ERROR CORRECTING CODES AS BIPARTITE GRAPHS

BURTON ROSENBERG
UNIVERSITY OF MIAMI

Contents

1. Preliminaries

Let (P, D, E) be a bipartite graph, where D are the nodes, P are parity nodes and $E \subseteq P \times D$ the edge set. The data nodes can be marked or unmarked. This is captured by attaching a marking function $\mu : D \to \{\, 0, 1 \,\}$. Only the data nodes are marked, not the parity nodes.

Definition 1.1. Given a bipartite graph (P, D, E), the *neighborhood* $N(p)$ of a node $p \in P$ is defined as,

$$N(p) = \{\, d \in D \mid (p, d) \in E \,\}.$$

Definition 1.2. Given a bipartite graph (P, D, E) and a marking $\mu : D \to \{\, 0, 1 \,\}$, the graph satisfies the *parity constraint* if the number of marked data nodes in the neighborhood of any parity node is even,

$$\forall p \in P \;\; \big|\{\, d \in N(p) \mid \mu(d) = 1 \,\}\big| = 0 \pmod 2$$

There is another way to picture the marking function μ. Suppose $|D| = n$ and fix an order $d_1, d_2, \ldots, d_n$ for the nodes in D. Let $\bar{\mu} \in \mathbb{Z}_2^n$ defined as,

$$\bar{\mu} = \big(\, \mu(d_1), \mu(d_2), \ldots, \mu(d_n) \,\big).$$

Date: December 1, 2023.

Addition is defined for two markings $\bar{\mu}_1$ and $\bar{\mu}_2$ as the bitwise exclusive or of the vectors as vectors in $\mathbb{Z}_2^n$,

$$\bar{\mu}_1 \oplus \bar{\mu}_2 = \big(\bar{\mu}_1(1) \oplus \bar{\mu}_2(1), \bar{\mu}_1(2) \oplus \bar{\mu}_2(2), \ldots, \bar{\mu}_1(n) \oplus \bar{\mu}_2(n) \big)$$

Since exclusive or is a group operating, this set of markings is a group.

To refer in either the marking of nodes in the graph or a vector in $\mathbb{Z}_2^n$, we call either of these a *word*. That is, a word can mean either the set of marked nodes $\mu \subset D$ or the vector of markings $\bar{\mu} \in \mathbb{Z}_2^n$, according to context. Those words that satisfy the parity condition are called *codewords*. The set of all words along with the subset of codewords is called a *code*.

Theorem 1.1. The set of words of a graph is a group and the set of codewords are a subgroup of that group.

Proof: That is, the empty marking $\bar{0} = \big(0, 0, \ldots, 0 \big)$ satisfies the parity condition, and given markings $\bar{\mu}_1$ and $\bar{\mu}_2$, each which satisfy the parity condition, then the sum $\bar{\mu}_1 \oplus \bar{\mu}_2$ satisfies the parity condition.

In the vector representation of a word, each index is called a *bit*, so we can speak of an n bit code. For a *systematic code*, among the n bits, k can be set arbitrarily while the remaining $n - k$ are set to complete the n bits to a unique codeword. These $n - k$ dependent bits are called the *parity bits*. In this case the code is called an (n, k) code.

Definition 1.3. A linear code is a (n, k) code if the set of data nodes is n, $|D| = n$ and the set of codewords is $|C| = 2^k$.

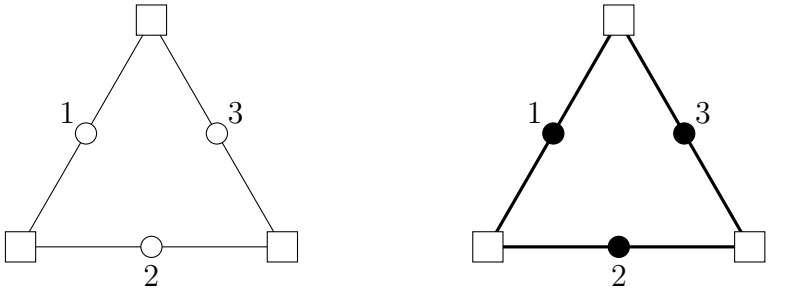
The bipartite graph for the code appears in the literature with the name a *Tanner graph*.

2. Simplex codes

A simple way to construct a linear code to take the frame of a simplex, associate at parity nodes for each vertex and a data node for each edge. Among the markings that satisfy the parity condition are loops — sequences of edges, consecutive on the body of the simplex, and closing simply. Then the algebra of loops is to compose two loops eliminating any edge that is covered twice.

2.1. **A** $(3, 1)$ **simplex code.** The case of a 2-simplex is very simple. A 2-simplex is a triangle. There is only one loop, that made by the entirety of edges. There is also the case of no loop. so the algebra is the two elements of $\mathbb{Z}_2^3$, $0 = (0, 0, 0)$ and $1 = (1, 1, 1)$. Note that,

$$1 \oplus 1 = (1, 1, 1) \oplus (1, 1, 1) = (1 \oplus 1, 1 \oplus 1, 1 \oplus 1) = (0, 0, 0) = 0.$$



(A) $l_0 = (0, 0, 0)$ (B) $l_1 = (1, 1, 1)$

FIGURE 1. The 2-simplex code with all possible codewords

This code is a $(3, 1)$ *repetition code*, in that the codewords are one bit repeated in three positions of the codeword. If as a result of data corruption there occurs an non-codeword, the majority value guides the correction. For instance, if the corruption gives the element $(1, 0, 1)$, the estimate of the correct codeword is $(1, 1, 1)$, as that needs fewer bit corrections than the only alternative, $(0, 0, 0)$.

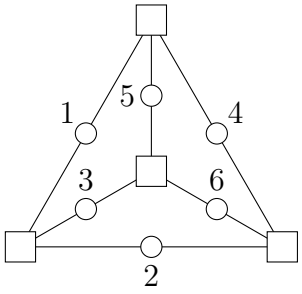
2.2. **A** $(6, 3)$ **simplex code.** A 3-simplex is the simplest space-filling polygon, consisting of 4 vertices and 6 edges. A code is the choosing of a set of edges, the edge for which the node on the edge is marked. For the parity condition, each vertex of the simplex must have an even number of edges selected. Since each vertex as degree 3, either no edges or two edges are selected. If two, then this parity vertex is on a cycle of edges forming a loop. Therefore each codeword is a loop, and all loops are codewords.

Referring to Figure 2 we demonstration all eight codewords. There is the zero codeword,

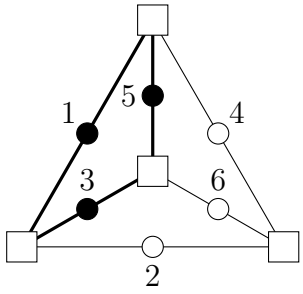
$$l_0 = (0, 0, 0, 0, 0, 0).$$

There are four codewords which are loops of length three,

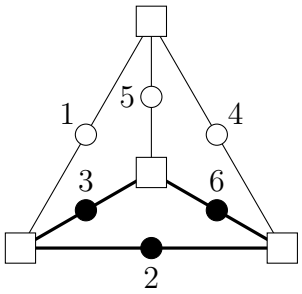
$$\begin{aligned} l_1 &= (1, 0, 1, 0, 1, 0) \\ l_2 &= (0, 1, 1, 0, 0, 1) \\ l_4 &= (0, 0, 0, 1, 1, 1) \\ l_7 &= (1, 1, 0, 1, 0, 0) \end{aligned}$$



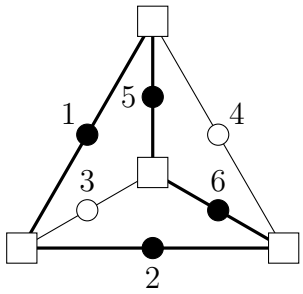
(A) $l_0 = (0,0,0,0,0,0)$



(B) $l_1 = (1,0,1,0,1,0)$



(C) $l_2 = (0,1,1,0,0,1)$



(D) $l_3 = (1,1,0,0,1,1)$

FIGURE 2. The 3-simplex code with l_0, l_1, l_2 and $l_3 = l_1 \oplus l_2$

There are three codewords which are loops of length four,

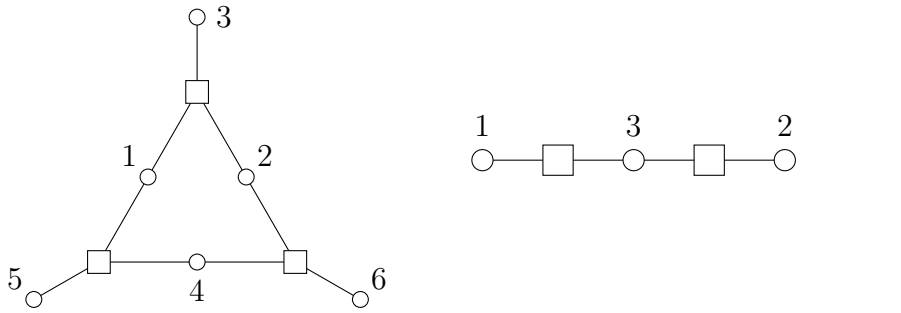
$$\begin{aligned} l_3 &= (1,1,0,0,1,1) \\ l_5 &= (1,0,1,1,0,1) \\ l_6 &= (0,1,1,1,1,0) \end{aligned}$$

Figure 2 illustrates codewords l_0, l_1, l_2 and l_3. Note that,

$$\begin{aligned} l_1 \oplus l_2 &= (1,0,1,0,1,0) \oplus (0,1,1,0,0,1) \\ &= (1 \oplus 0, 0 \oplus 1, 1 \oplus 1, 0 \oplus 0, 1 \oplus 0, 0 \oplus 1) \\ &= (1,1,0,0,1,1) \\ &= l_3. \end{aligned}$$

In general, the addition of loops parallels the addition of the loops' indices, when that addition is understood as a bit-wise exclusive or. Two additional examples are,

$$\begin{aligned} l_7 &= l_1 \oplus l_2 \oplus l_4 \\ l_0 &= l_3 \oplus l_5 \oplus l_6 \end{aligned}$$



(A) Reduced $(6,3)$ simplex code (B) Reduced $(3,1)$ simplex code

FIGURE 3. Reduced simplex codes

Each of the eight loops is the sum of the loops l_1, l_2 and l_3 by expressing i in binary

$$i = b_1 + b_2\, 2 + b_4\, 4$$

and summing,

$$l_i = (b_1\, l_1) \oplus (b_2\, l_2) \oplus (b_4\, l_4)$$

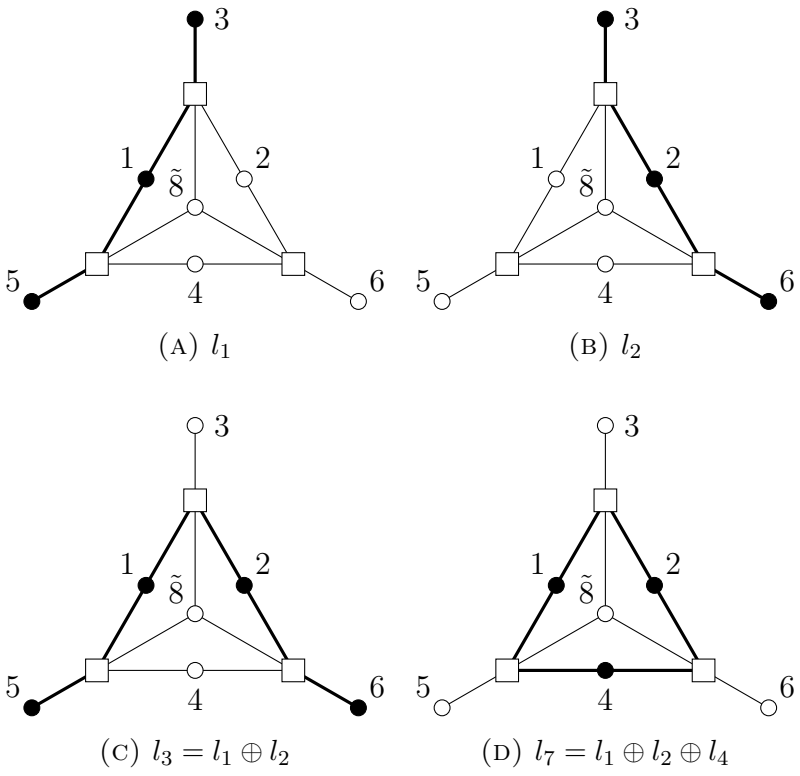
Given a loop l_i its index is computed just from the status of nodes 1, 2 and 4 being in the loop or not.

The marking vector is of dimension 6. We have associated codewords with the free choice of three boolean variables. Therefore this is a $(6,3)$ code. It is a code capable of correcting one bit error (see below for the discussion of error correction).

2.3. **Reduced Simplex Code.** The simplex code has a redundant parity constraint. Therefore we can remove one parity node and still have the same code, as the constraint imposed upon by the removed node is always satisfied. The reduced graphs are shown in figure 3.

Theorem 2.1. In a simplex code, one parity node is redundant. If the parity condition is verified for all but one parity node. it is satisfied for all nodes.

Proof: To simplify the description, we consider the vertices and edges of the simplex. The nodes on the edges are identified with the edge. Given any codeword, remove all edges from the graph other than edges in the codeword. Set apart one vertex v; and summing the edge degree over all all other vertices the parity condition gives an even number. Now remove edges of which nether endpoint is v. This maintains the sum of edges as even, until the only edges left are to v. Therefore the parity condition holds for v.



(A) l_1 (B) l_2

(C) $l_3 = l_1 \oplus l_2$ (D) $l_7 = l_1 \oplus l_2 \oplus l_4$

FIGURE 4. Hamming Code, l_1 through l_7

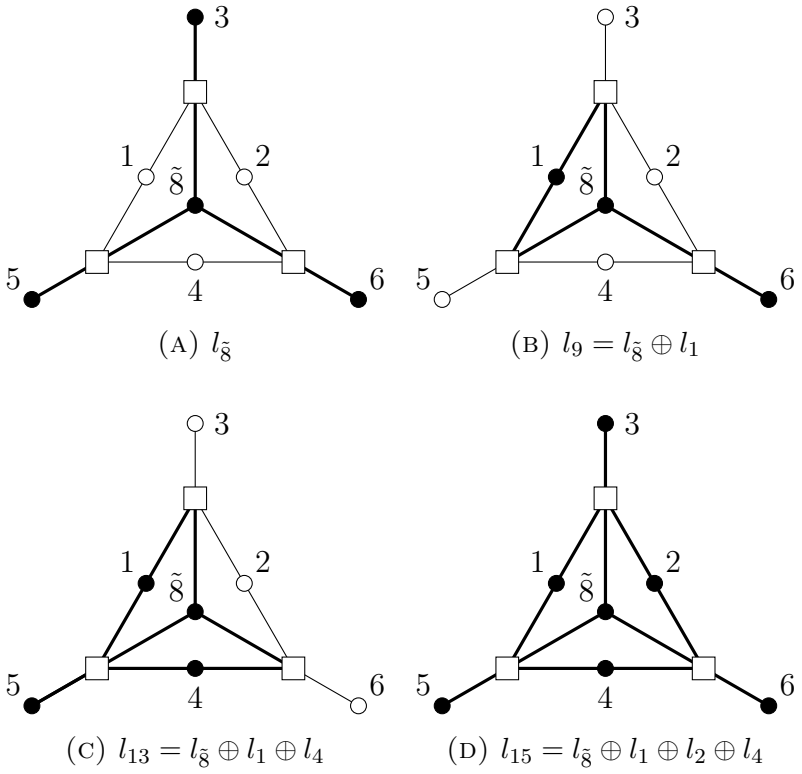
3. The $(7, 4)$ Hamming code

One of the most commonly taught codes in the Hamming Code. As with the simplex code, which is a family of codes depending on the dimension of the simplex, the Hamming code is a family of codes of which the smallest interesting case is the $(7, 4)$ hamming code.

Figures 4 and 5 show the graph, with round data nodes and square parity nodes. The marked data nodes are shown filled and for emphasis the edge on which the node sits is also highlighted. There are 16 possible codewords. We isolate 4 codewords that by adding among them can generate all 16. While there are 7 data nodes, the 7-th is named $\tilde{8}$ as an aid to generating codewords 8 through 15.

- There is the zero codeword

$$l_0 = (0, 0, \ldots, 0)$$



(A) $l_{\tilde{8}}$ (B) $l_9 = l_{\tilde{8}} \oplus l_1$

(C) $l_{13} = l_{\tilde{8}} \oplus l_1 \oplus l_4$ (D) $l_{15} = l_{\tilde{8}} \oplus l_1 \oplus l_2 \oplus l_4$

FIGURE 5. Hamming Code, $l_{\tilde{8}}$ through l_{15}.

- There are four codewords we single out as they generate through addition all the other code words,

$$\begin{aligned} l_1 &= (1,0,1,0,1,0,0) \\ l_2 &= (0,1,1,0,0,1,0) \\ l_4 &= (0,0,0,1,1,1,0) \\ l_{\tilde{8}} &= (0,0,1,0,1,1,0) \end{aligned}$$

The seventh bit is labeled label $\tilde{8}$ for its numerical value in calculating the remaining 11 code words. For code word l_i, express i in binary as,

$$i = b_1 + 2\,b_2 + 4\,b_4 + 8\,b_8$$

then,

$$l_i = (b_1\,l_1) \oplus (b_2\,l_2) \oplus (b_4\,l_4) \oplus (b_8\,l_{\tilde{8}})$$

We now describe the remaining 11 codewords.

- There are three like l_3 (Figure 4c)

(A) $l_0 = (0,0,0)$ (B) $l_1 = (1,1,1)$

FIGURE 6. The $(3,1)$ hamming code

- There is l_7. (Figure 4d)
- There are three like l_9. (Figure 5b)
- There are three like l_{13}. (Figure 5c)
- There is l_{15} (Figure 5d)

3.1. **General Hamming codes.**

Definition 3.1. Let P be a finite set of parity nodes. Let $S = 2^P \setminus \emptyset$. be the set of all non-empty subset of P. Let $D = \{ d_s \,|\, s \in S \}$ be a set of elements indexed by S. Let the edge set E be defined,

$$E = \bigcup_{d_s \in D} \{ (d_s, p) \,|\, p \in s \}$$

The (P, D, E) is the $(n, n-k)$ hamming code, where $n = |D|$ and $k = |P|$.

Example: In the $(7,4)$ code, nodes $3, 5$ and 6 each represented a single element of P, nodes $1, 2$ and 4 represented the 2-element subsets of P, and node $\tilde{8}$ represented the set P.

Example: The smallest hamming code sets $k = 2$. Then

$$\begin{aligned} P &= \{ 1, 2 \} \\ D &= \{ \{1\}, \{2\}, \{1,2\} \} \end{aligned}$$

All the codewords possible are show in figure 6. It is again a repetition code.

Parity and data bits: In all these constructions, the singleton sets are in one to one correspondence with the parity nodes, and are hence free to set the parity correctly after the other nodes have been marked. Hence they are considered *parity bits* and the other nodes are message bits.

Also note that any one bit error will light up a combination of parity bits that is the subset of parity bits represented by that data bit. This makes it possible and easy to correct any one bit error.

The code rate ir s the ration of message bits to parity bits. In the case of this code, the code rate is,

$$r = \frac{n-k}{n} = 1 - \frac{k}{2^k - 1} \sim 1 - \frac{\log n}{n}$$

Hence as n codes to infinity the code rate goes to 1, and practically all the bits a data bits.

4. Message encoding, transmission and correction

Definition 4.1. The *weight* w of a marking $\bar{\mu} \in \mathbb{Z}_2^n$ is the number of 1's in the marking,

$$w(\bar{\mu}) = \left| \{\, i \in [1, n] \,|\, \bar{\mu}_i = 1 \,\} \right|$$

Definition 4.2. The *hamming distance* between two markings $\bar{\mu}_1$ and $\bar{\mu}_2$ is the weight of their difference (which is also the sum),

$$h(\bar{\mu}_1, \bar{\mu}_2) = w(\, \bar{\mu}_1 \oplus \bar{\mu}_2 \,).$$

The hamming distance between the markings is the number of places the markings disagree.

Definition 4.3. Let (P, D, E) be a (n, k) linear code, where W is the set of all vectors in $\mathbb{Z}_2^n$ and $C \subseteq$ be the set of all codewords. The code is a *one bit error correcting code* if there exists as function f such that,

$$\forall c \in C \wedge \forall e \in W, w(e) \leq 1, f(c \oplus e) = c.$$

Theorem 4.1. A code with codewords C is a one bit error correcting code if and only if,

$$\forall c, c' \in C,\ h(c, c') > 2.$$

Theorem 4.2. The $(7, 4)$ hamming code, the $(6, 3)$ simplex code and the $(3, 1)$ repetition code are 1 bit error correcting codes.