A Low-Density Parity-Check Code Tutorial
Part I - Introduction and Overview

William Ryan, Associate Professor
Electrical and Computer Engineering Department
The University of Arizona
Box 210104
Tucson, AZ 85721
ryan@ece.arizona.edu
April 2002

Block Code Fundamentals

- we will consider only (n,k) linear block codes over the binary field $\mathbb{F}_2 = \{0, 1\}$
- $\mathbb{F}_2^n$ the n-dimensional vector space over $\mathbb{F}_2$
- the elements of $\mathbb{F}_2^n$ are the 2^n n-tuples $\mathbf{v} = [v_0, v_1, \ldots, v_{n-1}]$
- Definition: An (n, k) linear block code $C$ with data word length k and codeword length n is a k-dimensional subspace of $\mathbb{F}_2^n$
- there are $2^k$ datawords $\mathbf{u} = [u_0, u_1, \ldots, u_{k-1}]$ and $2^k$ corresponding codewords $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ in the code $C$.

- since $C$ is a subspace of dimension k, $\exists k$ linearly independent vectors $\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}$ which span $C$
- the correspondence (mapping) $\mathbf{u} \rightarrow \mathbf{c}$ is thus naturally written as

$$\mathbf{c} = \mathbf{u}_0 \mathbf{g}_0 + \cdots + \mathbf{u}_{k-1} \mathbf{g}_{k-1}$$

- in matrix form, this is

$$\mathbf{c} = \mathbf{u} \mathbf{G}$$

where

$$\mathbf{G} = \begin{bmatrix}
- \mathbf{g}_0 & - \\
- \mathbf{g}_1 & - \\
\vdots & \\
- \mathbf{g}_{k-1} & - \\
\end{bmatrix}_{k \times n}$$

is the so-called generator matrix for $C$

- $\{\mathbf{g}_i\}$ being linearly independent
  $\Rightarrow$ $G$ has rank $k$
  $\Rightarrow$ $G$ may be row reduced and put in the form

$$G = [I : P]$$

(after possible column swapping which permutes the order of the bits in the code words)

- the null space $C^\perp$ of the subspace $C$ has dimension n-k and is spanned by n-k (linearly independent vectors $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{n-k-1}$
- since each $\mathbf{h}_i \in C^\perp$, we must have for any $\mathbf{c} \in C$ that

$$\mathbf{c} \mathbf{h}_i^T = 0, \forall i$$

- further, if $\mathbf{x} \not\in \mathbb{F}_2^n$, but $\mathbf{x} \not\in C$, then $\mathbf{x} \mathbf{h}_i^T \neq 0, \forall i$
• we may put this in a more compact matrix form by defining a so-called parity-check matrix $H$.

$$H = \begin{bmatrix} - \bar{h}_0 & - \\ - \bar{h}_1 & - \\ \vdots & \vdots \\ - \bar{h}_{n-k-1} & - \end{bmatrix}_{(n-k) \times n},$$

so that

$$\epsilon H^T = \vec{0}$$

if and only if $\epsilon \in C$.

• suppose $\epsilon$ has $w$ 1’s (i.e., the Hamming weight of $\epsilon$, $W_H(\epsilon) = w$) and the locations of those 1’s are $P_1, P_2, \ldots, P_w$.

• then the computation $\epsilon H^T = \vec{0}$ effectively adds $W$ rows of $H^T$, rows $P_1, P_2, \ldots, P_w$, to obtain the vector $\vec{0}$; one important consequence of this fact is that the minimum distance $d_{\min} (= \min \text{ weight } W_{\min})$ of $C$ is exactly the minimum number of rows of $H^T$ which can be added together to obtain $\vec{0}$.

---

**Low-Density Parity-Check Codes**

• note that the parity-check matrix $H$ is so called because it performs $m := n-k$ separate parity checks on a received word $y = \epsilon + \vec{c}$.

**Example** with $H^T$ as given above, the $n-k=3$ parity checks implied by $y H^T = \vec{0}$ are

$$
\begin{align*}
0 & + y_1 + y_2 + y_4 \equiv 0 \\
0 & + y_1 + y_3 + y_5 \equiv 0 \\
0 & + y_2 + y_3 + y_6 \equiv 0
\end{align*}
$$

• a low-density parity-check (LDPC) code is a linear block code for which the parity-check matrix $H$ has a low density of 1’s.

---

**Example (7,4) Hamming Code**

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

• we can see that no two rows sum to $\vec{0}$, but row 0 + row 1 + row 6 = $\vec{0}$

$$\Rightarrow d_{\min} = 3$$

---

**Definition** a regular $(n,k)$ LDPC code is a linear block code whose parity-check matrix $H$ contains exactly $W_r = W_c (n/m)$ 1’s per row, where $W_c << m$.

**Remarks**

• note multiplying both sides of $W_c << m$ by $n/m$ implies $W_r << n$.

• the code rate $r = k/n$ can be computed from

$$r = \frac{W_c}{W_r} = 1 - \frac{W_c}{W_r}$$

• $W_c \geq 3$ is a necessity for good codes (Gallager) if $H$ is low density, but if the number of 1’s per column or row is not constant, the code is an irregular LDPC code.

• LDPC codes were invented by Robert Gallager of MIT in his PhD dissertation (1960). They received virtually no attention from the coding community until the mid-1990’s.

---
• note from the result developed earlier,

\[ d_{\text{min}} = \min \{ W_H (\tau), \tau \neq 0: \tau H^T = \mathbf{0} \} \]

we should expect reasonably designed LDPC codes to have large \( d_{\text{min}} \)

• this is because the operation \( \tau H^T \) adds selected rows of \( H^T \) (columns of \( H \)) and it would take a large number of such columns to sum to \( \mathbf{0} \) if \( H \) is sparsely populated with 1’s.

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Example \( W_c = 3 \)

• note any two columns have an overlap of at most one 1; also the sparse property allows us to minimize such overlap

• a consequence of this is that the sum of the columns shown is nonzero

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

• the two classes of nodes in a Tanner graph are the (code bit nodes) or variable nodes and the check nodes (or function nodes)

• the Tanner graph of a code is drawn according to the following rule:

  check node \( j \) is connected to bit node \( i \)

  whenever element \( H_{ji} \) in \( H \) is a 1

• one may deduce from this that there are \( m = n-k \) check nodes and \( n \) bit nodes

• further, the \( m \) rows of \( H \) specify the \( m \) check node connections, and the \( n \) columns of \( H \) specify the \( n \) bit node connections

\[
\begin{bmatrix}
\end{bmatrix}
\]

Representation of Linear Block Codes via Tanner Graphs

• one of the very few researchers who studied LDPC codes prior to the recent resurgence is Michael Tanner of UC Santa Cruz

• Tanner considered LDPC codes (and a generalization) and showed how they may be represented effectively by a so-called bipartite graph, now call a Tanner graph

Definition a bipartite graph is a graph (nodes or vertices connected by undirected edges) whose nodes may be separated into two classes, and where edges may only connect two nodes not residing in the same class
Example  (10, 5) block code with $W_c = 2$ and $W_r = W_c(n/m) = 4$.

\[
H = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}
\]

- observe that nodes $c_0$, $c_1$, $c_2$, and $c_3$ are connected to node $f_0$ in accordance with the fact that in the first row of $H$, $h_{00} = h_{01} = h_{02} = h_{03} = 1$ (all others equal zero)

- observe that the first row and first column of $H$ are assigned an index of 0

- observe an analogous situation for $f_1$, $f_2$, $f_3$, and $f_4$.

- thus, as follows from the fact that $cH^T = \tilde{u}$, the bit values connected to the same check node must sum to zero

---

**Definition** a cycle of length $l$ in a Tanner graph is a path comprising $l$ edges which closes back on itself

- the Tanner graph in the above example possesses a length-6 cycle as made evident by the 6 bold edges in the figure

**Definition** the girth of a Tanner graph is the minimum cycle length of the graph

- the shortest possible cycle in a bipartite graph is clearly a length-4 cycle

- length-4 cycles manifest themselves in the $H$ matrix as four 1’s that lie on the corners of a submatrix of $H$:

\[
H = \begin{bmatrix}
\cdots & a & b \\
1 & 1 & \cdots \\
s & 1 & 1 \\
\cdots & \cdots & \cdots \\
\end{bmatrix}
\]

- length-6 cycles are not quite as easily found in an $H$ matrix:

\[
H = \begin{bmatrix}
r & 1 & 1 \\
1 & 1 & 1 \\
s & 1 & 1 \\
\cdots & \cdots & \cdots \\
\end{bmatrix}
\]

- we are interested in cycles, particularly short cycles, because they have a negative impact on the decoding algorithm for LDPC codes as will be made evident below
**Encoding**

- As indicated above, once $H$ is generated, it may be put in the form $\tilde{H} = [\bar{F}^T : I]$ from which the systematic form of the generator matrix is obtained:

$$G = [I : P]$$

- Encoding is performed via

$$e = \pi G = [\pi : \pi P],$$

although this is more complex than it appears for capacity-approaching LDPC codes (n large)

**Example** Consider a (10000, 5000) linear block code. Then $G = [I : P]$ is $5000 \times 10000$ and $P$ is $5000 \times 5000$. We may assume that the density of ones in $P$ is $\sim 0.5$.

$$\Rightarrow \text{there are } \sim 0.5(5000)^2 = 12.5 \times 10^6 \text{ ones in } P$$

$$\Rightarrow \sim 12.5 \times 10^6 \text{ addition (XOR) operations are required to encode one codeword}$$

----------

**Selected Results**

- We present here selected performance curves from the literature to demonstrate the efficacy of LDPC codes

- The papers from which these plots were taken are listed in the reference section at the end of the note set

- We indicate the paper each plot is taken from to ensure proper credit is given

**MacKay (March 1999, Trans IT)**

- MacKay (and others) re-invented LDPC codes in the late 1990’s

- Here are selected figures from his paper (see his paper for code construction details; his codes are regular or nearly regular)

![Fig. 11. Short-blocklength Gallager codes' performance over Gaussian channel (solid curve) compared with that of standard textbook codes (dashed curve). Vertical axis shows empirical bit error probability. It should be emphasized that all the block errors in the experiments with Gallager codes were observed errors; the decoding algorithm reported the fact that it had failed. Textbook codes as in Fig. 9. Gallager codes: From left to right the codes had the following parameters (N, T, E): (1500, 104, 0.5) (Concentric (A), (504, 251, 0.5) (1A).](image)
Irregular LDPC Codes

- our discussions above favored regular LDPC codes for their simplicity, although we gave examples of irregular LDPC codes
- recall an LDPC code is irregular if the number of 1’s per column of $H$ and/or the number of 1’s per row of $H$ is allowed to vary
- in terms of the Tanner graph, this means that the bit node degree and/or the check node degree is allowed to vary (the degree of a node is the number of edges connected to it)
- a number of researchers have examined the optimal degree distribution among nodes:
  - Luby, et al., Trans. IT, February 2001
  - Richardson, et al., Trans. IT, February 2001

Richardson et al. (cont’d)

- plot below: turbo codes (dashed) and irregular LDPC codes (solid); for block lengths of $n=10^4, 10^5$, and $10^6$; all rates are $\frac{1}{2}$
Chung et al. Irregular LDPC Code

- the plot below is of two separate ½ (10^7, 12/10^7) irregular LDPC codes

Kou et al. LDPC Codes (GlobeCom 2000, a ISO Trans. IT 2001)

- various LDPC codes based on Euclidean geometries (EG) and Projective geometries (PG)

Figure 4: Bit-error probabilities of the (255, 175) EG-LDPC code, (273,191) PG-LDPC code and two computed searched (273,191) Gallager codes.

Kou et al. (cont'd)

Figure 8: Bit- and block-error probabilities of the (16383,14179) EG-LDPC code.

Kou et al. (cont’d)

Figure 10: Bit- and block-error probabilities of the (65520,61425) EG-LDPC code based on IDBP.