AGENDA

1. RANDOM CHANNEL MODEL
2. SHANNON's THM (statement)
3. CONCATENATED CODES ACHIEVE CAPACITY (BUT...)

1. RANDOM CHANNEL MODEL

- So far, we have focused on the trade-off between RATE and DISTANCE.
- We chose DISTANCE because it nicely captures WORST-CASE error/erasure tolerance.
- Moreover, DISTANCE was nice for applications like compressed sensing and group testing.

- HOWEVER, the worst-case error model is pretty pessimistic. This motivates a RANDOMIZED MODEL for errors.

Note. The RANDOM (or STOCHASTIC or SHANNON) model is extremely well-studied and we will largely ignore it in this class. See EE 376A (Information Theory) or EE 387 (Modern Coding Theory) for more on this very cool topic!
The model is this:

**DEF.** A **MEMORYLESS CHANNEL** $W$ with input alphabet $X$ and output alphabet $Y$ is specified by a probability distribution,

$$W(y|x) = \text{"the probability that } y \text{ came out of } W \text{ given that } x \text{ went in."}$$

**EXAMPLE:** $X = \{0, 1\}$, and $W(y|x) = \begin{cases} p & y = x \\ 1-p & y = \overline{x} \end{cases}$ for $p \in (0,1)$.

Thus, $W$ flips a bit with probability $p$.

We draw this as:

```
0 \quad 1
\quad \quad
\quad \quad
1-p \\
\quad \quad
1 \\
```

That is, $W$ **ERASES** a bit with probability $p$.

**EXAMPLE:** $X = \{0, 1\}$, $Y = \{0, 1, \overline{1}\}$, with $W$ given by:

```
0 \quad 1
\quad \quad
\quad \quad
1-p \\
\quad \quad
1 \\
```

These channels are "memoryless" because they act on one bit at a time, independently.

Our picture of error correcting codes thus looks like:

```
message \xrightarrow{\text{encoder}} codeword \xrightarrow{W} \text{corrupted codeword} \xrightarrow{\text{decoder}} \text{hopefully equal to } x...
```
Let $C \subseteq \mathcal{X}^n$ be an error correcting code with encoding map $\text{Enc}: \mathcal{X}^k \rightarrow \mathcal{X}^n$ and decoding map $\text{DEC}: \mathcal{Y}^n \rightarrow \mathcal{X}^k$.

Let $W$ be a channel with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$.

The **failure probability** of $C$ on $W$ is at most $\eta$ if for all $x \in \mathcal{X}^k$,

$$P_W(\text{Dec}(W(\text{Enc}(x))) \neq x) \leq \eta$$

This is a random variable which represents the output of the channel $W$ on the input $\text{Enc}(x)$.

Shannon showed that every channel has a **capacity**, $C \in [0,1]$, so that transmitting at rate $R>C$ reliably is impossible, but transmitting at rate $R<C$ is possible.

This is what Shannon's Theorem says for the BSC:

**Thm (Shannon's Channel Coding Thm for the BSC)**:

For all $p \in [0, \frac{1}{2})$ and all $\epsilon \in [0, \frac{1}{2} - p]$, the following holds for large enough $n$:

1. For $\frac{k}{n} \leq (1 - H_2(p + \epsilon)) \cdot n$, there exist $S > 0$, and $\text{Enc}: \{0,1\}^k \rightarrow \{0,1\}^n$, $\text{Dec}: \{0,1\}^n \rightarrow \{0,1\}^k$ such that for all $x \in \{0,1\}^k$,

$$P_{\text{BSC}_p}\{\text{Dec}\left(\text{BSC}_p(\text{Enc}(x))\right) \neq x\} \leq 2^{-\delta n}$$

aka, if the rate is a smidge below $1-H_2(p)$, the failure prob can be really tiny.

2. If $\frac{k}{n} \geq (1 - H_2(p + \epsilon)) \cdot n$, then for all such $\text{Enc}, \text{Dec},$

$$P_{\text{BSC}_p}\{\text{Dec}\left(\text{BSC}_p(\text{Enc}(x))\right) \neq x\} \geq \frac{1}{2}$$

aka, if the rate is a smidge above $1-H_2(p)$, the failure prob is at least $\frac{1}{2}$. 
ASIDE. More generally, Shannon's Theorem says that the capacity of \( W \) is
\[
C = \max_{p(x)} I(X; Y)
\]
where \( I(X; Y) \) is the mutual information.

The proof of Shannon's theorem is best done with information theory (see EE 376).

Here's a handwavy sketch of an argument to prove the BSC case directly:

pf (sketch)

1. A random code works great for the achievability result.
2. For the impossibility result, consider dividing up \( 0,1,1\) into a bunch of chunks, \( D_x = \{ y \mid \text{Dec}(y) = x \} \).

\[
\begin{align*}
D_x &= \{ y \mid \text{Dec}(y) = x \} \\
\{ 0,1,1 \} &= \{ 0,1,1 \}
\end{align*}
\]

Now, consider what happens to \( \text{Enc}(x) \) when it goes through the BSC:

Thus, we better have that most of \( S_x \) is contained in \( D_x \), or there would be some big probability of failure.
But then we should have, for all $x$:

$$\text{Vol} \left( D_x \right) \geq \text{Vol} \left( S_x \right) = \text{Vol}_2 \left( \mathbf{p}_n, n \right) = 2^n \cdot H(p)$$

So then how many $D_x$'s could possibly fit in $\Sigma_1^n$? At most $\frac{2^n}{2^{n \cdot H(p)}} = 2^{n \cdot H(p)}$.

So $|C| \leq 2^{n \cdot H(p)}$, so $R \leq 1 - H(p)$ [or $n$].

So, to recover from a $p$-fraction of errors:

Natural question: What if I want to efficiently decode from random errors.
**CONCATENATED CODES achieve CAPACITY**

**THM.** For every $p$ and every $\epsilon \in (0, 1 - H_2(p))$, and all large enough $n$, there is a binary linear code $C \subseteq \{0,1\}^n$ with rate $R > 1 - H_2(p) - \epsilon$, so that:

(a) $C$ can be constructed in time $\text{poly}(n) + 2^{O(\epsilon^2 n)}$
(b) $C$ can be encoded in time $O(n^2)$
(c) There is a decoding alg $\text{DEC}_C$ for $C$ that runs in time $\text{poly}(n) + n 2^{O(\epsilon^2 n)}$

and has failure probability at most $2^{-\Omega(\epsilon^2 n)}$ over BSC$(p)$.

Thus, this code “achieves capacity” on the BSC, in the sense that the rate can get arbitrarily close to $1 - H_2(p)$.

**DRAWBACK!** As the rate gets close to $1 - H_2(p)$, the running time of these algorithms blow up exponentially in $1/\epsilon$.

Whether or not this could be avoided (with efficient algos) was open for a long time... but then in 2009 Arikan introduced **Polar Codes** which will do it. We might talk about polar codes later in class.

But for now let’s prove (or, sketch the proof of) this theorem.

It turns out, we’ve already seen the answer! Concatenated codes!
PROOF SKETCH for the THEOREM:

Choose a parameter $\gamma \text{ TBD}$.

**CODE CONSTRUCTION:** Concatenated Code with:

<table>
<thead>
<tr>
<th>Code</th>
<th>Dimension</th>
<th>Blocklen</th>
<th>Rate</th>
<th>Decoding Time</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{in}$</td>
<td>$k_{in}$</td>
<td>$n_{in}$</td>
<td>$2^{1-H_2(p) - \epsilon/2}$</td>
<td>$T_{in}(n_{in})$</td>
<td>Fail prob on BSC: $\gamma/2$</td>
</tr>
<tr>
<td>$C_{out}$</td>
<td>$k_{out}$</td>
<td>$n_{out}$</td>
<td>$2^{k_{in}}$</td>
<td>$1 - \epsilon/2$</td>
<td>$T_{out}(n_{out})$</td>
</tr>
</tbody>
</table>

We'll see how to get these in a moment...

**RATE** is $R = (1 - H_2(p) - \epsilon/2) (1 - \epsilon/2) \geq 1 - H_2(p) - \epsilon$.

**DECODING ALG** is the one that wasn't a good idea last week:

Given $\{(y_1, ..., y_{n_{out}}) \in \mathbb{F}_2^{n_{in}}\}_{n_{out}}$

FOR $i = 1, ..., n_{out}$:

use $C_{in}$'s decoder to obtain $y_i' = DEC_{in}(y_i) \in \mathbb{F}_2^{k_{in}}$ or $\mathbb{F}_2^{k_{out}}$

Decode $y' = (y_1', ..., y_{n_{out}}')$ using $DEC_{out}$ (Cout's decoder), and

RETURN ($DEC_{out}(y')$)

Say $DEC_{in}$ takes time $T_{in}(n_{in})$, $DEC_{out}$ takes time $T_{out}(n_{out})$.

Then the decoding time is

$$\text{Decoding Time} = O(n_{out} \cdot T_{in}(k_{in}) + T_{out}(n_{out})) = \text{TBD}$$

**ENCODING TIME** = $O(n^2)$, since the code is linear

**CONSTRUCTION TIME**: TBD
Error Probability:

Say that \( C_{\text{out}} \) has relative distance \( \gamma \). Then

\[
P\{ \text{decoder fails} \} = P\{ > \gamma n \ \text{blocks are incorrectly decoded by} \ C_{\text{in}} \}
\]

For a fixed block \( i \), \( P\{ \ C_{\text{in}} \text{decodes the } i^{th} \text{block incorrectly} \} \leq \gamma / 2 \).

Each bit is flipped independently, so

\[
\text{BSC}(\frac{1}{2}) \sim \text{BSC}(\frac{1}{2}) \times \text{BSC}(\frac{1}{2}) \times \cdots \times \text{BSC}(\frac{1}{2})
\]

So \( P\{ \geq \gamma \text{ blocks are in error} \} \)

\[
\leq P\{ \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} (1 - \prod_{j \text{ on block } i} \frac{1}{2}) > \gamma \}
\]

\[
\leq P\{ \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} (1 - \prod_{j \text{ on block } i} \frac{1}{2}) > 2 \cdot \text{IE} \left[ \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} (1 - \prod_{j \text{ on block } i} \frac{1}{2}) \right] \}
\]

\[
\leq \exp \left( -\gamma \cdot n_{\text{out}} / 6 \right)
\]

So the error probability is indeed exponentially small.

But now.... What codes to use for \( C_{\text{in}}, C_{\text{out}} \)?

Inner Code: Just like before, let's try ALL the binary linear codes.

Construction Time: \( 2^{O(n_{\text{in}})} \). There are \( 2^{k_{\text{in}}} \) codes to check, and it takes \( 2^{k_{\text{in}}} 2^{O(n_{\text{in}})} \) time to compute the error probability for each one.

Decoding Time: \( 2^{O(k_{\text{in}})} \) to try all the codewords and find the closest one.

Fun Exercise: A random linear code of rate \( 1-H(n) - 6z \) (probably) has fail prob. \( 2^{-O(n)} \)

So choose \( k_{\text{in}} = \Omega(\frac{n}{\log n}) \) and we know there exists a binary linear code that works.
**OUTER CODE:**

**TRY 1:** **REED-SOLOMON.** Actually, NO! Like we saw last week, this would require \( n_{\text{out}} = 2^{k_{\text{in}}^2} \) but then the construction time would be \( 2^{O(k_{\text{in}}^2)} = n_{\text{out}}^{\log(n_{\text{out}})} \), and we get a quasipolynomial-time construction.

Before, we got around this by coming up with a slightly better construction of \( C_{\text{in}} \), that took time \( 2^{O(k_{\text{in}})} \) instead of \( 2^{O(k_{\text{in}}^2)} \). Here, we'll mess with the outer code instead.

**TRY 2:** **BINARY CODES on the ZYABLOV BOUND.**

Meta-logic: We need an efficiently encodable/decodable asymptotically good code.

So far, we have seen:
- RS CODES \( \overleftarrow{\text{We just said NOT this.}} \)
- Concatenated RS CODES \( \overrightarrow{\text{Better be this.}} \)

So let \( C_{\text{out}} \) be an explicit binary code on the Zyablov bound.

![Diagram of binary code construction](image-url)

\( \downarrow \text{Forget about the blocks} \)
\( \downarrow \text{Make new blocks} \)
\( \downarrow \text{Treat each new block as a symbol} \)

This stuff does not change the rate of the code

Since the rate and distance don't get worse, this code still lies at or above the Zyablov bd. To decode, just run the red stuff backwards and then use the decoder from Lec 7.
PARAMETERS:

- Choose $C'_{out}$ to be a binary code on the Zyablov bd (from Lecture 7).
- Choose $k_{in} = \Theta\left(\frac{\log(k_{out})}{\epsilon^4}\right)$ ← This is what we needed for the inner code to exist.
- Make $C_{out} \in \left[\mathbb{F}_{k_{in}}^n\right]^{k_{out}}$ by chopping up $C_{out}$ into chunks of size $k_{in}$.
- Now let's pick $\gamma$. We have

$$S_{out} = (1 - R_{rs}) H^{-1}(1 - r) \leftarrow \text{Zyablov Bd}$$

and recall we want $S_{out} = 2\gamma$.

So choose $R_{rs} = 1 - 2\sqrt{\gamma}$ and $r$ s.t. $H^{-1\gamma}(1 - r) = \sqrt{\gamma}$

This means $r = 1 - O\left(\sqrt{\frac{1}{\gamma}}\right)$

and that implies

$$R_{out} = R_{rs} \cdot r = (1 - 2\sqrt{\gamma})(1 - O\left(\sqrt{\frac{1}{\gamma}}\right))$$

$$= 1 - O\left(\frac{1}{\sqrt{\gamma}}\right).$$

We wanted $R_{out} \geq 1 - \epsilon/2$, which means that we should choose $\gamma$ s.t.

$$\frac{\epsilon}{2} = O\left(\frac{1}{\sqrt{\gamma}}\right).$$

$\gamma = O\left(\epsilon^3\right)$ works, so let's do that.
With our choice of $\gamma = \varepsilon^3$, let's go back and compute stuff.

<table>
<thead>
<tr>
<th>Code</th>
<th>Dimension</th>
<th>Blocklen.</th>
<th>$\Sigma$</th>
<th>Rate</th>
<th>Decoding Time</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cin</td>
<td>$k_{in} = \Theta(q^{1/2})$</td>
<td>$n_{in} = \Theta(k_{in})$</td>
<td>2</td>
<td>$1 - H_2(p) - \varepsilon/2$</td>
<td>$T_{in} (n_{in}) = 2^{O(n_{in})} 2^{O(q^{1/2})}$</td>
<td>Fall prob on BSC: $\gamma/2$</td>
</tr>
<tr>
<td>Caut</td>
<td>$k_{out} = n_{out} = \Theta(\varepsilon n^{1/2})$</td>
<td>2</td>
<td>$1 - \varepsilon/2$</td>
<td>$T_{out} (n_{out}) = \text{poly}(n_{out})$</td>
<td>Distance: $2\gamma$</td>
<td></td>
</tr>
</tbody>
</table>

So:

**Decoding Time**

$$\text{Decoding Time} = O\left(n_{out} \cdot T_{in} (k_{in}) + T_{out} (n_{out})\right) = \text{poly}(n) + n \cdot 2^{O(\frac{q^{1/2} n^2}{\varepsilon^2})}$$

**Failure Probability**

$$\exp\left(-\sqrt{n_{out}/6}\right) = \exp\left(-\Omega\left(\frac{\varepsilon^3 n^{1/2}}{q^{1/2}}\right)\right) = \exp(-O(\varepsilon^3 n))$$

**Construction Time**

$$2^{O(n_{in}^{1/2})} + \text{poly}(n_{out}) = 2^{O(\varepsilon^{-4} q^{5/2})} + \text{poly}(n) = 2^{O(\varepsilon^{-4} q^{5/2})} + \text{poly}(n).$$

And this gives all the things we claimed.

**Questions to Ponder:**

1. Which model (Shannon or Hamming) do you find more compelling?
2. Flesh out the details of our proof of Shannon's Thm for the BSC.
3. Why do we ask for failure probability $2^{-s_2(n)}$? Is $\varepsilon_1^{1000}$ okay?
4. Can you make the (something) RS approach work for achieving capacity on the BSC?