4.6 Lecture 25 (30/Nov): Limited linear independence, limited statistical independence, error correcting codes.

4.6.1 Generator matrix and parity check matrix

Error detection can be performed with the aid of the *parity check matrix* M:

$$\text{Left Nullspace}(M) = \text{Rowspace}(C)$$

$$\begin{pmatrix} \text{generator matrix} \\ C \end{pmatrix} \begin{pmatrix} \text{parity} \\ \text{check} \\ \text{matrix} \\ M \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$wM = 0 \iff w \in \text{Rowspace}(C) \iff w \text{ is a codeword}$$

$$\left.\begin{array}{l}\text{Every vector in} \\ \text{Rowspace}(C) \text{ has} \\ \text{weight } \geq k+1\end{array}\right\} \iff \left\{\begin{array}{l}\text{Every } k \text{ rows} \\ \text{of } M \text{ are linearly} \\ \text{independent}\end{array}\right.$$

In coding theory terms, this is an $(n, m, k+1)$ code over $GF(2)$. (Unfortunately, coding theorists conventionally use the letters (n, k, d) but we have $k+1$ reserved for the least weight, because we're following the conventional terminology from "k-wise independence".)

For any fixed values of n and k, the code is most efficient when the message length m, which is the number of rows of C, is as large as possible; equivalently, the number of columns of M, $\ell = n - m$, is as small as possible. So we'll want to design a matrix M with few columns in which every k rows are linearly independent.

But first, let's see a connection between linear and statistical independence.

Let B be a $k \times \ell$ matrix over $GF(2)$, with full row rank. (So $k \leq \ell$.)

If $x \in_U (GF(2))^\ell$ then $y = Bx \in_U (GF(2))^k$,

$$\begin{pmatrix} y \end{pmatrix} = \begin{pmatrix} B \end{pmatrix} \begin{pmatrix} x \end{pmatrix}$$

because the pre-image of any y is an affine subspace (a translate of the right nullspace of B). (We already made this observation in the context of Freivalds' verification algorithm, Theorem 26.)

Now, if we have a matrix M with n rows, of which every k are linearly independent, then every k bits of $z = Mx$ are uniformly distributed in $(GF(2))^k$.

$$\begin{pmatrix} z \end{pmatrix} = \begin{pmatrix} M \end{pmatrix} \begin{pmatrix} x \end{pmatrix}$$

We've exhibited *dual applications of the parity check matrix:*

- Action on row vectors: checking validity of a received word w as a codeword. ($s = wM$ is called the "syndrome" of w; in the case of non-codewords, i.e., $s \neq 0$, one of the ways to decode is to maintain a table containing for every s, the least-weight vector η s.t. $\eta M = s$. Then $w - \eta$ is the closest codeword to w. This table-lookup method is practical for some very high rate codes, where there are not many possible vectors s.)
- Action on column vectors: converting few uniform iid bits into many k-wise independent uniform bits.

Now we can see an entire sample space on n bits that are uniform and k-wise-independent. At the right end we place the uniform distribution on all 2^ℓ vectors of the vector space $GF(2)^\ell$.

$$\begin{pmatrix} \\ \Omega \\ \\ \end{pmatrix} = \begin{pmatrix} \\ M \\ \\ \end{pmatrix} \begin{pmatrix} 0 & 0 & \dots & 1 & 1 \\ \dots & \text{unif.} & \text{dist.} & \text{on} & \text{cols} \\ 0 & 1 & \dots & 0 & 1 \end{pmatrix}$$

Ω is the uniform distribution on the columns on the LHS.

Maximizing the transmission rate $\frac{m}{n} = \frac{n-\ell}{n}$ of a binary, k-error-detecting code, is equivalent to *minimizing* the size $|\Omega| = 2^\ell$ of a linear k-wise independent binary sample space.

So how big does $|\Omega|$ have to be?

Theorem 87

1. *For all n there is a sample space of size $O(n^{\lfloor k/2 \rfloor})$ with n uniform k-wise independent bits.*

   For larger ranges one has: For all n there is a sample space of size $O(2^{k \max\{m, \lceil \lg n \rceil\}})$ with n k-wise independent rvs, each uniformly distributed on $[2^m]$.

2. *For all n, any sample space on n k-wise independent bits, none of which is a.s. constant, has size $\Omega(n^{\lfloor k/2 \rfloor})$.*

We show Part 1 in Sec. 4.6.3; Part 2 will be on the problem set.

First though, returning to the subject of codes, there is a question worth asking even though we don't need it for our current purpose:

4.6.2 Constructing C from M

Suppose we have constructed a parity check matrix M. How do we then get a generator matrix C?

One should note that over a finite field, Gram-Schmidt does not work. Gram-Schmidt would have allowed us to produce new vectors which are both orthogonal to the columns of M and linearly independent of them. But this is generally not possible: the row space of C and the column space of M do not necessarily span the n-dimensional space. For example over $GF(2)$ we may have

$$C = \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

However, Gaussian elimination does work over finite fields, and that is what is essential.

Specifically, given $n \times a$ M and $n \times b$ N, $b < n - a$, N of full column rank (i.e., rank b), we show how to construct a vector c s.t. $cM = 0$ and c' is linearly independent of the columns of N. (Then adjoin c' to N and repeat.)

Perform Gaussian elimination on the columns of N so that it is lower triangular, with a nonzero diagonal. (That is, allowed operations are adding columns to one another, and permuting rows. When permuting rows of N, permute the rows of M to match.) Obviously this does not change the column space of N (except for the simultaneous permution of rows in N and M).

Now take the submatrix of M consisting of the $a+1$ rows $b+1, \ldots, b+a+1$. By Gaussian elimination on these rows we can find a linear dependence among them. Extending that dependence to the n-dimensional space with 0 coefficients elsewhere yields a vector c s.t. $cM = 0$ and s.t. the support of c is disjoint from the first b coordinates. Then c is linearly independent of the column space of N because the restriction of N to its first b rows is nonsingular, so any linear combination of the rows of N has a nonzero value somewhere among its first b entries.

4.6.3 Proof of Thm (87) Part (1): Upper bound on the size of k-wise independent sample spaces

(We'll do this carefully for producing binary rvs and only mention at the end what should be done for larger ranges.)

This construction uses the finite fields whose cardinalities are powers of 2. These are called extension fields of $GF(2)$. If you are not familiar with this, just keep in mind that for each integer $r \geq 1$ there is a (unique) field with 2^r elements. We can add, subtract, multiply and divide these without leaving the set; in particular, in the usual way of representing the elements of the field as bit strings of length r, addition is simply XOR addition.[8] Specifically, we can think of the elements of $GF(2^r)$ as the polynomials of degree $\leq r-1$ over $GF(2)$, taken modulo some fixed irreducible polynomial p of degree r. That is, a field element c has the form $c = c_{r-1}x^{r-1} + \ldots + c_1 x + c_0 \pmod{p(x)}$, and our usual way of representing this element is through the mapping $\beta : GF(2^r) \to (GF(2))^r$ given by $\beta(c) = (c_{r-1}, \ldots, c_0)$. (I.e., the list of coefficients.)

But all we really need today are three things: (a) Like $GF(2)$, $GF(2^r)$ is a field of characteristic 2, i.e., $2x = 0$. (b) For matrices over $GF(2^r)$ the usual concepts of linear independence and rank apply. (c) β is injective, linear (namely $\beta(c) + \beta(c') = \beta(c + c')$), and $\beta(1) = 0 \ldots 01$.

Now, round n up to the nearest $n = 2^r - 1$, and let $a_1, \ldots, a_n$ denote the nonzero elements of the field. Let M_1 be the following Vandermonde matrix over the field $GF(2^r)$:

$$M_1 = \begin{pmatrix} 1 & a_1 & a_1^2 & \ldots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \ldots & a_2^{k-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & a_n & a_n^2 & \ldots & a_n^{k-1} \end{pmatrix}$$

Exercise: Every k rows of M_1 are linearly independent over $GF(2^r)$. (Form any such submatrix B, say that using the first k rows. Verify that $\mathrm{Det}(B) = \prod_{i<j}(a_j - a_i)$.)

M_1 is an $n \times k$ matrix over $GF(2^r)$. Next, expand each of its entries as a row vector of bits, thus forming an $n \times kr$ matrix M_2 over $GF(2)$:

$$M_2 = \begin{pmatrix} \beta(1) = 001 & \beta(a_1) = 001 & \ldots & \beta(a_1^{k-1}) = 001 \\ \beta(1) = 001 & \beta(a_2) = 010 & \ldots & \beta(a_2^{k-1}) = \ldots \\ \ldots & \ldots & \ldots & \ldots \\ \beta(1) = 001 & \beta(a_n) = 111 & \ldots & \beta(a_n^{k-1}) = \ldots \end{pmatrix}$$

[8] See any introduction to Algebra, for instance Artin [9].

Corollary: Every k rows of M_2 are linearly independent over $GF(2)$.

Actually it is possible to even further reduce the number of columns while retaining the corollary.

First, we can drop the leading 0's in the first entry.

Second, we can strike out all batches of columns generated by positive even powers.

$$M_3 = \begin{pmatrix} 1 & \beta(a_1) = 001 & \beta(a_1^3) = 001 & \dots & \dots \\ 1 & \beta(a_2) = 010 & \beta(a_2^3) = \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \beta(a_n) = 111 & \beta(a_n^3) = \dots & \dots & \dots \end{pmatrix}$$

Lemma 88 *Every set of rows that is linearly independent (over $GF(2^r)$) in M_1 is also linearly independent (over $GF(2)$) in M_3. Hence every k rows of M_3 are linearly independent.*

Proof: Let a set of rows R be independent in M_1; we show the same is true in M_3. Since M_3 is over $GF(2)$, this is equivalent to saying that for every $\emptyset \subset S \subseteq R$, the sum of the rows S in M_3 is nonzero.

So we are to show that S independent in M_1 has nonzero sum in M_3. Independence in M_1 implies in particular that the sum in M_2 of the rows in S is nonzero.

If $|S|$ is odd then the same sum in M_3 has a nonzero first entry and we are done.

Otherwise, let $t > 0$ be the smallest value such that $\sum_{i \in S} a_i^t \neq 0$; it is enough to show that t is odd. Suppose not, so $t = 2t'$. Then, since Characteristic$(GF(2^r)) = 2$,

$$\sum_{i \in S} a_i^{2t'} = \left(\sum_{i \in S} a_i^{t'} \right)^2$$

so $\sum_{i \in S} a_i^{t'} \neq 0$, contradicting minimality of t. □

Finally for the binary construction, recalling that $n = 2^r - 1$, we have $|\Omega| = 2^{1+r\lfloor k/2 \rfloor} \in O(n^{\lfloor k/2 \rfloor})$.

Comment: If you want n k-wise independent bits with nonuniform marginals, then this construction doesn't work. The best general construction, due to Koller and Megiddo [64], is of size $O(n^k)$.

Larger ranges: this is actually simpler because we're not achieving the factor-2 savings in the exponent. Let r, as in the statement, be $r = \max\{m, \lceil \lg n \rceil\}$. Just form the matrix M_1.

4.6.4 Back to Gale-Berlekamp

We now see a deterministic polynomial-time algorithm for playing the Gale-Berlekamp game. As we demonstrated last time, it is enough to use a 4-wise independent sample space in order to achieve $\Omega(n^{3/2})$ expected performance. The above construction gives us a 4-wise independent sample space of size $O(n^2)$. All we have to do is exhaustively list the points of the sample space until we find one with performance $\Omega(n^{3/2})$.

4.6.5 Back to MIS

For MIS we need only pairwise independence, but want the marking probabilities p_v to be more varied (approximately $\frac{1}{2d_v+1}$). This, however, is easy to achieve: use the matrix M_1, with $k = 2$, without modifying to M_2 and M_3. This generates for each v an element in the field $GF(2^r)$; these elements are pairwise independent; and one can designate for each v a fraction of approximately $\frac{1}{2d_v+1}$ elements which cause v to be marked. The deterministic algorithm is therefore as described in Sec. 4.4.4, with a space of size $O(n^2)$.