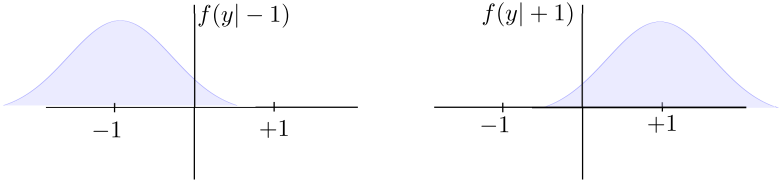
BP decoding of LDPC codes

Introduction to Graphical Models and Inference for Communications

UC3M

March 3, 2018

Binary-input discrete AWGN (BIAWGN) channel



Assuming a uniform prior distribution $P(S=1) = P(S=-1) = \frac{1}{2}$...

$$P(S=1|y) = \frac{f(y|1)}{f(y|1) + f(y|-1))}$$

Optimal decoder for the BIAWGN channel

- LDPC codeword of `n` bits.

$$\boldsymbol{s} = \begin{pmatrix} s_1 & s_2 & \dots & s_{\mathtt{n}} \end{pmatrix} \qquad \text{Vector of } \mathtt{n} \text{ encoded BPSK symbols.}$$
$$\boldsymbol{y} = \begin{pmatrix} y_1 & y_2 & \dots & y_{\mathtt{n}} \end{pmatrix} \qquad \text{Observation vector.}$$

- **Optimal LDPC decoder: bitwise-MAP decoder.** For $i = 1 \dots, n$,
  1. Compute
  $$P(S_i = +1|\boldsymbol{y})$$
  2. If $P(S_i = +1|\boldsymbol{y}) > 0.5$, then $\hat{S}_i = 1$.

Baye's rule ...

$$P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{s})P(\boldsymbol{S}=\boldsymbol{s})}{f(\boldsymbol{y})}, \quad P(\boldsymbol{S}=\boldsymbol{s}) = \frac{1}{2^{\mathtt{rn}}} \quad \forall \boldsymbol{s} \in \mathcal{C}, \qquad f(\boldsymbol{y}|\boldsymbol{s}) = \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)$$

$$P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{s})P(\boldsymbol{S}=\boldsymbol{s})}{f(\boldsymbol{y})}, \quad P(\boldsymbol{S}=\boldsymbol{s}) = \frac{1}{2^{\mathtt{rn}}} \quad \forall \boldsymbol{s} \in \mathcal{C}, \qquad f(\boldsymbol{y}|\boldsymbol{s}) = \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)$$

Therefore ....

$$\begin{aligned}
P(S_i = 1|\boldsymbol{y}) &= \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=1} P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{y}) \\
&= \frac{1}{f(\boldsymbol{y})}\frac{1}{2^{\mathtt{rn}}} \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j) \\
&\propto \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)
\end{aligned}$$

Optimal decoder for the BIAWGN channel

$$P(S_i = 1|\boldsymbol{y}) \propto \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j) \qquad P(S_i = -1|\boldsymbol{y}) \propto \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=-1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)$$

- The number of terms to be sum is roughly half of the codewords $2^{\mathtt{rn}-1}$.
- The decoding complexity grows exponentially fast with $\mathtt{n}$.
- Prohibitive complexity for $\mathtt{n} \geq 100$!!

Approximate decoder for the BIAWGN channel

- Belief Propagation (BP) algorithm to approximate marginals!
- Message-Passing description.
- Complexity grows linearly with the code length `n`!
- At each iteration, variable nodes in the Tanner graph send a "belief" (estimated probability) to check nodes and check nodes recompute a belief for each variable node.
- We proceed in this way for a given number of iterations $\ell_{\max}$.
- The associated factor graph is sparse, only contains very large cycles involving a large number of variables.
- BP estimates, based on local computations, are reasonably accurate.

Towards a BP formulation

$$P(S_i = 1|\boldsymbol{y}) \propto \sum_{\boldsymbol{s}\in\mathcal{C}:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)$$

$$= \sum_{\boldsymbol{s}\in\{0,1\}^n:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j)\mathbb{1}[\boldsymbol{Hx} = \mathbf{0} \pmod 2]$$

- $\mathbb{1}[\boldsymbol{Hx} = \mathbf{0} \pmod 2]$ can be decomposed as follows

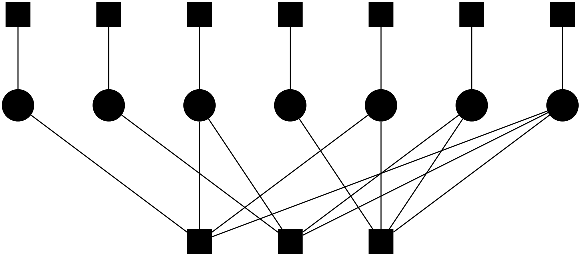
$$\prod_{q=1}^{\mathtt{k}} \mathbb{1}[\boldsymbol{h}_q\boldsymbol{x} = 0]$$

where $\boldsymbol{h}_q$ is the q-th row of $\boldsymbol{H}$ and $\mathtt{k}$ is the number of rows.

$$P(S_i = 1|\boldsymbol{y}) \propto \sum_{\boldsymbol{s}\in\{0,1\}^n:s_i=1} \prod_{j=1}^{\mathtt{n}} f(y_j|s_j) \prod_{q=1}^{\mathtt{k}} \mathbb{1}[\boldsymbol{h}_q\boldsymbol{x} = 0]$$

The LDPC Tanner graph

$$P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{y}) \propto \prod_{j=1}^{\mathtt{n}} f(y_j|s_j) \prod_{q=1}^{\mathtt{k}} \mathbb{1}[\boldsymbol{h}_q \boldsymbol{x} = 0]$$



From here you already know how to implement a BP decoder.

Excersise

Show that the BP message passing rules for binary LDPC codes can be simplified to the following experessions.

Update rule at variable nodes

Variable with $K+1$ neighbours sends a message to neighbour K+1:

$$l_{K+1} = \sum_{k=1}^{K} l_k,$$

where l_j, $j = 1, \ldots, K+1$ are real scalars.

Update rule at parity check nodes

Factor with $J+1$ neighbours sends a message to neighbour J+1:

$$l_{J+1} = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$$

where l_j, $j = 1, \ldots, J+1$ are real scalars.