Tensor Train and Taylor methods for LDPC decoding

Daniil Polykovskiy, Alexander Novikov

Apr. 15, 2016

Channel model

- Transmits only 0 and 1
- Flips with low probability p
- Bit stream is split into messages of length n

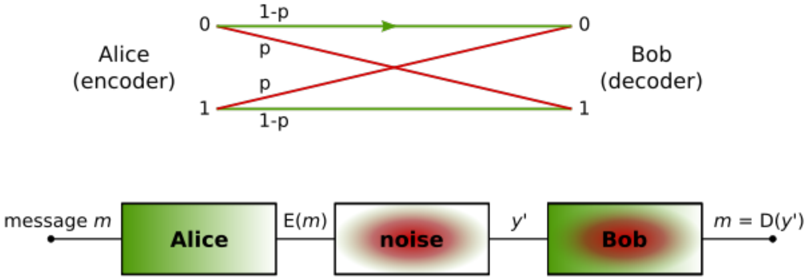
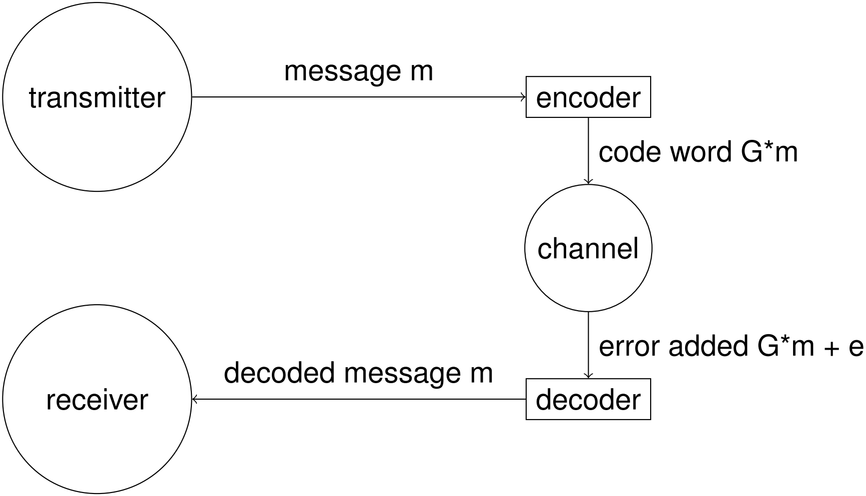


Figure: Binary symmetric channel

Communication system

- Redundancy is added at the encoder
- Error correction/detection at the receiver



LDPC codes

- Set of code words (C) is a subspace of $\{0,1\}^N$
- Linear codes: $\forall c_1, c_2 \in C: \quad c_1 + c_2 \in C$
- Parity check matrix $H \in \{0,1\}^{K \times N}$: $\quad H * c = 0 \quad \forall c \in C$
- Generator matrix $G \in \{0,1\}^{N \times n}$: $\quad G * m \in C \quad \forall m \in \{0,1\}^n$
- Orthogonality property: $H * G = 0$
- LDPC codes: H is sparse

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

Figure: An example of code-defining matrices

Examples

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

$$H * \hat{m} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Decoding algorithms

- Optimal decoder
  - Find the closest code word
- Heuristics
  - Bit-flipping
  - Simulated annealing
- Linear programming
- Maximum a posteriori decoder
  $m = \text{argmax}_{m \in C} P(m|\text{received\_message})$
  - Loopy BP
  - ? Tensor Train
  - ? Taylor

Tensor train

Tensor train and a bunch of formulas

- $m_i = \text{argmax}_{m_i \in \{0,1\}} P(m_i|c)$
- $m_i = \text{argmax}_{t \in \{0,1\}} \sum_{m:m_i=t} P(m|c)$
- $m_i = \text{argmax}_{t \in \{0,1\}} \sum_{m:m_i=t} P(c|m)P(m)$

Tensor train and a bunch of formulas

- $m_i = \text{argmax}_{m_i \in \{0,1\}} P(m_i|c)$
- $m_i = \text{argmax}_{t \in \{0,1\}} \sum_{m: m_i = t} P(m|c)$
- $m_i = \text{argmax}_{t \in \{0,1\}} \sum_{m: m_i = t} P(c|m)P(m)$
- $P(c|m) = \prod_{j=0}^{N} p^{m_j = c_j}(1-p)^{m_j \neq c_j}$
- $P(m) \propto 1\{H * m = 0\}(m)$
- $1\{H * m = 0\}[m_1, m_2, \ldots, m_N] = \prod_{i=1}^{K} \psi_i[m_1, m_2, \ldots, m_N]$
- $\psi_i[m_1, m_2, \ldots, m_N]$ — is i^{th} constraint satisfied

TT cores

Invariant: after multiplying first k cores, an obtained value should be a vector $[0, 1]$ if partial parity sum is 0 and $[1, 0]$ otherwise

$$\psi_i[m_1, m_2, \ldots, m_N] = [0, 1] * D^i_1[m_1] * D^i_2[m_2] * \cdots * D^i_N[m_N] * [0, 1]^T$$

$$D^i_j[1] = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H[i,j] = 0, \text{ no dependence on this bit} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, H[i,j] = 1, \text{ parity should flip} \end{cases}$$

$$D^i_j[0] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ because 0 does not change parity}$$

Kronecker product

Kronecker product of an $M \times N$ matrix A and a $P \times Q$ matrix B is the following $MP \times NQ$ matrix C:

$$A \otimes B = C = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

The Kronecker product is bilinear and associative. Also $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$.

TT meets Leopold Kronecker

$$
\begin{aligned}
1\{H * m = 0\}[m_1, m_2, \ldots, m_N] &= \prod_{i=1}^{K} \psi_i[m_1, \ldots, m_N] \\
&= \otimes_{i=1}^{K} \psi_i[m_1, \ldots, m_N] \\
&= \otimes_{i=1}^{K} (\prod_{j=1}^{N} D_j^i[m_j]) \\
&= \prod_{j=1}^{N} (\otimes_{i=1}^{K} D_j^i[m_j])
\end{aligned}
$$

$$P(c|m) 1\{H * m = 0\}(m) = \prod_{j=1}^{N} (\otimes_{i=1}^{K} D_j^i[m_j]) \cdot \prod_{i=1}^{N} P(c_i|m_i) = \prod_{j=1}^{N} A_j[m_j]$$

$$A_j[m_j] = (\otimes_{i=1}^{K} D_j^i[m_j]) \cdot P(m_j|c_j) \in \{0, 1\}^{2^K \times 2^K}$$

TT formulas. The final

- $m_i = \text{argmax}_{t \in \{0,1\}} \sum_{m: m_i = t} P(c|m) 1\{H * m = 0\}(m)$

$$m_i = \text{argmax}_{m_i \in \{0,1\}} \sum_{m_1, \ldots, m_{i-1}, m_{i+1}, \ldots, m_N} \prod_{j=1}^{N} A_j[m_j]$$

$$m_i = \text{argmax}_{m_i \in \{0,1\}} \prod_{j=1}^{i-1} (A_j[0] + A_j[1]) \cdot A_i[m_i] \cdot \prod_{j=i+1}^{N} (A_j[0] + A_j[1])$$

- Values of $A_1[m_1] * \ldots A_j[m_j]$ are stored in TT format with some upper bound on the maximum rank
- Final complexity: $O(N^2 r_{\text{max}}^4)$

TT experiments

Errors before decoding:	0	1	2	3	4	5	6
$r_{\mathsf{max}} = 2$	0	11	19	9	17	13	17
$r_{\mathsf{max}} = 5$	0	4	10	15	11	17	15
$r_{\mathsf{max}} = 10$	0	0	10	21	16	13	17
$r_{\mathsf{max}} = 100$	0	0	0	0	0	8	11
$r_{\mathsf{max}} = 500$	0	0	0	0	0	2	7
$r_{\mathsf{max}} = 1000$	0	0	0	0	0	0	5

Code: $n = 9$, $N = 30$

Taylor

Taylor basics

- Energy $E(m) = -\log P(c|m) + \text{const} = \begin{cases} -\sum_{i=1}^{N}[(m_i \neq c_i)\log p + (m_i = c_i)\log(1-p)], \text{ if H*m=0} \\ +\infty, \text{ otherwise} \end{cases}$.
- Interpret E as a variable ξ (m is random)
- $\xi_{i,t}$ in a similar way, but for E defined only on code words with $m_i = t$, $t \in \{0, 1\}$.

Chacacteristic function computation

1. $m_i = \text{argmax}_{t\in\{0,1\}} \sum_{m:m_i=t,H*m=0} \exp\{-E(m)\} = \text{argmax}_{t\in\{0,1\}} \mathbb{E}\exp\{-E(m)\} = \text{argmax}_{t\in\{0,1\}} \phi_{\xi_{i,t}}(j)$
   - $\phi_\xi(j) = \mathbb{E}exp(j^2\xi) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!}\mathbb{E}[\xi^k]$
   - Approximate by first T moments
2. $m_i = \log[\text{argmax}_{t\in\{0,1\}} \sum_{m:m_i=t,H*m=0} \exp\{-E(m)\}] = \text{argmax}_{t\in\{0,1\}} \log\phi_{\xi_{i,t}}(j)$
   - $\log\phi_\xi(j) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!}\kappa_k$
   - Approximate by first T cumulants (analyticaly computed from moments)

Complexity of both variants: $O(N^T T)$

Taylor experiments (1)

Approximation by T moments

Errors before decoding:	0	1	2	3	4	5	6
T=1	0	1	2	3	4	5	6
T=2	0	1	2	3	4	5	6
T=3	0	1	2	3	4	5	6
T=4	0	1	2	3	4	5	6
T=5	0	0	0	0	0	0	0
T=6	0	0	0	0	0	0	0

Code: $n = 9$, $N = 30$

Taylor experiments (2)

Approximation by T cumulants

Errors before decoding:	0	1	2	3	4	5	6
T=1	0	1	2	3	4	5	6
T=2	0	1	2	3	4	5	6
T=3	0	1	2	3	4	5	6
T=4	0	0	0	0	3	3	2
T=5	0	0	0	0	0	0	0
T=6	0	0	0	0	0	0	0

Code: $n = 9$, $N = 30$

Conclusion

- Asymptotic convergence to MAP decoder
- Taylor is parallelizable
- Possible use of Taylor in pipeline of decoders
- No complexity dependence on sparsity