Coding Theory in a Nutshell

The 20th century was the century of higher dimensions. Positions became phase spaces; species, ecosystems; voltages, signals. With large dimensions came new geometry — curvature, catastrophes, concentration — geometry underpinning the technologies of that age. We'll tell the story of how concentration of measure led to dense ball packing and thus to astonishingly efficient communication.

Following Shannon, we'll formalize protocols for redundantly communicating long signals across noisy channels. We seek concise (small redundancy), reliable (small error rate) protocols. As entropy clearly lower bounds concision, we ask: how does reliability deteriorate as we approach that bound? For example, how loose is the concision bound among very-reliable protocols? That bound eludes all low-dimensional examples we'll make by hand. And yet the bound is tight: miraculously, there are protocols arbitrarily close to the entropic bound with arbitrarily low error-rate!

Shannon's 1948 miracle is strikingly non-constructive: it gives no concrete optimal protocol. It would take another 2 billion seconds of dogged hunt for workers to construct a provably optimal protocol. After sketching this hunt's history, we'll construct that protocol, analogize to fast fourier transforms, and handwaive optimality.

A Noisy Channel

redundancy

Say we have a channel across which we transmit a stream of 0s and 1s. Alas, each transmitted bit has an independent chance $p = 1 - q < 1/2$ of flipping to the other value. Thus, to communicate some length-N message $m \in 2^N$, we transmit some longer, *coded* bitstring $y = \mathrm{enc}(m) \in 2^{RN}$. Here $R \in [1, \infty)$ is the *redundancy.* The receiver decodes a corrupted bitstring $\hat{y} = y \oplus z$, which it decodes via $\hat{m} = \mathrm{dec}(\hat{y}) \in 2^N$. This pair $(\mathrm{enc}, \mathrm{dec})$ is our *communication protocol*, or *code.*

$$m \xrightarrow{\mathrm{enc}} y \xrightarrow{\oplus z} \hat{y} \xrightarrow{\mathrm{dec}} \hat{m}$$

error rate

We seek concise (low R), reliable (low ϵ) protocols. Here ϵ is the chance of decoding m incorrectly averaged uniformly over all messages:

$$\epsilon = \mathbb{P}_m \mathbb{P}_z \left[m = \mathrm{dec}(\mathrm{enc}(m) \oplus z) \right]$$

We minimize ϵ with respect to a given encoder by decoding each received $\hat{y}$ to a minimum $\mathrm{dec}(\hat{y}) = \arg\min \delta(\mathrm{enc}(\hat{m}), \hat{y})$ of the hamming distance. We'll by default assume this dec; algorithmic concerns may suggest other decoders.

distance

We say our code enjoys *minimum distance* and *radius*:

$$\delta_\star = \min_{m \neq m'} \delta(\mathrm{enc}(m), \mathrm{enc}(m')) \qquad\qquad r_\star = \lceil \delta_\star / 2 \rceil$$

With $S_p^n = \sum_{0 \leq k < n} \binom{RN}{k} p^k q^{RN-k}$ the chance of less than k corruptions, we bound

$$p^{r_\star} q^{NR - r_\star} / 2^N \leq \epsilon \leq 1 - S_p^{r_\star}$$

For any fixed code, this *binomial bound* tightens as $p \to 0$: $\epsilon \in \Theta(p^{r_\star})$.

Protocols Matter

linear

A rich class of protocols arise with enc an injective linear map : $2^N \to 2^{RN}$; here, we regard the domain and codomain as vector spaces over the field with 2 elements. Due to linearity (and injectivity), the minimum distance coincides with the minimum hamming norm $|y|$ among images of nonzero messages.

A representative case is $N = 32$, $RN = 9 \cdot 32$, $p = 1/20$. That is, we wish to communicate four bytes — perhaps a word in a novel — by transmitting thirty six bytes, knowing that roughly fourteen bits will get corrupted. How small can we get ϵ?

repetition

What first comes to mind is the *repetition code*

$$y = \text{enc}(m) = (m, m, m, m, m, m, m, m, m)$$

We decode each bit of m by taking the majority value among the 9 received estimates of that bit. We get a bit wrong only when at least 5 estimates are corrupted (relatedly, $\delta_\star = 5$). The estimates are mutually independent and each estimate has error rate linear in p as $p \to 0$, so $\epsilon \sim 32\binom{9}{5}p^5 \sim 1.3 \cdot 10^{-3}$. We expect our novel to have an error every couple pages.

pairwise

But we can use our redundancy more effectively! Let

$$y = \text{enc}(m) = (m, Wm)$$

where W, a $(16{\cdot}16)\times 32$ matrix, gives for each "even-odd" pair $(2i, 2j+1) \in [0, 16)^2$ the sum $m_{2i} \oplus m_{2j+1}$. Intuitively, W "mixes up" m's bits, diluting each corruption across m's many bits until it is correctible.

One way to decode the bit $\hat{m}_0$ from $\hat{y} = (\tilde{m}, \tilde{Wm})$ is to take the majority value among the 17 estimates

$$\tilde{m}_0 \qquad \tilde{m}_1 \oplus (\widetilde{Wm})_{(0,1)} \qquad \tilde{m}_3 \oplus (\widetilde{Wm})_{(0,3)} \qquad \cdots$$

Likewise to decode the other bits. We get a bit wrong only when at least 9 estimates are corrupted (relatedly, $\delta_\star = 9$). The estimates are mutually independent and each estimate has error rate linear in p as $p \to 0$, so $\epsilon \sim 32\left(2^4\binom{8}{4} + 2^5\binom{8}{5}\right)p^9 \sim 4 \cdot 10^{-5}$, a drastic improvement over the repetition code. We expect only a few errors in our novel.

A Beautiful Code

hamming

Imagine a simplex on k vertices. There are $\binom{k}{d+1}$ cells of dimension d. Take $0 \leq D < k$; call cells of dimension $d < D$ *parity cells* and the remaining cells *data cells.* For each labeling of data cells by bits, we get a labeling of parity cells by bits: we label each p-cell by the mod-2 sum of the labels of the containing d-cells. We regard the full labeling as a bitstring of length $2^k - 1$. Let N count the data cells, let $RN = 2^k - 1$ count all cells, and let enc send a data labeling to a full labeling. This gives us a *generalized Hamming code.*

For example, with $D = 1$, we get the classic Hamming codes with $(N, RN, \delta_\star) = (2^k - 1 - k, 2^k - 1, 3)$. That $\delta_\star \geq 3$ reflects that if every atom appears in an even number of members of a nonempty set $\mathcal{S}$ of sets, then $\mathcal{S}$ has size at least 3. We may thus correct any single corruption.

Higher Ds give codes robust to more corruptions. For example, $(k, D) = (5, 2)$ gives a $(16, 31, 6)$ code?????? That $\delta_\star \geq 5$ reflects that if every size-≤ 2 set includes into an even number of members of a nonempty set $\mathcal{S}$ of sets, then $\mathcal{S}$ has size at least 5.

Obstructions to Concise, Reliable Codes

entropy

Which tuples N, RN, p, ϵ are feasible? Let Δ denote the vector difference $\Delta = m - \hat{m}$. Then $(\hat{y}, \Delta)$ determines $(m, z) = (\text{dec}(\hat{y}) \oplus \Delta, \hat{y} - \text{enc}(m))$. So, by subadditivity, data processing, and independence:

$$H(\hat{y}) + H(\Delta) \geq H(\hat{y}, \Delta) \geq H(m, z) = H(m) + H(z)$$

Plugging in $H(\hat{y}) \leq RN$, $H(\Delta) \leq H(\epsilon)N$, $H(m) = N$, and $H(z) = H(p)RN$, we see

$$R \geq \frac{1 - H(\epsilon)}{1 - H(p)}$$

In particular, we need $R \geq 1/(1 - H(p))$ to have any hope of a small ϵ.

volume

Besides the entropy bound, there are ball packing obstructions. Indeed, to get good rates for small p, we want large $r_\star$s. We want to pack 2^N — but by additivity of volume can pack at most $1/S^r_{1/2}$ — disjoint (open) hamming-balls of radius r.

In more detail: the median ball's volume is at most $2^{NR} \cdot 2^{N-1}$. In turn, the volume is at least $\binom{NR}{r}$ for any specific closed r-ball. So at least half of the balls have closed radii obeying $2^{NR}/2^{N-1} \geq \binom{NR}{r}$ or, by a Stirling approximation,

$$1 - (N-1)/NR \geq H(r/(NR))$$

For this median r we have $\epsilon \geq (1/2)p^{r+1}q^{NR-r-1}$.

geometry

Perhaps most interestingly, the hypercube's metric structure prevents us from achieving the volume bound. First, when $2^N \geq 3$ we must have $\delta_\star \leq (2/3)RN$. This is due to the hypercube's positive curvature; for, $\delta(a, z), \delta(b, z) \geq (1/2 + x)RN$ implies $\delta(a, b) \leq (1 - 2x)RN$. Though the volume bound allows $N, RN, \delta_\star = 2, 7, 5$, this curvature bound does not.

We may extend this idea to slightly smaller $x = \delta_\star/(RN)$. How many length-RN bitstrings have distance at least xRN from the RN-zeros string and from the $(1-x)RN$-ones-then-xRN-zeros string? The exact count is:

$$\sum_{0 \leq sRN \leq (1-x)RN} \sum_{0 \leq tRN \leq xRN} [\![s \geq t]\!][\![s + t \geq x]\!] \binom{(1-x)RN}{sRN} \binom{xRN}{tRN}$$

The summand is maximized near $s = x/2 = t$ at which point it has log roughly:

$$N \le \left(H\left(\frac{1}{2}\frac{x}{1-x}\right)(1-x)+x\right)RN$$

Say $y = x/(1-x) = 1/(1/x-1)$, $x = 1/(1+1/y) = y/(1+y)$, $1-x = 1/(1+y)$. Then $1/R \le (H(y/2)+y)/(1+y) \le 2/3+(2-y)^{3/4}/2$, so $2-(2(1/R-2/3))^{4/3} \ge y$, whence we get an upper bound on x.

a map

A Miracle

Polar Codes

Analysis of Polar Codes

Appendix A: Entropy and Binomials

Here, $H(z) = z\lg(\frac{1}{z})+(1-z)\lg(\frac{1}{1-z})$ is *entropy*.

Stirling says that $\binom{T}{xT} \sim 2^{H(x)T}/\sqrt{2\pi x(1-x)T}$, or, for small x:

$$\lg\binom{T}{xT} \approx H(x)T - \frac{1}{2}\lg(xT) - 0.66$$

This is good plus-minus 0.50 for $1 \le xT \le T/4$.

We find it useful to bound binomial sums. For $x < 1/2$: FIXX

$$\begin{aligned}
&q^{xT}\sum_{0\le i\le xT}\binom{T}{i}(p/q)^i \\
\le&\left(\frac{qx}{(1-x)}\right)^{xT}\cdot\sum_{0\le i\le xT}\binom{T}{xT}\left(\frac{(1-x)p}{xq}\right)^i \\
\le&\left(\frac{qx}{(1-x)}\right)^{xT}\cdot\binom{T}{xT}\cdot\frac{xq}{xq-(1-x)p}
\end{aligned}$$