Lecture 4: Channel Coding 1
Advanced Digital Communications (EQ2410)[1]

Ming Xiao
CommTh/EES/KTH

Wednesday, Jan. 28, 2014
10:00-12:00, V01

[1]Textbook: U. Madhow, *Fundamentals of Digital Communications*, 2008

Notes

KTH VETENSKAP OCH KONST
Lecture 4
Channel Coding 1
Ming Xiao
CommTh/EES/KTH
Overview
Linear Block Codes
Tanner Graph
LDPC Codes
Irregular LDPC Codes
LDPC Decoding
Density Evolution
Code Design

Overview

Lecture 1-3

- ISI channel and equalization

→ Signal processing methods to improve the received signal

Digital Communications

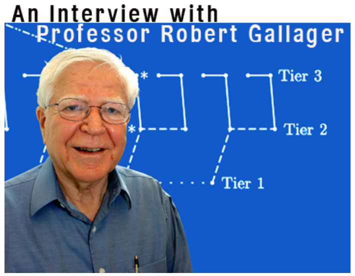
- Block codes
- Convolutional codes
- Random Coding (information theoretical concept)

Lecture 4: Channel Coding 1 (LDPC Codes)

Notes

Overview

- LDPC codes were invented by Robert G. Gallager in the 1960s and forgotten for three decades.



[source: http://lids.mit.edu/]

- After Turbo codes were invented 1993, LDPC codes found new attention.
- First channel codes, which provably allow to achieve the capacity limit of the binary erasure channel and to approach the capacity limit for other important channel models.

Notes

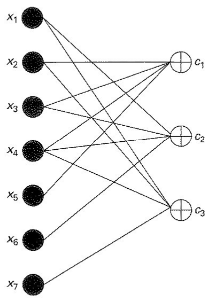
Linear Block Codes

- Information word $\mathbf{u} = [u_1, \ldots, u_k] \Rightarrow 2^k$ codewords $\mathbf{x} = [x_1, \ldots, x_n]$
- Code $\mathcal{C}$
  - Set of all codewords $\mathcal{C} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{2^k}\}$
  - Code rate $R = k/n$
  - A linear block code spans a k-dimensional subspace $\mathcal{C}$ in the n-dimensional binary space.
- Encoder
  - Mapping from the information word space into the codeword space
  - Linear encoding with generator matrix $\mathbf{G}$: $\mathbf{x} = \mathbf{uG}$, dimension $k \times n$
  - → The rows $\mathbf{v}_i$ of $\mathbf{G}$ are basis vectors of the subspace $\mathcal{C}$.
- Check matrix $\mathbf{H}$
  - Each codeword $\mathbf{x} \in \mathcal{C}$ satisfies $\mathbf{Hx}^T = \mathbf{HG}^T\mathbf{u} = \mathbf{0}$.
  - $\mathbf{H}$ spans the $(n-k)$-dimensional subspace $\mathcal{C}^\perp$ orthogonal to $\mathcal{C}$.
  - $\mathbf{H}$ is the generator matrix of the dual code $\mathcal{C}^\perp$ of the code $\mathcal{C}$.
  - Syndrome $\mathbf{c} = \mathbf{Hx}^T$; i.e., for all $\mathbf{x} \in \mathcal{C}$ we have $\mathbf{c} = \mathbf{0}$.
- Linearity
  - For $\mathbf{x}_0 = \mathbf{u}_0\mathbf{G}$ and $\mathbf{x}_1 = \mathbf{u}_1\mathbf{G}$ we can see that $\mathbf{x}_2 = \mathbf{x}_0 + \mathbf{x}_1 = \mathbf{u}_0\mathbf{G} + \mathbf{u}_1\mathbf{G} = (\mathbf{u}_0 + \mathbf{u}_1)\mathbf{G} \in \mathcal{C}$.
  - Convenient for performance evaluation: distance properties can be expressed by the weight distribution (e.g., $d_{min} = w_{min}$).

Notes

Tanner Graph

Bipartite graph representing the parity-check matrix.

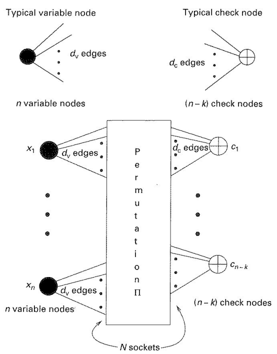


- Variable nodes (left) represent the code symbols x_i in $\mathbf{x}$.
- Check nodes (right) represent the symbols c_j of the syndrome $\mathbf{c}$.
- A variable node x_i is connected to a check node c_j by an edge in the graph if x_i is included in the check equation specifying c_j (i.e., if $H_{ji} = 1$).
- Degree of a node
  - Number of outgoing edges of a node
  - Variable node degree d_v
  - Check node degree d_c

Notes

LDPC Codes

Low-density parity-check (LDPC) codes



- Codes with a sparse parity-check matrix (i.e., only few elements $H_{ij} = 1$ in $\mathbf{H}$).
- Regular (d_v, d_c) LDPC code
  - Sparse $\mathbf{H}$ where each variable node has degree d_v and each check node has degree d_c.
- Code rate
  - Number of edges in the Tanner graph
  $$N = n \cdot d_v = (n-k) \cdot d_c$$
  - With $R = k/n$ we get
  $$R = 1 - \frac{d_v}{d_c}.$$

Code construction

- As suggested by the figure above, the problem of finding the $\mathbf{H}$ matrix can be interpreted as the problem of finding the edge permutation Π (edge interleaver).

Notes

Irregular LDPC Codes

- Variable-node and check-node degrees are not constant; the degrees are chosen according to a predefined degree distribution.
- Degree distribution for the variable-node degrees and check-node degrees
$$\lambda(x) = \sum_i \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_i \rho_i x^{i-1}$$
with coefficients
  - $\lambda_i = \Pr[\text{an edge is connected to a variable node with } d_v = i]$
  - $\rho_i = \Pr[\text{an edge is connected to a check node with } d_c = i]$

  Example, (3,6) LDPC code: $\lambda(x) = x^2$ and $\rho(x) = x^5$
- Code rate
  - Number of edges connected to degree-i variable nodes: $N\lambda_i$
  - Number of variable nodes with degree $d_v = i$: $N\lambda_i/i$

$$\Rightarrow \; n = N\sum_i \frac{\lambda_i}{i} = N\int_0^1 \lambda(x)dx \quad \text{and similarly} \quad (n-k) = N\sum_i \frac{\rho_i}{i} = N\int_0^1 \rho(x)dx$$

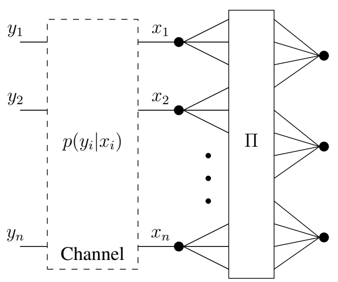
$$R = \frac{k}{n} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$$

- Fractions of degree-i variable nodes and degree-j check nodes
$$\tilde{\lambda}_i = \frac{\lambda_i/i}{\sum_l \lambda_l/l} \quad \text{and} \quad \tilde{\rho}_i = \frac{\rho_i/i}{\sum_l \rho_l/l}$$

Notes

LDPC Decoding

Iterative decoding on the Tanner graph



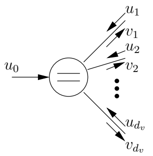
- Code symbols are transmitted over a channel characterized by $p(y_i|x_i)$ ($\rightarrow$ received symbols y_i).
- Nodes are replaced by local decoders.
  - $\rightarrow$ Variable node decoder (repetition code)
  - $\rightarrow$ Check node decoder (single-parity-check code)
- Decoders exchange "messages" along the edges (e.g., log-likelihood ratios or estimates of the bits).

Notes

LDPC Decoding
– Gallager's Algorithm A (suboptimal)

Assumption: BSC with error probability ϵ (i.e., $\Pr(x_i \neq y_i) = \epsilon$).

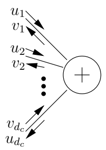
Variable-node decoder



- Message from the channel: $u_0 = y$
- Messages received by the variable node from the check nodes: u_j ("decoder input")
- Messages from the variable node to check nodes: v_i ("decoder output")

$$v_i = \begin{cases} \bar{u}_0 & u_1 = \ldots = u_{i-1} = u_{i+1} = \ldots = u_{d_v} = \bar{u}_0 \\ u_0 & \text{else} \end{cases} .$$

Check-node decoder



- Messages received from the variable nodes: v_i ("decoder input")
- Messages from the check node to the variable nodes: u_j ("decoder output")

$$u_j = \sum_{l=1, l\neq j}^{d_c} v_l \mod 2$$

Decoding is successful if all check equations after an iteration are fulfilled.

Notes

Lecture 4
Channel Coding 1

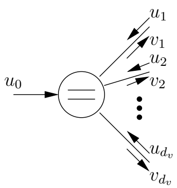
Ming Xiao
CommTh/EES/KTH

Overview
Linear Block Codes
Tanner Graph
LDPC Codes
Irregular LDPC Codes
LDPC Decoding
Algorithm A
Belief Propagation
Density Evolution
Code Design

LDPC Decoding
– Belief Propagation

- Variable-node decoder and check-node decoder are realized by the respective soft-input/soft-output decoders.
- Extrinsic log-likelihood ratios (LLRs) are exchanged.
- Suboptimal algorithm with close-to-optimal performance

Variable-node decoder



- Message from the channel:

$$u_0 = \log(p(y|x=0)/p(y|x=1))$$

  - BSC, $\Pr(y \neq x) = \epsilon$:

$$u_0 = (-1)^x \log((1-\epsilon)/\epsilon)$$

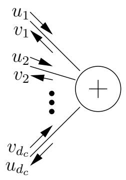
  - AWGN, $y = A(-1)^x + w$: $u_0 = 2A/\sigma^2 y$
- LLRs received by the variable node from the check nodes: u_q ("decoder input")
- LLRs from the variable node to check nodes: v_p ("decoder output")

$$v_p = u_0 + \sum_{q=1, q\neq p}^{d_v} u_q \quad \rightarrow \text{ extrinsic information}$$

Notes

LDPC Decoding
– Belief Propagation

Check-node decoder



- LLRs received from the variable nodes: v_q ("decoder input")
- LLRs u_p from the check node to the variable nodes ("decoder output") satisfy

$$\tanh\left(\frac{u_p}{2}\right) = \prod_{q=1,q\neq p}^{d_c} \tanh\left(\frac{v_q}{2}\right) \qquad (1)$$

or

$$u_p = 2 \cdot \tanh^{-1}\left(\prod_{q=1,q\neq p}^{d_c} \tanh\left(\frac{v_l}{2}\right)\right)$$

$\rightarrow$ extrinsic information!

Remark

- Given the LLR l for a bit b, the estimate of b given l is $E[b|l] = \tanh(b/2)$.
- Interpretation of Eq. (1): the expected value of the output LLR is given by the product of the expected values of the incoming LLRs.

Notes

Density Evolution
– General Idea

- Tool for analyzing iterative decoding and predicting the convergence of the iterative decoder.
- Track how the distribution of the messages u_i, v_j at the output of the component decoders evolve from iteration to iteration.
- Without loss of generality the analysis can be restricted to the case where the all-zero codeword is transmitted.
- To simplify the analysis, one typically parameterizes the densities by a single parameter (approximation, only optimal in special cases):
  - AWGN channel and message passing with LLRs: variance or mean of the LLRs (both are coupled; see problem 7.12(f) in the textbook).
  - BSC channel and binary messages (e.g., Algorithm A): error probability (optimal).
  - Binary erasure channel (messages are either the erasure symbol or the correct bit): erasure probability (optimal).
- EXIT charts (see Chapter 7.2.5): special case of density evolution where the densities are represented by their mutual information.

Notes

Density Evolution
– Algorithm A

- Binary messages are exchanged.
- Assuming that the all-zero codeword was transmitted, the error probabilities $p(l), q(l)$ at the decoder outputs during the l-th iteration are:

$$p(l) = \Pr[\text{message sent by variable node in iteration } l \text{ is } 1]$$
$$q(l) = \Pr[\text{message sent by check node in iteration } l \text{ is } 1]$$

- Analysis for the check-node decoder, l-th iteration
  - Input to the check-node decoder: binary messages with error probability $p(l)$
  - Output message at edge i is incorrect if the input to the check decoder on the remaining edges $j \neq i$ includes an odd number of errors.
  - Marginalizing over all error events yields

$$q(l) = \sum_{j=1, j \text{ odd}}^{d_c-1} \binom{d_c-1}{j} p(l)^j (1-p(l))^{d_c-1-j} = \frac{1-(1-2p(l))^{d_c-1}}{2}$$

Notes

Density Evolution
– Algorithm A

- Analysis for the variable node decoder, l-th iteration
  - Input to the variable-node decoder: binary messages with error probability $q(l)$
  - Output message at edge i is incorrect if
    1. Channel message u_0 is right and all incoming messages u_j at edges $j \neq i$ are wrong, or
    2. Channel message u_0 is wrong and not all incoming messages u_j at edges $j \neq i$ are right.
  - It follows that

$$p(l) = p(0)[1-(1-q(l))^{d_v-1}] + (1-p(0))q(l)^{d_v-1}$$

(with the error probability of the channel $p(0) = \epsilon$)
  - Combining the terms for $p(l)$ and $q(l)$ yields

$$p(l) = p(0) - p(0)\left(\frac{1+(1-2p(l-1))^{d_c-1}}{2}\right)^{d_v-1} + (1-p(0))\left(\frac{1-(1-2p(l-1))^{d_c-1}}{2}\right)^{d_v-1}$$

→ If $p(l) \to 0$ as $l \to \infty$, Algorithm A converges to the correct solution.

Notes

Density Evolution
– Belief Propagation for AWGN Channels

- AWGN channel: $u_0 = 2/\sigma^2 y$ $(A = 1)$
  Considering that the all-zero codeword was transmitted, we get $u_0 \sim \mathcal{N}(2/\sigma^2, 2 \cdot (2/\sigma^2)) = \mathcal{N}(m_{u_0}, 2m_{u_0})$, with $m_{u_0} = 2/\sigma^2$.
- Gaussian assumption
  - The messages u_i, v_j at the outputs of the check-node and variable-node decoders are Gaussian with means m_{u_i}, m_{v_j} and variances $2m_{u_i}, 2m_{v_j}$.
  - → Density evolution by tracking the means $m_{u_i}(l), m_{v_j}(l)$ over the number of iterations l.
- Variable-node decoder: $m_v(l) = m_{u_0} + (d_v - 1)m_u(l-1)$ by considering independence of the incoming messages.
- Check-node decoders: quite involved....

→ If $m_u(l) \to \infty$ as $l \to \infty$, belief propagation converges to the correct solution.

Notes

Code Design

- Choose the degree distributions $\lambda(x), \rho(x)$ such that the rate R is maximized while the chosen decoder converges provably to the correct solution for the given channel (i.e., $p(l) \to 0$ for Algorithm A, $m_u(l) \to \infty$ for belief propagation).
- So far, density evolution for regular LDPC codes; for irregular codes the error probabilities or means can be obtained by averaging over the degree distributions.

  Example: Algorithm A:

$$
\begin{aligned}
p(l) &= \sum_i p(l|d_v = i)\lambda_i \\
q(l) &= \sum_i q(l|d_c = i)\rho_i
\end{aligned}
$$

- Finding $\mathbf{G}$: generate $\mathbf{H}$ satisfying $\lambda(x), \rho(x)$, bring it into a systematic format, and generate $\mathbf{G}$.

Notes