Lecture 2: More limits on codes

Anup Rao

September 30, 2019

Last time, we discussed the following estimate for the volume of balls in the Hamming metric. We defined

$$h_q(\epsilon) = (1-\epsilon)\log_q\left(\frac{1}{1-\epsilon}\right) + \epsilon\log_q\left(\frac{q-1}{\epsilon}\right).$$

The quantity $h_q(\epsilon)$ is proportional to the maximum entropy of a random variable taking q values, subject to the constraint that the random variable must be equal to the first value with probability at least $1-\epsilon$.

When $r \leq n(1-1/q)$, we shall prove

$$q^{n\cdot h_q(r/n) - O(\log_q n)} \leq \mathsf{vol}(r) \leq q^{n\cdot h_q(r/n)}. \tag{1}$$

We used the above estimate to prove the Hamming bound:

$$R + h_q(\delta/2) \leq 1 + O(\log_q(n)/n).$$

We begin today be using bounds on the volume to brute force search for a code.

The Gilbert-Varshamov bound (Greedy Code)

Consider the code of distance d obtained by greedily picking strings. Pick an arbitrary string $x \in \Sigma^n$ and let that be the first codeword. Delete $B(x,d)$ from the ambient set, and find another string in what remains. If we continue in this way, we will get a code with $|\Sigma|^n/\mathsf{vol}(d)$ codewords, and the distance is d by construction.

The lower bound we prove on the size of the greedy code is called the Gilbert-Varshamov bound.

So, using (1), the number of codewords will be at least

$$q^n/q^{h_q(d/n)} = q^{n(1-h_q(d/n))}.$$

As q gets larger and larger $\delta + R$ approaches 1, and this sum cannot be larger by the Singleton bound. In other words, we have found a code with rate $R = 1 - h_q(\delta)$, and relative distance δ. To summarize, given a particular constant δ, our bounds show that if we take the best family of codes with this relative distance must have rate R (here we take the infimum of all values of R as $n \to \infty$) satisfying:

$$1 - h_q(\delta) \leq R \leq 1 - h_q(\delta/2).$$

Random Codes and Random Errors

We have seen two bounds on the rate of a code so far. Instead of picking the codewords greedily, we could pick them completely at random. This idea does really well, especially when the errors themselves are random. Let us analyze what happens.

Suppose we use a random code to transmit messages of length Rn over the alphabet Σ. Suppose each transmitted symbol is corrupted to a random value with probability p. What is the probability that the transmitted message is decoded incorrectly?

An obvious way to do the decoding is to find the codeword that is closest to the received word in Hamming distance, breaking ties arbitrarily. If this process decodes the wrong message, then one of two things must have happened:

- Either the actual number of errors exceeded $(p+\epsilon)n$,
- Or more than one codeword lies in $B(y,(p+\epsilon)n)$, where y is the input to the decoder.

By the Chernoff bound, the probability of the first event is at most

$$\exp(-(\epsilon/p)^2 pn/3) \leq \exp(-\epsilon^2 n/(3p)).$$

The probability of the second event is at most

$$|C| \cdot \mathsf{vol}((p+\epsilon)n)/q^n \leq q^{n(R+h_q(p+\epsilon)-1)}.$$

So, if we set $p = \delta - \epsilon$, we obtain a code with rate approaching $1-h_q(\delta)$, and yet it tolerates $\delta-\epsilon$ fraction of errors with high probability. This is much better than the guarantee of the greedy code, which could only guarantee a rate of $1-h_q(\delta)$ if the fraction of errors was less than $\delta/2$. The point is that because the errors are random and the code is random, even though the distance is d, you can tolerate nearly d errors, because the errors are unlikely to be pathological enough to lead to a mistake when decoding.

Next, we discuss more sophisticated bounds on codes using Euclidean geometry.

Geometry of Codes

Codes correspond to some very interesting geometric objects — you can map codewords to vectors that are far apart. Suppose we have a binary code $C \subseteq \{0,1\}^n$ of distance d. For each codeword

$x \in C$, let $v(x) \in \mathbb{R}^n$ be given by $v(x)_i = (-1)^{x_i}$. Then the distance of the code implies that if $x, y \in C$ are distinct codewods,

$$\langle v(x), v(y) \rangle = \sum_{i=1}^{n} v(x)_i v(y)_i = (n - \Delta(x,y)) - \Delta(x,y) \leq n - 2d.$$

That is an interesting property for a set of vectors to have: if $\delta \geq 1/2$, all of the angles between the vectors are more than 90 degrees. There cannot be too many such vectors:

Note that since $\|v(x)\| = \|v(y)\| = \sqrt{n}$, asserting that $\langle v(x), v(y) \rangle \leq n - 2d$ is equivalent to saying that $\|v(x) - v(y)\|^2 = \langle v(x) - v(y), v(x) - v(y) \rangle \geq 2n - 2(n - 2d) = 4d$. If we normalized these vectors by their length, we get 2^{Rn} unit vectors whose pairwise distances are all at least $2\sqrt{\delta}$. Intuitively, you cannot have too many such vectors in an n-dimensional space.

Lemma 1 (Plotkin Bound). *Suppose $v_1, v_2, \ldots, v_m \in \mathbb{R}^n$ are non-zero vectors such that $\langle v_i, v_j \rangle \leq 0$ when $i \neq j$. Then $m \leq 2n$.*

Proof. We proceed by induction on n. When $n = 1$, clearly $m \leq 2$, since given any 3 non-zero numbers, at least 2 must have the same sign. When $n > 1$, write $v_i = (\alpha_i, w_i)$, where here $\alpha_i \in \mathbb{R}$ is the first coordinate and $w_i \in \mathbb{R}^{n-1}$ denotes the remaining coordinates. We have

$$\langle v_i, v_j \rangle = \alpha_i \alpha_j + \langle w_i, w_j \rangle.$$

What is the example showing that Lemma 1 is tight?

First observe that without loss of generality $w_1 = 0$ and $\alpha_1 > 0$. This is because we can always rotate all of the vectors so that v_1 has this form, and this does not affect the inner products between the vectors. By assumption, for all $i \neq 1$, $\alpha_i \alpha_1 = \langle v_i, v_1 \rangle \leq 0$, so $\alpha_i \leq 0$ for all $i > 1$.

There can be at most one $i > 1$ with $w_i = 0$, since if there were 2 such vectors w_i, w_j, then $\alpha_1, \alpha_i, \alpha_j$ would violate the base case. When $i, j > 1$, $\alpha_i \alpha_j \geq 0$, so $\langle w_i, w_j \rangle$ must be non-positive when i, j are distinct. Thus, we find $m - 2$ vectors in $\mathbb{R}^{n-1}$ that have pairwise non-positive inner products. By induction we get $m - 2 \leq 2(n-1)$, proving that $m \leq 2n$. □

As a corollary we get:

Theorem 2 (Plotkin Bound). *If C is a binary code with relative distance $\delta \geq 1/2$, then $|C| \leq 2n$.*

To give a bound that applies even when $\delta < 1/2$, we need to find a clever mapping from the codewords to the vector space. Suppose we have a binary code $C \subseteq \{0,1\}^n$ as above. Pick a random string $z \in \{0,1\}^n$ and consider the set $C \cap B(z, w)$, for a parameter w that we set later. The expected size of this set is $|C| \cdot \mathsf{vol}(w)/2^n$, which is still quite large if w is chosen to be close to $n/2$. In effect, we have found a subset of the code of this size which has the same distance, yet all the codewords have weight at most w.

For each codeword x in this set, define the vector $u(x)$ where for each i, we set $u(x) = v(x) - \alpha \cdot v(z)$, for some scalar α that we fix below. Now, for two distinct codewords x, y of weight w,

Recall that we defined $v(x) \in \mathbb{R}^n$ by $v(x)_i = (-1)^{x_i}$.

$$\begin{aligned}
&\langle u(x), u(y)\rangle \\
&= \langle v(x) - \alpha \cdot v(z), v(y) - \alpha \cdot v(z)\rangle \\
&= \langle v(x), v(y)\rangle - \alpha\langle v(x), v(z)\rangle - \alpha\langle v(z), v(y)\rangle + \alpha^2 n \\
&\leq n - 2d - 2\alpha(n - 2w) + \alpha^2 n \\
&= \alpha\Big(\frac{n-2d}{\alpha} + \alpha n - (n - 2w)\Big)
\end{aligned}$$

It is best to set $\alpha = \sqrt{\frac{n-2d}{n}} = \sqrt{1-2\delta}$. Then the above quantity is

$$= \alpha\Big(2\sqrt{n(n-2d)} - (n-2w)\Big).$$

If we set $w = n/2 - \sqrt{n(n-2d)}/2$, the above quantity is non-positive. To summarize, we have found $|C| \cdot \mathsf{vol}(w)/2^n$ vectors with non-positive inner products. Using (1) and Lemma 1 gives that if $d/n \leq 1/2$:

In the exercises, you will be asked to generalize these ideas to the case of alphabets of size q.

$$|C| \leq (2n) \cdot 2^n/\mathsf{vol}(w) \leq (2n) \cdot 2^{n(1-h_2(1/2-\sqrt{1-2(d/n)}/2))+O(\log n)}.$$

Stated a different way, we have established that:

Theorem 3 (Elias-Bassalygo bound). *Every binary code with $\delta \leq 1/2$ must satisfy:*

$$R + h_2\Big(\frac{1-\sqrt{1-2\delta}}{2}\Big) \leq 1 + O(\log n)/n.$$