Chap. 3 Low-Density Parity-Check Codes

□ Review of Block Codes

◇ Block code

- Alphabet $\mathcal{A}$: finite field $\mathbb{F}_q$, finite ring $\mathbb{Z}_q$, etc.

- (n, M) block code $\mathcal{C}$ over $\mathcal{A}$
  $=$ the set of M vectors of length n with components in $\mathcal{A}$

- Rate $= \frac{\log_q M}{n}$

- A vector $\mathbf{c} \in \mathcal{C}$ is called a codeword or a code vector of $\mathcal{C}$.

◇ An $[n, k]$ linear block code $\mathcal{C}$ over $\mathbb{F}_2$

- $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_2^n$

- $\mathbb{F}_2^n \triangleq \{(x_1, x_2, \cdots, x_n) \mid x_i \in \mathbb{F}_2\}$: n-dimensional vector space over $\mathbb{F}_2$

- $n \triangleq$ length (or code length) of the code

- $k \triangleq$ dimension of the code

- Code rate $= \frac{k}{n}$

- Conditions for a linear code (or a subspace)

  1) $\mathbf{c}_1 + \mathbf{c}_2 \in \mathcal{C}, \quad \forall\ \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$
  2) $a\mathbf{c} \in \mathcal{C}, \quad \forall\ \mathbf{c} \in \mathcal{C}, \ \forall\ a \in \mathbb{F}_q$

◇ Generator matrix G for $\mathcal{C}$

- Since $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_2^n$, there is a basis for $\mathcal{C}$, say,

$$\{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k\}.$$

Note:

1) $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$ are linearly independent over $\mathbb{F}_q$.
2) $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$ span $\mathcal{C}$, i.e. $\mathcal{C} =< \mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k >$.

- Any codeword $\mathbf{c} = (c_1, c_2, \cdots, c_n)$ can be expressed uniquely as a linear combination of $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k$. i.e.,

$$\mathbf{c} = m_1\mathbf{g}_1 + m_2\mathbf{g}_2 + \cdots + m_k\mathbf{g}_k.$$

Therefore,

$$\begin{aligned} \mathbf{c} &= \begin{bmatrix} m_1 & m_2 & \cdots & m_k \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_k \end{bmatrix} \\ &= \begin{bmatrix} m_1 & m_2 & \cdots & m_k \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ & & \vdots & \\ g_{k1} & g_{k2} & \cdots & g_{km} \end{bmatrix} \\ \text{or} \quad \mathbf{c} &= \mathbf{m}G. \end{aligned}$$

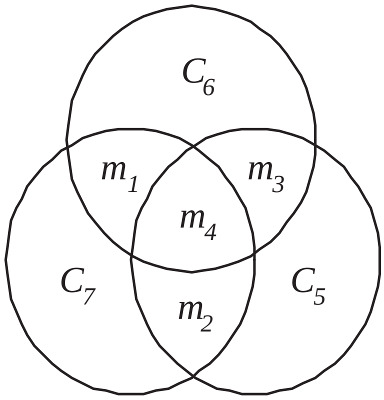
Note: 1) G is called a *generator matrix* of the code $\mathcal{C}$
2) $\mathcal{C}$ = the row space of $G = \{\mathbf{m}G \mid \mathbf{m} \in \mathbb{F}_2^k\}$
3) $\mathbf{m} \in \mathbb{F}_2^k$: a message to be encoded.

Remark:

1) Elementary row operations on G does not change the code $\mathcal{C}$ because $\mathcal{C} = \mathcal{R}(G)$
2) Any generator matrix can be reduced to a "*row-reduced echelon form*"
3) A linear block code can always be considered to be epuivalent to a systematic code by applying elementary column operations, if necessary.

$$\Rightarrow \quad G = [I_k : P]$$

Example: [7,4] Hamming code



Encoding rule:

$$\underbrace{\begin{aligned} c_1 &= m_1 \\ c_2 &= m_2 \\ c_3 &= m_3 \\ c_4 &= m_4 \end{aligned}}_{\text{information symbols}} \qquad \underbrace{\begin{aligned} c_5 &= m_2 + m_3 + m_4 \pmod 2 \\ c_6 &= m_1 + m_3 + m_4 \pmod 2 \\ c_7 &= m_1 + m_2 + m_4 \pmod 2 \end{aligned}}_{\text{redundant symbols}}$$

The codeword $\mathbf{c}$ can be expressed as

$$\begin{aligned} \mathbf{c} &= \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_7 \end{bmatrix} \\ &= \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \\ \Rightarrow \quad \mathbf{c} &= \mathbf{m}G \end{aligned}$$

On the other hand, $\mathbf{c}$ satisfies the following equations

$$\begin{aligned} c_2 + c_3 + c_4 + c_5 &= 0 \\ c_1 + c_3 + c_4 + c_6 &= 0 \\ c_1 + c_2 + c_4 + c_7 &= 0 \end{aligned}$$

$$\Rightarrow \quad \begin{bmatrix} 0 & 1 & 1 & 1 & \vdots & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & \vdots & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & \vdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{or} \quad H\mathbf{c}^t = 0$$

- In general,

  a matrix H with the property that $H\mathbf{c}^t = 0$ if and only if $\mathbf{c} \in \mathcal{C}$ is called a parity-check matrix for $\mathcal{C}$.

  Remark:

  1) $\mathcal{C}$ is the null space of H, denoted by $\mathcal{C} = \mathcal{N}(H)$, that is,

  $$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_2^n \mid H\mathbf{c}^t = 0\}.$$

  2) For any $[n, k]$ systematic code,

  $$G = [I_k \vdots P] \leftrightarrow H = [-P \vdots I_{n-k}]$$

  3) $GH^T = 0$

◇ **Minimum distance of $\mathcal{C}$**

- $w_H(\mathbf{x}) \triangleq$ Hamming weight of $\mathbf{x}$
  $=$ the number of nonzero symbols in $\mathbf{x}$
- $d_H(\mathbf{x}, \mathbf{y}) \triangleq$ Hamming distance between $\mathbf{x}$ and $\mathbf{y}$
  $= w_H(\mathbf{x} - \mathbf{y})$
- $d_H(\mathbf{x}, \mathbf{y})$ is a metric:

  a) $d_H(\mathbf{x}, \mathbf{y}) \geq 0$ with equality iff $\mathbf{x} = \mathbf{y}$
  b) $d_H(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{y}, \mathbf{x})$
  c) $d_H(\mathbf{x}, \mathbf{y}) \leq d_H(\mathbf{x}, \mathbf{z}) + d_H(\mathbf{z}, \mathbf{y})$
- $d \triangleq$ minimum distance of $\mathcal{C}$
  $= \min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \neq \mathbf{y};\ \mathbf{x}, \mathbf{y} \in \mathcal{C}\}$
  $= \min\{w_H(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}\}$ (for a linear code)

Theorem 1 *There exists a codeword of weight s in $\mathcal{C}$ iff there exists s columns in H which are linearly dependent.*

Corollary 2 *The minimum distance d of $\mathcal{C}$ is the minimum number of columns in H which are linearly dependent (over $\mathbb{F}_q$).*

◇ **Repetition code vs. Even parity check code**

$$\left.\begin{array}{c} G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \\[2ex] H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ & & & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix} \end{array}\right\} [n, 1, n] \text{ code}$$

◇ **Dual code of $\mathcal{C}$:**

$$\begin{aligned} \mathcal{C}^{\perp} &= \text{dual code of } \mathcal{C} \\ &= \{\mathbf{x} \in \mathbb{F}_2^{\mathbf{n}} \mid \mathbf{x}\,\mathbf{c}^{\mathbf{t}} = \mathbf{0}, \ \forall\ \mathbf{c} \in \mathcal{C}\} \end{aligned}$$

□ Low-Density Parity-Check Codes

◇ Milestone References

1) R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, pp. 21-28, Jan. 1962.
2) R. G. Gallager, *Low-density Parity-check Codes*. MIT Press, 1963. (An expanded and revised version of Ph.D.thesis (1960))

◇ The Noisy Channel Coding Theorem (Shannon)

If properly coded information is transmitted at a rate R below channel capacity C, then *the probability P_e of decoding error can be made to approach zero exponentially with the code length*.

Note:

1) code length n $\uparrow$ $\Rightarrow$ decoding complexity $\uparrow$
(computation time or equipment costs)

2) If $R < C$, then

$$P_e < e^{-nE(R)}$$

where $E(R)$ is the error exponent.

◇ Elias (1955)

If *a typical parity-check code of long block length* is used on a binary symmetric channel, and if the code rate is between critical rate and channel capacity, then *the probability of decoding error will be almost as small as that for the best possible code of that rate and block length*.

Note:

In general, decoding of parity-check codes (or linear codes) is not simple.
$\Rightarrow$ Need special classes of parity-check codes.
"Ensemble of LDPC codes"

◇ Low-density parity-check codes

- A low-density parity-check code is a linear block code whose parity check matrix contains most $0's$ and only a small number of $1's$.

"sparseness" or *"low-density of 1's"*

- A regular (N, j, k) LDPC code is a linear block code of length N, whose parity-check matrix H contains exactly j $1's$ in each column and k $1's$ in each row.

Note:

1) $j, k \ll N$ and $j, k \ll N - K$.
2) Assuming that H has full rank, say $N - K$,

$$jN = (N - K)k.$$

Therefore, the *code rate* R is given by

$$R \triangleq \frac{K}{N} = \frac{k-j}{k}$$

where K is the dimension of the code.

3) $j \geq 3$ and $k > j$ in most applications.
4) For $j > 3$ and a sufficiently low rate, the probability of error over a BSC decreases at least exponentially with $\sqrt{N}$.

◇ **Remark on the Historical Backgrounds**

- *Gallager (1960) invented LDPC codes and their iterative decoding.*
- Dark Ages
- Zyablov and Pinsker (1975): Flipping. Linear fraction
- *Tanner (1981): codes defined on graphs*
- *Pearl (1986): Belief propagation*
- Sourlas (1989): codes and random fields
- *Berrou, Galvieux, Thitimajshima (1993): turbo codes*

— — — — — ○ — — — — — — ○ — — — — — — ○ — — — — — —

- *Rediscovery of LDPC codes*

  Reinvented - Mackay and Neal (1996)

  Decoding on Graphs - Wiberg (1996)

  Expander graphs - Spielman, Sipser (1996)

  Tornado codes (over the erasure channel)

  Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann (1997)

  Irregular graphs for error correcting

  Luby, Mitzenmacher, Shokrollahi, Spielman (1998)

Note:

1) Regular LDPC codes are *not optimum* (in the sense of minimizing probability of decoding error for a given block length).

2) The maximum rate at which these codes can be used is bounded below channel capacity.

3) A very simple decoding scheme exists.

◇ Methods of Analysis of Codes

- Individual codes: difficult if $N \to \infty$.
- An ensemble of LDPC codes: simpler (by statistical statements)

◇ Construction of an Ensemble of (N, j, k) LDPC codes

1) H is divided into j submatrices, each containing a single 1 in each column.

2) The first of these submatrices contains all its 1's in descending order, i.e., the ith row contains 1's in columns $(i-1)k+1$ to ik.

3) The other submatrices are merely column permutations of the first.

Example:

An $(j, 6)$ LDPC code can be constructed by

$$H = \left[\begin{array}{cccc} 111111 & & & \\ & 111111 & & \\ & & 111111 & \\ & & & 111111 \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ & \pi_2(H_1) & & \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ & \vdots & & \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ & \pi_j(H_1) & & \end{array}\right] = \left[\begin{array}{c} \\ H_1 \\ \\ \cdots\cdots \\ \pi_2(H_1) \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ \pi_j(H_1) \end{array}\right]$$

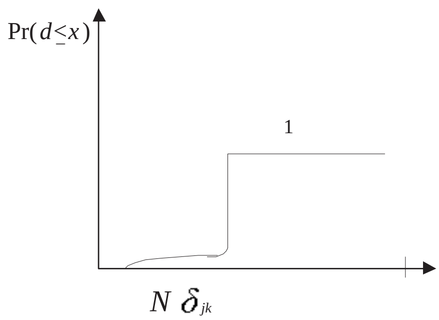
where π_i is a column permutation of H_1

◇ Minimum Distance of LDPC Codes

- The minimum distance d of a code in this ensemble is a *random variable*.
- The distribution function of this random variable can be overbounded by a function.
- For a fixed $j \geq 3$ and $k > j$, as $N \to \infty$,

$$\Pr(d \leq x) \approx u(x - N\delta_{jk})$$

where δ_{jk} is a fixed fraction and $u(\cdot)$ is the unit-step function.



- For large N, practically all the codes in the ensemble have a minimum distance of at least $N\delta_{jk}$.

j	k	Rate	δ_{jk}	δ
5	6	0.167	0.255	0.263
4	5	0.2	0.210	0.241
3	4	0.25	0.122	0.214
4	6	0.333	0.129	0.173
3	5	0.4	0.044	0.145
3	6	0.5	0.023	0.11

δ_{jk}: the ratio of typical minimum distance to block length for an (N, j, k) code
δ: the same ratio for an ordinary parity-check code of the same rate
(See Fig 3, Gallager (1962, IT))

- Loss of rate associated with regular LDPC Codes

j	k	Rate for (N, j, k) code	Rate for an equivalent optimum code of the same exponent
3	6	0.5	0.555
3	5	0.4	0.43
4	6	0.333	0.343
3	4	0.25	0.266

(See Fig 4, Gallager (1962, IT))

- Over a reasonable range of channel transition probabilities, the low-density code has a probability of decoding error that decreases exponentially with block length and the exponent is the same as that for the optimum code of slightly higher rate.

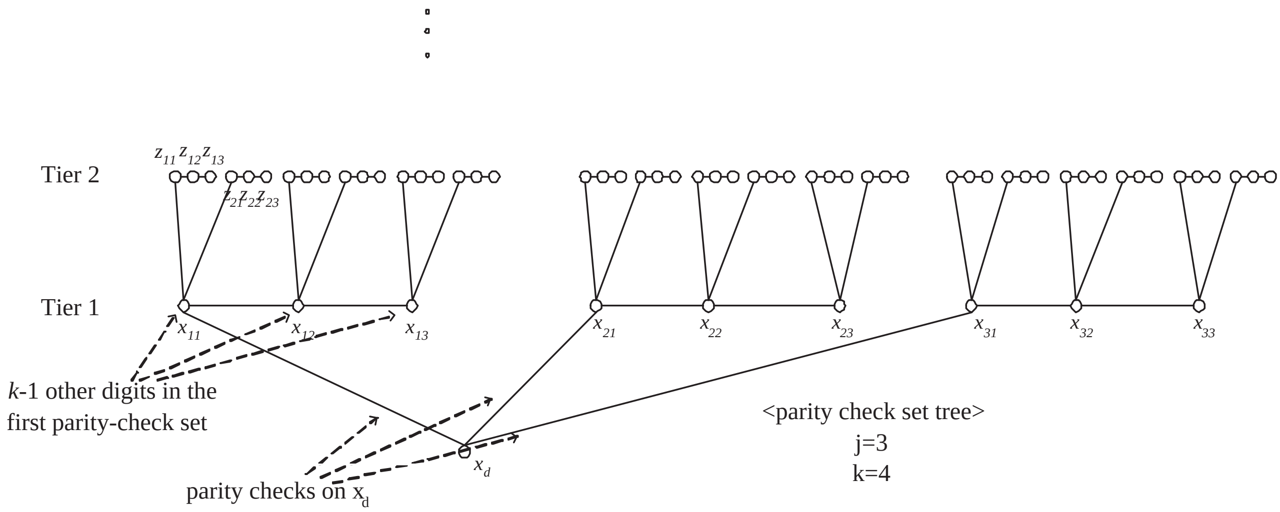
Remark: For long binary BCH codes,

$$\frac{d}{n} \to 0 \quad \text{as} \quad n \to \infty \quad \text{(Berlekamp)}$$

□ Hard-decision decoding (Gallager Decoding A)

- BSC at rates far below channel capacity
- Decoding procedure:
  1) Compute all the parity checks.
  2) Change any digit that is contained in more than some fixed number of unsatisfied parity-check equations
  3) Recompute the parity checks using these new values
  4) Repeat 2) and 3) until the parity checks are all satisfied.

- *Parity-check set tree* ($j = 3$ and $k = 4$ case)



- Parity-check equations on x_d:

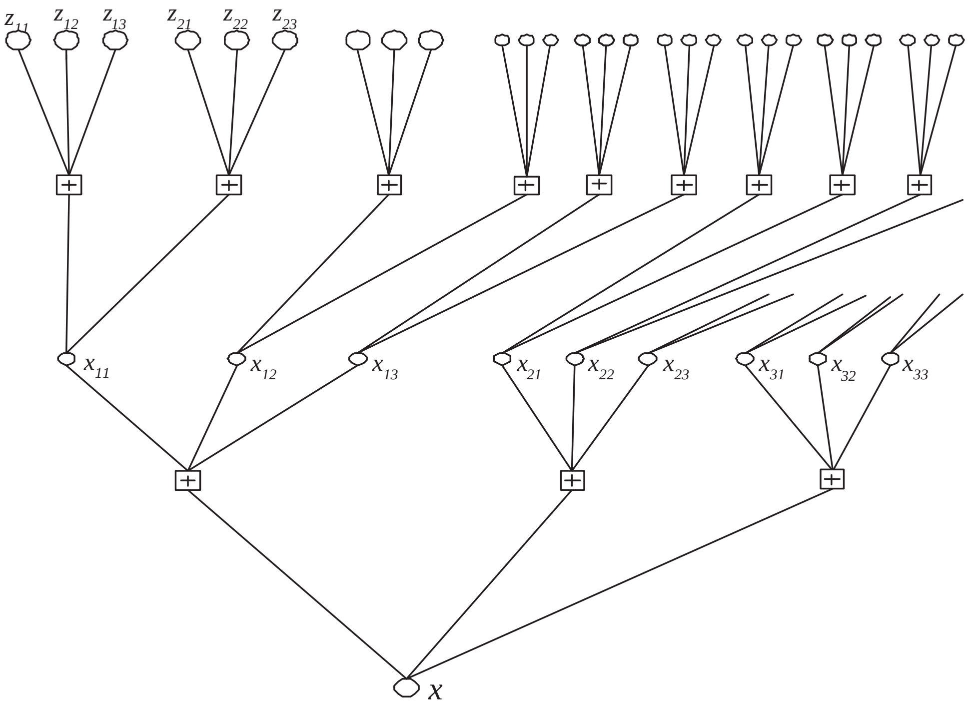
$$\begin{cases} x_d + x_{11} + x_{12} + x_{13} = 0 \\ x_d + x_{21} + x_{22} + x_{23} = 0 \\ x_d + x_{31} + x_{32} + x_{33} = 0 \end{cases}$$

- Parity-check equations on x_{11}:

$$\begin{cases} x_{11} + z_{11} + z_{12} + z_{13} = 0 \\ x_{11} + z_{21} + z_{22} + z_{23} = 0 \\ x_{11} + x_{12} + x_{13} + x_d = 0 \end{cases}$$

◇ Factor graph

The factor graph shows the tree structure from node x.



□ Probabilistic Decoding (Belief-Propagation Decoding)

◇ Problem Formulation

- Define

$$\begin{aligned} \{y\} &\triangleq \text{the set of received symbols} \\ S &\triangleq \text{the event that the transmitted digits satisfy} \\ &\quad\ \text{the } j \text{ parity-check equations on digit } d. \end{aligned}$$

- Assume the *Ensemble of events* in which

  1) the transmitted digits in the positions of the first tier of digit d are independent equiprobable binary digits; and
  2) the probabilities of the received symbols in these positions are determined by the channel transition probabilities $P_x(y)$.

- **Goal**: Compute $\Pr(x_d = 1 | \{y\}, S)$.

Theorem 3 *Let*

$$\begin{aligned} P_d &\triangleq \Pr(x_d = 1 | y_d) \\ P_{il} &\triangleq \Pr(x_{il} = 1 | y_{il}) \end{aligned}$$

where

$$x_{il} = l^{th} \text{ digit in the } i^{th} \text{ parity-check set of the first tier.}$$

Then

$$\frac{\Pr(x_d = 0 | \{y\}, S)}{\Pr(x_d = 1 | \{y\}, S)} = \frac{1 - P_d}{P_d} \prod_{i=1}^{j} \left[\frac{1 + \prod_{l=1}^{k-1}(1 - 2P_{il})}{1 - \prod_{l=1}^{k-1}(1 - 2P_{il})} \right].$$

Lemma 4 (Sum of I.I.D. Random Variables) *Let z be the sum of independent and identically distributed (i.i.d.) random variables, given by*

$$z = z_1 + z_2 + \cdots + z_m,$$

where z_i's are independent random variable taking on 0 and 1 with $\Pr(z_i = 1) \triangleq p_i$. Then

$$\Pr(z = \textit{even}) = \frac{1 + \prod_{i=1}^{m}(1 - 2p_i)}{2}.$$

Proof. Note that

$$\begin{aligned} \prod_{i=1}^{m} [(1 - p_i) + p_i t] &= \sum_{l=0}^{m} A_l t^l, \\ \prod_{i=1}^{m} [(1 - p_i) - p_i t] &= \sum_{l=0}^{m} B_l t^l \end{aligned}$$

where

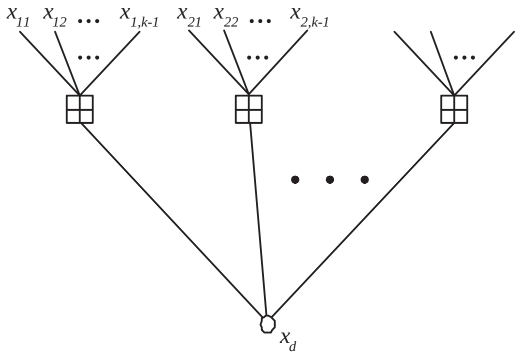
$$A_l = \Pr(z = l) \quad \text{and} \quad B_l = \begin{cases} \Pr(z = l), & l = \text{even} \\ -\Pr(z = l), & l = \text{odd}. \end{cases}$$

Therefore,

$$\begin{aligned} \Pr(z = \text{even}) &= \sum_{l=0,\ l=\text{even}}^{m} \Pr(z = l) \\ &= \frac{1}{2}\left[\prod_{i=1}^{m}(1 - p_i + p_i t) + \prod_{i=1}^{m}(1 - p_i - p_i t)\right]_{t=1} \\ &= \frac{1 + \prod_{i=1}^{m}(1 - 2p_i)}{2}. \end{aligned}$$

□

Proof of Theorem:



Assuming that x_{ij}'s are statistically independent,

$$\begin{aligned}
\Pr\left(S \,|\, x_d = 0, \{y\}\right) &= \Pr\left(\sum_{l=1}^{k-1} x_{il} = \text{even}, \forall\, i = 1, \cdots, j \,|\, x_d = 0, \{y\}\right) \\
&= \prod_{i=1}^{j} \left[\frac{1 + \prod_{l=1}^{k-1}(1 - 2P_{il})}{2}\right] \\
\Pr\left(S \,|\, x_d = 1, \{y\}\right) &= \Pr\left(\sum_{l=1}^{k-1} x_{il} = \text{odd}, \forall\, i = 1, \cdots, j \,|\, x_d = 1, \{y\}\right) \\
&= \prod_{i=1}^{j} \left[\frac{1 - \prod_{l=1}^{k-1}(1 - 2P_{il})}{2}\right]
\end{aligned}$$

by Lemma 4.

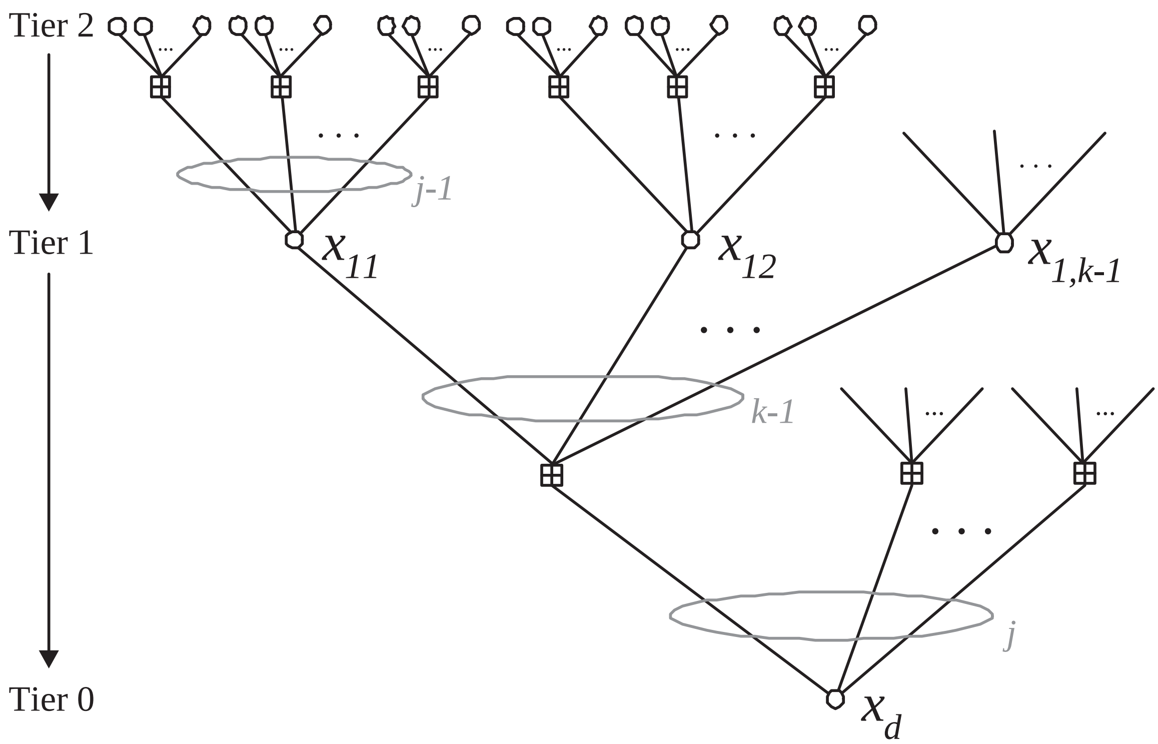
Therefore,

$$\begin{aligned}
\frac{\Pr(x_d = 0|\{y\}, S)}{\Pr(x_d = 1|\{y\}, S)} &= \frac{1 - P_d}{P_d} \cdot \frac{\Pr(S \,|\, x_d = 0, \{y\})}{\Pr(S \,|\, x_d = 1, \{y\})} \\
&= \frac{1 - P_d}{P_d} \prod_{i=1}^{j} \left[\frac{1 + \prod_{l=1}^{k-1}(1 - 2P_{il})}{1 - \prod_{l=1}^{k-1}(1 - 2P_{il})}\right].
\end{aligned}$$

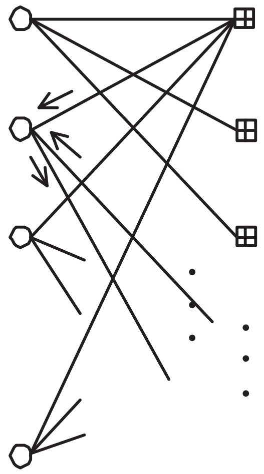
Note:

- The complexity of decoding in many-tier case can be solved from the 1-tier case by a simple *iterative technique*.

- 2-tier case



□ General Decoding Procedure



Remarks on Gallager Decoding

1) If the decoding is successful, $P_d \rightarrow 0$ or 1 (depending on the transmitted digit) as the number of iterations is increased.

2) This procedure is only valid for as many iterations as meet the independence assumption.

3) cycle $\leftrightarrow$ independency.

4) Reasonable assumption: The dependencies have a relatively minor effect and tend to cancel each other out somewhat.

5) The computation per digit per iteration is independent of block length.

6) The average number of iterations required to decode is bounded by a quantity proportional to the $\log(\log n)$.

□ Decoding in terms of LLR

Define

$$\ln \frac{1-P_d}{P_d} \triangleq \alpha_d \beta_d, \qquad \ln \frac{1-P_{il}}{P_{il}} \triangleq \alpha_{il}\beta_{il}$$

and

$$\ln \frac{\Pr(x_d = 0|\{y\}, S)}{\Pr(x_d = 1|\{y\}, S)} \triangleq \alpha'_d \beta'_d$$

$$(\alpha = \text{sign} \quad \text{and} \quad \beta = \text{magnitude})$$

Then

$$\alpha'_d \beta'_d = \alpha_d \beta_d + \sum_{i=1}^{j} \left(\prod_{l=1}^{k-1} \alpha_{il} \right) f \left(\sum_{l=1}^{k-1} f(\beta_{il}) \right) \tag{1}$$

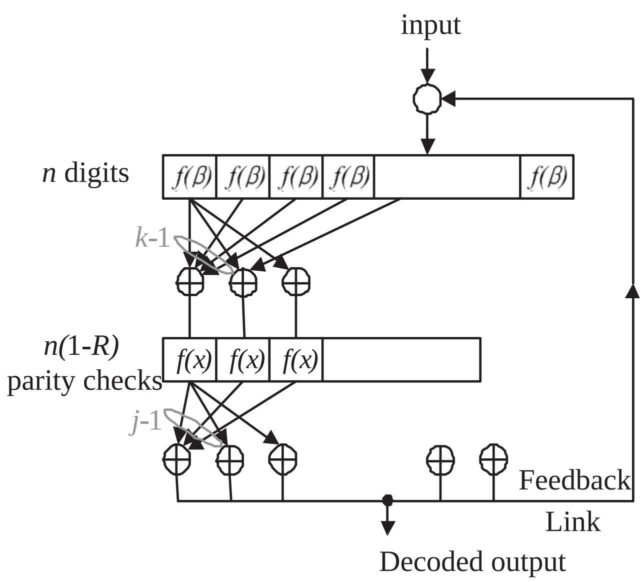
where

$$f(\beta) = \ln \frac{e^\beta + 1}{e^\beta - 1}$$

Exercise: Derive (1).

Note:

- Serial computing
- Parallel computing $\rightarrow$ fast



Derivation of Equation (1):

$$\ln \frac{1-P_{il}}{P_{il}} = \alpha_{il}\beta_{il} \quad \Rightarrow \quad 1-2P_{il} = \frac{e^{\alpha_{il}\beta_{il}}-1}{e^{\alpha_{il}\beta_{il}}+1} = \alpha_{il}\frac{e^{\beta_{il}}-1}{e^{\beta_{il}}+1} = \alpha_{il}e^{-f(\beta_{il})}.$$

Using this representation, we get

$$\prod_{l=1}^{k-1}(1-2P_{il}) = \left(\prod_{l=1}^{k-1}\alpha_{il}\right) e^{-\sum_{l=1}^{k-1} f(\beta_{il})} \triangleq \alpha e^{-\beta}.$$

where

$$\alpha = \prod_{l=1}^{k-1}\alpha_{il}, \quad \beta = \prod_{l=1}^{k-1} f(\beta_{il}).$$

Hence,

$$\ln \frac{1+\prod_{l=1}^{k-1}(1-2P_{il})}{1-\prod_{l=1}^{k-1}(1-2P_{il})} = \ln \frac{1+\alpha e^{-\beta}}{1-\alpha e^{-\beta}} = \alpha \ln \frac{1+e^{-\beta}}{1-e^{-\beta}} = \alpha f(\beta).$$

$$\therefore \alpha'_d\beta'_d = \alpha_d\beta_d + \sum_{i=1}^{j}\left(\prod_{l=1}^{k-1}\alpha_{il}\right) f\left(\sum_{l=1}^{k-1} f(\beta_{il})\right).$$

□ Probability of Error in Gallager Decoding A

Assumption

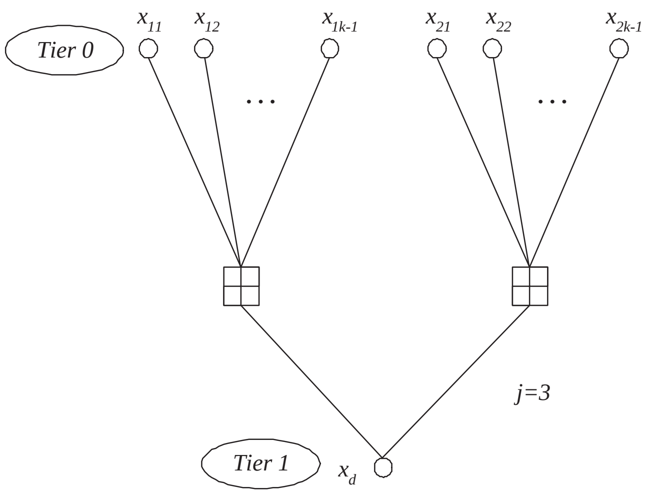
1) BSC with crossover probability p_0.
2) (n, j, k) LDPC code with $j = 3$, for simplicity.

Decoding rule

If *both of two* check-equations containing x_d are not satisfied, say,

$$x_d + x_{11} + \cdots + x_{1,k-1} \neq 0$$
$$x_d + x_{21} + \cdots + x_{2,k-1} \neq 0$$

then $x_d \quad \rightarrow \quad x_d + 1$.



Computation of related probabilities

- $\Pr\{$even number of errors among $k-1$ digits$\}$

$$= \frac{1 + (1-2p_0)^{k-1}}{2}.$$

- $\Pr\{$odd number of errors among $k-1$ digits$\}$

$$= \frac{1 - (1-2p_0)^{k-1}}{2}.$$

- $\Pr\{$a digit in the first tier is received in error and then corrected$\}$

$$= p_0 \cdot \left[\frac{1+(1-2p_0)^{k-1}}{2}\right]^2.$$

- $\Pr$ $\{$a digit in the first tier is received correctly but then changed because of unsatisfied parity checks$\}$

$$= (1-p_0) \cdot \left[\frac{1-(1-2p_0)^{k-1}}{2}\right]^2.$$

- Let p_1 be the probability of error of a digit in the first tier after applying this decoding process.

  Then

$$\begin{aligned} p_1 &= p_0 \cdot \Pr\{\text{no correction}\} + (1-p_0) \cdot \Pr\{\text{correction}\} \\ &= p_0 \left(1 - \left[\frac{1+(1-2p_0)^{k-1}}{2}\right]^2\right) + (1-p_0) \left[\frac{1-(1-2p_0)^{k-1}}{2}\right]^2. \end{aligned}$$

- Using the induction, we get the following recursion:

$$p_{i+1} = p_0 \left(1 - \left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^2\right) + (1-p_0) \left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^2$$

  where p_i is *the probability of error after processing of a digit in the ith tier*.

Decoding convergence

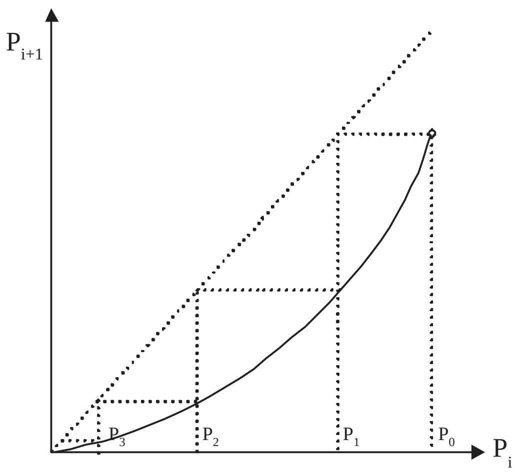
- For sufficiently small p_0,

$$p_i \rightarrow 0 \quad \text{as} \quad i \rightarrow \infty.$$

- Note that

$$\left.\begin{array}{ll} 0 < p_{i+1} < p_i & \text{for} \;\; 0 < p_i \le p_0 \\ p_{i+1} = p_i & \text{for} \;\; p_i = 0 \end{array}\right\} \quad \Rightarrow \quad p_i \rightarrow 0.$$

- Density evolution from p_i to p_{i+1}:



For sufficiently small p_0, the recursion for p_i can be approximated as

$$p_{i+1} \approx p_i \cdot 2(k-1)p_0.$$

Therefore,

$$p_i \approx c\,[\,2(k-1)p_0\,]^i$$

for a constant c.

Maximum p_0 for weak bound decoding convergence in regular LDPC codes

j	k	Rate	p_0
3	6	0.5	0.04
3	5	0.4	0.061
4	6	0.333	0.075
3	4	0.25	0.106

□ General Case

Decoding rule:

$$x \;\to\; x+1 \pmod 2 \quad \text{if the number of unsatisfied parity-checks} \geq b.$$

Recursion for p_i:

$$\begin{aligned} p_{i+1} \;=\;& p_0 - p_0 \sum_{l=b}^{j-1} \binom{j-1}{l} \left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^{l} \left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^{j-1-l} \\ &+(1-p_0) \sum_{l=b}^{j-1} \binom{j-1}{l} \left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^{l} \left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^{j-1-l}. \end{aligned}$$

Convergence of p_i:

- To minimize p_i, find the smallest integers b for which

$$\frac{1-p_0}{p_0} \leq \left[\frac{1+(1-2p_i)^{k-1}}{1-(1-2p_i)^{k-1}}\right]^{2b-j+1}.$$

  Note that $\quad p_i \downarrow \;\Rightarrow\; b \downarrow$

- Optimal choice for b, if p_i is sufficiently small:

$$b = \begin{cases} j/2, & j \text{ even} \\ (j+1)/2, & j \text{ odd.} \end{cases}$$

- Then the recursion for p_i can be approximated as

$$p_{i+1} = \begin{cases} p_0 \binom{j-1}{\frac{j-1}{2}} (k-1)^{(j-1)/2}\, p_i^{(j-1)/2} \;+\; \text{higher order terms} \quad (j \text{ odd}) \\ \\ \binom{j-1}{\frac{j}{2}} (k-1)^{j/2}\, p_i^{j/2} \;+\; \text{higher order terms} \quad (j \text{ even}). \end{cases}$$

  Therefore, p_i can be approximated as

$$p_i \;\leq\; \begin{cases} \exp\left[-C_{jk} \left(\frac{j-1}{2}\right)^{i}\right], & j \text{ odd} \\ \\ \exp\left[-C_{jk} \left(\frac{j}{2}\right)^{i}\right], & j \text{ even.} \end{cases}$$

Number of iterations to guarantee the independency:

- Since there are $(j-1)^m(k-1)^m$ digits in the mth tier of a tree,

$$n \geq (j-1)^m(k-1)^m$$

  for independent digits.

- Then

$$\frac{\ln n}{\ln(j-1)(k-1)} \geq m \geq \frac{\ln\left(\frac{n}{2k}-\frac{n}{2j(k-1)}\right)}{2\ln(k-1)(j-1)}.$$

- The probability p_m of error after processing of a digit in the mth tier can be upper bounded by

$$p_m \leq \exp\left(-c_{jk}\left[\frac{n}{2k}-\frac{n}{2j(k-1)}\right]^\alpha\right)$$

  where

$$\alpha = \begin{cases} \frac{\ln\frac{j-1}{2}}{2\ln(k-1)(j-1)}, & j \text{ odd} \\ \frac{\ln\frac{j}{2}}{2\ln(k-1)(j-1)}, & j \text{ even.} \end{cases}$$

- For $j > 3$,

$$p_m \leq \exp(-c\sqrt{n}\,).$$

Remark:

1) Another way to evaluate the probabilistic decoding scheme is to *calculate the probability distributions of the log-likelihood ratios (LLRs) for a number of iterations*.

2) The above approach is also another *density evolution*.