CS 229r Essential Coding Theory	February 3, 2020
Lecture 3	
Instructor: Madhu Sudan	*Scribe: Kenz Kallal*

1 Converse to Shannon's theorem

Last lecture, we started considering the binary symmetric channel with parameter p.

Definition 1.1. Let $p \in [0,1]$. The *binary symmetric channel with parameter p* is the probabilistic function $BSC_p : \{0,1\}^n \to \{0,1\}^n$ which flips each bit independently with probability p.

Recall that the point is that you are trying to communicate over the binary symmetric channel, and you'd like to be able to do so with error probability that goes to zero (hopefully very fast) as $n \to \infty$. In formal terms, for each integer $n \geq 1$, we need another positive integer k_n (hopefully going to infinity as $n \to \infty$) and an encoding function

$$E_n : \{0,1\}^{k_n} \to \{0,1\}^n$$

plus a decoding function

$$D_n : \{0,1\}^n \to \{0,1\}^{k_n}.$$

The process of communication over the binary symmetric channel is as follows: a message $m \in \{0,1\}^{k_n}$ (which we randomize over when we consider the error probability) gets sent to $E_n(m) \in \{0,1\}^n$, which the other side reads as

$$BSC_p(E_n(m))$$

(this depends probabilistically on the BSC, as well as m if we choose to also randomize over the message). Then the other side tries to decode this by applying D_n, so we say that the message m gets decoded correctly if

$$m = D_n(BSP_p(E_n(m))).$$

To reiterate:

Definition 1.2. The *error probability* over a uniformly distributed message $m \in \{0,1\}^{k_n}$ of a code (E_n, D_n) is

$$\Pr_{m, BSC_p}\left[D_n(BSC_p(E_n(m))) \neq m\right].$$

For a fixed message m (which we will not consider except for in the exercises), it is defined the same way except the probability is taken only over the randomization from the binary symmetric channel (since m is taken to be fixed).

The result of Shannon [Sha48] which is a main topic of today and yesterday's lectures is that the binary symmetric channel has "capacity" $1 - H(p)$ where H is the classical binary entropy function. There are two parts to this statement:

1. We can accomplish error-probability going to zero as $n \to \infty$, while at the same time having

$$\liminf_{n\to\infty} \frac{k_n}{n}$$

   arbitrarily close to $1 - H(p)$.

2. It is impossible to accomplish this if

$$\liminf_{n\to\infty} \frac{k_n}{n} > 1 - H(p).$$

Last class, we proved a slightly more quantitative version of the first part, which is stated as follows:

Theorem 1.3. *For all $p \in [0,1]$, and $\epsilon > 0$, for sufficiently large n there exist k_n and coding schemes (E_n, D_n) [with E_n taking k_n bits to n bits and vice versa for D_n as explained above] such that*

1. $k \geq (1 - H(p) - \epsilon)n$
2. $\Pr_{m, BSC_p}[D_n(BSC_p(E_n(m))) \neq m] \leq \exp(-O_{p,\epsilon}(n))$.

Proof. This was done last class; recall that the key step was to choose the coding scheme randomly in a natural way and use the probabilistic method. □

NB: asking the rate $\liminf \frac{k_n}{n}$ to be closer to $1 - H(p)$, i.e. decreasing ϵ, requires the error probability (or at least the bound on it from Theorem 1.3) to decrease slower (though for any fixed $\epsilon > 0$ and p it goes to 0 exponentially in n; this is the meaning of the notation $O_{p,\epsilon}(n)$).

Exercise 1.4. Note that the error probability being bounded by Theorem 1.3 as $n \to \infty$ is also over a uniform random message $m \in \{0,1\}^{k_n}$. Is the result of Theorem 1.3 still true if we instead look at a fixed m and the error probability where the only randomness comes from the binary symmetric channel? (prove or disprove and salvage if possible). NB: this is much better because it guarantees that no matter what we want to send, the error probability is small. Otherwise it might be true that most inputs in $\{0,1\}^{k_n}$ have small probability of error, but some could be much more likely.

Now we move on to the second claim: how do we know that we can't achieve a rate better than $1 - H(p)$? This is the "converse" part of Shannon's result that we promised to cover today. Here is the converse result and its proof.

Theorem 1.5. *For all $p \in [0,1]$ and $\epsilon > 0$, for all sufficiently large n, any encoding schemes (E_n, D_n) [between $\{0,1\}^{k_n}$ and $\{0,1\}^n$] with the property that $k_n \geq (1 - H(p) + \epsilon)n$ must have very bad error probability, in the sense that*

$$\Pr_{m, BSC_p}[D_n(BSC_p(E_n(m))) = m] \leq \exp(-O_{p,\epsilon}(n)).$$

Proof. First, as usual we throw away some atypical cases. In particular, the number of errors (which we call ℓ in this proof) introduced to the encoded message $E_n(m) \in \{0,1\}^n$ by BSC_p is a sum of n independent 0-1 random variables each with expectation p. So by a Chernoff bound,

$$\Pr_{BSC_p}[\ell \notin [(p - \epsilon)n, (p + \epsilon)n]] \leq \exp(-O_{p,\epsilon}(n)).$$

This means that we can condition on $(p - \epsilon)n \leq \ell \leq (p + \epsilon)n$ without affecting the error probability by anything that will change the type of the bound we want.

The second half of the argument is short but somewhat subtle. It is based on the following fact: For $m \in \{0,1\}^{k_n}$, the event of m being decoded correctly is equivalent to the event that $D_n(r) = m$, where r is the message which is actually received, namely $r = BSC_p(E_n(m))$ (this is just restating Definition 1.2). This is in turn equivalent to the event that there exists an $r \in \{0,1\}^n$ such that $D(r) = m$ and r is received[1]. By the union bound, it follows that

$$\Pr_{m, BSC_p}[D_n(BSC_p(E_n(m))) = m | \ell = \ell_0] \leq \sum_{r \in \{0,1\}^n} \Pr_{m, BSC_p}[D(r) = m \text{ and } r = BSC_p(E_n(m)) | \ell = \ell_0]. \quad (1)$$

But for any fixed $r \in \{0,1\}^n$, $D(r)$ is fixed, and therefore (since m is uniformly distributed on $\{0,1\}^{k_n}$) the probability that $D_n(r) = m$ is just 2^{-k_n}. Conditioned on $\ell = \ell_0$ and $D_n(r) = m$, the probability that $r = BSC_p(E_n(m))$ is the probability that $r = BSC_p(E_n(D_n(r)))$, which is at most $\frac{1}{\binom{n}{\ell_0}}$ since at most one

[1] This is obvious but maybe somewhat counterintuitive since there is really at most one possible value for r, namely $BSC_p(E_n(m))$ so it's maybe a somewhat clever trick to think about it this way.

of the ways to flip exactly ℓ_0 bits of $E_n(D_n(r))$ actually results in r (in particular all of those ways result in a different string and are equally likely[2]). So we have for any $r \in \{0,1\}^n$,

$$\begin{aligned}
&\Pr_{m,BSC_p}[D(r) = m \text{ and } r = BSC_p(E_n(m))|\ell = \ell_0] \\
&\qquad = \Pr_{m,BSC_p}[D(r) = m|\ell = \ell_0] \Pr_{m,BSC_p}[r = BSC_p(E_n(m))|D(r) = m \text{ and } \ell = \ell_0] \\
&\qquad = \Pr_{m}[D(r) = m] \Pr_{BSC_p}[r = BSC_p(E_n(D(r)))|\ell = \ell_0] \\
&\qquad \leq \frac{1}{2^{k_n}} \cdot \frac{1}{\binom{n}{\ell_0}}
\end{aligned}$$

where in the middle we have slightly simplified the expressions by removing conditions (and things being randomized over) with which the events in question has no dependence on, and used the logic from above for computing the two probabilities before the last line. It follows from Eq. (1) that we can exploit this to bound the probability of correctly decoding by

$$\Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|\ell = \ell_0] \leq \sum_{r \in \{0,1\}^n} \frac{1}{2^{k_n}\binom{n}{\ell_0}} = 2^{n-k_n}\binom{n}{\ell_0}^{-1}. \tag{2}$$

Now we combine this with the first part of the proof, which said that $\Pr_{BSC_p}[\ell \neq [(p-\epsilon)n, (p+\epsilon)n]] \leq \exp(-O_{p,\epsilon}(n))$. The point of this is that it means it suffices to show that

$$\Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|(p-\epsilon)n \leq \ell \leq (p+\epsilon)n] \leq \exp(-O_{p,\epsilon}(n)),$$

since $\Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m]$ is equal to $\Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|(p-\epsilon)n \leq \ell \leq (p+\epsilon)n]$ times $\Pr_{BSC_p}[(p-\epsilon)n \leq \ell \leq (p+\epsilon)n]$ plus some other probability (the same one as before except conditioned on ℓ being outside the typical range) times $\Pr_{BSC_p}[\ell \notin [(p-\epsilon)n, (p+\epsilon)n]] < \exp(-O_{p,\epsilon}(n))$; this means that since all probabilities are at most 1, proving this would give us a bound of $2\exp(-O_\epsilon(n)) = \exp(-O_{p,\epsilon}(n))$ as desired. This bound follows directly from what we have already done, namely Eq. (2):

$$\begin{aligned}
&Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|(p-\epsilon)n \leq \ell \leq (p+\epsilon)n] \\
&\qquad = \sum_{(p-\epsilon)n \leq \ell_0 \leq (p+\epsilon)n} \Pr_{BSC_p}[\ell = \ell_0] Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|\ell = \ell_0] \\
&\qquad \leq \sum_{(p-\epsilon)n \leq \ell_0 \leq (p+\epsilon)n} Pr_{m,BSC_p}[D_n(BSC_p(E_n(m))) = m|\ell = \ell_0] \\
&\qquad \leq (2\epsilon n + 2)2^{n-k_n} \left(\min_{(p-\epsilon)n \leq \ell_0 \leq (p+\epsilon)n} \binom{n}{\ell_0} \right)^{-1} \\
&\qquad \leq (2\epsilon n + 2)2^{n-k_n} 2^{-(1+o(1))H(p-\epsilon)n}
\end{aligned}$$

where in the last step we have assumed without loss of generality that $p < 1/2$ so that $H(p+\epsilon) > H(p-\epsilon)$, and we have used the Stirling approximation. This bound might be completely trivial, except for the fact that we have assumed that the rate of the code is good enough that $k_n \geq (1 - H(p) + \epsilon)n$, which means that our bound on the probability of success is at most

$$(2\epsilon n + 2)2^{(H(p)-\epsilon)n} 2^{-(1+o(1))H(p-\epsilon)n}.$$

By the convexity of $H(p)$, we know that $H(p-\epsilon) > H(p) - \epsilon$, so we write $H(p-\epsilon) = H(p) - \epsilon'$ where $\epsilon' < \epsilon$ is positive, and see that our bound is now

$$(2\epsilon n + 2)2^{H(p)n - \epsilon n - H(p)n + \epsilon' n - o(1)H(p-\epsilon)n} = (2\epsilon n + 2)2^{(\epsilon' - \epsilon - o(1)H(p-\epsilon))n}$$

[2]Technically you need to check the following trivial fact: the distribution of the error (represented as a string in $\{0,1\}^n$) from BSC_p applied to an element of length n, conditioned on there being ℓ_0 total errors, is the same as the uniform distribution on the set of all elements of $\{0,1\}^n$ where the sum of the digits is ℓ_0.

which is clearly $\leq \exp(-O_{p,\epsilon}(n))$ as desired. □

Exercise 1.6. Work out the details of the Stirling estimate at the end of the proof of Theorem 1.5. In particular, we have used the fact that

$$\binom{n}{\lfloor pn \rfloor} = 2^{(1+o(1))H(p)}$$

as $n \to \infty$

Exercise 1.7. Is it possible to modify Theorem 1.5 to make sure such high probability of error happens for each possible input (and not just a random input)?

Note that we have not yet provided any algorithmic way to construct the codes whose existence is guaranteed by the first part of Shannon's theorem. A more natural setting for this is the adversarial error model of Hamming [Ham50], which we have already started discussing.

2 More bounds on error-correcting codes: Gilbert, Singleton, and Hamming

Now we are back in the adversarial error model. Let q be a positive integer, Σ an alphabet with q letters, and $E : \Sigma^k \to \Sigma^n$ an injective encoding function. Recall that the image of E is the set of *codewords* of the code, which we call $C \subset \Sigma^n$. Since E is injective, $|C| = q^k$. We also defined the *distance* of the code,

$$d = \Delta(C) = \min_{x \neq y \in C} \Delta(x, y)$$

where $\Delta(x, y)$ denotes the usual Hamming distance. A code with those invariants is called an $(n, k, d)_q$ code. Its *rate* is $R := k/n$, and its *normalized distance* is $\delta := d/n$.

Remark In reality, we really want to know the rate and normalized distance of a *family* of codes, namely a sequence of codes (E_n, D_n) which encode k_n bits as n bits for a sequence of positive integers $n \to \infty$. Then the rate is really

$$R := \liminf_{n \to \infty} \frac{k_n}{n}$$

and the normalized distance is really

$$\delta := \liminf_{n \to \infty} \frac{d_n}{n}.$$

Compare this to the definition of "capacity" from Section 1. As we have already seen, the distinction between a single code and a family thereof will just be a matter of constructing n-bit codes for infinitely many n (which we want to do anyway) and keeping track of the asymptotics of k_n and d_n.

We can naturally ask:

Question 2.1. Which pairs $(R, \delta) \in [0, 1] \times [0, 1]$ are achievable by error-correcting codes?

Exercise 2.2. Explicitly construct codes achieving $(R, \delta) = (0, 1)$ and $(1, 0)$. This shows that the x- and y-axes are achievable.

On the flipside of this exercise, one of the main goals is to be able to construct *asymptotically good codes*:

Definition 2.3. A family of codes is *asymptotically good* if $R > 0$ and $\delta > 0$.

Exercise 2.4. Prove by taking random codes in $\{0, 1\}^n$ with $2^{\epsilon^2 n/100}$ codewords that we can achieve $\delta \geq \frac{1}{2} - \epsilon$.

The first result we present about the achievable region is due to Gilbert [Gil52].

Theorem 2.5 (Gilbert's theorem)**.** *Let n and d be positive integers. Then there exists a code $C \subset \{0,1\}^n$ with distance at least d such that*

$$|C| \geq \frac{2^n}{\mathrm{Vol}_2(n, d-1)}.$$

Proof. The proof is by greedy construction: just pick one element of $\{0,1\}^n$ at a time, eliminating the closed Hamming ball of radius $d-1$ around that point every time. Stop once there are no elements left (this clearly terminates). This way, each chosen element is guaranteed to be distance at least d away from each previously chosen element (it is not in the set of elements of distance $\leq d-1$ away, so it is of distance at least d), so our code C (consisting of all the chosen elements) indeed has distance at least d. To bound $|C|$, just notice that taking away the closed Hamming ball of radius $d-1$ takes away at most $\mathrm{Vol}_2(n, d-1)$ additional elements of $\{0,1\}^n$, so the number of elements we are able to choose (i.e. the number of steps in the greedy algorithm, i.e. the number of balls removed until there are no elements left) is at least

$$\frac{|\{0,1\}^n|}{\mathrm{Vol}_2(n, d-1)} = \frac{2^n}{\mathrm{Vol}_2(n, d-1)}$$

□

Corollary 2.6. *The region $R \leq 1 - H(\delta)$ is achievable for $\delta < 1/2$.*

Proof. This is just because of the asymptotic expression

$$\liminf_{n\to\infty} \frac{\log_2 |C|}{n} \geq 1 - H(\delta)$$

because of the fact that

$$\mathrm{Vol}_2(n, d-1) \leq 2^{nH((d-1)/n)}.$$

So if we take $d = \delta n$ then we can construct a family of codes with normalized distance δ and rate at least $1 - H(\delta)$ [NB: the only reason we must take $\delta < 1/2$ is because our bound for the volume of the Hamming ball requires it]. Finally, recall that if (R, δ) is achievable then so is (R', δ') if $R' \leq R$ and $\delta' \leq \delta$. □

Exercise 2.7. Here we considered $q = 2$. What happens if you try and repeat the argument of Theorem 2.5 for alphabets of size $q > 2$? The same exercise applies to everything that follows this.

So already Theorem 2.5 guarantees the existence of many asymptotically good codes. Next, we present a result due to Singleton [Sin64], which cuts the admissible region in $[0,1] \times [0,1]$.

Theorem 2.8 (the Singleton bound)**.** *Let Σ be an arbitrary alphabet with q letters, and $E : \Sigma^k \to \Sigma^n$ an injective map inducing a code of distance d. Then $d \leq n - k + 1$.*

Proof. This is an easy consequence of the pidgeonhole principle. Consider the projection $\pi : \Sigma^n \to \Sigma^{k-1}$, say, to the first $k-1$ coordinates. Then

$$\pi \circ E : \Sigma^k \to \Sigma^{k-1}$$

is a map of sets from a set of size q^k to a set of size q^{k-1}. So by the pidgeonhole principle, there must exist two distinct $m, m' \in \Sigma^k$ such that $\pi(E(m)) = \pi(E(m'))$, in other words $E(m)$ and $E(m')$ agree in the first $k-1$ coordinates. It follows that the distance between the two codewords $E(m)$ and $E(m')$ is at most $n - (k-1) = n - k + 1$. As a result, the distance of the code induced by E is

$$d = \min_{x \neq y \in E(\Sigma^k)} \Delta(x, y) \leq \Delta(E(m), E(m')) \leq n - k + 1$$

as desired. □

Corollary 2.9. *The region $\delta < R$ is not achievable.*

Returning to the case $q = 2$, we recall Hamming's sphere packing bound, and see that it yields an improvement on the understanding of the achievable region from Theorem 2.9.

Theorem 2.10 (the Hamming bound)**.** *Suppose $C \subseteq \{0,1\}^n$ is a code of distance d. Then*

$$|C| \leq \frac{2^n}{\mathrm{Vol}_2\left(n, \lfloor \frac{d-1}{2} \rfloor\right)}.$$

Proof. This is another sphere-packing argument (in fact we probably already wrote this exact bound down in a previous lecture). Around each codeword in C, we can draw a Hamming ball of radius $(d-1)/2$, and be guaranteed (by definition of the distance d) that these balls are nonoverlapping. As a result, their volumes (all equal to $\mathrm{Vol}_2(n, \lfloor \frac{d-1}{2} \rfloor)$) add up to be at most $|\{0,1\}^n| = 2^n$. There are $|C|$ of these nonoverlapping balls, which immediately gives the desired inequality. □

Corollary 2.11. *The region $R > 1 - H(\delta/2)$ is not achievable.*

Proof. Same as Corollary 2.6. □

3 Linear Codes

At the end of today's lecture, we start the discussion on linear codes (to be continued next time). The goal for now is to come up with a linear version of Gilbert's construction in Theorem 2.5, in order to achieve a better rate. The result we are going towards is due to Varshamov [Var57], and is the same as Theorem 2.5 except the $d-1$ is replaced with $d-2$:

Theorem 3.1. *the Varshamov bound Let n, d be positive integers. Then there exists a code $C \subset \{0,1\}^n$ of distance at least d and*

$$|C| \geq \frac{2^n}{\mathrm{Vol}_2(n, d-2)}.$$

The way this is done is by constructing C as a linear code over the finite field $\mathbf{F}_2$. To do linear algebra over finite fields, we first need to answer the basic question from algebra

Question 3.2. What are the finite fields?

I will completely answer this question in a series of exercises. I will assume we know what the following definitions are:

- Fields and extensions of fields;
- The definition of a separable polynomial;
- The definition of a splitting field, and the fact that they are unique up to isomorphism.

Exercise 3.3. Construct a finite field of order p for each prime p. Call it $\mathbf{F}_p$.

Exercise 3.4. Show that $\mathbf{F}_p$ is the only finite field of order p, up to isomorphism.

Exercise 3.5. Let F, L be fields, and $f : F \to L$ a nonzero ring homomorphism between them. Show that f is injective.

Exercise 3.6. Show that every finite field F of characteristic p admits an injective homomorphism $\mathbf{F}_p \to F$. Hint: this is tautological if you know what the characteristic of a field is.

So in order to understand the finite fields, it suffices to understand the finite extensions of $\mathbf{F}_p$.

Exercise 3.7. Let F be a finite field. Show that $|F|$ must be a power of a prime.

Exercise 3.8. Let F be a finite field of characteristic p. Show that the map $x \mapsto x^p$ is an automorphism of F.

An extremely useful way to go from the finite field $\mathbf{F}_p$ to the finite fields containing it is to use the Frobenius automorphism. Here is how the picture goes: the algebraic closure $\overline{\mathbf{F}}_p$ is the union of all the finite fields containing $\mathbf{F}_p$. The (Galois) automorphism group $\mathrm{Aut}(\overline{\mathbf{F}}_p/\mathbf{F}_p)$ is topologically generated by the Frobenius element $x \mapsto x^p$. In particular, this is why we are able to construct the finite extensions of $\mathbf{F}_p$ by taking fixed fields of powers of the Frobenius. A less abstract way to say this is that if F is a degree-n extension of $\mathbf{F}_p$, then $\mathrm{Aut}(F/\mathbf{F}_p)$ should be cyclic of order n and generated by the Frobenius. In particular, we know that if $F/\mathbf{F}_p$ is of degree n, then the n-th power of the Frobenius, i.e. $x \mapsto x^{p^n}$, should be trivial. This is because the multiplicative group of any finite field is cyclic. In fact, that is a good exercise:

Exercise 3.9. Let F be any field, and G a finite subgroup of $F^\times$. Then G is cyclic. Hint: you might find it helpful to use the structure theorem for finite abelian groups: $G \cong \mathbf{Z}/a_1\mathbf{Z} \times \cdots \mathbf{Z}/a_r\mathbf{Z}$ for some positive integers $a_1, \ldots, a_r$.

So we might expect the finite extensions of $\mathbf{F}_p$ to just be the zero sets in an algebraic closure of polynomials like $x^{p^n} - x$. One elegant way to go about proving their existence and uniqueness is to construct them as splitting fields of these polynomials.

Exercise 3.10. Show that the polynomial $X^{p^n} - X \in \mathbf{F}_p[X]$ is separable. Hint: look at the derivative.

Exercise 3.11. Let $F/\mathbf{F}_p$ be the splitting field for $X^{p^n} - X$. Show that every element of F is a root of $X^{p^n} - X$, and therefore (by the previous exercise and the definition of a splitting field), F is equal to the set of roots of $X^{p^n} - X$, and thus $|F| = p^n$.

Congratulations! You have constructed a finite field of size p^n for every prime p and positive integer n. You've also shown that the finite field of size p is unique up to isomorphism. For the finite fields of size p^n, the uniqueness is for the following reason:

Exercise 3.12. Let $K/\mathbf{F}_p$ be a finite extension of degree n. Show that K is a splitting field for $X^{p^n} - X$. Hint: you've already done this.

Since splitting fields are unique up to isomorphism, this implies that the finite field of size p^n is unique up to isomorphism. Note that $\mathbf{F}_{p^n}$ is NOT the same thing as $\mathbf{Z}/p^n\mathbf{Z}$, which cannot be given the structure of a field because it has zero-divisors. One elegant way to construct the finite field of order p^n is to take $\mathbf{F}_[X]$ and quotient out by an irreducible polynomial of degree n.

Now we come back to the topic of linear codes. The point is as follows. Let $\mathbf{F}_q$ be the finite field of size q, where q is a prime power. A *linear code* is a code with alphabet $\Sigma = \mathbf{F}_q$, where the encoding map

$$E : \Sigma^k \to \Sigma^n$$

is linear as a map of $\mathbf{F}_q$-vector spaces. For today, we just make two observations.

Lemma 3.13. *Let C be the linear code induced by E. Then*

$$\Delta(C) = \min_{x \neq 0 \in C} \Delta(x, 0).$$

Proof. This is because $\Delta(x, y) = \Delta(x - y, 0)$. The fact that $C = E(\mathbf{F}_q^k)$ means that it is a linear subspace of $\mathbf{F}_q^n$, which means that the set of elements $x - y$ for $x, y \in C$ is C again, so the result follows from the definition of $\Delta(C)$. □

Also, given the map of vector spaces $E : \mathbf{F}_q^k \to \mathbf{F}_q^n$, we can consider the resulting projection map $H : \mathbf{F}_q^n \to \mathrm{coker}(E) = \mathbf{F}_q^n/E(\mathbf{F}_q^k)$ which is surjective and therefore has rank $n - k$. By the definition of the projection, we have

Lemma 3.14. *The code $C = \{E(x) : x \in \mathbf{F}_q^k\}$ is also equal to the kernel of H.*

Note that we explained how to compute H in problem set 0. In more concrete terms (also in language which is dual to what I've written here), we view E as a $k \times n$ matrix of rank k whose action is given on row vectors $x \in \mathbf{F}_q^k$ by $x \mapsto x \cdot E$. Problem set 0 shows how to compute an $n \times (n-k)$ matrix H of rank $n-k$ such that $G \cdot H = 0$.

References

[Gil52] E.N. Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

[Ham50] R.W. Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[Sin64] R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[Var57] R.R. Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.