Foundations of Probabilistic Proofs (Fall 2022)
Note 9: Low-Degree Testing
Date: 2022.10.18

This note contains definitions, theorems, facts, *etc.* that are not fully explained in lectures due to limited time. If you think there are anything missing or any mistakes, please contact ziyi.guan@epfl.ch.

1 Matrix representation for linear code

We introduce the linear codes in note 7, now we give an alternative representation for them, i.e. the generator matrix and parity-check matrix.

1.1 Generator matrix

Definition 1. *A* **linear code** C *is a code such that for any* $c_1, c_2 \in \mathrm{C}$, $c_1 + c_2 \in \mathrm{C}$.

Recall that in note 7 we show that a linear code C of block length n and message length k over a finite field $\mathbb{F}$ is a k-dimensional linear subspace of $\mathbb{F}^n$. Let $\{c_i\}_{1 \le i \le k}$ be a basis of C where $c_i \in \mathrm{C}$ (and $c_i \in \mathbb{F}^n$) for all $1 \le i \le k$, then:

- $\{c_i\}_{1 \le i \le k}$ are linearly independent.
- $\mathrm{C} = span(c_1, \ldots, c_k)$.

We claim that there is a matrix $G \in \mathbb{F}^{n \times k}$ such that $\mathrm{C} = \{Gx : x \in \mathbb{F}^k\}$. Such G is called a ***generator matrix*** for C and $\mathrm{C} = \mathrm{Img}(G)$. For example, let $G \in \mathbb{F}^{n \times k}$ be such that the columns of G are the basis vectors.

Generator matrix can be viewed as a representation of a linear code and is useful for the encoding of messages. However, it can be shown that a linear code has more than one generator matrices. In other words, generator matrices are not unique (one can simply reorder the columns and get another generator matrix).

Example 1. *Recall the example* 2 *in note* 7. *Suppose* $\mathsf{Enc}\colon \{0,1\}^3 \to \{0,1\}^7$ *with* $\mathsf{Enc}(x_1, x_2, x_3) \coloneqq (x_1, x_2, x_3, x_4, x_2 + x_3 + x_4, x_1 + x_3 + x_4, x_1 + x_2 + x_4)$. *Let* $\mathrm{C} \coloneqq \mathrm{Img}(\mathsf{Enc})$. *Then we can find two generator matrices for* C*:*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \quad G' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

One can easily verify that both of them are generator matrices of C *by checking that every column of* G *and* G' *is a codeword in* C *and they are linearly independent.*

Nonetheless, note that we can use a standard form:

$$G \coloneqq [I_k|P]^\intercal$$

where I_k is the $k \times k$ identity matrix and P a $k \times (n-k)$ martix. For all $1 \leq i \leq k$, the i^{th} column of G is the codeword in C encoded from message $x \in \mathbb{F}^k$ with $x_i \coloneqq 1$ and $x_j \coloneqq 0$ for all $j \neq i$.

1.2 Parity-check matrix

Moreover, linear codes are often associated with their parity-check matrix.

Definition 2. *A matrix is called a* **parity-check matrix**, *denoted H, for linear code* C *if $H \in \mathbb{F}^{(n-k)\times n}$ is such that $c \in \mathrm{C}$ if and only if $Hc = \mathbf{0}$.*

Now we describe how to compute the parity-check matrix for a linear code. Let C be a linear code and $G \coloneqq [I_k|P]^\intercal$ be its generator matrix. Then its parity-check matrix (in standard form) is given by:

$$H \coloneqq [-P^\intercal|I_{n-k}]$$

To justify that this computation gives us the desired property, we introduce dual codes for linear codes.

Definition 3. *Let* C *be a linear code of block length n and message length k over a finite field $\mathbb{F}$, then the* **dual code** *of* C *is* $\mathrm{C}^\perp \coloneqq \{x \in \mathbb{F}^n : \langle x, c\rangle = 0, \forall c \in \mathrm{C}\}$.

$\mathrm{C}^\perp$ is a linear code of block length n and message length $n-k$ over $\mathbb{F}$, because:

- $\mathrm{C}^\perp \subseteq \mathbb{F}^n$ and $\left|\mathrm{C}^\perp\right| = |\mathbb{F}|^{n-k}$.
- for any $c_1, c_2 \in \mathrm{C}^\perp$, $\langle c_1 + c_2, c\rangle = \langle c_1, c\rangle + \langle c_2, c\rangle = 0, \forall c \in \mathrm{C}$, i.e. $c_1 + c_2 \in \mathrm{C}^\perp$.

We claim that the parity-check matrix of C is a generator matrix of $\mathrm{C}^\perp$. The proof of the following lemma is left as an exercise to the reader.

Lemma. *$H \coloneqq [-P^\intercal|I_{n-k}] \in \mathbb{F}^{(n-k)\times n}$ is a parity-check matrix of* C, *and $H^\intercal$ is a generator matrix of* $\mathrm{C}^\perp$.

Example 2. *The generator matrix and parity-check matrix in standard form of* C *defined in Example 1 are:*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The reason that parity-check matrices are important for linear codes is that it can be used in error-correction. Consider a binary linear code C. Suppose a codeword $c \in \mathrm{C}$ is corrupted at one location i, i.e. $\tilde{c} = c + \boldsymbol{e}_i$, we can use H to correct $\tilde{c}$:

$$H\tilde{c} = H(c + \boldsymbol{e}_i) = Hc + H\boldsymbol{e}_i = H\boldsymbol{e}_i$$

which is the i^{th} column of H. Therefore, we can find the location of the error and then correct it.

Example 3. *Let c be a codeword in the code defined in Example 1. Assume that c is corrupted at one location, which gives $\tilde{c} = (1,1,1,1,1,1,0)^\intercal$.*

$$H\tilde{c} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot (1,1,1,1,1,1,0)^\intercal = (0,0,1)^\intercal.$$

$H\tilde{c}$ is equal to the 7^{th} column of H, thus we know that $\boldsymbol{e}_i = (0,0,0,0,0,0,1)$ and $c = (1,1,1,1,1,1,1)^\intercal$.

2 Multivariate Lagrange interpolation

In note 4 we introduce Lagrange interpolating polynomials. Now we look more carefully in the multivariate Langrange interpolating polynomials. In particular, we explain that the number of points needed to interpolate a polynomial with n variables and degree at most d is $\binom{n+d}{d}$.

We claim that we can construct a unique interpolating polynomial of total degree at most d using a set of distinct pairs $\{(x_i, y_i)\}_{1 \le i \le \rho}$ where $x_i \in \mathbb{F}^n, y_i \in \mathbb{F}$ where $\rho = \binom{n+d}{d}$.

- There are $\binom{n+d}{d}$ monomials for a n-variate polynomial with total degree at most d: the number of monomials is equal to the total number of ordered partitions $\{0, 1, \ldots, d\}$ into n nonnegative parts, which is $\binom{n+d}{d}$.
- p has ρ monomials, so we can write $p(x_1, \ldots, x_n) \coloneqq \sum_{i=1}^{\rho} \alpha_i x^{\boldsymbol{e}_i}$ where
  - $\boldsymbol{e}_i \coloneqq (e_{i,1}, \ldots, e_{i,n})$ where $e_{i,j}$ is the exponent of x_j in i^{th} term of p.
  - $x^{\boldsymbol{e}_i} \coloneqq \prod_{j=1}^{n} x_j^{e_{i,j}}$. Note that $\sum_{j=1}^{n} e_{i,j} \le d$ for all i (because p's degree $\le d$).
  - Using $\{(x_i, y_i)\}_{1 \le i \le \rho}$ to compute the coefficients α_i's.

3 The Reed-Solomon codes and the Reed-Muller codes

In the lecture, we introduce low-degree testing, which determines if a given polynomial has degree at most d. We call these polynomials low-degree polynomials. In this section, we introduce two codes associated with low-degree polynomials, the Reed-Solomon codes and the Reed-Muller code.

3.1 Reed-Solomon codes

We first consider the univariate low-degree polynomials. We claim that the set of all univariate low-degree polynomials is a linear code.

Definition 4. *Let $\mathbb{F}$ be a finite field. Let $\mathbb{F}[X]_{<d}$ be the set of all univariate polynomials over $\mathbb{F}$ with degree less than d. Let $S = \{\alpha_1, \ldots, \alpha_n\}$ be a subset of $\mathbb{F}$. The* **Reed-Solomon code** *(RS code) of message length d and block length n is*

$$\mathrm{RS}(S, n, d) = \{(f(\alpha_1), \ldots, f(\alpha_n)) : f \in \mathbb{F}[X]_{<d}\} \ .$$

Remark 1. *Note that we can use d elements in $\mathbb{F}$ to describe a polynomial in $\mathbb{F}[X]_{<d}$. Therefore, the message length of a RS code is d, and the block length is $|S| = n$. Moreover, this definition suggests a natural encoding for RS codes:*

$$\boldsymbol{x} = (x_1, \ldots, x_d) \mapsto (f_x(\alpha_1), \ldots, f_x(\alpha_n)) \ ,$$

where $f_x(X) = x_1 + x_2X + x_3X^2 + \cdots + x_dX^{d-1}$.

Now we discuss the properties of the RS codes. Let C be a RS code with message length d and block length n.

- C is a linear code: $\mathrm{C}(f+g) = \mathrm{C}(f) + \mathrm{C}(g)$ for all $f, g \in \mathbb{F}[X]_{<d}$.
- The generator matrix of C is the Vandermonde matrix:

$$G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{d-1} & \alpha_2^{d-1} & \alpha_3^{d-1} & \dots & \alpha_n^{d-1} \end{bmatrix} .$$

- C has distance $n - d + 1$:
  - Two distinct polynomials of degree less than d agree at most $d-1$ points.
  - Two distinct codewords in C disagree at least $n - d + 1$ points.

3.2 Reed-Muller codes

The Reed-Solomon codes use the univariate low-degree polynomials, in this section, we generalize to the multivariate case. We consider the set of multivariate polynomials with total degree d.

Definition 5 (Reed-Muller code)**.** *Let* $\mathbb{F}$ *be a finite field and let* $q = |\mathbb{F}|$*. Let* $\mathbb{F}[X_1, \dots, X_m]_{\leq d}$ *with* $d \leq q$ *be the set of all polynomials with* m *variables of total degree at most* d*. The* **Reed-Muller code** *(RM code) with* m *variables and total degree* d *over* $\mathbb{F}$ *is*

$$\mathrm{RM}_q(m, d) = \{(f(\boldsymbol{\alpha_1}), \cdots, f(\boldsymbol{\alpha_{q^m}})) : f \in \mathbb{F}[X_1, \dots, X_m]_{\leq d}\} .$$

Now we discuss the properties of the RM codes. Let C be a m-variate RM code of total degree d over $\mathbb{F}$.

- C is a linear codes, the proof is similar to the one for Reed-Solomon codes.
- C has block length $|\mathbb{F}|^m$ and message length $\binom{m+d}{d}$ as explained in Section 2.
  - C has rate that is inverse-polynomial. Better than Hadamard code, which has inverse-exponential rate.
- Reed-Muller codes generalize Reed-Solomon codes and the Hadamard codes:
  - $\mathrm{RM}_q(1, d)$ gives us the Reed-Solomon code.
  - $\mathrm{RM}_2(m, 1)$ gives us the Hadamard Code.
- The relative distance of the Reed-Muller code is $1 - \frac{d}{|\mathbb{F}|}$.
  - From the Schwartz-Zippel Lemma, $\Pr[f(a_1, \dots, a_m) \neq 0] \geq 1 - \frac{d}{|\mathbb{F}|}$ for $f \in \mathbb{F}[X_1, \dots, X_m]_{\leq d}$.
- Reed-Muller code is locally testable.

4 Local characterization of low-degree polynomials

In the lecture, we use interpolation for univariate low degree testing. However, it cannot be generalized to multivariate case because the exponential number of queries. Therefore, we use the Rubinfeld-Sudan test instead, which can be directly extended to the multivariate case with small overhead. This test is motivated by a local characterization of low-degree polynomials.

Lemma. *For every prime q such that $d+1 < q$, for every univariate polynomial $f \coloneqq \mathbb{F}_q \to \mathbb{F}_q$, $f \in \mathbb{F}_q^{\leq d}[x]$ iff for every $a \in \mathbb{F}_q, \sum_{i=0}^{d+1} c_{d,i} f(a+i) = 0$, where $c_{d,i} \coloneqq (-1)^{i+1}\binom{d+1}{i}$ for $i \in [d+1]$.*

Proof. We prove by induction on the degree d.

- Base case $d = 0$: In this case f is a constant and $(c_{0,0}, c_{0,1}) = (-1, 1)$, thus $c_{0,0}f(a) + c_{0,1}f(a+1) = 0$ follows immediately.
- Inductive step: We assume that the lemma is true for all polynomials of degree at most $d-1$ and prove for f of degree at most d.
  - Let $g(x) = f(x+1) - f(x)$: $\deg(g) \leq \deg(f) - 1$.
    - $f(x) = \sum_{i=0}^{d} \alpha_i x^i \implies g(x) = \sum_{i=0}^{d} \alpha_i[(x+1)^i - x^i] = \sum_{i=0}^{d} \alpha_i \sum_{j=0}^{i-1} \binom{i}{j} x^j$
  - By the induction hypothesis, we know that for every $a \in \mathbb{F}_q$, $\sum_{i=0}^{d} c_{d-1,i} g(a+i) = 0$. We rewrite the equality to finish the proof:

$$
\begin{aligned}
0 &= \sum_{i=0}^{d} c_{d-1,i} g(a+i) \\
&= \sum_{i=0}^{d} (-1)^{i+1} \binom{d}{i} (f(a+i+1) - f(a+i)) \\
&= \sum_{i=0}^{d} (-1)^{i+1} \binom{d}{i} f(a+i+1) - \sum_{i=0}^{d} (-1)^{i+1} \binom{d}{i} f(a+i) \\
&= \sum_{i=1}^{d+1} (-1)^{i} \binom{d}{i-1} f(a+i) + \sum_{i=0}^{d} (-1)^{i} \binom{d}{i} f(a+i) \\
&= (-1)^{d+1} f(a+d+1) + f(a) + \sum_{i=1}^{d} (-1)^{i} (\binom{d}{i-1} + \binom{d}{i}) f(a+i) \\
&= -\sum_{i=0}^{d+1} (-1)^{i+1} \binom{d+1}{i} f(a+i) \\
&= -\sum_{i=0}^{d+1} c_{d,i} f(a+i)
\end{aligned}
$$

□