A Universal Theory of Pseudocodewords

Judy L. Walker
University of Nebraska

Joint work with
Lance C. Pérez
Nate Axvig, Emily Price, Eric Psota, Deanna Turk
A binary linear code of length $n$ and dimension $k$ is a $k$-dimensional subspace $C$ of $\mathbb{F}_2^n$.

\[
\begin{align*}
\text{Information} & \quad \xrightarrow{\text{Encoder}} \quad \text{Codeword} \\
\ i \in \mathbb{F}_2^k & \quad \xrightarrow{\text{Channel}} \quad y = x + e \in \mathbb{F}_2^n \\
\text{Received Word} & \quad \xrightarrow{\text{Decoder}} \quad \hat{x} \in C \subset \mathbb{F}_2^n \\
\ y = x + e & \quad \xrightarrow{\text{Un-encoder}} \quad \hat{i} \in \mathbb{F}_2^k \\
\end{align*}
\]
We can describe any binary linear code \( C \) as the kernel of a matrix \( H \), called a \textit{parity check matrix} for \( C \):

\[
C = \{ x \in \mathbb{F}_2^n \mid Hx^\top = 0 \}.
\]

**Note:** Given \( C \), there are many choices for \( H \). Some are “better” than others. Everything we are about to say depends on picking a specific \( H \).
Let $H = (h_{ji})$ be an $r \times n$ matrix of 0’s and 1’s. To $H$, we associate the bipartite Tanner graph $T$:

- **left vertices** $\leftrightarrow$ columns of $H$
  - $n$ “bit nodes” $X_1, \ldots, X_n$

- **right vertices** $\leftrightarrow$ rows of $H$
  - $r$ “check nodes” $f_1, \ldots, f_r$

- **edges** $\leftrightarrow$ 1’s in $H$
  - $\{X_i, f_j\}$ is an edge $\iff h_{ji} = 1$
  $\iff$ the $i^{th}$ bit is involved in the $j^{th}$ check
Remark

Note that the Tanner graph $T$ records the matrix $H$, and hence the code $C$, graphically:

- a binary assignment of the bit nodes $(c_1, \ldots, c_n)$ is a codeword in $C$ if and only if
- the binary sum of the values at the neighbors of each check node is zero.
If $H$ (and hence $T$) is sparse, then $C$ is called a low density parity check (LDPC) code.

**Recall:** Given $C$, there are many choices for $H$. Some are better than others.

LDPC codes come equipped with an iterative message-passing decoding algorithm which is extremely efficient and corrects, with high probability, many more error patterns than guaranteed by the minimum distance.
Intuition

- Received word $\Rightarrow$ assignments to the bit nodes
- Bit nodes broadcast to check nodes.
- Check nodes make estimates based on what they receive.
- Check nodes broadcast back to bit nodes.
- Bit nodes make estimates based on what they receive.
- Repeat.
Intuition II

The decoding algorithm acts *locally*: at each stage, decisions are made at each vertex, based only on information coming from its neighbors.

The algorithm *cannot distinguish* between the original Tanner graph and any finite, unramified cover of the Tanner graph.

*Every* codeword in *every* code $\tilde{C}$ corresponding to *every* cover $\tilde{T}$ of the Tanner graph is competing with the codewords in $C$ to be the best explanation of the received word.
Definition: Let \( \tilde{c} = (c_{(1,1)} : \cdots : c_{(1,M)}, \cdots, c_{(n,1)} : \cdots : c_{(n,M)}) \) be a codeword in the code corresponding to some finite cover of \( T \). The \textit{(graph cover) pseudocodeword} associated to \( \tilde{c} \) is the vector

\[
p = p(\tilde{c}) = (p_1, \ldots, p_n) \in \mathbb{N}^n
\]

with

\[
p_i = \# \{ k | c_{(i,k)} \neq 0 \}.
\]
If every variable node of $T$ has degree 2, we call $C$ a cycle code. In this case, there is an associated normal graph $N$ and we have:

**Theorem** (Koetter/Li/Vontobel/Walker, 2004)

Let $C$ be a cycle code of length $n$ with normal graph $N$. Then $(p_1, \ldots, p_n)$ is a pseudocodeword if and only if $u_1^{p_1} \cdots u_n^{p_n}$ occurs in the power series expansion of $\zeta_N(u_1, \ldots, u_n)$ with nonzero coefficient, where $\zeta_N$ is the edge zeta function of $N$.

There is a more general version of this theorem too.
**Definition:** Let $H = (h_{ji})$ be an $r \times n$ matrix of 0’s and 1’s. The *fundamental cone* of $H$ is the subset $\mathcal{K}(H) \subset \mathbb{R}^n$ given by

$$\left\{ (v_1, \ldots, v_n) \mid \sum_{i \neq i'} h_{ji} v_i' \geq h_{ji} v_i \text{ for all } i, j \right\}$$

**Theorem:** (Koetter/Li/Vontobel/Walker, 2004)
Let $p = (p_1, \ldots, p_n) \in \mathbb{N}^n$. The following are equivalent:
- $p$ is a pseudocodeword.
- $p \in \mathcal{K}(H)$ and $Hp^t = 0 \in \mathbb{F}_2^r$. 

A Universal Theory of Pseudocodewords – p. 11/2
Linear Programming (Feldman)

- Associated to $H$ is a fundamental polytope $\mathcal{P}(H) \subseteq [0, 1]^n$.
- Associated to the channel output is a cost function.
- Linear programming decoding minimizes the cost function on $\mathcal{P}(H)$.
  - Every codeword is a vertex of $\mathcal{P}(H)$.
  - There are other vertices too $\implies$ linear programming pseudocodewords.
Connections I (Vontobel and Koetter)

- The conic hull of $\mathcal{P}(H)$ is $\mathcal{K}(H)$.
- Points in the polytope $\mathcal{P}(H)$ are the same as the (normalized) graph cover pseudocodewords.
  - The required cover of $T$ might be disconnected.
**Graph Cover Decoding**

**Definition:** *Graph Cover Decoding* is given by the following decision rule:

If the vector $y$ is received, then among all vectors $\tilde{x}$ which are codewords in codes corresponding to covers (of arbitrary degree $M$) of $T$, choose the one which maximizes the quantity

$$\frac{1}{M} \log \prod_{i=1}^{n} \prod_{m=1}^{M} P_{Y|X}(y_i|x_{i,m}),$$

where $P_{Y|X}(y|x)$ is the channel law, and return the normalized pseudocodeword corresponding to $\tilde{x}$. 
Vontobel and Koetter proved:

**Theorem:** *Graph Cover* Decoding and *Linear Programming* Decoding are equivalent. That is, given a received vector $y$, *Graph Cover* Decoding and *Linear Programming* Decoding always return the same vector of rational numbers between 0 and 1.
Vontobel and Koetter proved:

**Theorem:** *Graph Cover* Decoding and *Linear Programming* Decoding are equivalent. That is, given a received vector $y$, *Graph Cover* Decoding and *Linear Programming* Decoding always return the same vector of rational numbers between 0 and 1.

**Beautiful!**
Reality Check

All of this is based on our intuition that, because iterative message passing decoding acts locally on the Tanner graph $T$, it simultaneously sees all finite covers of $T$.

*Think this through* . . .

If iterative message passing decoders are approximations of Graph Cover Decoding, then
Reality Check (con’t)

- Graph Cover Decoding (i.e., Linear Programming Decoding) *should always outperform* any iterative message passing decoder.

- The decoding errors (at least the majority of them) in iterative message passing decoding *should be due to* graph cover pseudocodewords.

*Neither of these things is true!*
A Simulation Result

Probability of Error for LP, SP, and MS decoding of Turbo Code

- Word Error LP
- Word Error SP
- Word Error MS
- Bit Error LP
- Bit Error SP
- Bit Error MS

SNR $E_b/N_0$ (dB)
Back to Basics (Wiberg)

- Any iterative message passing algorithm on a Tanner graph can be viewed as actually acting on a computation tree built from the graph. The depth of the tree depends on the number of iterations one wants to perform.

- For the Min-Sum and Sum-Product algorithms, Wiberg gives explicit cost functions, determined by channel output. Performing these algorithms is equivalent to finding the configuration on the computation tree which has lowest cost.
Back to Basics (Wiberg, con’t)

- Every codeword induces a configuration on each computation tree.
- There are configurations other than those induced by codewords \(\Rightarrow\) computation tree pseudocodewords.
Where to go from here?

Observation: The “infinite computation tree” is the universal cover of the Tanner graph.

Observation: Every graph cover pseudocodeword induces a configuration on the universal cover.

Observation: Every configuration on the universal cover truncates to a configuration on every computation tree, and every configuration on every computation tree can be obtained in this way.
Normalized Cost

Restrict to the Min-Sum case. Wiberg’s computation tree cost function is the sum of local costs, over all variable nodes in the computation tree. The number of variable nodes grows exponentially with the size of the tree.

**Definition:** Let $c$ be a configuration on a computation tree with $N$ variable nodes. The *normalized cost* of $c$ is $\frac{1}{N} G(c)$, where $G(c)$ is the value of Wiberg’s cost function at the configuration $c$. 
Cost on the Universal Cover

**Definition:** Let $c$ be a configuration on the universal cover and, for $D \geq 1$, let $c^{(D)}$ be the truncation of $c$ to the depth-$D$ computation tree. Let $G_D(c)$ be the normalized cost of $c^{(D)}$. Declare the *cost* of $c$ to be

$$G(c) := \lim_{D \to \infty} G_D(c)$$

provided that this limit exists. Otherwise, the cost of $c$ is undefined.
Universal Cover Decoding

**Conjecture:** Cost is often defined. In particular, if $c$ is induced by a graph cover pseudocodeword, then $G(c)$ exists.

**Definition:** *Universal Cover Decoding* is given by the following decision rule:

Among all configurations $c$ on the universal cover such that $G(c)$ exists, choose the one of minimal cost.
Connections III

- It follows from the work of Wiberg that Min-Sum is an approximation of Universal Cover Decoding, since Min-Sum finds the minimal cost configuration on a finite truncation of the universal cover.

- Graph Cover Decoding (and hence Linear Programming Decoding) is an approximation of Universal Cover Decoding, since every finite (connected) cover of $T$ is a surjective image of the universal cover of $T$. 
Connections III (con’t)

• Computation tree pseudocodewords and graph cover pseudocodewords are related, but usually not directly.
  • Kelley and Sridhara introduced the notion of *locally consistent* computation tree pseudocodewords and showed they could be realized as graph cover pseudocodewords.