6.454 Pseudocodewords Presentation

Todd P. Coleman

September 23, 2003

1 Decoding Binary Linear Codes Defined on Graphs

An (n, k, d) binary linear block code can be defined in terms of the set of elements of F_2^n that satisfy a set of constraints. One way to express this is to express this in terms of an $n - k$ by n parity-check matrix H:

$$\mathcal{C} = \{c \in \{0,1\}^n \;\; s.t. \;\; Hx = 0\}$$

Graphical representations denote the dependencies between codewords based upon the constraints they must satisfy. Each constraint corresponds to a (smaller) linear code.

When transmitting across a memoryless channel, the receiver observes measurement y_i at each node i. A sufficient statistic for decoding is $\gamma_i = \ln\left(\frac{P(y_i|c_i=0)}{P(y_i|c_i=1)}\right)$ for each i. We may describe

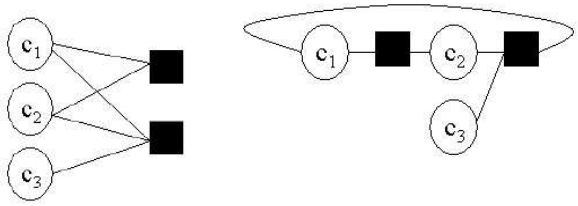

Figure 1: An example of a graph G corresponding to a blbc. Squares denote parity checks, circles denote code symbols

ML-decoding as follows:

$$\begin{aligned}
x^* &= \arg\max_{c\in\mathcal{C}} \left(\prod_{i=1}^{n} P(y_i|c_i) \right) \\
&= \arg\min_{c\in\mathcal{C}} \left(-\ln \prod_{i=1}^{n} P(y_i|c_i) \right) \\
&= \arg\min_{c\in\mathcal{C}} \left(-\sum_{i=1}^{n} \ln P(y_i|c_i) \right) \\
&= \arg\min_{c\in\mathcal{C}} \left(\sum_{i=1}^{n} \ln P(y_i|0) - \ln P(y_i|c_i) \right) \\
&= \arg\min_{c\in\mathcal{C}} \left(\sum_{i:y_i=1} \ln \left(\frac{P(y_i|0)}{P(y_i|1)} \right) \right) \\
&= \arg\min_{c\in\mathcal{C}} \left(\sum_{i=1}^{n} \gamma_i c_i \right)
\end{aligned}$$

Thus we see that ML decoding can be thought of as a combinatorial optimization problem where the objective function is linear.

Note that we can cast such an optimization problem as a linear program:

$$x^* = \arg \min_{c\in CH(\mathcal{C})} \left(\sum_{i=1}^{n} \gamma_i c_i \right)$$

However, the polyhedron corresponding to the feasible set is the convex hull of the elements of the code, and it has an exponential number of vertices.

2 Feldman LP feasible set formulation [Fel03]:

Feldman's LP formulation operates on the graph G corresponding to the parity-check representation of the code. Note that each parity check node j involved in $\delta_j = N(j)$ variable nodes defines a $(\delta_j, \delta_j - 1, 2)$ code $\mathcal{C}_j$. Each of the $2^{\delta_j - 1}$ codewords correspond to a different set S (called a *configuration*) of bits that has a parity sum to 0. We'll call the set of all possible sets $E(\mathcal{C}_j)$. Note that it's always the case that $\emptyset \in E(\mathcal{C}_j)$. So we may define a configuration variable $x_j = \{x_{j,S_1}, x_{j,S_2} \ldots, \}$ that must satisfy $0 \leq x_{j,S} \leq 1$ and also, because for any codeword realization, only one realization is on, they must satisfy

$$\sum_{S\in E(\mathcal{C}_j)} x_{j,S} = 1.$$

Finally, we must impose the constraint that for each variable i, symbol value c_i must be consistent with the selected configuration for each parity check:

$$c_i = \sum_{s\in E(\mathcal{C}_j),\ S\ni i} x_{j,S} \quad \forall i \in N(j).$$

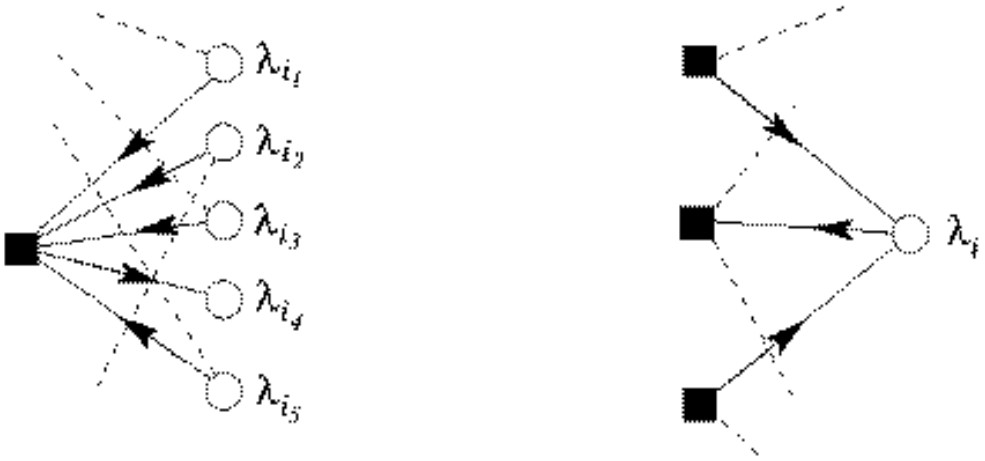

Figure 2: locally operating algorithms

resulting feasible set $\mathcal{P}^{\text{Feldman}}(G)$:

$$\begin{aligned}
\forall j \in \{1,\ldots,n-k\} : \mathcal{P}_j^{\text{Feldman}}(G) &= \Big\{ c \in \mathbb{R}^n s.t. : \\
&\quad x_j \in \mathbb{R}^{2^{\delta_j - 1}} s.t. : \\
&\quad \sum_{S \in E(\mathcal{C}_j)} x_{j,S} = 1, \\
&\quad 0 \le x_{j,S} \le 1 \ \ \forall S \in E(\mathcal{C}_j), \\
&\quad c_i = \sum_{s \in E(\mathcal{C}_j),\, S \ni i} x_{j,S} \ \ \forall i \in N(j) \Big\} \\
\mathcal{P}^{\text{Feldman}}(G) &= \bigcap_{j=1}^{n-k} \mathcal{P}_j^{\text{Feldman}}(G)
\end{aligned}$$

This formulation exhibits ML-certificate property: if a solution is integral, guaranteed to be ML. Otherwise, a fractional solution is spit out, which denotes a failure.

2.1 Iterative decoding

For any graphical realization $G = (V, E)$ of a code $\mathcal{C}$, iterative decoding algorithms (such as sum-product and min-sum) have been introduced to perform approximate optimal decoding. These algorithms iteratively perform computations locally at variable and constraint code nodes. When applied to acyclic graphical realizations, the min-sum algorithm corresponds to ML decoding. Decoding is not guaranteed to be optimal when graph has cycles. In practice, however, it has been observed to work extremely well.

For any fixed decoding schedule S, and a graphical realization G, we can define a *computation tree* $\overline{G}_{k,r,S}$ associated with vertex k after r iterations by constructing a tree with root node $\overline{c}_k$

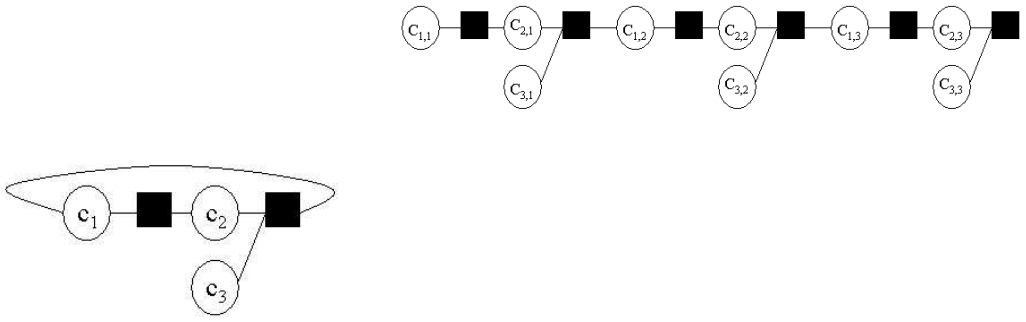

Figure 3: Tail-biting representation (left) and computation tree (right) associated with vertex c_1 for 3 cycles of decoding

corresponding to c_k in G and then recursively adding edges and leaf nodes in $\overline{G}_{k,r,S}$ corresponding to messages passed in the iterative decoder. We see that $\overline{G}_{k,r,S}$ is a graphical realization for some new code $\overline{\mathcal{C}}_{k,r,S}$.

Analyzing the computation tree is beneficial because

a) any local iterative algorithm utilizing schedule S cannot distinguish between operating on G or $\overline{G}_{k,r,S}$ when the observed LLRs are replicated at repeated code symbol nodes.

b) the computation tree is acyclic so we know that local iterative algorithms mentioned earlier are performing optimal decoding on it.

Thus we see that the suboptimality of iterative algorithms on codes with cycles corresponds to 'pseudo-codewords' that compete with true codewords in the decoding algorithm.

Every vertex in G corresponds to a set (possibly larger than 1) of vertices in $\overline{G}_{k,r,S}$. For example, we see that for the tail-biting trellis [FKKR01, BJFV01] representation corresponding to graph G (see figure 3), a computation tree $\overline{G}_{k,r,S}$ can be obtained by unwrapping G into a conventional trellis on an ordered time axis. In this case, we see that for a graph G with $|V|$ vertices, after r full cycles, the computation tree $\overline{G}_{k,r,S}$ has $r|V|$ vertices, and each vertex is repeated r times. Every codeword in C has a representation in $\overline{\mathcal{C}}_{k,r,S}$ (simply repeat the codeword r times). In figure 3, we see that

$$(110) \in C \Rightarrow (110\ 110\ 110) \in \overline{\mathcal{C}}_{1,3,S}.$$

However, there are codewords in $\overline{\mathcal{C}}_{1,3,S}$ such as (110 111 000) that do not correspond to codewords in C. The universal cover $\mathcal{U}(G,S)$ corresponds to the codewords of the computation tree $\overline{G}_{k,r,S}$ for each k with schedule S for infinitely many iterations:

$$\mathcal{U}(G,S) = \bigcup_{r=1}^{\infty} \bigcup_{k=1}^{|V|} \overline{\mathcal{C}}_{k,r,S}$$

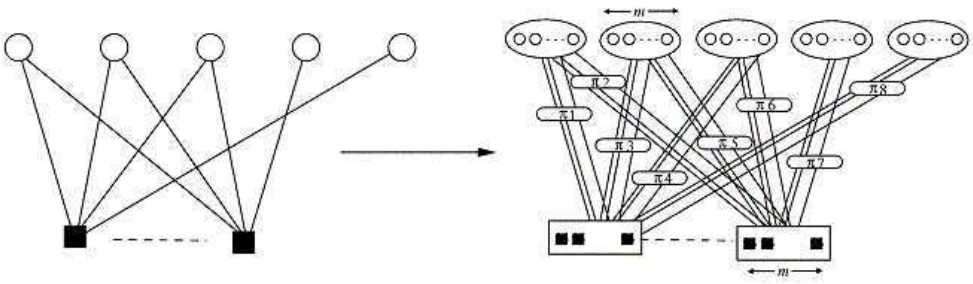

Figure 4: An example of an m-cover

For graphs with more than one cycle, characterizing this exactly can become cumbersome [BJFV01].

2.2 Finite-degree covers [KV03]:

We may now generalize the idea of pseudocodewords that evolve from computation trees. However, a subset of the universal cover can be characterized analytically by the use of graph covers.

A graph cover is an m-fold replication of the original graph where edges connecting vertices preserve the graphical structure in the original graph with respect to each individual node. Formal definition:

Definition A finite-degree m cover of $G = (V, E)$ is a graph $\hat{G}$ with vertex set $\hat{V} = \cup_{i=0}^{l-1}$ where each set $\hat{V}_i = \{\hat{v}_{i,0}, \hat{v}_{i,1}, \ldots, \hat{v}_{i,m-1}\}$ contains exactly m vertices. The edge set $\hat{E}$ of $\hat{G}$ is chosen so that for each vertex $\hat{v}_{i,s} \in \hat{V}$, $|N(\hat{v}_{i,s})| = |N(v_i)|$ and $N(\hat{v}_{i,s})$ contains exactly one vertex $\hat{v}_{j,r}$ for all j such that $v_j \in N(v_i)$.

If G is a parity-check graph representation for the code $\mathcal{C}$, then a degree-m cover $\hat{G}$ is a parity-check graph representation for the code $\hat{\mathcal{C}}$ of length mn. Vertices in $\hat{V}_i$ are denoted as

$\hat{x}_{i,0}, \hat{x}_{i,1}, \ldots, \hat{x}_{i,m-1}$ for lifted variable nodes x_i and likewise $\hat{f}_{i,0}, \hat{f}_{i,1}, \ldots, \hat{f}_{i,m-1}$ denote the same for parity-check nodes f_i. Likelihoods of replicated nodes are replicated. Note that by the definition of a graph cover, any original codeword, after being repeated m times, is also a codeword in $\hat{\mathcal{C}}$. We see that any codeword $\hat{c}$ in an m-cover contributes a log-likelihood cost of

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\gamma_i\hat{c}_{i,j} \;=\; m\sum_{i=1}^{n}\gamma_i\hat{c}_{i,j}\frac{|\{l:\hat{c}_{i,l}=1\}|}{m}$$

This motivates the following definition of pseudocodewords:

Definition A *pseudocodeword* $w(\hat{c}) \in [0,1]^n$ corresponds to a codeword $\hat{c} \in \hat{\mathcal{C}}$ for some m that reflects the effect on the likelihood ratio in the objective function:

$$\omega_i(\hat{c}) \triangleq \frac{|\{l:\hat{c}_{i,l}=1\}|}{m}$$

it is the fraction of times a variable node in $\hat{\mathcal{C}}$ takes on the value 1. We have that $\omega(\hat{c}) = (\omega_1(\hat{c}), \omega_i(\hat{c}), \ldots, \omega_n(\hat{c}))$.

Note that a pseudocodeword $\omega(\overline{c})$ can be analagously defined for elements $\overline{c}$ of $\overline{G}_{k,r,S}$ in the previous section.

2.3 Koetter,Vontobel's fundamental polytope construction [KV03]

At parity-check j, start with the parity-check code C_j which is connected to $\delta_j = |N(j)|$ variable nodes. This is a $(\delta_j, \delta_j - 1, 2)$ binary linear code with $2^{\delta_j - 1}$ codewords. The degree-m cover of this graph is simply an m-fold copy of the original graph. Any individual copy of the code can support a codeword in C_j so it follows that the set of pseudocodewords $\omega(\hat{c})$ is given by

$$\left\{\frac{\sum_{i=1}^{m} c^{(i)}}{m} \; s.t. \; c^{(i)} \in C_j\right\}$$

We can then define a 2^{δ_j-1} x δ_j matrix P_{δ_j} which contains for each row a codeword of C_j. It follows that after taking the closure of the union of possible pseudocodewords for all m, the *fundamental polytope corresponding to check* j $\mathcal{P}_j^{\text{Koetter}}(\mathcal{C}_j)$ is given by

$$\mathcal{P}_j^{\text{Koetter}}(\mathcal{C}_j) = \left\{\omega \in \mathbb{R}^{\delta_j} \; s.t. \; \omega = xP_{\delta_j}, x \in \mathbb{R}^{2^{\delta_j-1}}, 0 \le x_i \le 1, \sum_i x_i = 1\right\}$$

If we define $\omega_{|V}$ to be the restriction of ω to the coordinates in V then it follows that the fundamental polytope $\mathcal{P}^{\text{Koetter}}(G)$ is given by

$$\mathcal{P}^{\text{Koetter}}(G) = \left\{\omega \in \mathbb{R}^n \; s.t. \; \omega_{|N(j)} \in \mathcal{P}_j^{\text{Koetter}}(\mathcal{C}_j) \;\; j = 1, \ldots, n-k\right\}.$$

Observing carefully, we see that $\mathcal{P}^{\mathbf{Koetter}}(G) = \mathcal{P}^{\mathbf{Feldman}}(G)$:

$$\begin{aligned}
\mathcal{P}^{\text{Koetter}}(G) = \Big\{\omega \in \mathbb{R}^n \quad s.t. \quad & \omega_{|N(j)} = x^{(j)} P_{\delta_j} \\
& x^{(j)} \in \mathbb{R}^{2^{\delta_j - 1}}, \\
& 0 \le x_i^{(j)} \le 1, \\
& \sum_i x_i^{(j)} = 1, \\
& j = 1, \ldots, n-k \Big\} \\
\mathcal{P}^{\text{Feldman}}(G) = \Big\{c \in \mathbb{R}^n \; s.t. \quad & x_j \in \mathbb{R}^{2^{\delta_j - 1}} \\
& \sum_{S \in E(\mathcal{C}_j)} x_{j,S} = 1, \\
& c_i = \sum_{s \in E(\mathcal{C}_j), S \ni i} x_{j,S} \;\; \forall i \in N(j) \\
& 0 \le x_{j,S} \le 1 \;\; \forall S \in E(\mathcal{C}_j), \\
& j \in \{1, \ldots, n-k\} \Big\}
\end{aligned}$$

Furthermore, we note that since each $\omega \in \mathcal{P}^{\text{Koetter}}(G)$ satisfies $0 \le \omega_i \le 1$, and any nonempty bounded polyhedron is the convex hull of its extreme points, we see that the pseudocodewords of interest are the extreme points of $\mathcal{P}^{\text{Koetter}}(G)$.

It follows that a vector of log-likelihoods γ and its lifting $\hat{\gamma}$ satisfy, for any two codewords $\hat{c}$ and $\hat{c}'$:

$$P(\hat{c}|\hat{\gamma}) > P(\hat{c}'|\hat{\gamma}) \Leftrightarrow \langle \omega(\hat{c}), \gamma \rangle < \langle \omega(\hat{c}'), \gamma \rangle.$$

If we assume the 0 vector was transmitted, then the *decision region* is separated by the hyperplane

$$\langle \omega(\hat{c}), \gamma \rangle = 0.$$

Definition For a particular channel, the distance from the transmitted signal point to this hyperplane is denoted as the *pseudo-distance* or *pseudo-weight*. The *pseudo-weight* along an AWGN channel is given by [FKKR01]:

$$w_{AWGN}(\omega) = \left(\frac{\|\omega\|_1}{\|\omega\|_2} \right)^2.$$

Note that if c is a codeword with Hamming weight $\omega_H(c)$, then $\omega = c$ and $w_{AWGN}(c) = w_H(c)$. For an AWGN channel with antipodal signaling, the squared Euclidean distance to the boundary $\langle w(\hat{c}), \gamma \rangle = 0$ from the signal point 1^n is given by $w_{AWGN}(\omega)$. *This motivates trying to characterize the minimum pseudo-weight of all non-zero codewords of finite covers, given by* $w_{AWGN,min}(\mathcal{C})$.

Stepping back for a second, we see from the previous section that for any locally iterative decoding schedule S, we may draw out the corresponding computation tree. If we let the number

of iterations go to infinity you wind up with an infinite graph on which you the algorithms attempt to find the 'best' codeword. The shape of the leaves in this graph may depend on the update schedule. For any fixed schedule, any pseudocodeword in a finite m-cover can be located in the computation tree to give a valid codeword in the tree code (this was pointed out to me by Ralf Koetter in a personal communication). This codeword starts competing with the true transmitted codeword and will influence the decoding decision on the infinite graph. However, the universal cover $\mathcal{U}(G,S)$ may admit pseudocodewords that do not appear as pseudocodewords in the union of all finite covers. We see that the min-sum algorithm when applied to graph G using schedule S is performing the optimization

$$\begin{aligned} \min \quad & \gamma'\omega \\ s.t. \quad & \omega \in cl\{\omega\left(\mathcal{U}\left(G,S\right)\right)\} \end{aligned}$$

where for a set B we mean $\omega(B) = \{\omega(b),\ b \in B\}$ and Koetter's fundamental polytope $\mathcal{P}_j^{\text{Koetter}}(G)$ satisfies, for any schedule S,

$$\mathcal{P}_j^{\text{Koetter}}(G) \subset cl\{\omega\left(\mathcal{U}\left(G,S\right)\right)\}.$$

It is useful to note that across an AWGN channel, the space of LLRs is proportional to the pseudo-weight. However, for different channels, the relation is not necessarily linear. Expressions for the BSC and BEC that have been provided in [FKKR01].

Definition The decision region with respect to 0 is the region $\mathcal{D}_0$ in Euclidean space such that for any $\gamma \in \mathcal{D}_0$, the all-0 word is more likely than any other codeword $\hat{c}$ in any finite cover:

$$\mathcal{D}_0 = \{\gamma \in \mathbb{R}^n :\ \langle \omega, \gamma \rangle > 0\ \forall \omega \in \mathcal{P}_j^{\text{Koetter}}(G)\}.$$

The set of pseudo-codewords is channel-independent and is characterized by the fundamental cone. However, the decision boundaries are channel-dependent. The relation between pseudo-codewords and pseudo-weight is channel-dependent. Since on an AWGN channel, there is a linear relationship between the Euclidean space representation of the received signal y and γ, the decision region for this channel is a polytope. However, for different channels, the decision region is not necessarily a polytope.

Also, it is worthwhile to note that the minimal pseudo-weight can be upper-bounded for *any* (j,k) with $j \geq 3$ regular LDPCs (by a simple computation-tree construction) by a sublinear term in n. This is in contrast to Gallager's PhD results, where he showed that with high probability, linear minimum distance can be attained.

On the BEC: a failure in iterative decoding occurs iff a *stopping set*[DPT$^+$02] exists amongst erased bits.

Definition A stopping set S is a subset of the code bits such that all the checks in the neighborhoods of the each code bit have degree (with respect to S) greater than one.

Stopping sets are special cases of pseudocodewords on the BEC, and Feldman's LP decoding algorithm fail iff a stopping set amongst erased bits exists. The two decoding algorithms exhibit the same convergence results. Thus it appears as if on the binary erasure channel, using the parity-check representation for G, we have that

$$\mathcal{P}^{\text{Koetter}}(G) = \mathcal{P}^{\text{Feldman}}(G) = cl\{w\left(\mathcal{U}\left(G,S\right)\right)\}.$$

Other references: [FWK03]

References

[BJFV01] R. Koetter B. J. Frey and A. Vardy. Signal space characterization of iterative decoding. *IEEE Transactions on Information Theory*, 47(2):766–781, 2001.

[DPT+02] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke. Finite-length analysis of low-density parity-checkcodes on the binary erasure channel. *IEEE Transactions on Information Theory*, 48:15701579, 2002.

[Fel03] J. Feldman. *Decoding Error-Correcting Codes via Linear Programming*. PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2003.

[FKKR01] G. Forney, R. Koetter, J. Kschischang, and A. Reznik. On the effective weights of pseudocodewords for codes defined on graphs with cycles. *Codes, Systems and graphical models*, pages 101–112, 2001.

[FWK03] J. Feldman, M. Wainwright, and D. R. Karger. Using linear programming to decode linear codes. *Proceedings of Conference on Information Sciences and Systems, The John Hopkins University*, March 2003.

[KV03] R. Koetter and P. O. Vontobel. Graph-covers and iterative decoding of finite length codes. *Proceedings of Turbo conference, Brest*, 2003.