Information and Coding Theory **Autumn 2022**

Lecture 17: November 29, 2022

Lecturer: Madhur Tulsiani

1 Linear Codes

A linear code $C \subseteq \mathbb{F}_q^n$ is a subspace of $\mathbb{F}_q^n$, viewed as a vector space over the finite field $\mathbb{F}_q$. We will always take q to be a prime number, with addition and multiplication in $\mathbb{F}_q$ defined modulo q (although the discussion can also be extended to the case when q is a prime power). If $\dim(C) = k$, we can think of C as encoding a message in $\mathbb{F}_q^k$ by *linearly* mapping it to an element $x \in C$. Overloading notation to denote $\mathsf{Enc}(w) \in C$ by $C(w)$, the encoding map $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ satisfies

$$C(\alpha \cdot u + \beta \cdot v) \; = \; \alpha \cdot C(u) + \beta \cdot C(v) \qquad \forall\, u, v \in \mathbb{F}_q^k,\ \alpha, \beta \in \mathbb{F}_q\,.$$

Since a linear encoding is a linear map from a finite dimensional vector space to another, we can write it as a matrix of finite size. That is, there is a corresponding $G \in \mathbb{F}_q^{n \times k}$ s.t. $C(w) = Gw$ for all $w \in \mathbb{F}_q^k$. This matrix is referred to as a generator matrix for the code C.

If the encoding map is injective (which is the bare minimum for a good code), then the rank of G must be k (otherwise there exist $u, v \in \mathbb{F}_q^k$ such that $Gu = Gv$). Hence, the null space of G^T has dimension $n - k$. This defines another useful matrix, known as the parity check matrix of the code.

Definition 1.1 (Parity Check Matrix). *Let $b_1, \ldots, b_{n-k} \in \mathbb{F}_q^n$ be a basis for the null space of G^T corresponding to a linear code C. Then $H \in \mathbb{F}_q^{(n-k) \times n}$, defined by*

$$H^T = \left[\ b_1 \mid b_2 \mid \ldots \mid b_{n-k}\ \right]$$

is called a parity check matrix for C.

Remark 1.2. *As defined above, the generator and parity-check matrices for a code are not unique. However, the column span of G is unique (is equal to C), and so is the row-span of H. In many cases however, there is a canonical definition of the generator or parity-check matrix based on the construction of the code, which may be referred to as* the *generator or parity-check matrix.*

Since $G^T H^T = 0 \Leftrightarrow HG = 0$, we have $(HG)w = 0$ for all $x \in \mathbb{F}_q^k$, i.e., $Hx = 0$ for all $x \in C$. Moreover, since the columns of H^T are a basis for the null-space of G^T, we have that

$$x \in C \;\Leftrightarrow\; Hx = 0\,.$$

So the parity check matrix gives us a way to quickly check a codeword, by checking the parities of some bits of x (each row of H gives a parity constraint on x). Also, one can equivalently define a linear code by either giving G or the parity check matrix H.

Note that for linear codes, encoding $w \in F_2^k$ to the codeword $Gw \in C$ can always be done in polynomial time, by simply multiplying with the matrix G. Also, given $x \in C$, one can always find $w \in F_2^k$ such that $Gw = x$, either by Gaussian elimination, or (equivalently) by multiplying $x = Gw$ by an appropriate matrix G^* such that $G^*Gw = w$ for all $w \in \mathbb{F}_2^k$. Since we will only be concerned with polynomial time decoding in our discussion of codes, we can view the decoding problem as: given y which is a corruption of x, find x. The problem of going from $x \in \mathbb{F}_2^n$ to $w \in \mathbb{F}_2^k$ can always be solved for linear codes, as outlined above. Of course, if one is interested in a more fine-grained analysis of the decoding complexity, one needs to look carefully at the structure of the matrix G^*, but we will restrict our notion of efficiency to polynomial time.

1.1 Hamming Code

Consider the following code from $\mathbb{F}_2^4$ to $\mathbb{F}_2^7$, known as the Hamming Code.

Example 1.3. *Let $C : \mathbb{F}_2^4 \to \mathbb{F}_2^7$, where*

$$C(x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4, x_2 + x_3 + x_4, x_1 + x_3 + x_4, x_1 + x_2 + x_4).$$

Note that each element of the image is a linear function of the x_i's, i.e., one can express C with matrix multiplication as follows:

$$C(x_1, x_2, x_3, x_4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Example 1.4. *The parity check matrix of our example Hamming Code is:*

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Note that the i^{th} column is the integer i in binary. One can easily check that $HG = 0$.

Now suppose $x = (x_1, \ldots, x_7)^T$ is our codeword and we make a single error in the i^{th} entry. Then the output codeword with the error is

$$x + e_i = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_7 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

and $H(x + e_i) = Hx + He_i = He_i = H_i$, the i^{th} column of H, which reads i in binary. So this is a very efficient decoding algorithm just based on parity checking. Thus, the Hamming code can correct one *arbitrary* error in any position. One can generalize the Hamming code to larger message and block lengths, we can create a parity matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, where the i^{th} column reads i in binary.

2 Polar codes

We will briefly outline a beautiful recent construction of codes by Arikan [Ari09] (based on information-theoretic considerations) which also achieve capacity for the binary symmetric channel, and allow for encoding and decoding algorithms running in time $O(n \log n)$. We will discuss the basic intuition behind this construction, but will not discuss the details of the analysis. An excellent reference for a more thorough treatment, is the recent book by Guruswami, Rudra and Sudan [GRS12], also linked from the course webpage.

Recall that in the proof of achieving capacity using random codes, we claimed that for codes with rate $1 - H_2(p) - \varepsilon$, the error probability is small for block-length $n \geq C/\varepsilon^2$. In fact this dependence on ε can also be proved to be optimal. Arikan's "polar codes" also achieve this polynomial convergence to capacity, and one can show that for polar codes it suffices to take $n \geq C_0/\varepsilon^c$ for some constant c, although c is larger than 2. There has also been recent work on modifying the construction, to allow for a constant $c = 2 + \alpha$ for an arbitrary small $\alpha > 0$ [GRY20].

2.1 Codes via linear compression

We start with a reduction from the problem of designing codes for error-correction, to the problem of compressing a random input $z \sim (\mathsf{Bern}(p))^n$, so that with high probability, z can be recovered from its compression i.e., we want to compression and decompression maps $\mathsf{Com} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $\mathsf{Decom} : \mathbb{F}_2^m \to \mathbb{F}_2^n$, so that

$$\mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom}(\mathsf{Com}(Z)) \neq Z] \;\longrightarrow\; 0\,.$$

Moreover, we will want m to be nearly optimal i.e., $m \approx H(Z) = H_2(p) \cdot n$.

Note that the problem is the same as source coding if we allow for arbitrary compression schemes. However, we will require the compression map to be *linear*, which can specified as $z \mapsto Hz$ for an appropriate matrix H. Thus, the goal is to find a matrix $H \in \mathbb{F}_2^{m \times n}$ for $m \leq (H_2(p) + \varepsilon) \cdot n$, and a decompression map $\mathsf{Decom} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ such that

$$\mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom}(HZ) \neq Z] \longrightarrow 0 .$$

We claim that such a linear compression scheme immediately implies the existence of a code for the binary symmetric channel, with near-optimal rate.

Proposition 2.1. *Let $H \in \mathbb{F}_2^{m \times n}$ and $\mathsf{Decom} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ define a linear compression scheme as above. Then the linear code $C = \{x \mid Hx = 0\}$ has a decoding algorithm with vanishing probability of error, for transmission through the channel BSC(p).*

Proof: Recall that a codeword x transmitted through the binary symmetric channel, we can write the received word as $y = x + z$, where $z \sim (\mathsf{Bern}(p))^n$. We can then define the decoding algorithm as

$$\mathsf{Dec}(y) := y + \mathsf{Decom}(Hy) .$$

Note that $Hy = H(x+z) = Hz$ since $Hx = 0$. Also, if Hz is correctly decompressed to z, we indeed recover x, since $y + \mathsf{Decom}(Hz) = y + z = x$. Thus, we have

$$\forall x \in \mathbb{F}_2^n \quad \mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Dec}(x+Z) \neq x] = \mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom} HZ \neq Z] \longrightarrow 0 .$$

■

Since the code C above has $\dim(C) = n - m \geq (1 - H_2(p) - \varepsilon) \cdot n$ if $m \leq (H_2(p) + \varepsilon) \cdot n$, the above code has near-optimal rate, if the compression scheme is near-optimal.

2.2 Linear compression from entropy polarization

We now reduce from the problem of constructing a linear compression scheme, to designing an *invertible* matrix, such that the all the entropy of $Z \sim (\mathsf{Bern}(p))^n$ is "contained" only in m bits of $W = PZ$.

Before formalizing this, we briefly consider the reverse direction. Let $H \in \mathbb{F}_2^{m \times n}$ be a matrix defining a linear compression scheme, as discussed earlier. In particular, let $m \leq (H_2(p) + \varepsilon) \cdot n$ and let

$$\mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom}(HZ) \neq Z] \leq \delta .$$

Also, assume that H has full row-rank i.e., $\mathrm{im}(H) = \mathbb{F}_2^m$. Let $H' \in \mathbb{F}^{(n-m) \times n}$ be such that the rows of H and H' together span all of F_2^n. Then the matrix $P \in F_2^{n \times n}$ with first m rows from H and the last $n-m$ rows from H' is an invertible matrix. Thus, we have that for $W = PZ$

$$H(W) \;=\; H(Z) \;=\; n \cdot H_2(p)\,.$$

Note that $H(W)$ above denotes the entropy of the random variable W, and not the matrix H. Unfortunately, the common notation for both parity-check matrices and entropy, is H. However, in the rest of the lecture, we will only deal with a matrix P, and use $H(\cdot)$ to denote entropy. Using the chain rule, we can write the entropy of W as

$$n \cdot H_2(p) \;=\; H(W) \;=\; \sum_{i=1}^{m} H(W_i \mid W_{<i}) + \sum_{i=m+1}^{n} H(W_i \mid W_{<i}) \;=\; H(W_{\le m}) + H(W_{>m} \mid W_{\le m})\,.$$

Since the first m entries of W (corresponding to the first m rows of P) allow for a decompression of Z, we expect that $H(W_i \mid W_{<i}) \approx 0$ for all $i > m$, and hence $H(W_i \mid W_{<i}) \approx 1$ for $i \le m$. In fact, one can prove the following using Fano's inequality.

Exercise 2.2. *Prove that $H(W_{>m} \mid W_{\le m}) \;\le\; H_2(\delta) + \delta \cdot n$.*

Thus, while Z satisfies $H(Z_i \mid Z_{<i}) \;=\; H_2(p)$ for all i (since bits in Z are independent), entropies for the bits of W (after conditioning on previous bits) are very close to 0 or 1. This is the phenomenon, we refer to as *entropy polarization*. We now define it formally for arbitrary matrices P.

Definition 2.3. *An invertible matrix $P \in \mathbb{F}_2^{n \times n}$ is said to be (ε, τ)-*polarizing *for the random variable $Z \sim (\mathsf{Bern}(p))^n$ if for*

$$W = PZ \qquad \text{and} \qquad S_\tau \;=\; \{i \in [n] \mid H(W_i \mid W_{<i}) \ge \tau\}\;,$$

we have that $|S_\tau| \le (H_2(p) + \varepsilon) \cdot n$.

Check that the above bound on $|S_\tau|$ is nearly optimal, for small ε and τ.

Exercise 2.4. *Prove that for any invertible matrix P, we have $|S_\tau| \ge (H_2(p) - \tau) \cdot n$.*

Polarizing matrices imply linear compression schemes

While the argument we sketched above shows that linear compression schemes can be used to obtain polarizing matrices, we are more interested in the reverse direction. Below, we show that polarizing matrices can be used to obtain linear compression schemes, which will then reduce our problem of designing codes, to that of constructing polarizing matrices.

Given an (ε, τ)-polarizing matrix P and $S_\tau \;=\; \{i \in [n] \mid H(W_i \mid W_{<i}) \ge \tau\}$ as before (for $W = PZ$) we now define compression and decompression maps below.

- **Compression**: We take $\mathsf{Com}(Z) = Y = P_{S_\tau} Z$, where P_{S_τ} denotes the sub-matrix of P, restricted to the rows in S_τ. This defines a linear compression map $\mathsf{Com} : \mathbb{F}_2^n \to \mathbb{F}_2^{|S_\tau|}$.
- **Decompression**: Given $Y \in F_2^{|S_\tau|}$, we construct an estimator $\widehat{W}$ for $W = PZ$, and defined $\mathsf{Decom}(Y) = P^{-1}\widehat{W}$. The estimator $\widehat{W}$ is computed as below.

  For $i = 1$ to n

  - if $i \in S_\tau$, take $\widehat{W}_i = Y_i$.
  - else take $\widehat{W}_i = \mathrm{argmax}_{b \in \{0,1\}} \left\{ \mathbb{P}\left[W_i = b \mid W_{<i} = \widehat{W}_{<i} \right] \right\}$.

Thus, we take the compression map to be the entries of $W = PZ$ corresponding only to indices in S_τ. The decompression algorithm, computes an estimator $\widehat{W}$, which copies the bits in S_τ, and fills in the rest of the bits by choosing for each i the most likely value, given the estimate for the previous bits. Note that for an (ε, τ)-polarizing matrix, we have $|S_\tau| \leq (H_2 + \varepsilon) \cdot n$, and thus the compression is near-optimal as desired. We will argue that

$$\mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom}(\mathsf{Com}(Z)) \neq Z] \ \leq\ n \cdot \tau\,,$$

which will imply a compression scheme with vanishing error when $\tau = o(1/n)$. The proof the above bound is left as an exercise, which can be completed using the following.

Exercise 2.5. *Prove that*

$$\mathop{\mathbb{P}}_{Z \sim (\mathsf{Bern}(p))^n} [\mathsf{Decom}(\mathsf{Com}(Z)) \neq Z] \ \leq\ \sum_{i=1}^{n} \mathbb{P}\left[\widehat{W}_i \neq W_i \mid \widehat{W}_{<i} = W_{<i} \right] .$$

You may also need the following observation.

Exercise 2.6. *For a binary random variable X taking values in $\{0,1\}$, prove that*

$$H(X) \leq \alpha \quad \Rightarrow \quad \max\{\mathbb{P}[X=0], \mathbb{P}[X=1]\} \ \geq\ 1 - \alpha\,.$$

2.3 A small (slightly) polarizing matrix

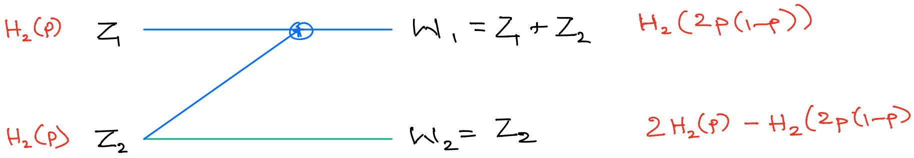
The polarizing matrix $P \in \mathbb{F}_2^{n \times n}$ will be constructed bu recursively applying a simple transform $P_2 \in \mathbb{F}_2^{2\times 2}$. Consider the transform

$$P_2 \ = \ \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{which maps} \quad \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \to \begin{bmatrix} Z_1 + Z_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$$

For $Z \sim (\mathsf{Bern}(p))^2$, we have that $H(Z_1) = H(Z_2 \mid Z_1) = H_2(p)$. On the other hand $W_1 = Z_1 + Z_2$ is 1 with probability $2p(1-p)$ and 0 otherwise. We thus have

$$\begin{aligned} H(W_1) &= H_2(2p(1-p)) & > H_2(p) \\ H(W_2 \mid W_1) &= 2H_2(p) - H_2(2p(1-p)) & < H_2(p)\,. \end{aligned}$$

Thus, the entropies $H(W_1)$ and $H(W_2 \mid W_1)$ are slightly closer to 1 and 0 respectively, compared to $H(Z_1)$ and $H(Z_2 \mid Z_1)$, assuming of course that $p \neq 1/2$ (otherwise there can be no compression and no good codes anyway). We represent this transformation pictorially as below.



2.4 Recursive construction of polarizing matrices

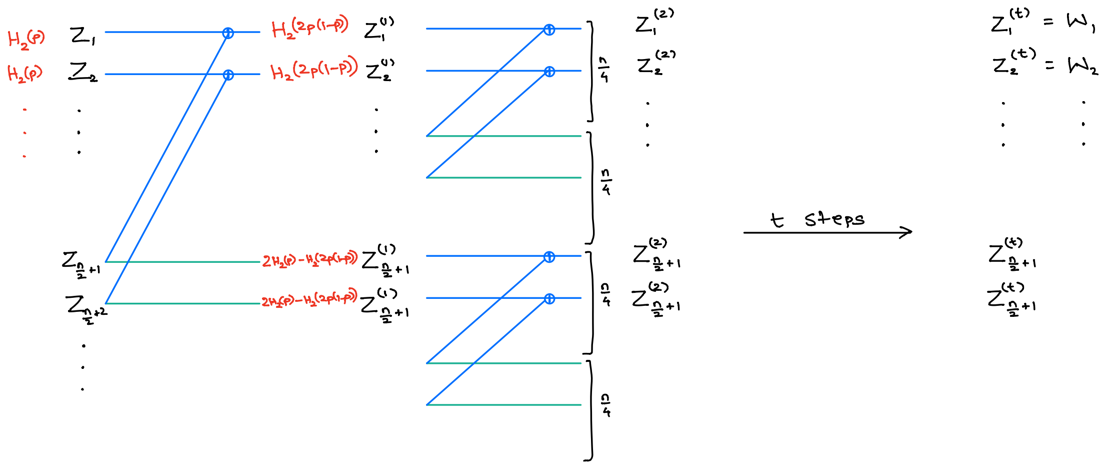
The final construction is obtained by recursing the above construction of 2×2 matrices. Let $n = 2^t$ for some $t \in \mathbb{N}$. We define the polarizing matrices recursively as

$$P_n = \begin{bmatrix} P_{n/2} & P_{n/2} \\ P_{n/2} & 0 \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

The following is a good exercise for understanding the recursive structure of the matrix.

Exercise 2.7. *Prove that for $z \in \mathbb{F}_2^n$, the multiplication Pz can be computed in time $O(n \log(n))$ where $P = P_n \in \mathbb{F}_2^{n \times n}$ is the matrix as defined above.*

The matrix can be thought of as a circuit, which applies the transformation P_2 in t layers as indicated in the following diagram.


$H_2(p)$ Z_1
$H_2(p)$ Z_2
$Z_{\frac{n}{2}+1}$
$Z_{\frac{n}{2}+2}$
$H_2(2p(1-p))$ $Z_1^{(1)}$
$H_2(2p(1-p))$ $Z_2^{(1)}$
$2H_2(p)-H_2(2p(1-p))$ $Z^{(1)}_{\frac{n}{2}+1}$
$2H_2(p)-H_2(2p(1-p))$ $Z^{(1)}_{\frac{n}{2}+1}$
$\frac{n}{4}$
$\frac{n}{4}$
$\frac{n}{4}$
$\frac{n}{4}$
$Z_1^{(2)}$
$Z_2^{(2)}$
$Z^{(2)}_{\frac{n}{2}+1}$
$Z^{(2)}_{\frac{n}{2}+1}$
t steps
$Z_1^{(t)} = W_1$
$Z_2^{(t)} = W_2$
$Z^{(t)}_{\frac{n}{2}+1}$
$Z^{(t)}_{\frac{n}{2}+1}$

We denote by $Z_i^{(j)}$ the random variables obtained in the j-th layer of the transformation. The analysis of polarization (which we will not be able to discuss) considers the sequence of random variables $X_j = H(Z_i \mid Z_{<i})$ for $j = 1, \ldots, t$, which tracks the entropy in a randomly chosen row i of the above diagram. One obtains the following result via a (somewhat involved) martingale analysis.

Theorem 2.8 (Speed of polarization)**.** *For all $\gamma > 0$, there exist constant $\alpha \in (0,1), \beta > 0$ such that for all $t \in \mathbb{N}$, we have*

$$\mathbb{P}\left[X_t \in (\gamma^t, 1-\gamma^t)\right] \;\leq\; \beta \cdot \alpha^t\,.$$

The important part of the above theorem is the freedom to choose γ, which lets us obtain a small τ in the (ε, τ)-polarization property. For example, choosing $\gamma = 1/4$, yields that (for $t = \log n$) the fraction of entropies in the interval $(1/n^2, 1-1/n^2)$ is small, which gives $\tau = 1/n^2 = o(1/n)$. Details of the above analysis can be found in [GRS12], which is also an excellent reference for coding theory in general.

References

[Ari09] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory*, 55(7):3051–3073, 2009. 3

[GRS12] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential Coding Theory. 2012. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/index.html. 3, 8

[GRY20] Venkatesan Guruswami, Andrii Riazanov, and Min Ye. Arikan meets Shannon: Polar codes with near-optimal convergence to channel capacity. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 552–564, 2020. 3