Soft iterative decoding of block codes, part II

- Review
  - Probability
  - Repetition code => bit node
  - Parity-check code => check node
  - Hamming code
- AWGN channel
- LDPC code
  - Message-passing algorithm
  - Performance plots
- References

John L. Fan
Iospan Wireless, Inc.

www.johnfan.com

Types of probability

- **intrinsic** probability : *a priori*,
  - what was known before event *N*: *a=P(x)*
- **posterior** probability : *a posteriori*,
  - what is known after event *N*: *p=P(x|N)*
- **extrinsic** probability :
  - what has been learned from event *N*

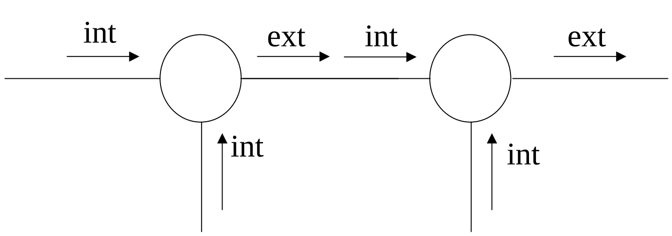
$$p=\frac{ae}{ae+(1-a)(1-e)}$$

$$LLR(p)=LLR(a)+LLR(e)$$

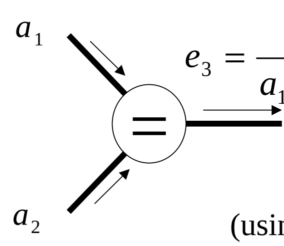
- Log-likelihood ratio $LLR(p)=\log\frac{p}{1-p}$
- "Soft bit" $2p-1=\tanh\left(\frac{1}{2}LLR(p)\right)$
- Hyperbolic tangent $\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$

Graphs

- Composed of nodes and edges
- To each edge is associated a variable
- To each node is associated a "code"
  - The set of edges connected to the node corresponds to the variables in the code
- Soft decoding performed based on the "code" at node N:
  - Input: the **intrinsic** probability w.r.t. node N
  - Output: the **extrinsic** probability w.r.t. node N
- Messages passed in both directions on edges
- Extrinsic output from one node is intrinsic input for other node

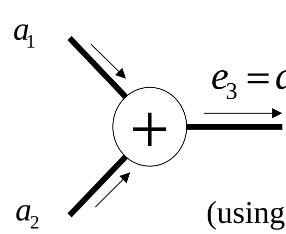


Bit and parity-check nodes (probabilities)



$$e_3=\frac{a_1a_2}{a_1a_2+(1-a_1)(1-a_2)}$$

"Bit node"
(using a repetition code)

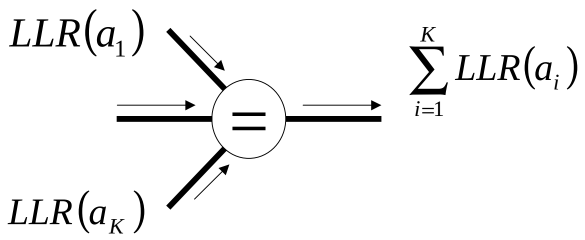


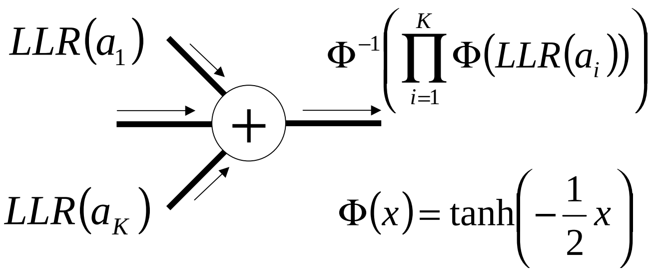
$$e_3=a_1(1-a_2)+(1-a_1)a_2$$

"Check node"
(using a parity-check code)

Bit and parity-check nodes (log-likelihood ratio)

- The log-likelihood ratio is defined for binary variables:





$$\Phi(x) = \tanh\left(-\frac{1}{2}x\right)$$

Proof of equivalence for K=2

- Claim: the following two statements are equivalent.

$$e_3 = a_1(1-a_2) + (1-a_1)a_2$$

$$LLR(e_3) = \Phi^{-1}\left(\prod_{i=1}^{2} \Phi(LLR(a_i))\right)$$

- Proof:

$$\Phi(LLR(e_3)) = \prod_{i=1}^{2} \Phi(LLR(a_i))$$

$$1 - 2e_3 = (1-2a_1)(1-2a_2)$$

$$e_3 = \frac{1}{2}(1 - (1 - 2a_1 - 2a_2 + 4a_1a_2))$$

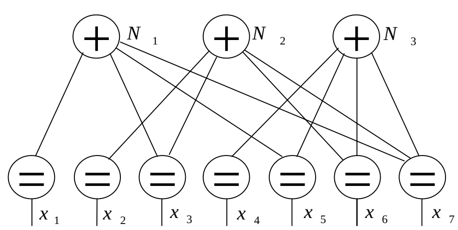
$$e_3 = a_1 + a_2 - 2a_1a_2$$

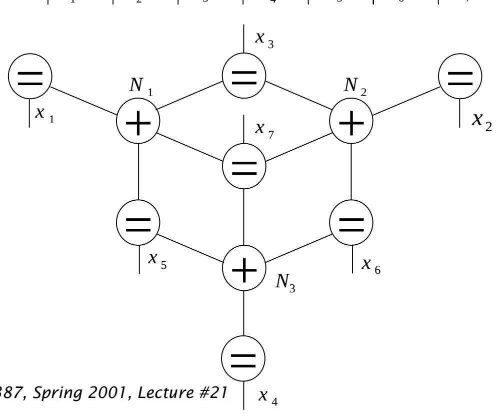
$$e_3 = (1-a_1)a_2 + a_1(1-a_2)$$

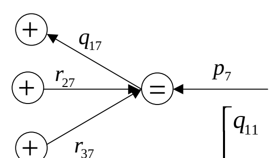
- Note: this generalizes to K edges

Hamming code

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







$$LLR(q_{ij}) \leftarrow \sum_{\substack{j' \in Col[i] \\ j' \neq j}} LLR(r_{ij'}) + LLR(p_i)$$

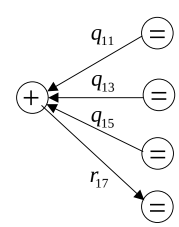
$$\begin{bmatrix} q_{11} & & q_{13} & & q_{15} & & q_{17} \\ & q_{22} & q_{23} & & & q_{26} & q_{27} \\ & & & q_{34} & q_{35} & q_{36} & q_{37} \end{bmatrix}$$

Bit node (repetition code)

bit-to-check message

$$\begin{bmatrix} r_{11} & & r_{13} & & r_{15} & & r_{17} \\ & r_{22} & r_{23} & & & r_{26} & r_{27} \\ & & & r_{34} & r_{35} & r_{36} & r_{37} \end{bmatrix}$$

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \end{bmatrix}$$



$$LLR(r_{ij}) \leftarrow \Phi^{-1}\left(\prod_{\substack{i' \in Row[j] \\ i' \neq i}} \Phi(LLR(q_{i'j}))\right)$$

$$\begin{bmatrix} q_{11} & & q_{13} & & q_{15} & & q_{17} \\ & q_{22} & q_{23} & & & q_{26} & q_{27} \\ & & & q_{34} & q_{35} & q_{36} & q_{37} \end{bmatrix}$$

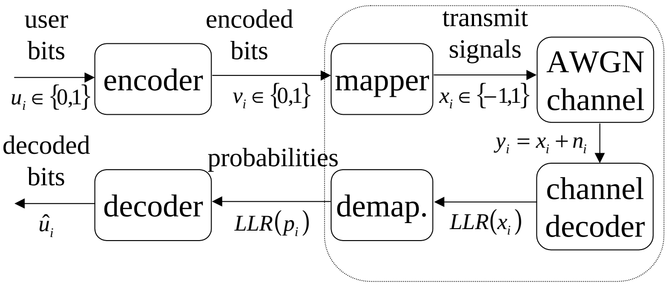
Check node (parity-check code)

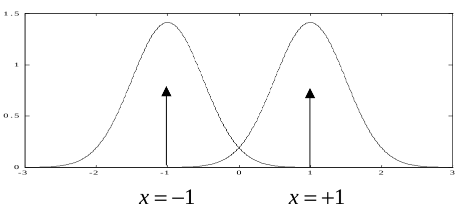
check-to-bit message

$$\begin{bmatrix} r_{11} & & r_{13} & & r_{15} & & r_{17} \\ & r_{22} & r_{23} & & & r_{26} & r_{27} \\ & & & r_{34} & r_{35} & r_{36} & r_{37} \end{bmatrix}$$

AWGN channel

- Additive white Gaussian noise
- $y_i = x_i + n_i$, where the noise has Gaussian distribution, with variance σ^2





Using Gaussian distributions,

$$P(y_i \mid x_i = +1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - 1)^2\right)$$

$$P(y_i \mid x_i = -1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i + 1)^2\right)$$

By Bayes' theorem,

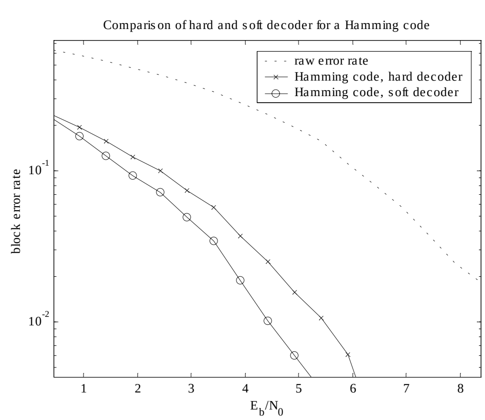
$$P(x_i = +1 \mid y_i) = \frac{P(y_i \mid x_i = +1) P(x_i = +1)}{P(y_i)}$$

So the log-likelihood ratio (LLR) is simply given by:

$$\begin{aligned} LLR(x_i) &= \log \frac{P(x_i = +1 \mid y_i)}{P(x_i = -1 \mid y_i)} \\ &= -\frac{1}{2\sigma^2}\left((y_i - 1)^2 - (y_i + 1)^2\right) \\ &= \frac{2}{\sigma^2} y_i \end{aligned}$$

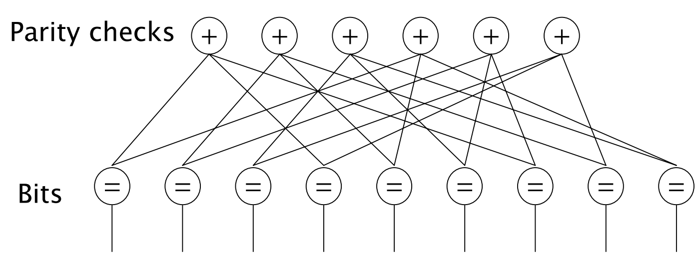
Hamming code: performance

- Rate = 4/7
- Binary symmetric channel
  - Detects 2 errors
  - Corrects 1 errors
- Binary erasure channel
  - Detects 7 erasures
  - Corrects 3 erasures
- Binary-input AWGN channel:



Low-density parity-check codes

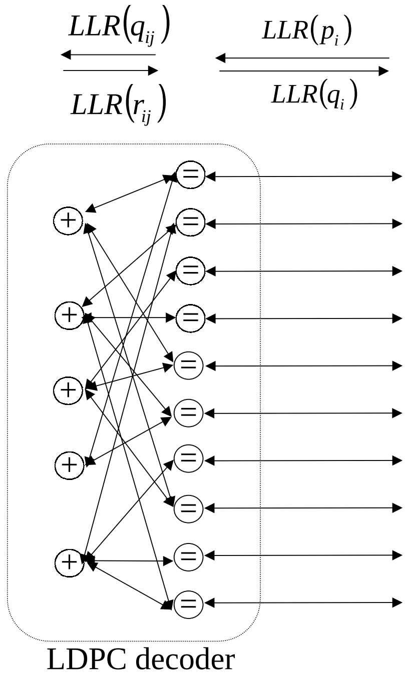
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



LDPC codes

- Code constructions
  - "Low density" (sparse), avoid small cycles
  - Randomly chosen codes work well!
  - Optimal constructions with irregular degrees
- LDPC encoding
  - use generator matrix (e.g., constructed from H using Gaussian elimination)
  - systematic codes are often used
- LDPC decoding
  - Message-passing between bit and check nodes
  - Combine many simple decoders to approximate a very complicated one
  - *Shannon capacity is achieved!*
- Benefits over turbo codes
  - More efficient (fewer operations)
  - Parallel computation (for hardware)
  - Stopping criterion (fewer iterations on average)
  - Patent-free

Graph for LDPC code



Summary of LDPC decoding

- Initialize

$$LLR(p_i)=\frac{2}{\sigma^2}y_i \qquad LLR(r_{ji})=0$$

- Iterate (repeat until done)
  - Messages from bit nodes to check nodes

$$LLR(q_{ij})\leftarrow \sum_{\substack{j'\in Col[i]\\ j'\neq j}} LLR(r_{ij'})+LLR(p_i)$$

  - Messages from check nodes to bit nodes

$$LLR(r_{ij})\leftarrow \Phi^{-1}\left(\prod_{\substack{i'\in Row[j]\\ i'\neq i}}\Phi\left(LLR(q_{i'j})\right)\right)$$

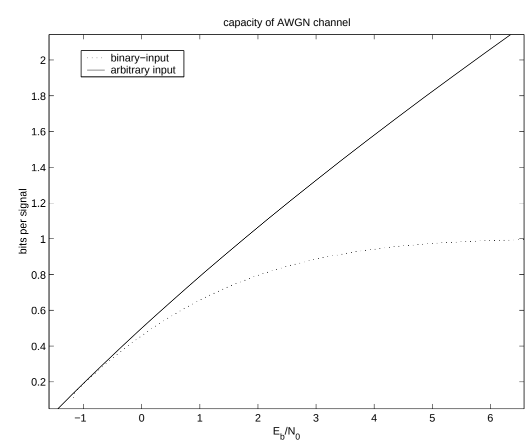
$$\Phi(x)=\tanh\left(-\tfrac{1}{2}x\right)$$

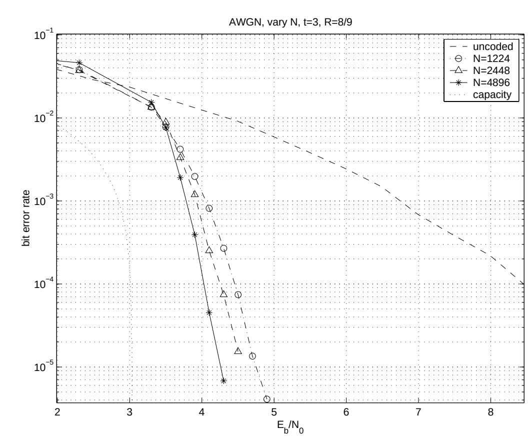
- Decoder output

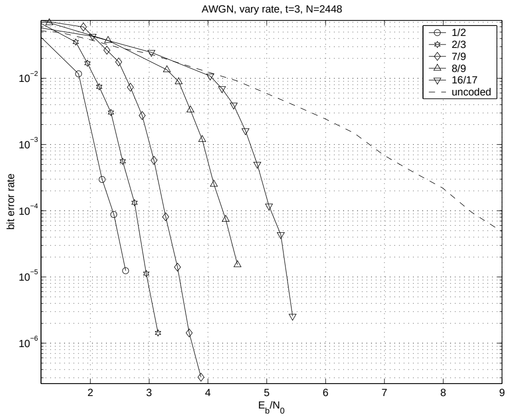
posterior extrinsic intrinsic

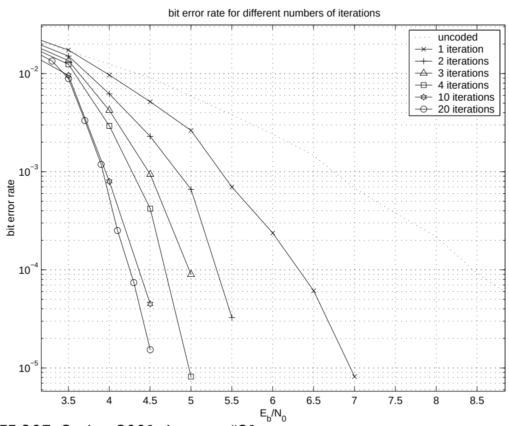
$$LLR(q_i)\leftarrow \sum_{j'\in Col[i]} LLR(r_{ij'})+LLR(p_i)$$

$$\hat{u}_i=\begin{cases}1 & \textbf{if } LLR(q_i)>0\\ 0 & \textbf{if } LLR(q_i)<0\end{cases}$$









References

- Books
  - R.G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MIT Press, 1963.
  - B.J. Frey, *Graphical Models for Machine Learning and Digital Communications*, MIT Press, 1998.
  - C. Heegard and S.B. Wicker, *Turbo codes,* Kluwer Academic Publishers, 1999.
  - B. Vucetic and J. Yuan, *Turbo Codes - Principles and Applications*, Kluwer Academic Publishers, 2000.
  - K.M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection - Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic Publishers, 2000.
  - J.L. Fan, *Constrained Coding and Soft Iterative Decoding*, Kluwer Academic Publishers, 2001.
  - D.J.C. MacKay, *Information Theory, Pattern Recognition, and Neural Networks*, http://wol.ra.phy.cam.ac.uk/mackay/itprnn
- Articles
  - D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399-431, March 1999.
  - G.D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520-548, Feb. 2001.
  - Special issue on codes, graphs and iterative decoding, *IEEE Transactions in Information Theory*, February 2001