot\mu_Y(0) && x'=1,x''=0,z=1 \\
&+\delta(1)\cdot\mu_X(0)\cdot\mu_Y(1) && x'=0,x''=1,z=1 \\
&+\delta(0)\cdot\mu_X(1)\cdot\mu_Y(1) && x'=1,x''=1,z=0
\end{aligned} \tag{10}
$$

$$
\begin{bmatrix} \mu_Z(0) \\ \mu_Z(1) \end{bmatrix} = \begin{bmatrix} \mu_X(0)\cdot\mu_Y(0)+\mu_X(1)\cdot\mu_Y(1) \\ \mu_X(1)\cdot\mu_Y(0)+\mu_X(0)\cdot\mu_Y(1) \end{bmatrix} \tag{11}
$$

2 Iterative Decoding of a linear Code

Consider a simple binary code

$$
C = \{c_1, c_2, c_3, c_4\} = \{(0,0,0,0),(0,1,1,1),(1,0,1,1),(1,1,0,0)\}. \tag{12}
$$

Assume that a codeword (X_1, X_2, X_3, X_4) is transmitted over a binary symmetric channel with a crossover probability of $\epsilon = 0.1$ and assume that $(Y_1, Y_2, Y_3, Y_4) = (0, 0, 1, 0)$ is received. In the following example the *a posteriori* probability $p(x_l|y_1, \ldots, y_4)$ for $l = 1, \ldots, 4$ is calculated.

First it is necessary to check if the given code is complete. Therefore a linear combination of codewords should result in an other existing codeword. As shown in equation 13 this holds for the given code C.

$$
\begin{aligned}
c_2 \oplus c_3 &= c_4 \\
c_3 \oplus c_4 &= c_2 \\
c_2 \oplus c_4 &= c_3
\end{aligned} \tag{13}
$$

Since the combination of two codewords creates an existing codeword it is sufficient to use two codewords for the generator matrix G, i.e:

$$
G = \begin{bmatrix} c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \tag{14}
$$

because two codewords span the whole vecorspace of C. To construct the parity check matrix H, the generator matrix G has to be transformed into its *systematic* form:

$$
G = [I_k|P] = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \tag{15}
$$

where I_k is the identity matrix. The generator matrix generates the corresponding codeword c_i for a particular message m_i:

$$C = M \cdot G \tag{16}$$

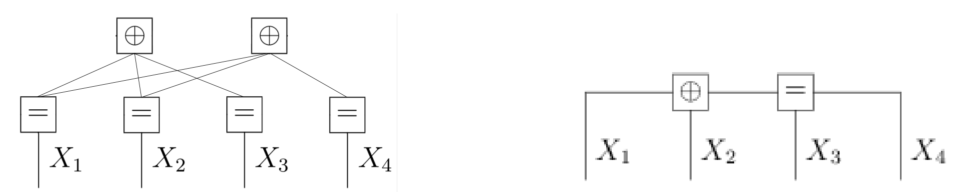
$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \cdot G$$

$$\begin{bmatrix} (0,0,0,0) \\ (0,1,1,1) \\ (1,0,1,1) \\ (1,1,0,0) \end{bmatrix} = \begin{bmatrix} (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

The parity check matrix H can be written as:

$$H = \left[-P^{\mathrm{T}} | I_{n-k}\right] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \tag{17}$$

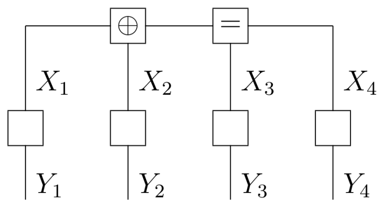
where $-P = P$ for binary code. The factor graph (FG) of the code is shown in figure 1(a). This FG can be transformed into a graph without cycles as shown in figure 1(b). A FG without cycles is needed to be able to calculate the belief propagation without iteration.



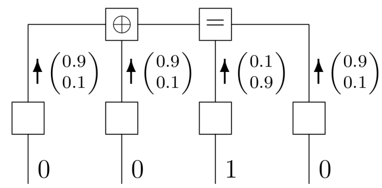
(a) with cycles (b) without cycles

Figure 1: Factor graph of the code

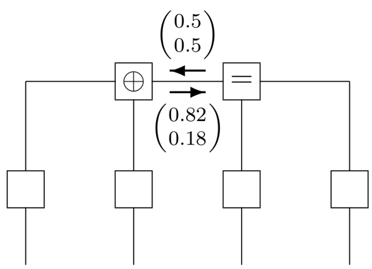
Next the belief propagation algorithm is applied to the joint code/channel FG shown in figure 2(a). The messages μ are represented as $\begin{pmatrix} \mu(0) \\ \mu(1) \end{pmatrix}$, scaled in order to get $\mu(0) + \mu(1) = 1$. The message propagation is shown in figure 2(b) - 2(d).



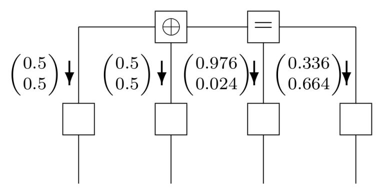
(a) joint code/channel model



(b) message propagation



(c) message propagation



(d) message propagation

Figure 2: Belief propagation

The *a posteriori* probability $p(x_l|y_1, \ldots, y_4)$ for $l = 1, \ldots, 4$ is the product of the incomming and outgoing messages. The result is shown in figure 3.

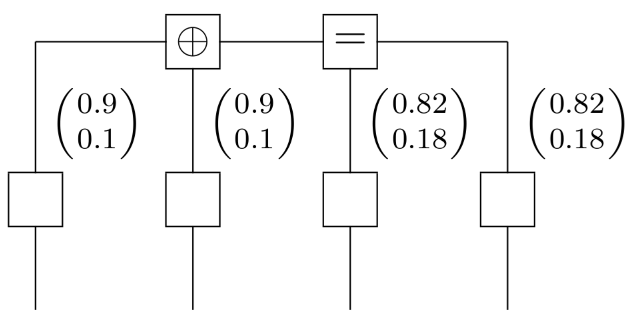


Figure 3: A posteriori probabilities $p(x_l|y_1, \ldots, y_4)$ for $l = 1, \ldots, 4$.

With applying a threshold of 0.5 on the probabilities, the decoded message is $(X_1, X_2, X_3, X_4) = (0, 0, 0, 0)$.