MPRI **Year 2020–21**

Module 2.13.2 : Error correcting codes and applications to cryptography

Mid-term exam, November 26

You have 2h30 Including time for scanning/taking pictures.
You can answer either in French or in English.

Exercise 1. True or false. You should justify your answers.

1. A linear code has a unique generator matrix.

Answer : **False**, for instance

$$\mathbf{G}_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{G}_2 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

generate the same code while being distinct.

2. A linear code has a unique parity–check matrix.

Answer : **False**, a parity–check matrix is a generator matrix of the dual code, so the answer is the same as for the previous question.

3. Given a linear code with a generator matrix **G**, multiplying **G** on the right by a non-singular matrix does not change the code.

Answer : **False**, Consider the binary code with generator matrix

$$\mathbf{G} = (0\ 1)$$

and the nonsingular matrix $\mathbf{S} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ then the code with generator matrix $\mathbf{GS} = (1\ 0)$ is not the same.

4. Given a linear code with a generator matrix **G**, multiplying **G** on the left by a non-singular matrix does not change the code.

Answer : **True**, left multiplying by a non singular matrix replaces the set of rows by another one which spans the very same space and hence the same code.

5. There is no linear code with parameters $[n, k, n - k + 2]$.

Answer : **True**, this is due to Singleton bound.

6. There is no linear code whose parameters exceed the Gilbert-Varshamov bound.

Answer : **False**, Gilbert Varshamov bound is a lower bound : *i.e.* there exist codes whose parameters reach or possibly strictly exceed GV bound.

7. Asymptotic Plotkin bound is always sharper than asymptotic Singleton bound.

Answer : **True**, asymptotic Singleton bound asserts that $R \leqslant 1 - \delta$ while Plotkin asserts that $R \leqslant \min(1 - \delta(\frac{q-1}{q}), 0)$. The latter is sharper than the former.

8. The weight distribution of a linear code of length n and dimension $n-3$ can be computed in polynomial time in n.

Answer : **True**, the weight distribution of a linear code of dimension 3 can be computed in polynomial time in n by enumerating all the codewords and computing their weight which costs $O(nq^3)$ operations. Then, for a code of dimension $n-3$, one can compute its dual using Gaussian elimination. Then, the weight distribution of its dual can be computed in polynomial time and one deduces the weight distribution of the original code by applying McWilliams Theorem.

9. For any linear code, decoding up to half the minimum distance can be done in polynomial time.

Answer : **False**, for an arbitrary code, decoding is a difficult problem. Even the estimate of the minimum distance is a difficult problem.

10. For any linear code of length n and dimension k, computing a codeword of weight $\leqslant n-k+1$ can be done in polynomial time.

Answer : **True**, by computing a row echelon form of a generator matrix, the last row has weight at most $n-k+1$ (this is a constructive proof of Singleton bound).

Exercise 2. A *Boolean function* in m variables is an m-variable polynomial which is a sum of monomials $X_1^{i_1} \cdots X_m^{i_m}$ where $i_1, \ldots, i_m \in \{0,1\}$. The *degree* of a monomial $X_1^{i_1} \cdots X_m^{i_m}$ is the sum $i_1 + \cdots + i_m$. The *degree* of a Boolean function is the maximum degree of its monomials. For instance, the Boolean functions :

$$F(X_1, X_2, X_3) = X_1 + X_1X_2 + X_2X_3 \quad \text{and} \quad G(X_1, X_2, X_3, X_4) = X_1X_3X_4 + X_2X_3 + 1$$

have respective degrees 2 and 3. By convention, the degree of the function 0 is set to $-\infty$. The space of Boolean functions of degree $\leqslant r$ in m variables is denoted by $\mathcal{B}_r(m)$ and the whole space of Boolean functions in m variables is denoted by $\mathcal{B}(m)$.

Question 1. Give the full list of the elements of the sets $\mathcal{B}_0(2)$ and $\mathcal{B}_1(2)$.

Answer :

$$\mathcal{B}_0(2) = \{0, 1\},$$
$$\mathcal{B}_1(2) = \{0, 1, X_1, X_2, X_1 + 1, X_2 + 1, X_1 + X_2, X_1 + X_2 + 1\}.$$

Question 2. Prove that

(a) for any $0 \leqslant r < m$, we have $\dim_{\mathbb{F}_2} \mathcal{B}_r(m) = \sum_{j=0}^{r} \binom{m}{j}$.

Answer : The space is spanned by monomials involving up to r distinct variables. For any $0 \leqslant j \leqslant r$, the number of degree j monomials involving distinct variables is $\binom{m}{j}$. Hence the result.

(b) $\dim_{\mathbb{F}_2} \mathcal{B}(m) = 2^m$.

Answer : From Newton binomial formula,

$$\sum_{j=0}^{m} \binom{m}{j} = \sum_{j=0}^{m} \binom{m}{j} 1^j \cdot 1^{m-j} = (1+1)^m = 2^m.$$

Fix integers $m > 0$ and $r > 0$, then the Reed–Muller code $\mathcal{R}(r, m)$ is defined as

$$\mathcal{R}(r,m) := \left\{ (P(x_1, \dots, x_m))_{(x_1,\dots,x_m) \in \mathbb{F}_2^m} \mid P \in \mathcal{B}_r(m) \right\}.$$

where the elements of $\mathbb{F}_2^m$ are sorted in the lexicographic order. For instance for $m = 3$, the elements of $\mathbb{F}_2^3$ are sorted as :

$$(000) \prec (100) \prec (010) \prec (110) \prec (001) \prec (101) \prec (011) \prec (111).$$

Question 3. Prove that for any $m \geqslant 0$, the code $\mathcal{R}(0, m)$ is the repetition code of length 2^m.

Answer : Note first that $\mathcal{B}_0(m)$ is spanned by the constant boolean function 1. Then, the code $\mathcal{R}(0, m)$ is spanned by the evaluation of this function which is nothing but the all–one codeword. Hence this code is the repetition code of length 2^m.

Question 4. Give a generator matrix of the code $\mathcal{R}(1, 3)$.

Answer :

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We focus on the encoding of the code $\mathcal{R}(1, m)$, which is given by the map

$$\mathrm{Enc}_m : \left\{ \begin{array}{ccc} \mathcal{B}_1(m) & \longrightarrow & \mathbb{F}_2^{2^m} \\ F = a_0 + a_1 X_1 + \cdots + a_m X_m & \longmapsto & (F(x_1, \dots, x_m))_{(x_1,\dots,x_m) \in \mathbb{F}_2^m}. \end{array} \right.$$

Question 5. Prove that a naive encoding of the code $\mathcal{R}(1, m)$ has a complexity of $O(n \log n)$, where $n = 2^m$ denotes the code length.

Answer : The Naive encoding consists in performing the multiplication of a row vector of length $m + 1$ on the right by an $(m + 1) \times 2^m$ binary matrix, which costs $O(m2^m) = O(n \log n)$ binary operations.

Question 6. This encoding may be improved using the following principle :

(a) Prove that, given $F_{m-1} = a_0 + a_1 X_1 + \cdots + a_{m-1} X_{m-1}$ and $F = F_{m-1} + a_m X_m$, then

$$\mathrm{Enc}_m(F) = (\underbrace{\mathrm{Enc}_{m-1}(F_{m-1})}_{\text{length } 2^{m-1}} \mid \underbrace{\mathrm{Enc}_{m-1}(F_{m-1})}_{\text{length } 2^{m-1}} + \underbrace{(a_m, \ldots, a_m)}_{\text{length } 2^{m-1}}), \tag{1}$$

where the "|" stands for the concatenation of codewords.

Answer : Since the entries are indexed by vectors sorted in the lexicographic order, the 2^{m-1} first ones correspond to elements of $\mathbb{F}_2^m$ whose last entry is 0 and the 2^{m-1} last ones to elements whose last entry is 1. Moreover, If $(v_i)_{i \in \{1 \ldots 2^{m-1}\}}$ is the sequence of vectors of $\mathbb{F}_2^{m-1}$ sorted in the lexicographic order, then the concatenation of the sequences $((v_i|0))_i$ and $((v_i|1))_i$ yields the list of elements of $\mathbb{F}_2^m$ sorted in the lexicographic order.
Consequently,

(*i*) for a Boolean function $H \in \mathcal{B}_r(m-1)$ such that $\mathrm{Enc}_{m-1}(H) = u$, the same function regarded as a function of $\mathcal{B}_r(m)$ (but which does not depend on X_m) yields the vector

$$\mathrm{Enc}_m(H) = (u|u) \in \mathcal{R}(r, m).$$

(*ii*) On the other hand, the evaluation vector of $a_m X_m$ is nothing but $(0 \ldots 0 \mid a_m \ldots a_m)$.

By linearity, we deduce the expected result from the two previous observations.

(b) Deduce from the previous question a faster encoding algorithm.

Answer : The previous result gives trivially a recursive algorithm for encoding.

(c) Prove that this faster encoding has complexity $O(n)$, where $n = 2^m$ denotes the code length.

Answer : The cost $C(m)$ of the encoding of F equals :
— the cost $C(m-1)$ of the encoding of F_{m-1} ;
— plus the cost of computing $\mathrm{Enc}_{m-1}(F_{m-1}) + (a_m \cdots a_m)$ which is 2^{m-1} operations. This yields

$$\forall m \geqslant 2, \quad C(m) = 2^{m-1} + C(m-1)$$

which entails :

$$C(m) = \left(\sum_{j=1}^{m-1} 2^j \right) + O(1) = O(2^m).$$

We now focus on the decoding of these codes. Indexing words of $\mathbb{F}_2^{2^m}$ with elements of $\mathbb{F}_2^m$ sorted in the lexicographic order, for any $\mathbf{c} \in \mathbb{F}_2^{2^m}$, one defines

$$\Delta_\alpha(\mathbf{c}) := \left(\mathbf{c}_{(x_1+\alpha_1, \ldots, x_m+\alpha_m)} + \mathbf{c}_{(x_1, \ldots, x_m)}\right)_{(x_1, \ldots, x_m) \in \mathbb{F}_2^m}$$

Question 8. Let $\mathbf{c} = (1\ 1\ 0\ 1\ 0\ 0\ 0\ 1) \in \mathbb{F}_2^8$ and $\alpha = (1\ 0\ 1) \in \mathbb{F}_2^3$. Compute $\Delta_\alpha(\mathbf{c})$.

Answer : $\Delta_\alpha(\mathbf{c}) = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$.

Question 9. If $\mathbf{c} \in \mathbb{F}_2^{2^m}$ has weight t,

(a) prove that for any $\alpha \in \mathbb{F}_2^m$ the vector $\Delta_\alpha(\mathbf{c})$ has weight less than or equal to $2t$;

Answer : For any $v \in \mathbb{F}_2^m$, denote by $\mathbf{c}_v \in \mathbb{F}_2$ the entry of $\mathbf{c}$ indexed by v. Let $v_1, \dots, v_t \in \mathbb{F}_2^m$ be the indexes such that $\mathbf{c}_{v_i} = 1$. Then the possible indexes at which $\Delta_\alpha(\mathbf{c})$ might be nonzero are $v_1, \dots, v_t, v_1 + \alpha, \dots, v_t + \alpha$ whose number is at most $2t$.

(b) Give an example where this $2t$ is reached (*Hint. choose a small value of m to design your example*)

Answer : The example of Question 8.

Question 10. Let $\mathbf{b}_1, \dots, \mathbf{b}_m$ be the canonical basis of $\mathbb{F}_2^m$. Let $F = a_0 + a_1X_1 + \dots + a_mX_m \in \mathcal{B}_1(m)$ and $\mathbf{c} = \mathrm{Enc}_m(F) \in \mathcal{R}(1, m)$. Prove that for any $i \in \{1, \dots, m\}$, we have

$$\Delta_{\mathbf{b}_i}(\mathbf{c}) = (a_i, \dots, a_i).$$

Answer : Let $F \in \mathcal{B}_1(m)$ such that $\mathbf{c} = \mathrm{Enc}_m(F)$. Let $v \in \mathbb{F}_2^m$, then

$$\left(\Delta_{\mathbf{b}_i}(\mathbf{c})\right)_v = F(v) + F(v + \mathbf{b}_i)$$

Moreover,

$$\begin{aligned} F(v) + F(v + \mathbf{b}_i) = {} & a_0 + a_1v_1 + \dots + a_iv_i + \dots + a_mv_m \\ & a_0 + a_1v_1 + \dots + a_i(v_i + 1) + \dots + a_mv_m \\ = {} & a_i. \end{aligned}$$

Hence the result.

Question 11. Suppose you received a corrupted codeword $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} = \mathrm{Enc}_m(F) \in \mathcal{R}(1, m)$ and $\mathbf{e}$ has Hamming weight $w_{\mathrm{H}}(\mathbf{e}) \leqslant 2^{m-2} - 1$.

(a) Explain how to recover $a_1, \dots, a_m$ using derivations and detail why the bound on the weight of $\mathbf{e}$ asserts that these coefficients are uniquely recovered.

Answer : Compute the derivatives $\Delta_{\mathbf{b}_i}(\mathbf{y})$. By linearity of the $\Delta_{\mathbf{b}_i}$ operator, we have $\Delta_{\mathbf{b}_i}(\mathbf{y}) = \Delta_{\mathbf{b}_i}(\mathbf{c}) + \Delta_{\mathbf{b}_i}(\mathbf{e})$. Moreover, $\Delta_{\mathbf{b}_i}(\mathbf{c}) = (a_i \cdots a_i)$ and, from Question 9a the weight of $\Delta_{\mathbf{b}_i}(\mathbf{e})$ is at most $2^{m-1} - 2$. Therefore, a_i can be deduced from $\Delta_{\mathbf{b}_i}(\mathbf{y})$ by majority voting.

(b) Once $a_1, \dots, a_m$ are known, explain how to find a_0.

Answer : Compute the encoding $\mathbf{c}'$ of the Boolean function $F' = a_1X_1 + \dots + a_mX_m$. Then compute $\mathbf{y} - \mathbf{c}' = \mathrm{Enc}_m(a_0) + \mathbf{e}$. Here again, a majority voting permits to recover a_0.

(c) Give the complexity of this decoding algorithm.

Answer :

— The computation of the $\Delta_{\mathbf{b}_i}(\mathbf{y})$ for $i \in \{1, \ldots, m\}$ costs $O(m2^m)$ binary operations and the m majority voting processes also cost $O(m2^m)$;

— The calculation of $\mathbf{c}'$ is an encoding, which can be performed in $O(2^m)$ operations according to Question 6c.

This gives an overall complexity of $O(m2^m)$ operations.

(d) Looking at Theorem 10.8 of these notes, what can you say about the decoding radius (*i.e.* the amount of errors it corrects) of this algorithm ?

Answer : The algorithm corrects up to half the minimum distance.