Foreword

This chapter is based on lecture notes from coding theory courses taught by Venkatesan Guruswami at University at Washington and CMU; by Atri Rudra at University at Buffalo, SUNY and by Madhu Sudan at MIT.

This version is dated **February 5, 2017**. For the latest version, please go to

http://www.cse.buffalo.edu/ atri/courses/coding-theory/book/

The material in this chapter is supported in part by the National Science Foundation under CAREER grant CCF-0844796. Any opinions, findings and conclusions or recomendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Chapter 4

What Can and Cannot Be Done-I

In this chapter, we will try to tackle Question 2.5.1. We will approach this trade-off in the following way:

> If we fix the relative distance of the code to be δ, what is the best rate R that we can achieve?

Note that an upper bound on R is a *negative* result, while a lower bound on R is a *positive* result.

In this chapter, we will consider only one positive result, i.e. a lower bound on R called the Gilbert-Varshamov bound in Section 4.2. In Section 4.1, we recall a negative result that we have already seen– Hamming bound and state its asymptotic version to obtain an upper bound on R. We will consider two other upper bounds: the Singleton bound (Section 4.3), which gives a tight upper bound for large enough alphabets (but not binary codes) and the Plotkin bound (Section 4.4).

4.1 Asymptotic Version of the Hamming Bound

We have already seen an upper bound in Section 1.7 due to Hamming. However, we had stated this as an upper bound on the dimension k in terms of n, q and d. We begin by considering the trade-off between R and δ given by the Hamming bound. Recall that Theorem 1.7.1 states the following:

$$\frac{k}{n} \leq 1 - \frac{\log_q Vol_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right)}{n}$$

Recall that Proposition 3.3.1 states following lower bound on the volume of a Hamming ball:

$$Vol_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) \geq q^{H_q\left(\frac{\delta}{2}\right)n - o(n)},$$

which implies the following asymptotic version of the Hamming bound:

$$R \leq 1 - H_q\left(\frac{\delta}{2}\right) + o(1).$$

See Figure 4.1 for a pictorial description of the Hamming bound for binary codes.

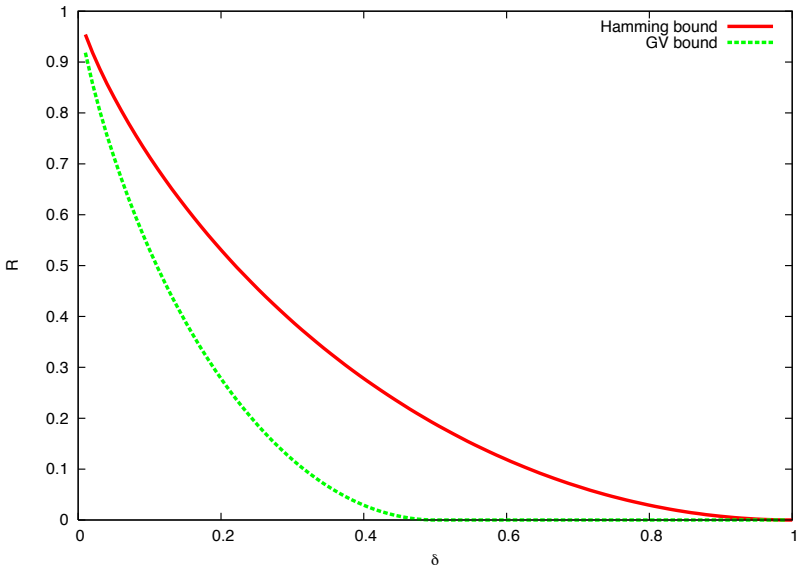

Figure 4.1: The Hamming and GV bounds for binary codes. Note that any point below the GV bound is achievable by some code while no point above the Hamming bound is achievable by any code. In this part of the book we would like to push the GV bound as much up as possible while at the same time try and push down the Hamming bound as much as possible.

4.2 Gilbert-Varshamov Bound

Next, we will switch gears by proving our first non-trivial lower bound on R in terms of δ. (In fact, this is the only positive result on the R vs δ tradeoff question that we will see in this book.) In particular, we will prove the following result:

Theorem 4.2.1 (Gilbert-Varshamov Bound)**.** *Let $q \geq 2$. For every $0 \leq \delta < 1 - \frac{1}{q}$, and $0 < \varepsilon \leq 1 - H_q(\delta)$, there exists a code with rate $R \geq 1 - H_q(\delta) - \varepsilon$ and relative distance δ.*

The bound is generally referred to as the GV bound. For a pictorial description of the GV bound for binary codes, see Figure 4.1. We will present the proofs for general codes and linear codes in Sections 4.2.1 and 4.2.2 respectively.

4.2.1 Greedy Construction

We will prove Theorem 4.2.1 for general codes by the following greedy construction (where $d = \delta n$): start with the empty code C and then keep on adding vectors not in C that are at Hamming distance at least d from all the existing codewords in C. Algorithm 5 presents a formal description of the algorithm and Figure 4.2 illustrates the first few executions of this algorithm.

We claim that Algorithm 5 terminates and the C that it outputs has distance d. The latter is true by step 2, which makes sure that in Step 3 we never add a vector $\mathbf{c}$ that will make the distance of C fall below d. For the former claim, note that, if we cannot add $\mathbf{v}$ at some point, we

Algorithm 5 Gilbert's Greedy Code Construction

INPUT: n, q, d

OUTPUT: A code $C \subseteq [q]^n$ of distance d

1: $C \leftarrow \emptyset$
2: WHILE there exists a $\mathbf{v} \in [q]^n$ such that $\Delta(\mathbf{v}, \mathbf{c}) \geq d$ for every $\mathbf{c} \in C$ DO
3: Add $\mathbf{v}$ to C
4: RETURN C

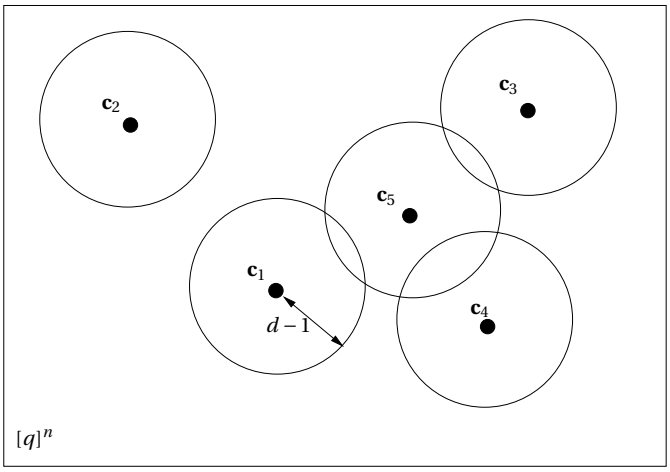

Figure 4.2: An illustration of Gilbert's greedy algorithm (Algorithm 5) for the first five iterations.

cannot add it later. Indeed, since we only add vectors to C, if a vector $\mathbf{v} \in [q]^n$ is ruled out in a certain iteration of Step 2 because $\Delta(\mathbf{c}, \mathbf{v}) < d$, then in all future iterations, we have $\Delta(\mathbf{v}, \mathbf{c}) < d$ and thus, this $\mathbf{v}$ will never be added in Step 3 in any future iteration.

The running time of Algorithm 5 is $q^{O(n)}$. To see this note that Step 2 in the worst-case could be repeated for every vector in $[q]^n$, that is at most q^n times. In a naive implementation, for each iteration, we cycle through all vectors in $[q]^n$ and for each vector $\mathbf{v} \in [q]^n$, iterate through all (at most q^n) vectors $\mathbf{c} \in C$ to check whether $\Delta(\mathbf{c}, \mathbf{v}) < d$. If no such $\mathbf{c}$ exists, then we add $\mathbf{v}$ to C otherwise, we move to the next $\mathbf{v}$. However, note that we can do slightly better– since we know that once a $\mathbf{v}$ is "rejected" in an iteration, it'll keep on being rejected in the future iterations, we can fix up an ordering of vectors in $[q]^n$ and for each vector $\mathbf{v}$ in this order, check whether it can be added to C or not. If so, we add $\mathbf{v}$ to C, else we move to the next vector in the order. This algorithm has time complexity $O(nq^{2n})$, which is still $q^{O(n)}$.

Further, we claim that after termination of Algorithm 5

$$\bigcup_{\mathbf{c} \in C} B(\mathbf{c}, d-1) = [q]^n.$$

This is because if not, then there exists a vector $\mathbf{v} \in [q]^n \setminus C$, such that $\Delta(\mathbf{v}, \mathbf{c}) \geq d$ and hence $\mathbf{v}$ can

be added to C. However, this contradicts the fact that Algorithm 5 has terminated. Therefore,

$$\left|\bigcup_{\mathbf{c}\in C} B(\mathbf{c}, d-1)\right| = q^n. \tag{4.1}$$

It is not too hard to see that

$$\sum_{\mathbf{c}\in C} |B(\mathbf{c}, d-1)| \geq \left|\bigcup_{\mathbf{c}\in C} B(\mathbf{c}, d-1)\right|,$$

which by (4.1) implies that

$$\sum_{\mathbf{c}\in C} |B(\mathbf{c}, d-1)| \geq q^n$$

or since the volume of a Hamming ball is translation invariant,

$$\sum_{\mathbf{c}\in C} Vol_q(d-1, n) \geq q^n.$$

Since $\sum_{\mathbf{c}\in C} Vol_q(d-1,n) = Vol_q(d-1,n)\cdot|C|$, we have

$$\begin{aligned} |C| &\geq \frac{q^n}{Vol_q(d-1,n)} \\ &\geq \frac{q^n}{q^{nH_q(\delta)}} \qquad (4.2) \\ &= q^{n(1-H_q(\delta))}, \end{aligned}$$

as desired. In the above, (4.2) follows from the fact that

$$\begin{aligned} Vol_q(d-1,n) &\leq Vol_q(\delta n, n) \\ &\leq q^{nH_q(\delta)}, \end{aligned} \tag{4.3}$$

where the second inequality follows from the upper bound on the volume of a Hamming ball in Proposition 3.3.1.

It is worth noting that the code from Algorithm 5 is not guaranteed to have any special structure. In particular, even storing the code can take exponential space. We have seen in Proposition 2.3.1 that linear codes have a much more succinct representation. Thus, a natural question is:

Question 4.2.1. *Do linear codes achieve the $R \geq 1 - H_q(\delta)$ tradeoff that the greedy construction achieves?*

Next, we will answer the question in the affirmative.

4.2.2 Linear Code Construction

Now we will show that a random linear code, with high probability, lies on the GV bound. The construction is a use of the probabilistic method (Section 3.2).

By Proposition 2.3.4, we are done if we can show that there exists a $k \times n$ matrix $\mathbf{G}$ of full rank (for $k = (1 - H_q(\delta) - \varepsilon)n$) such that

$$\text{For every } \mathbf{m} \in \mathbb{F}_q^k \setminus \{\mathbf{0}\}, wt(\mathbf{mG}) \geq d.$$

We will prove the existence of such a $\mathbf{G}$ by the probabilistic method. Pick a random linear code by picking a random $k \times n$ matrix $\mathbf{G}$ where each of kn entries is chosen uniformly and independently at random from $\mathbb{F}_q$. Fix $\mathbf{m} \in \mathbb{F}_q^k \setminus \{\mathbf{0}\}$. Recall that by Lemma 3.1.8, for a random $\mathbf{G}$, $\mathbf{mG}$ is a uniformly random vector from $\mathbb{F}_q^n$. Thus, we have

$$\begin{aligned} Pr[wt(\mathbf{mG}) < d] &= \frac{Vol_q(d-1,n)}{q^n} \\ &\leq \frac{q^{nH_q(\delta)}}{q^n}, \end{aligned} \tag{4.4}$$

where (4.4) follows from (4.3). Thus, by the union bound (Lemma 3.1.3)

$$\begin{aligned} Pr[\text{There exists a non-zero } \mathbf{m}, wt(\mathbf{mG}) < d] &\leq q^k q^{-n(1-H_q(\delta))} \\ &= q^{-\varepsilon \cdot n}, \end{aligned}$$

where the equality follows by choosing $k = (1 - H_q(\delta) - \varepsilon)n$. Since $q^{-\varepsilon n} \ll 1$, by the probabilistic method, there exists a linear code C with relative distance δ.

All that's left is to argue that the code C has dimension at least $k = (1 - H_q(\delta) - \varepsilon)n$. To show this we need to show that the chosen generator matrix $\mathbf{G}$ has full rank. Note that there is a non-zero probability that a uniformly matrix $\mathbf{G}$ does not have full rank. There are two ways to deal with this. First, we can show that with high probability a random $\mathbf{G}$ does have full rank, so that $|C| = q^k$. However, the proof above has already shown that, with high probability, the distance is greater than zero, which implies that distinct messages will be mapped to distinct codewords and thus $|C| = q^k$. In other words, C does indeed have dimension k, as desired

Discussion. We now digress a bit to discuss some consequences of the proofs of the GV bound.

We first note the probabilistic method proof shows something stronger than Theorem 4.2.1: *most* linear codes (with appropriate parameters) meet the Gilbert-Varshamov bound.

Note that we can also pick a random linear code by picking a random $(n-k) \times n$ parity check matrix. This also leads to a proof of the GV bound: see Exercise 4.1.

Finally, we note that Theorem 4.2.1 requires $\delta < 1 - \frac{1}{q}$. An inspection of Gilbert and Varshamov's proofs shows that the only reason the proof required that $\delta \leq 1 - \frac{1}{q}$ was because it is needed for the volume bound (recall the bound in Proposition 3.3.1): $Vol_q(\delta n, n) \leq q^{H_q(\delta)n}$– to hold. It is natural to wonder if the above is just an artifact of the proof or, for example,

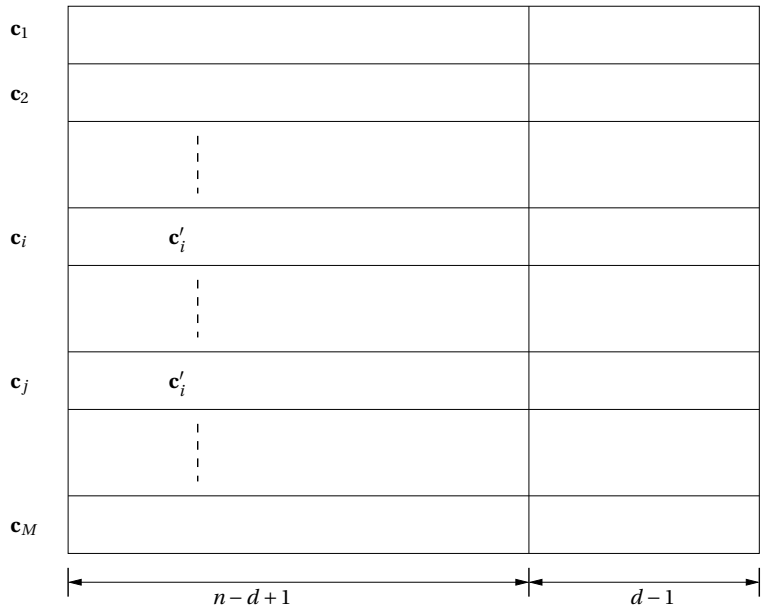

Figure 4.3: Construction of a new code in the proof of the Singleton bound.

Question 4.2.2. *Does there exists a code with $R > 0$ and $\delta > 1 - \frac{1}{q}$?*

We will return to this question in Section 4.4.

4.3 Singleton Bound

We will now change gears again and prove an upper bound on R (for fixed δ). We start by proving the Singleton bound.

Theorem 4.3.1 (Singleton Bound)**.** *For every $(n,k,d)_q$ code,*

$$k \leq n - d + 1.$$

Proof. Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M$ be the codewords of an $(n,k,d)_q$ code C. Note that we need to show $M \leq q^{n-d+1}$. To this end, we define $\mathbf{c}'_i$ to be the prefix of the codeword $\mathbf{c}_i$ of length $n-d+1$ for every $i \in [M]$. See Figure 4.3 for a pictorial description.

We now claim that for every $i \neq j$, $\mathbf{c}'_i \neq \mathbf{c}'_j$. For the sake of contradiction, assume that there exits an $i \neq j$ such that $\mathbf{c}'_i = \mathbf{c}'_j$. Note that this implies that $\mathbf{c}_i$ and $\mathbf{c}_j$ agree in all the first $n-d+1$ positions, which in turn implies that $\Delta(\mathbf{c}_i, \mathbf{c}_j) \leq d-1$. This contradicts the fact that C has distance d. Thus, M is the number of prefixes of codewords in C of length $n-d+1$, which implies that $M \leq q^{n-d+1}$ as desired. □

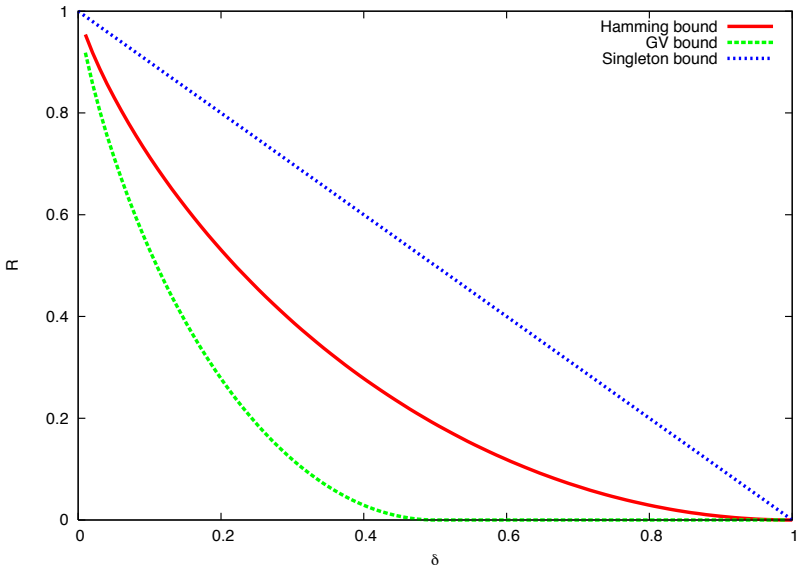

Figure 4.4: The Hamming, GV and Singleton bound for binary codes.

Note that the asymptotic version of the Singleton bound states that $k/n \leq 1 - d/n + 1/n$. In other words,

$$R \leq 1 - \delta + o(1).$$

Figure 4.4 presents a pictorial description of the asymptotic version of the Singleton bound. It is worth noting that the bound is *independent* of the alphabet size. As is evident from Figure 4.4, the Singleton bound is worse than the Hamming bound for binary codes. However, this bound is better for larger alphabet sizes. In fact, we will look at a family of codes called Reed-Solomon codes in Chapter 5 that meets the Singleton bound. However, the alphabet size of the Reed-Solomon codes increases with the block length n. Thus, a natural follow-up question is the following:

Question 4.3.1. *Given a fixed $q \geq 2$, does there exist a q-ary code that meets the Singleton bound?*

We'll see an answer to this question in the next section.

4.4 Plotkin Bound

In this section, we will study the Plotkin bound, which will answer Questions 4.2.2 and 4.3.1. We start by stating the bound.

Theorem 4.4.1 (Plotkin bound)**.** *The following holds for any code $C \subseteq [q]^n$ with distance d:*

1. *If $d = \left(1 - \frac{1}{q}\right) n$, $|C| \leq 2qn$.*
2. *If $d > \left(1 - \frac{1}{q}\right) n$, $|C| \leq \frac{qd}{qd-(q-1)n}$.*

Note that the Plotkin bound (Theorem 4.4.1) implies that a code with relative distance $\delta \geq 1 - \frac{1}{q}$, must necessarily have $R = 0$, which answers Question 4.2.2 in the negative.

Before we prove Theorem 4.4.1, we make couple of remarks. We first note that the upper bound in the first part of Theorem 4.4.1 can be improved to $2n$ for $q = 2$. (See Exercise 4.12.) Second, it can be shown that this bound is tight– see Exercise 4.13. Third, the statement of Theorem 4.4.1 gives a trade-off only for relative distance greater than $1 - 1/q$. However, as the following corollary shows, the result can be extended to work for $0 \leq \delta \leq 1 - 1/q$. (See Figure 4.5 for an illustration for binary codes.)

Corollary 4.4.2. *For any q-ary code with distance δ, $R \leq 1 - \left(\frac{q}{q-1}\right)\delta + o(1)$.*

Proof. The proof proceeds by shortening the codewords. We group the codewords so that they agree on the first $n - n'$ symbols, where $n' = \left\lfloor \frac{qd}{q-1} \right\rfloor - 1$. (We will see later why this choice of n' makes sense.) In particular, for any $\mathbf{x} \in [q]^{n-n'}$, define

$$C_{\mathbf{x}} = \{(c_{n-n'+1}, \ldots c_n) \mid (c_1 \ldots c_N) \in C, (c_1 \ldots c_{n-n'}) = \mathbf{x}\}.$$

Define $d = \delta n$. For all $\mathbf{x}$, $C_{\mathbf{x}}$ has distance d as C has distance d.[1] Additionally, it has block length $n' < (\frac{q}{q-1})d$ and thus, $d > \left(1 - \frac{1}{q}\right) n'$. By Theorem 4.4.1, this implies that

$$|C_{\mathbf{x}}| \leq \frac{qd}{qd - (q-1)n'} \leq qd, \tag{4.5}$$

where the second inequality follows from the fact that $qd - (q-1)n'$ is an integer.

Note that by the definition of $C_{\mathbf{x}}$:

$$|C| = \sum_{\mathbf{x} \in [q]^{n-n'}} |C_{\mathbf{x}}|,$$

which by (4.5) implies that

$$|C| \leq \sum_{\mathbf{x} \in [q]^{n-n'}} qd = q^{n-n'} \cdot qd \leq q^{n - \frac{q}{q-1}d + o(n)} = q^{n\left(1 - \delta \cdot \frac{q}{q-1} + o(1)\right)}.$$

In other words, $R \leq 1 - \left(\frac{q}{q-1}\right)\delta + o(1)$ as desired. □

[1] If for some $\mathbf{x}$, $\mathbf{c}_1 \neq \mathbf{c}_2 \in C_{\mathbf{x}}$, $\Delta(\mathbf{c}_1, \mathbf{c}_2) < d$, then $\Delta((\mathbf{x}, \mathbf{c}_1), (\mathbf{x}, \mathbf{c}_2)) < d$, which implies that the distance of C is less than d (as by definition of $C_{\mathbf{x}}$, both $(\mathbf{x}, \mathbf{c}_1), (\mathbf{x}, \mathbf{c}_2) \in C$).

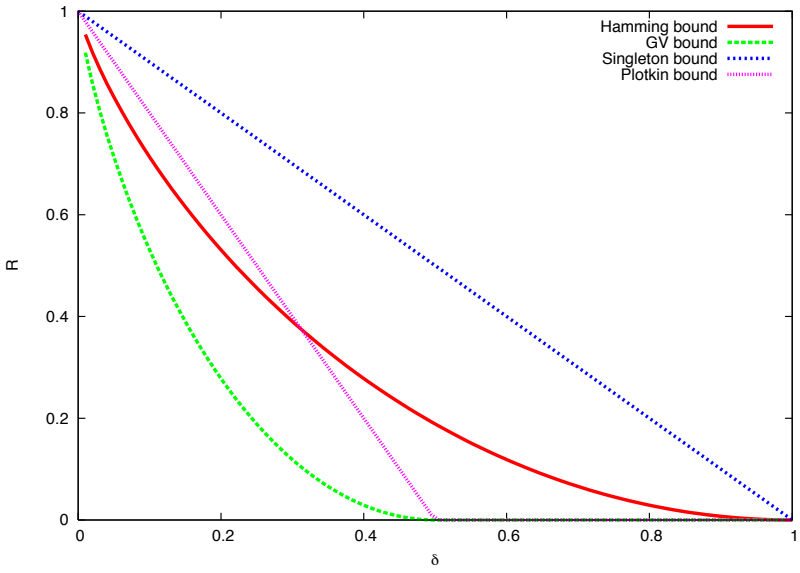

Figure 4.5: The current bounds on the rate R vs. relative distance δ for binary codes. The GV bound is a lower bound on rate while the other three bounds are upper bounds on R.

Note that Corollary 4.4.2 implies that for any q-ary code of rate R and relative distance δ (where q is a *constant* independent of the block length of the code), $R < 1 - \delta$. In other words, this answers Question 4.3.1 in the negative.

Let us pause for a bit at this point and recollect the bounds on R versus δ that we have proved till now. Figure 4.5 depicts all the bounds we have seen till now (for $q = 2$). The GV bound is the best known lower bound at the time of writing of this book. Better upper bounds are known and we will see one such trade-off (called the Elias-Bassalygo bound) in Section 8.1.

Now, we turn to the proof of Theorem 4.4.1, for which we will need two more lemmas.

The first lemma deals with vectors over real spaces. We quickly recap the necessary definitions. Consider a vector $\mathbf{v}$ in $\mathbb{R}^n$, that is, a tuple of n real numbers. This vector has (Euclidean) norm $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$, and is a unit vector if and only if its norm is 1. The inner product of two vectors, $\mathbf{u}$ and $\mathbf{v}$, is $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_i u_i \cdot v_i$. The following lemma gives a bound on the number of vectors that can exist such that every pair is at an obtuse angle with each other.

Lemma 4.4.3 (Geometric Lemma)**.** *Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m \in \mathbb{R}^N$ be non-zero vectors.*

1. *If $\langle \mathbf{v}_i, \mathbf{v}_j \rangle \leq 0$ for all $i \neq j$, then $m \leq 2N$.*
2. *Let $\mathbf{v}_i$ be unit vectors for $1 \leq i \leq m$. Further, if $\langle \mathbf{v}_i, \mathbf{v}_j \rangle \leq -\varepsilon < 0$ for all $i \neq j$, then $m \leq 1 + \frac{1}{\varepsilon}$.*[2]

(Item 1 is tight: see Exercise 4.14.) The proof of the Plotkin bound will need the existence of a map from codewords to real vectors with certain properties, which the next lemma guarantees.

[2]Note that since $\mathbf{v}_i$ and $\mathbf{v}_j$ are both unit vectors, $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ is the cosine of the angle between them.

Lemma 4.4.4 (Mapping Lemma)**.** *Let $C \subseteq [q]^n$. Then there exists a function $f : C \longrightarrow \mathbb{R}^{nq}$ such that*

1. *For every $\mathbf{c} \in C$, $\|f(\mathbf{c})\| = 1$.*
2. *For every $\mathbf{c}_1 \neq \mathbf{c}_2$ such that $\mathbf{c}_1, \mathbf{c}_2 \in C$, $\langle f(\mathbf{c}_1), f(\mathbf{c}_2)\rangle = 1 - \left(\frac{q}{q-1}\right)\left(\frac{\Delta(\mathbf{c}_1,\mathbf{c}_2)}{n}\right)$.*

We defer the proofs of the lemmas above to the end of the section. We are now in a position to prove Theorem 4.4.1.

Proof of Theorem 4.4.1 Let $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$. For all $i \neq j$,

$$\left\langle f(\mathbf{c}_i), f(\mathbf{c}_j)\right\rangle \leq 1 - \left(\frac{q}{q-1}\right)\frac{\Delta(\mathbf{c}_i,\mathbf{c}_j)}{n} \leq 1 - \left(\frac{q}{q-1}\right)\frac{d}{n}.$$

The first inequality holds by Lemma 4.4.4, and the second holds as C has distance d.

For part 1, if $d = \left(1 - \frac{1}{q}\right)n = \frac{(q-1)n}{q}$, then for all $i \neq j$,

$$\left\langle f(\mathbf{c}_i), f(\mathbf{c}_j)\right\rangle \leq 0$$

and so by the first part of Lemma 4.4.3, $m \leq 2nq$, as desired.

For part 2, $d > \left(\frac{q-1}{q}\right)n$ and so for all $i \neq j$,

$$\langle f(\mathbf{c}_i), f(\mathbf{c}_j)\rangle \leq 1 - \left(\frac{q}{q-1}\right)\frac{d}{n} = -\left(\frac{qd-(q-1)n}{(q-1)n}\right)$$

and, since $\varepsilon \stackrel{\text{def}}{=} \left(\frac{qd-(q-1)n}{(q-1)n}\right) > 0$, we can apply the second part of Lemma 4.4.3. Thus, $m \leq 1 + \frac{(q-1)n}{qd-(q-1)n} = \frac{qd}{qd-(q-1)n}$, as desired □

4.4.1 Proof of Geometric and Mapping Lemmas

Next, we prove Lemma 4.4.3.

Proof of Lemma 4.4.3. We begin with a proof of the first result. The proof is by induction on n. Note that in the base case of $N = 0$, we have $m = 0$, which satisfies the claimed inequality $m \leq 2N$.

In the general case, we have $m \geq 1$ non-zero vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^N$ such that for every $i \neq j$,

$$\langle \mathbf{v}_i, \mathbf{v}_j\rangle \leq 0. \tag{4.6}$$

Since rotating all the vectors by the same amount does not change the sign of the inner product (nor does scaling any of the vectors), w.l.o.g. we can assume that $\mathbf{v}_m = \langle 1, 0, \ldots, 0\rangle$. For $1 \leq i \leq m-1$, denote the vectors as $\mathbf{v}_i = \langle \alpha_i, \mathbf{y}_i\rangle$, for some $\alpha_i \in \mathbb{R}$ and $\mathbf{y}_i \in \mathbb{R}^{N-1}$. Now, for any $i \neq 1$, $\langle \mathbf{v}_1, \mathbf{v}_i\rangle = 1 \cdot \alpha_i + \sum_{i=2}^{m} 0 = \alpha_i$. However, note that (4.6) implies that $\langle \mathbf{v}_1, \mathbf{v}_i\rangle \leq 0$, which in turn implies that

$$\alpha_i \leq 0. \tag{4.7}$$

Next, we claim that at most one of $\mathbf{y}_1, \ldots, \mathbf{y}_{m-1}$ can be the all zeroes vector, $\mathbf{0}$. If not, assume w.l.o.g., that $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{0}$. This in turn implies that

$$\begin{aligned}\langle \mathbf{v}_1, \mathbf{v}_2 \rangle &= \alpha_1 \cdot \alpha_2 + \langle \mathbf{y}_1, \mathbf{y}_2 \rangle \\ &= \alpha_1 \cdot \alpha_2 + 0 \\ &= \alpha_1 \cdot \alpha_2 \\ &> 0,\end{aligned}$$

where the last inequality follows from the subsequent argument. As $\mathbf{v}_1 = \langle \alpha_1, \mathbf{0} \rangle$ and $\mathbf{v}_2 = \langle \alpha_2, \mathbf{0} \rangle$ are non-zero, this implies that $\alpha_1, \alpha_2 \neq 0$. (4.7) then implies that $\alpha_1, \alpha_2 < 0$. However, $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle > 0$ contradicts (4.6).

Thus, w.l.o.g., assume that $\mathbf{v}_1, \ldots, \mathbf{v}_{m-2}$ are all non-zero vectors. Further, note that for every $i \neq j \in [m-2]$, $\langle \mathbf{y}_i, \mathbf{y}_j \rangle = \langle \mathbf{v}_i, \mathbf{v}_j \rangle - \alpha_i \cdot \alpha_j \leq \langle \mathbf{v}_i, \mathbf{v}_j \rangle \leq 0$. Thus, we have reduced problem on m vectors with dimension N to an equivalent problem on $m-2$ vectors with dimension dimension $N-1$. If we continue this process, we can conclude that every loss in dimension of the vector results in twice in loss in the numbers of the vectors in the set. Induction then implies that $m \leq 2N$, as desired.

We now move on to the proof of the second part. Define $\mathbf{z} = \mathbf{v}_1 + \ldots + \mathbf{v}_m$. Now consider the following sequence of relationships:

$$\|\mathbf{z}\|^2 = \sum_{i=1}^{m} \|\mathbf{v}_i\|^2 + 2 \sum_{i<j} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \leq m + 2 \cdot \binom{m}{2} \cdot (-\varepsilon) = m(1 - \varepsilon m + \varepsilon).$$

The inequality follows from the facts that each $\mathbf{v}_i$ is a unit vector and the assumption that for every $i \neq j$, $\langle \mathbf{v}_i . \mathbf{v}_j \rangle \leq -\varepsilon$. As $\|\mathbf{z}\|^2 \geq 0$,

$$m(1 - \varepsilon m + \varepsilon) \geq 0.$$

Thus, we have $m \leq 1 + \frac{1}{\varepsilon}$, as desired. □

Finally, we prove Lemma 4.4.4.

Proof of Lemma 4.4.4. We begin by defining a map $\phi : [q] \to \mathbb{R}^q$ with certain properties. Then we apply ϕ to all the coordinates of a codeword to define the map $f : \mathbb{R}^q \to \mathbb{R}^{nq}$ that satisfies the claimed properties. We now fill in the details.

Define $\phi : [q] \to \mathbb{R}^q$ as follows. For every $i \in [q]$, we define

$$\phi(i) = \left\langle \frac{1}{q}, \frac{1}{q}, \ldots, \underbrace{\frac{-(q-1)}{q}}_{i^{\text{th}}\text{position}}, \ldots \frac{1}{q} \right\rangle.$$

That is, all but the i'th position in $\phi(i) \in \mathbb{R}^q$ has a value of $1/q$ and the ith position has value $-(q-1)/q$.

Next, we record two properties of ϕ that follow immediately from its definition. For every $i \in [q]$,

$$\phi(i)^2 = \frac{(q-1)}{q^2} + \frac{(q-1)^2}{q^2} = \frac{(q-1)}{q}. \tag{4.8}$$

Also for every $i \neq j \in [q]$,

$$\langle \phi(i), \phi(j) \rangle = \frac{(q-2)}{q^2} - \frac{2(q-1)}{q^2} = -\frac{1}{q}. \tag{4.9}$$

We are now ready to define our final map $f : C \to \mathbb{R}^{nq}$. For every $\mathbf{c} = (c_1, \ldots, c_n) \in C$, define

$$f(\mathbf{c}) = \sqrt{\frac{q}{n(q-1)}} \cdot \big(\phi(c_1), \phi(c_2), \ldots, \phi(c_n)\big).$$

(The multiplicative factor $\sqrt{\frac{q}{n(q-1)}}$ is to ensure that $f(\mathbf{c})$ for any $\mathbf{c} \in C$ is a unit vector.)

To complete the proof, we will show that f satisfies the claimed properties. We begin with condition 1. Note that

$$\|f(\mathbf{c})\|^2 = \frac{q}{(q-1)n} \cdot \sum_{i=1}^{n} \big|\phi(i)\big|^2 = 1,$$

where the first equality follows from the definition of f and the second equality follows from (4.8).

We now turn to the second condition. For notational convenience define $\mathbf{c}_1 = (x_1, \ldots, x_n)$ and $\mathbf{c}_2 = (y_1, \ldots, y_n)$. Consider the following sequence of relations:

$$\begin{aligned}
\big\langle f(\mathbf{c}_1), f(\mathbf{c}_2) \big\rangle &= \sum_{\ell=1}^{n} \big\langle f(x_\ell), f(y_\ell) \big\rangle \\
&= \left[\sum_{\ell : x_\ell \neq y_\ell} \big\langle \phi(x_\ell), \phi(y_\ell) \big\rangle + \sum_{\ell : x_\ell = y_\ell} \big\langle \phi(x_\ell), \phi(y_\ell) \big\rangle \right] \cdot \left(\frac{q}{n(q-1)} \right) \\
&= \left[\sum_{\ell : x_\ell \neq y_\ell} \left(\frac{-1}{q} \right) + \sum_{\ell : x_\ell = y_\ell} \left(\frac{q-1}{q} \right) \right] \cdot \left(\frac{q}{n(q-1)} \right) \qquad (4.10) \\
&= \left[\Delta(\mathbf{c}_1, \mathbf{c}_2) \left(\frac{-1}{q} \right) + (n - \Delta(\mathbf{c}_1, \mathbf{c}_2)) \left(\frac{q-1}{q} \right) \right] \cdot \left(\frac{q}{n(q-1)} \right) \qquad (4.11) \\
&= 1 - \Delta(\mathbf{c}_1, \mathbf{c}_2) \left(\frac{q}{n(q-1)} \right) \left[\frac{1}{q} + \frac{q-1}{q} \right] \\
&= 1 - \left(\frac{q}{q-1} \right) \left(\frac{\Delta(\mathbf{c}_1, \mathbf{c}_2)}{n} \right),
\end{aligned}$$

as desired. In the above, (4.10) is obtained using (4.9) and (4.8) while (4.11) follows from the definition of the Hamming distance. □

4.5 Exercises

Exercise 4.1. Pick a $(n-k) \times n$ matrix H over $\mathbb{F}_q$ at random. Show that with high probability the code whose parity check matrix is H achieves the GV bound.

Exercise 4.2. Recall the definition of an ε-biased space from Exercise 2.14. Show that there exists an ε-biased space of size $O(k/\varepsilon^2)$.

Hint: Recall part 1 of Exercise 2.14.

Exercise 4.3. Argue that a random linear code as well as its dual both lie on the corresponding GV bound.

Exercise 4.4. In Section 4.2.2, we saw that random *linear* code meets the GV bound. It is natural to ask the question for general random codes. (By a random $(n,k)_q$ code, we mean the following: for each of the q^k messages, pick a random vector from $[q]^n$. Further, the choices for each codeword is independent.) We will do so in this problem.

1. Prove that a random q-ary code with rate $R > 0$ with high probability has relative distance $\delta \geq H_q^{-1}(1-2R-\varepsilon)$. Note that this is worse than the bound for random linear codes in Theorem 4.2.1.

2. Prove that with high probability the relative distance of a random q-ary code of rate R is at most $H_q^{-1}(1-2R)+\varepsilon$. In other words, general random codes are worse than random linear codes in terms of their distance.
   Hint: Use Chebyshev's inequality.

Exercise 4.5. We saw that Algorithm 5 can compute an $(n,k)_q$ code on the GV bound in time $q^{O(n)}$. Now the construction for linear codes is a randomized construction and it is natural to ask how quickly can we compute an $[n,k]_q$ code that meets the GV bound. In this problem, we will see that this can also be done in $q^{O(n)}$ deterministic time, though the deterministic algorithm is not that straight-forward anymore.

1. Argue that Theorem 4.2.1 gives a $q^{O(kn)}$ time algorithm that constructs an $[n,k]_q$ code on the GV bound. (Thus, the goal of this problem is to "shave" off a factor of k from the exponent.)

2. A $k \times n$ *Toeplitz Matrix* $A = \{A_{i,j}\}_{i=1,\,j=1}^{k,\,n}$ satisfies the property that $A_{i,j} = A_{i-1,j-1}$. In other words, any diagonal has the same value. For example, the following is a 4×6 Toeplitz matrix:
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 1 & 2 & 3 & 4 & 5 \\ 8 & 7 & 1 & 2 & 3 & 4 \\ 9 & 8 & 7 & 1 & 2 & 3 \end{pmatrix}$$
   A *random* $k \times n$ Toeplitz matrix $T \in \mathbb{F}_q^{k \times n}$ is chosen by picking the entries in the first row and column uniformly (and independently) at random.

Prove the following claim: For any non-zero $\mathbf{m} \in \mathbb{F}_q^k$, the vector $\mathbf{m} \cdot T$ is uniformly distributed over $\mathbb{F}_q^n$, that is for every $\mathbf{y} \in \mathbb{F}_q^n$, $\Pr\left[\mathbf{m} \cdot T = \mathbf{y}\right] = q^{-n}$.

3. Briefly argue why the claim in part 2 implies that a random code defined by picking its generator matrix as a random Toeplitz matrix with high probability lies on the GV bound.

4. Conclude that an $[n,k]_q$ code on the GV bound can be constructed in time $q^{O(k+n)}$.

Exercise 4.6. Show that one can construct the parity check matrix of an $[n,k]_q$ code that lies on the GV bound in time $q^{O(n)}$.

Exercise 4.7. So far in Exercises 4.5 and 4.6, we have seen two constructions of $[n,k]_q$ code on the GV bound that can be constructed in $q^{O(n)}$ time. For constant rate codes, at the time of writing of this book, this is fastest known construction of any code that meets the GV bound. For $k = o(n)$, there is a better construction known, which we explore in this exercise.

We begin with some notation. For the rest of the exercise we will target a distance of $d = \delta n$. Given a message $\mathbf{m} \in \mathbb{F}_q^k$ and an $[n,k]_q$ code C, define the indicator variable:

$$W_{\mathbf{m}}(C) = \begin{cases} 1 & \text{if } wt(C(\mathbf{m})) < d \\ 0 & \text{otherwise.} \end{cases}$$

Further, define

$$D(C) = \sum_{\mathbf{m} \in \mathbb{F}_q^k \setminus \{\mathbf{0}\}} W_{\mathbf{m}}(C).$$

We will also use $D(G)$ and $W_{\mathbf{m}}(G)$ to denote the variables above for the code C generated by G.

Given an $k \times n$ matrix M, we will use M^i to denote the ith column of M and $M^{\leq i}$ to denote the column submatrix of M that contains the first i columns. Finally below we will use $\mathscr{G}$ to denote a uniformly random $k \times n$ generator matrix and G to denote a specific instantiation of the generator matrix. We will arrive at the final construction in a sequence of steps. In what follows define $k < (1 - H_q(\delta))n$ for large enough n.

1. Argue that C has a distance d if and only if $D(C) < 1$.

2. Argue that $\mathbb{E}[D(\mathscr{G})] < 1$.

3. Argue that for any $1 \leq i < n$ and fixed $k \times n$ matrix G,

$$\min_{\mathbf{v} \in \mathbb{F}_q^k} \mathbb{E}\left[D(\mathscr{G}) | \mathscr{G}^{\leq i} = G^{\leq i}, \mathscr{G}^{i+1} = \mathbf{v}\right] \leq \mathbb{E}\left[D(\mathscr{G}) | \mathscr{G}^{\leq i} = G^{\leq i}\right].$$

4. We are now ready to define the algorithm to compute the final generator matrix G: see Algorithm 6. Prove that Algorithm 6 outputs a matrix G such that the linear code generated by G is an $[n,k,\delta n]_q$ code. Conclude that this code lies on the GV bound.

Algorithm 6 $q^{O(k)}$ time algorithm to compute a code on the GV bound

INPUT: Integer parameters $1 \le k \ne n$ such that $k < (1 - H_q(\delta)n)$
OUTPUT: An $k \times n$ generator matrix G for a code with distance δn

1: Initialize G to be the all 0s matrix ▷ This initialization is arbitrary
2: FOR every $1 \le i \le n$ DO
3: $\quad G^i \leftarrow \arg\min_{\mathbf{v} \in \mathbb{F}_q^k} \mathbb{E}\left[D(\mathscr{G}) | \mathscr{G}^{\le i} = G^{\le i}, \mathscr{G}^{i+1} = \mathbf{v}\right]$
4: RETURN G

5. Finally, we will analyze the run time of Algorithm 6. Argue that Step 2 can be implemented in $\text{poly}\left(n, q^k\right)$ time. Conclude Algorithm 6 can be implemented in time $\text{poly}\left(n, q^k\right)$.

   Hint: It might be useful to maintain a data structure that keeps track of one number for every non-zero $\mathbf{m} \in \mathbb{F}_q^k$ throughout the run of Algorithm 6.

Exercise 4.8. In this problem we will derive the GV bound using a graph-theoretic proof, which is actually equivalent to the greedy proof we saw in Section 4.2.1. Let $1 \le d \le n$ and $q \ge 1$ be integers. Now consider the graph $G_{n,d,q} = (V, E)$, where the vertex set is the set of all vectors in $[q]^n$. Given two vertices $\mathbf{u} \ne \mathbf{v} \in [q]^n$, we have the edge $(u, v) \in E$ if and only if $\Delta(\mathbf{u}, \mathbf{v}) < d$. An *independent set* of a graph $G = (V, E)$ is a subset $I \subseteq V$ such that for every $u \ne v \in I$, we have that (u, v) is *not* an edge. We now consider the following sub-problems:

1. Argue that any independent set C of $G_{n,d,q}$ is a q-ary code of distance d.
2. The *degree* of a vertex in a graph G is the number of edges incident on that vertex. Let Δ be the maximum degree of any vertex in $G = (V, E)$.Then argue that G has an independent set of size at least $\frac{|V|}{\Delta+1}$.
3. Using parts 1 and 2 argue the GV bound.

Exercise 4.9. In this problem we will improve slightly on the GV bound using a more sophisticated graph-theoretic proof. Let $G_{n,d,q}$ and N and Δ be as in the previous exercise (Exercise 4.8). So far we used the fact that $G_{n,d,q}$ has many vertices and small degree to prove it has a large independent set, and thus to prove there is a large code of minimum distance d. In this exercise we will see how a better result can be obtained by counting the number of "triangles" in the graph. A triangle in a graph $G = (V, E)$ is a set $\{u, v, w\} \subset V$ of three vertices such that all three vertices are adjacent, i.e., $(u, v), (v, w), (w, u) \in E$. For simplicity we will focus on the case where $q = 2$ and $d = n/5$, and consider the limit as $n \to \infty$.

1. Prove that a graph on N vertices of maximum degree Δ has at most $O(N\Delta^2)$ triangles.
2. Prove that the number of triangle in graph $G_{n,d,2}$ is at most

$$2^n \cdot \sum_{0 \le e \le 3d/2} \binom{n}{e} \cdot 3^e.$$

Hint: Fix u and let e count the number of coordinates where at least one of v or w disagree with u. Prove that e is at most $3d/2$.

3. Simplify the expression in the case where $d = n/5$ to show that the number of triangles in $G_{n,n/5,2}$ is $O(N \cdot \Delta^{2-\eta})$ for some $\eta > 0$.

4. A famous result in the "probabilistic method" shows (and you don't have to prove this), that if a graph on N vertices of maximum degree Δ has at most $O(N \cdot \Delta^{2-\eta})$ triangles, then it has an independent set of size $\Omega(\frac{N}{\Delta}\log\Delta)$. Use this result to conclude that there is a binary code of block length n and distance $n/5$ of size $\Omega(n2^n/\binom{n}{n/5})$. (Note that this improves over the GV-bound by an $\Omega(n)$ factor.)

Exercise 4.10. Use part 2 from Exercise 1.7 to prove the Singleton bound.

Exercise 4.11. Let C be an $(n,k,d)_q$ code. Then prove that fixing any $n-d+1$ positions uniquely determines the corresponding codeword.

Exercise 4.12. Let C be a binary code of block length n and distance $n/2$. Then $|C| \leq 2n$. (Note that this is a factor 2 better than part 1 in Theorem 4.4.1.)

Exercise 4.13. Prove that the bound in Exercise 4.12 is tight– i.e. there exists binary codes C with block length n and distance $n/2$ such that $|C| = 2n$.

Exercise 4.14. Prove that part 1 of Lemma 4.4.3 is tight.

Exercise 4.15. In this exercise we will prove the Plotkin bound (at least part 2 of Theorem 4.4.1) via a purely combinatorial proof.

Given an $(n,k,d)_q$ code C with $d > \left(1-\frac{1}{q}\right)n$ define

$$S = \sum_{\mathbf{c}_1 \neq \mathbf{c}_2 \in C} \Delta(\mathbf{c}_1, \mathbf{c}_2).$$

For the rest of the problem think of C has an $|C| \times n$ matrix where each row corresponds to a codeword in C. Now consider the following:

1. Looking at the contribution of each column in the matrix above, argue that

$$S \leq \left(1-\frac{1}{q}\right) \cdot n|C|^2.$$

2. Look at the contribution of the rows in the matrix above, argue that

$$S \geq |C|\,(|C|-1) \cdot d.$$

3. Conclude part 2 of Theorem 4.4.1.

Exercise 4.16. In this exercise, we will prove the so called *Griesmer Bound*. For any $[n,k,d]_q$, prove that

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

Hint: Recall Exercise 2.17.

Exercise 4.17. Use Exercise 4.16 to prove part 2 of Theorem 4.4.1 for linear codes.

Exercise 4.18. Use Exercise 4.16 to prove Theorem 4.3.1 for linear code.

4.6 Bibliographic Notes

Theorem 4.2.1 was proved for general codes by Edgar Gilbert ([23]) and for linear codes by Rom Varshamov ([75]). Hence, the bound is called the Gilbert-Varshamov bound. The Singleton bound (Theorem 4.3.1) is due to Richard C. Singleton [66]. For larger (but still constant) values of q, better lower bounds than the GV bound are known. In particular, for any prime power $q \geq 49$, there exist linear codes, called *algebraic geometric* (or AG) codes that outperform the corresponding GV bound[3]. AG codes out of the scope of this book. One starting point could be the following [41].

The proof method illustrated in Exercise 4.15has a name– *double counting*: in this specific case this follows since we count S in two different ways.

[3]The lower bound of 49 comes about as AG codes are only defined for q being a square (i.e. $q = (q')^2$) and it turns out that $q' = 7$ is the smallest value where AG bound beats the GV bound.