Information Theory

Omar Fawzi*
http://perso.ens-lyon.fr/omar.fawzi/teaching/it/index.html
Master 1, ENS de Lyon

Contents

1 Introduction **2**
1.1 Encoding 1: the Trivial encoding . . . 3
1.2 Encoding 2: the Repetition code . . . 3
1.3 Encoding 3: the Block code . . . 4

2 Information measure **5**
2.1 Probability notations . . . 5
2.2 Entropy of event . . . 6
2.3 Joint entropy and conditional entropy . . . 7

3 Data compression **9**
3.1 Settings . . . 9
3.2 Variable length compression . . . 10
3.2.1 General compressors . . . 10
3.2.2 Uniquely decodable and prefix-free compression . . . 11
3.3 Fixed-length almost lossless compression . . . 14
3.4 Universal compression . . . 19
3.4.1 Arithmetic code . . . 19
3.4.2 Lempel-Ziv coding . . . 20

4 Noisy channel coding **20**
4.1 Setting . . . 20
4.2 Converse bounds . . . 23
4.3 Achievability bound . . . 25

5 Information and combinatorics **29**

6 Error correcting code **31**
6.1 General error-correcting codes . . . 31
6.1.1 General bounds on the best codes . . . 33
6.2 Linear error correcting code . . . 34
6.3 Reed-Solomon codes . . . 38
6.4 Concatenation of codes . . . 42
6.5 An application of ECC . . . 44

7 Introduction to Quantum Information **49**

*omar.fawzi@ens-lyon.fr

1 Introduction

Midterm exam: Friday 4th November, 10:00 a.m.
This course will mainly be a mathematical one, with only a few practical courses.

Background needed:

- A bit of Probability theory
- Linear algebra (finite field)

"Information theory" comes from *Shannon* in 1948 as "Communication theory":

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." where "point" is to be taken at the broad sense.

Ex:

- Point 1: memory at t_1, Point 2: memory at t_2.
- Point 1: DNA of the parent cell, Point 2: DNA of the daughter cell.

$$\text{Point 1} \;—\; \boxed{\text{Channel}} \;—\; \text{Point 2}$$

Two fields of solution:

- Improve the channel
- Accept an error model as given and build a system on top of it to transform it into a reliable one.

$$s \quad — \quad t \quad — \boxed{\text{Noise}} — \quad r \quad — \quad \hat{s}$$

The goal is to achieve $s = \hat{s}$.

As a designer: Find "good" encoding and decoding function. We want $\mathbb{P}(s \neq \hat{s})$ small.

Example: Suppose I have memory cells storing 1 bit suffers from noise. After one year, the bits flip with probability $f \in [0, 1]$.
We model this channel:

$$W(y|x) = \text{prob that output} = y \text{ for input } x$$

For this channel, $W(0|0) = 1 - f$, $W(1|1) = 1 - f$, $W(1|0) = f$, $W(0|1) = f$.

Think that $f = 0.1$, and that we want to tore a file with $n = 10^6$ bits.

1.1 Encoding 1: the Trivial encoding

$$
\begin{array}{ccccc}
s_1 & \xrightarrow{(=)} & t_1 & \text{———}\boxed{\text{Noise}}\text{———} & r_1 \\
s_2 & \xrightarrow{(=)} & t_2 & \text{———}\boxed{\text{Noise}}\text{———} & r_2 \\
\vdots & & \vdots & & \vdots \\
s_n & \xrightarrow{(=)} & t_n & \text{———}\boxed{\text{Noise}}\text{———} & r_n
\end{array}
$$

Decoding $\hat{s} = r_i$.
<u>Bit error:</u> $\mathbb{P}(s_i \neq \hat{s}_i) = f$
<u>Aside:</u> How different are s and $\hat{s}$ distance between s and $\hat{s}$ follows a Binomial(n, f) distribution.

$$\mathbb{E}(\#flips) = nf, \qquad Var(\#flips) = nf(1-f) \tag{1}$$

With high probability, $\#flips \in [nf - 10\sqrt{nf(1-f)},\ nf + 10\sqrt{nf(1-f)}]$
<u>Block error:</u>

$$
\begin{aligned}
\mathbb{P}(s \neq \hat{s}) &= 1 - \mathbb{P}(s \neq \hat{s}) \\
&= 1 - \mathbb{P}(\forall i \in \{1, ..., n\}, s_i = \hat{s}_i) \\
&= 1 - (1-f)^n
\end{aligned}
$$

For this to be small, need $nf \ll 1$. For $n = 10^6$, need $f = 10^-8$.
<u>Rate:</u> $\frac{\#bits\ in\ file}{\#cells\ used} = \frac{n}{n} = 1$.

1.2 Encoding 2: the Repetition code

Encode each bit of file in 3 different cells.

$$
\begin{aligned}
0 &\to 000 \\
1 &\to 111
\end{aligned}
\tag{R_3}
$$

<u>Rate:</u> $\frac{1}{3}$

s	0	1	0
t	000	111	000
r	001	111	010
$\hat{s}$	0	1	0

Decoding: Majority vote.

<u>Bit error</u>

$$
\begin{aligned}
\mathbb{P}(s_1 \neq \hat{s}_1) &= \mathbb{P}(\geq 2\ flips) \\
&= 3f^2(1-f) + f^3 \\
&= 3f^2 - 2f^3 \\
&< f (for\ f < 1/2)
\end{aligned}
$$

Better than trivial encoding. For $f = 0.1$, $\mathbb{P}(s_1 \neq \hat{s}_1) = 0.028$.

Block Error:

$$\mathbb{P}(s \neq \hat{s}) = 1 - \mathbb{P}(\forall i \in \{1, ..., n\}, s_i = \hat{s}_i$$
$$1 - (1 - 3f^2 + 2f^3)^n$$

Slightly better but not so good.

HW: Generalize to N repetitions.

1.3 Encoding 3: the Block code

Make block of size 4 and encode each one: (7,4)-Hamming code.

$$\begin{array}{ccccc}
 & & t_1 & \text{---}\boxed{\text{Noise}}\text{---} & r_1 \\
s_1 & & t_2 & \cdots & \\
s_2 & \text{---}\boxed{\text{Encode}}\text{---} & t_3 & \cdots & \\
s_3 & & t_4 & \cdots & \\
s_4 & & t_5 & \cdots & \\
 & & t_6 & \cdots & \\
 & & t_7 & \cdots &
\end{array}$$

$$\begin{aligned}
t_1 &= s_1 \\
t_2 &= s_2 \\
t_3 &= s_3 \\
t_4 &= s_4 \\
t_5 &= s_1 \oplus s_2 \oplus s_3 \\
t_6 &= s_2 \oplus s_3 \oplus s_4 \\
t_7 &= s_1 \oplus s_3 \oplus s_4
\end{aligned}$$

Rate: $\frac{4}{7}$

Decode: $r_1,\ r_2,\ ...\ r_7$

Ex:

$$\overbrace{1000}^{s} \to \overbrace{1000101}^{t}$$

Decoding: Flip the bit that is in all violated circles and not in good circle.

If ≤ 1 error, recover $T_1, ..., t_7$ from $r_1, ...r_7$.

Bit error: One can show that

$$\mathbb{P}(s_i \neq \hat{s_i}) \leq 9f^2 + O(f^3)$$

Block error:

$$
\begin{aligned}
\mathbb{P}(s \neq \hat{s}) &= 1 - \mathbb{P}(\forall i \in \{1, ..., \frac{n}{4}\}, \forall j \in \{0,1,2,3\}, s_{4i+j} = \hat{s}_{4i+j}) \\
&= 1 - \prod_{i=1}^{n/4} \mathbb{P}(\forall j \in 0,1,2,3,4, s_{4i+j} = \hat{s}_{4i+j}) \\
&\leq 1 - \mathbb{P}(\leq 1 \text{ error in a block})^{n/4} \\
&= 1 - (1 - \left(\binom{7}{2} f^2 (n-f)^5 - ...\right)^{n/4} \\
&= 1 - (1 - 21 f^2 - O(f^3))^{n/4}
\end{aligned}
$$

The conventional wisdom was: "to decrease error probability, we need to decrease the rate to 0. But Shannon showed that we can do much better. **We can make the error probability go arbitrary close to 0 with a constant rate > 0.** Even more, we can make the block error tate arbitrary close to zero at positive rate.

Ex: $f = 0.1$ File $n = 10^6$ bits.

Use $\simeq 2.10^6$ cells with very small block error.

2 Information measure

There are many approaches to define entropy, which mainly depends on the the question we ask.

Ex: Given data X, determine the minimum space needed to store X.

- Find the shortest description of X
  Solution: a description is an algorithm that computes X. This is the *Algorithmic complexity*, also called *Kolmogorov complexity*.

$X = 0...0$	"small"
$X = \pi$	"small"
$X = $"*random*"	"large"

  Problem: This in not computable.
- More useful approach of Shannon
  Entropy = measure of likelihood of X (Thus we need a probability model).

2.1 Probability notations

All system are finite $(\Omega, \mathcal{E}, \mathbb{P})$. X random variable in $\mathcal{X}$. We note $P_X(x) = \mathbb{P}(X = x)$.

For joint random variables, we note:

$$
\begin{aligned}
P_{XY}(x,y) &= \mathbb{P}(X = x, Y = y) \\
P_{X|Y=y}(x) &= \mathbb{P}(X = x | Y = y) \\
P_X^{\times n} &= P_X \times P_X \times ... \times P_X \quad n \text{ times} \\
\mathbb{E}(X) &= \sum_{x \in \mathcal{X}} x P_Y(x)
\end{aligned}
$$

2.2 Entropy of event

$$h_X : \mathcal{E} \to \mathcal{R}_+ \cup \{\infty\}$$

1. Independence of representation: $h(E)$ only depends on $\mathbb{P}(E)$
2. Continuity with respect $\mathbb{P}$: h continuity in $\mathbb{P}$
3. Additivity: $h(E \cap E') = h(E) + h(E')$ if E and E' are independent
4. Normalization: $h(E) = 1$ if $\mathbb{P}(E) = \frac{1}{2}$

Propriety 1. *h_X satisfies 1, 2, 3, 4* $\Leftrightarrow h(E) = -\log_2 \mathbb{P}(E)$

Proof. Skipped. □

h is also called *surprisal*.
If X is a random variable, we define:

$$\begin{aligned} h_X(x) &= h_X(\{X = x\}) \\ &= -\log_2 P_X(x) \end{aligned}$$

$$\begin{aligned} h_X : \mathcal{X} &\to \mathbb{R}_+ \cup \{\infty\} \\ x &\mapsto -\log_2 P(x) \end{aligned}$$

$h(X)$ is a random variable. It's distribution is

Definition 1 (Shannon entropy)**.** *The Shannon Entropy of X is:*

$$\begin{aligned} H(X) &= \mathbb{E}(h_X(X)) \\ &= -\sum_{x \in \mathcal{X}} P_X(x) \log_2 P_X(x) \end{aligned}$$

Remarks

- Only depends on P_X and not on the values taken
- Units is "bits"
- $0 \log_2 0 = 0$

Remark on notation $P_X(X)$ this is $P_X : \mathcal{X} \to \mathbb{R}_+$ applied to the random variable X. It is <u>NOT</u> $\mathbb{P}(X = X) = 1$.

Propriety 2. *For any $x \in \mathcal{X}$:*

$$0 \leq H(X) \leq \log |\mathcal{X}|$$

With the equality cases $H(X) = 0$ if and only if X is constant and $H(X) = \log |\mathcal{X}|$ if and only if X is uniform on $\mathcal{X}$.

Proof.
- First inequality: easy

- $$H(X) = \mathbb{E}\left(\log_2 \frac{1}{P_X(X)}\right)$$

  As log is concave :

  $$\begin{aligned} &\leq \log_2 \mathbb{E}_X \left(\frac{1}{P_X(X)}\right) \\ &= \log_2 \sum_{x \in \mathcal{X}} P_X(x) - \frac{1}{P_X(x)} = \log_2 |\mathcal{X}| \end{aligned}$$

  Equality condition: all $P_X(x)$ are equal so P_X is the uniform distribution.

$\square$

Remark Expectation $\mathbb{E}(h_X(X))$ is not the only interesting quantity. For example

$$\begin{aligned} H_{\min}(X) &= \min_{x \in \mathcal{X}} h_X(x) \\ &= -\log_2 \max_x P_X(x) \end{aligned}$$

Ex If $X \in \{0, 1\}$ $P_X(0) = 1 - p$ and $P_X(1) = p$:

$$H(X) = -p \log_2 p - (1-p) \log(1-p)$$

2.3 Joint entropy and conditional entropy

Definition 2 (Joint entropy)**.** *Let $X \in \mathcal{X}$, $Y \in \mathcal{Y}$. The joint entropy $H(X, Y)$ is defined as:*

$$\underbrace{H(X,Y)}_{H(XY)} = -\sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y}}} P_{XY}(x,y) \log_2 P_{XY}(x,y)$$

Definition 3 (Conditional entropy)**.** *The conditional entropy $H(X|Y)$ is defined as:*

$$H(X|Y) = \sum_{y \in \mathcal{Y}} P_Y(y). \underbrace{H(P_{X|Y=y})}_{H(X|Y=y)}$$

Ex

- $X = Y$, then $H(X|Y) = \sum_{y \in \mathcal{Y}} P_Y(y) H(P_{X|Y=y}) = 0$
- Y and X are independent, then $H(X|Y) = \sum_{y \in \mathcal{Y}} P_Y(y) \underbrace{H(P_{X|Y=y})}_{=H(P_X)} = H(X)$

Propriety 3.

$$H(X|Y) = H(XY) - H(Y)$$

Proof.

$$\begin{aligned}
P_{XY}(x,y) &= P_Y(y)P_{X|Y=y}(x) \\
H(XY) &= -\sum_{x,y} P_{XY}(x,y)\log_2 P_Y(y)P_{X|Y=y}(x) \\
&= -\sum_{x,y} P_{XY}(x,y)\log_2 P_Y(y) \\
&\quad -\sum_{x,y} P_{XY}(x,y)\log_2 P_{X|Y=y}(x) \\
&= H(Y) \qquad\qquad \left(\text{as } \sum_x P_{XY}(x,y) = P_Y(y)\right) \\
&\quad -\sum_y P_Y(y) \underbrace{\sum_x P_{X|Y=y}(x)\log P_{X|Y=y}(x)}_{-H(X|Y=y)} \\
&= H(Y) + H(X|Y)
\end{aligned}$$

□

Definition 4 (The mutual information)**.**

$$\begin{aligned}
I(X:Y) &= H(X) - H(X|Y) \\
&= H(X) + H(Y) - H(XY) \\
I(X:Y) &= \sum_{x,y} P_{XY}\log_2 \frac{P_{XY}(xy)}{P_X(x)P_Y(y)}
\end{aligned}$$

Examples

- If $X = Y$, $I(Y:Y) = H(X)$
- If X and Y are independent, $I(X:Y) = 0$

Definition 5. *Let P and Q be distributed on $\mathcal{X}$. The relative entropy*

$$D(P||Q) = \sum_{x\in\mathcal{X}} P(x).\log_2 \frac{P(x)}{Q(x)}$$

Remark

- Common name Kullback-Leibler divergence.
- If $P(x) = 0$, $P(x)\log\frac{P(x)}{Q(x)} = 0$.
- If for some $X \in \mathcal{X}$, $P(x) > 0$ but $Q(x) = 0$, $D(P||Q) = \infty$.
- Not symmetric between P and Q
- $D(P||P) = 0$
- $I(X,Y) = D(P_{XY}||P_X \times P_Y)$

Propriety 4. *For any dist P, Q*

$$D(P||Q) \geq 0$$

with equality if and only of $P = Q$

Proof. Let $S = \{x : P(x) > 0\}$

$$\begin{aligned} D(P||Q) &= -\sum_{x\in S} P(x)\log_2 \frac{Q(x)}{P(x)} \\ &- \log_2 \text{ is convex} \\ &\geq -\log_2 \sum_{x\in S} P(x)\frac{Q(x)}{P(x)} \quad (1) \\ &= -\log_2 \sum_{x\in S} Q(x) \\ &\geq 0 \quad (2) \end{aligned}$$

Equality condition:

1. Strict convexity: $\frac{Q(x)}{P(x)} = C.\ \forall X \in S$
2. $\sum_{x\in S} Q(x) = 1$
   This implies that $Q = P$

□

Corollary 1. *For any* X, Y

$$I(X : Y) \geq 0 \quad (*)$$

with equality if and only if X *and* Y *are independent*

Proof. Just write $I(X : Y) = D(P_{XY}||P_X \times P_Y)$ □

Another way of writing (*)

$$\begin{aligned} H(X) &\geq H(X|Y) \\ H(X) + H(Y) &\geq H(XY) \end{aligned}$$

3 Data compression

3.1 Settings

Also called *source coding.*

In interesting data: not all possible sequences are expected.

Setting Source $X \in \mathcal{X}$ with distribution P_X

$$C : \mathcal{X} \to \{0,1\}^*$$

Two variants:

- Variable length compression
  $|C(x)|$ might be different from $|C(x')|$. Want to minimize, e.g., *expected* length $\mathbb{E}(|C(X)|)$
- Fixed-length compression, allow a probability of error δ and minimize the length.

3.2 Variable length compression

3.2.1 General compressors

Definition 6. *A variable length lossless compressor is a function $C : \mathcal{X} \to \{0,1\}^*$ such that there is a decompressor $D : \{0,1\}^* \to \mathcal{X}$ with $D \circ C(x) = x$ for all $x \in \mathcal{X}$*

Note

- Equivalent condition: C is injective
- For $x \in \mathcal{X}$, $C(x)$ is called a code word
  $\{C(x) : x \in \mathcal{X}\}$ is called code or codebook.

Objective Find C that minimizes $\mathbb{E}(|C(X)|)$

Theorem 1. *Let P_X be a distribution on $\mathcal{X}$ and $x_1, ..., x_{|\mathcal{X}|}$ such that $P_X(x_1) \geq P_X(x_2) \geq ... \geq P_X(x_{|\mathcal{X}|})$*
Then define $C^*(x_i) = w_i$ (the i-th bitstring in shortlex order).
C^* is an optimal compressor i.e.,

$$\mathbb{E}(|C^*(X)|) \leq \mathbb{E}(C(X))$$

for any lossless compressor C. We have

$$H(X) - \log_2(1 + \lfloor \log_2 |X| \rfloor) \leq \mathbb{E}(|C^*(X)|) \leq H(X)$$

Proof. Let C be a lossless compressor, C is injective

$$\begin{aligned} |\{x \in \mathcal{X} : |C(x)| \leq k\}| &\leq \sum_{l=0}^{k} 2^l \\ &= |\{x \in \mathcal{X} : |C^*(x)| \leq k\}| \end{aligned}$$

as C^* uses all the possible strings of length k. Because in addition, these codewords (bitstrings length $\leq k$) are assigned to the $2^{k+1} - 1$ elements with largest probability, we have

$$\begin{aligned} \sum_{x \in \mathcal{X} : |C(X)| \leq k} P_X(x) &\leq \sum_{x \in \mathcal{X} : |C^*(x)| \leq k} P_X(x) \\ \mathbb{E}(|C^*(X)|) &= \sum_{k=0}^{\infty} \mathbb{P}(|C^*(X)| > k) \\ \Big(\underline{\text{Aside:}} \text{ Ex:} \quad \mathbb{E}X &= \sum_{n \geq 1} \mathbb{P}(Y \geq n)\Big) \\ &= \sum_{k=0}^{\infty} \sum_{x \in \mathcal{X} : |C^*(x)| > k} P_X(x) \\ &= \sum_{k=0}^{\infty} \left(1 - \sum_{x : |C*(x)| \leq k} P_X(x)\right) \\ &\leq \sum_{k=0}^{\infty} \left(1 - \sum_{x : |C(x)| \leq k} P_X(x)\right) \\ &= \mathbb{E}(|C(X)|) \end{aligned}$$

- To relate to entropy:
  Observe that $|C^*(x_i)| = \lfloor \log_2(i) \rfloor$
  Note also that $P_X(x_i) \leq 1 - \sum_{j=1}^{i-1} P_X(x_j) \leq 1 - (i-1)P_X(x_i)$
  We get $P_X(x_i) \leq \frac{1}{i}$

$$
\begin{aligned}
\mathbb{E}(|C^*(X)|) &= \sum_{i=1}^{|\mathcal{X}|} P_X(x_i)\lfloor \log(i) \rfloor \\
&\leq - \sum_{i=1}^{|\mathcal{X}|} P_X(x_i) \log\left(\frac{1}{i}\right) \\
&\leq - \sum_{i=1}^{|\mathcal{X}|} P_X(x_1) \log P_X(x_j) \\
&\leq H(X)
\end{aligned}
$$

- Lower bound
  Let $L = |C^*(X)| \in \{0, 1, ..., \lfloor \log |\mathcal{X}| \rfloor\}$

$$
\begin{aligned}
H(X, L) &= H(X) + \underbrace{H(L|X)}_{\substack{=\sum_x P_X(x) H(P_{L|X=x}) \\ =0}} \\
&= H(X) \\
\text{As a result: } H(X) = H(X, L) &= H(L) + H(X \mid L) \\
&\leq \log_2(1 + \lfloor \log |\mathcal{X}| \rfloor) + \sum_{k=0}^{\lfloor \log |\mathcal{X}| \rfloor} P_L(k) H(X|L=k) \\
&\leq \log_2(1 + \log_2 |\mathcal{X}|) + \underbrace{\sum_{k=0}^{\lfloor \log |\mathcal{X}| \rfloor} P_L(k).k}_{=\mathbb{E}L = \mathbb{E}(|C^*(X)|)}
\end{aligned}
$$

□

3.2.2 Uniquely decodable and prefix-free compression

Let $C : \mathcal{A} \to \{0,1\}^*$, we can naturally define its extension on $\mathcal{A}^* = \bigcup\limits_{n \leq 1} \mathcal{A}^n$ by $C^+(a_1...a_n) = C(a_1).C(a_2)...C(a_n)$.

For C^+ to be lossless, we need C to be lossless, but it is not sufficient in general. Let $\mathcal{A} = \{a, b, c\}$

$$
\begin{aligned}
C(a) &= 0 \\
C(b) &= 010 \\
C(c) &= 01 \\
C^+ b &= 010 \\
C^+(ca) &= 010
\end{aligned}
$$

Definition 7 (Uniquely decodable compressor). *A compressor C is uniquely decodable if its extension C^+ is injective.*

Definition 8 (Prefix-free compressor)**.** *C is a prefix-free compressor if no codeword is a prefix of any other.*

$Code = \{C(a) : a \in \mathcal{A}\}$

Ex $\mathcal{A} = \{a, b, c\}, C(a) = 0, C(b) = 10, C(c) = 110$

Propriety 5. *Prefix-free $\Rightarrow$ uniquely decodable.*

Proof. To decompose $C(a_1)...C(a_n)$, $C(a_1)$ is the unique prefix of $C(a_1)...C(a_n)$ which is a codeword. □

Remark C might be uniquely decodable without being prefix-free.

$$C(a) = 10$$
$$C(b) = 11$$
$$C(c) = 110$$

Uniquely decodable is more general than prefix-free, but not very useful, because there is a correspondence between prefix-free code and uniquely decodable code.

Theorem 2. *Let $A \in \mathcal{A}$ be a random variable. The Huffman algorithm computes $O(|\mathcal{A}| \log |\mathcal{A}|)$ a prefix-free compressor $C_H : \mathcal{A} \to \{0,1\}^*$ with minimum expected length $\mathbb{E}|C_H(A)|$ among all possible prefix-free compressor.*
Moreover,

$$\mathbb{E}(|C_H(A)|) < H(A) + 1$$

The key observation is to find a correspondence between prefix-free code and a binary tree.

Bitstrings labelling leaves form a prefix-free code. In this representation, the expected length is $\sum_{a\in\mathcal{A}} P_A(a).depth(C(A))$.

Proof. Huffman: in tutorial and HW □

Lemma 1 (Kraft's inequality)**.** • *For any prefix-free compressor C with codeword lengths $l_a = |C(a)|$, we have*

$$\sum_{a\in\mathcal{A}} 2^{-l_a} \leq 1 \tag{*}$$

- *Conversely, given a set of length $\{l_a\}$ satisfying (*), we can construct a prefix compressor with $|C(a)| = l_a$.*

Proof. $\Rightarrow$ In terms of binary tree T, $l_a = depth(\underbrace{C(A)}_{\text{leaf}})$. I have exactly $\sum_{a\in\mathcal{A}} 2^{l_{max}-l_a}$ nodes at level l_{max}, but at most $2^{l_{max}}$ nodes at depth l_{max}, so

$$\sum_{a\in\mathcal{A}} 2^{l_{max}-l_a} \leq 2^{l_{max}}$$
$$\sum_{a\in\mathcal{A}} 2^{-l_a} \leq 1$$

$\Leftarrow$ Let $\{l_a\}$ satisfy $\sum_a 2^{-l_a} \leq 1$. We order the elements of $\mathcal{A}, \{a_1, ..., a_n\}$ so that $l_{a1} \leq ... \leq l_{a_{|\mathcal{A}|}}$

$$C(a_i) = \text{binary expansion of length } l_{a_i} \text{ of } \sum_{j=1}^{i-1} 2^{-l_{a_i}} < 1$$
$$= 0.\underbrace{01...0}_{l_{a_i}}$$

Want to show that C is prefix-free. Consider $C(a_i) = b_i$ and $C(a_k) = b_k$ for $k > i$.

$$b_k - b_i = \sum_{j=i}^{k-1} 2^{-l_{a_j}} \geq 2^{-l_{a_i}}$$

Any codeword that has $C(a_i)$ as a prefix is a binary expression of a number that is at most

$$b_i + \sum_{p=l_{a_i}+1}^{l_{max}} 2^{-p} < b_i + 2^{-l_{a_i}}$$

So $C(a_i)$ cannot be a prefix of $C(a_k)$. So C is prefix-free. □

Using lemma, we can write the minimum expected length as the following optimization program:

$$\begin{aligned} OPT = \ & \text{minimize} \sum_{a \in \mathcal{A}} P_A(a) l_a \\ & \text{subject to } l_a \in \mathbb{N}_+ \\ & \qquad \sum_{A \in \mathcal{A}} 2^{-l_a} \leq 1 \end{aligned}$$

Proof. of the theorem.
$\mathbb{E}(|C_H(A)|) = OPT$. We start by proving that $H(A) \leq OPT$. For that, we relax the condition $l_a \in \mathbb{N}_+$ to $l_a \in \mathbb{R}$. We change variables: $Q(a) = 2^{-l_a}$.
The program becomes:

$$\begin{aligned} OPT = \ & \text{minimize} \sum_{a \in \mathcal{A}} P_A(a).(-\log_2 Q(a)) \\ & \text{subject to} \sum_{A \in \mathcal{A}} Q(a) \leq 1 \end{aligned}$$

Recall that $D(P||Q) = \sum_a P(a) \log_2 P(a) - \sum_a P(a) \log_2 Q(a)$. So the objective function can be written as:

$$-\sum_a P_A(a) \log_2 P_A(a) + \sum_a P_A(a) \log_2 P_A(a) - \sum_a P_A(a) \log_2 Q(a) = H(P_A) + D(P_A||Q)$$

To show that $D(P_A||Q) \geq 0$, we consider $Q'(a) = \frac{Q(a)}{\sum_{a'} Q(a')}$ be the normalized version of Q.

$$\begin{aligned} D(P_A||Q) &= \sum_a P_A(a) \log_2 P_A(a) - \sum_a P_A(a) \log_2 (Q'(a). \sum_{a'} Q(a')) \\ &= \sum_a P_A(a) \log_2 P_A(a) - \sum_a \Big(P_A(a) \log_2 Q'(a) \Big) - \log_2 (\sum_{a'} Q(a')) \\ &\geq D(P_A||Q') \\ &\geq 0 \text{ by propriety of the relative entropy} \end{aligned}$$

So the value for the relaxed program is exactly $H(P_A)$.

Now we want to show that $OPT < H(A) + 1$.

From the lower bound proof, we choose $l_a = -\log_2 P_A(a)$, but might not be an integer. Let's choose $l_a = \lceil -\log_2 P_A(a) \rceil$. We have $\sum_a 2^{-l_a} \leq \sum_a P_A(a) = 1$, and the objective function has thus the value

$$\sum_a P_A(a).l_a = \sum_a P_A(a) \lceil -\log_2 P_A(a) \rceil < \sum_a P_A(a) \big(-\log_2 P_A(a) + 1 \big) = H(P_A) + 1$$

□

Propriety 6. *For any uniquely decodable compressor C with codeword lengths $\{l_a\}_{a \in \mathcal{A}}$, we have*

$$\sum_{a \in \mathcal{A}} 2^{-l_a} \leq 1$$

Proof. For a string $a^n = a_1 ... a_n$, define

$$C(a^n) = C(a_1) ... C(a_n)$$

and length

$$l_{a_n} = l_{a_1} + ... + l_{a_n}$$

$$\begin{aligned} \Big(\sum_{a \in \mathcal{A}} 2^{-l_a} \Big)^n &= \sum_{a \in \mathcal{A}} 2^{-l_{a^n}} \\ &= \sum_{m=1}^{n.l_{max}} N_m . 2^{-m} \\ \text{where } N_m &= |\{a^n \in \mathcal{A}^n : L_{a^n} = m\}| \end{aligned}$$

By unique decodability, $N_m \leq 2^m \leq n.l_{max}$.

So for all $n \geq 1$,

$$\sum_{a \in \mathcal{A}} 2^{-l_a} \leq \underbrace{(n.l_{max})^{1/n}}_{\xrightarrow[n \to \infty]{} 1}$$

□

3.3 Fixed-length almost lossless compression

Definition 9. *A fixed-length compressor for some $x \in \mathcal{X}$ of length l is a function $C : \mathcal{X} \to \{0,1\}^l$. It has an error probability $\leq \delta$ if there exists $D : \{0,1\}^l \to \mathcal{X}$ such that $\mathbb{P}(D \circ C(X) = X) \geq 1 - \delta$.*

We would like to determine

Definition 10.

$$l^{OPT}(X, \delta) = \min\{l : \textit{there is a length } l \textit{ compressor with error probability} \leq \delta\}$$

One natural encoding strategy is:

Sort elements in $\mathcal{X} = \{x_1, x_2, ..., x_{|\mathcal{X}|}\}$
so that $P_X(x_1) \geq P_X(x_2) \geq ... \geq P_X(x_{|\mathcal{X}|})$
$S^*_\delta = \emptyset$
for *i=1 to* $|\mathcal{X}|$ **do**
  $S^*_\delta \leftarrow S^*_\delta \cup \{x_i\}$
  if $\sum_{x \in S^*_\delta} P_X(x) \geq 1 - \delta$ **then**
    stop

Theorem 3.

$$l^{OPT} = \lceil \log_2 |S^*_\delta| \rceil$$

Proof. • Start with "$\leq$"
$S^*_\delta = \{x_1, ..., x_k\}$ for some k.
$l = \lceil \log_2 |S^*_\delta| \rceil$, we have $k = |S^*_\delta| \leq 2^e$.

$$C(x_i) = \begin{cases} \underbrace{bin^l(i-1)}_{\substack{\text{binary representation} \\ \text{of } i-1 \text{ with } l \text{ bits}}} & \text{if } i \leq k \\ 0^l & \text{if } i > k \end{cases}$$

Define $D(y) = x_i$ if $i - 1$ is the number in $\{0, 1, ..., 2^l - 1\}$ with binary representation y (y is a bitstring of length l).

$$\begin{aligned} \mathbb{P}(D \circ C(X) = X) &\geq \sum_{i=1}^{l} P_X(x_i) \\ &\geq 1 - \delta \end{aligned}$$

• Ineq "$\geq$"
Let C be a compressor with length l and error probability $\leq \delta$.
Define $S = \{x \in \mathcal{X} : D(C(X)) = x\}$
•$S \subset D(\{0,1\}^l)$ so $|S| \leq 2^l$
$\sum_{x \in S} P_X(x) = \mathbb{P}(D \circ C(X) = X) = 1 - \delta$
So $|S^*_\delta| \leq |S| \leq 2^l$ and so $\log |S^*_\delta| \leq l$ using the fact that S^*_δ is the smallest set with probability $\geq 1 - \delta$. □

Remark $\lceil \log_2 |S^*_\delta| \rceil$ can be very different from $H(X)$.
As an example: $\mathcal{X} = \{0, 1, ..., m\}$

$$X = \begin{cases} 0 & \text{with prob } 1 - \epsilon \\ i & \text{with prob } \frac{\epsilon}{m} \end{cases}$$

For $\delta = 0$, $\log_2 |S^*_\delta| = \log_2(m+1)$
But

$$\begin{aligned} H(X) &= -(1-\epsilon)\log_2(1-\epsilon) - \sum_{i=1}^{m} \frac{\epsilon}{m} \log_2 \frac{\epsilon}{m} \\ &= \underbrace{-(1-\epsilon)\log_2(1-\epsilon) - \epsilon \log_2 \epsilon}_{h_2(\epsilon)} + \epsilon \log_2 m \end{aligned}$$

But $\log_2 |S_\delta^*|$ is an entropic quantity in itself.
Define

$$H_0(X) = \log_2(|\sup P_X|) \qquad \text{(Hartley entropy)}$$

Note that H_0 has shared some proprieties with H:

- $H_0 = 0$ iff $X = x_0$ wp 1
- $H_0 = \log |\mathcal{X}|$ if X is uniform
- $H_0(X) \geq H(X)$ (Ex)

If allow "error probability" δ, we define *another* version

$$H_0^\delta = \min_{\sum_{x \in s_\delta} P_X(x) \geq 1-\delta} \log_2 |S_\delta|$$

Smoothing can have a strange effect on entropy. For X defined before,

$$H_0(X) = \log(1+m) \qquad H_0^\epsilon = 0$$

We also saw in homework an example: $X_i \hookrightarrow \mathcal{B}(p) \quad X^n = X_1...X_n \in \{0,1\}^n : H_0(X^n) = n$, but $H_0^\delta(X^n) \leq n \log p$
Recall that

$$H(X) = \mathbb{E}(\underbrace{h_X(X)}_{surprisal})$$

$$\text{with } h_X(X) = -\log_2(P_X(X))$$

Examples of $h_X(X)$

- $X \hookrightarrow \mathcal{U}(\mathcal{X})$
  $h_X(X) = \log |\mathcal{X}|$ with probability 1,
  In particular $H(X) = \mathbb{E}(\mathbb{h}_{\mathbb{X}}(\mathbb{X})) = \log |\mathcal{X}|$
- $\mathcal{X} = \{1, 2, ..., 2t\}$

$$P_X(X) = \begin{cases} \frac{3}{4t} & \text{for } X \in \{1, ..., t\} \\ \frac{1}{4t} & \text{for } X \in \{t+1, ..., 2t\} \end{cases}$$

$$\mathbb{P}(-\log_2 P_X(X) = \log_2 \frac{4t}{3}) = \sum_{\log \frac{1}{P_X(x)} = \log \frac{4t}{3} \Leftrightarrow P_X(x) = \frac{3}{4t}} P_X(x) = \frac{3}{4}$$

$$\mathbb{P}(-\log_2 P_X(X) = \log_2 4t) = \sum_{x : P_X(x) = \frac{1}{4t}} P_X(x) = \frac{1}{4}$$

$$\begin{aligned} H(X) &= t\frac{3}{4t} \log_2 \frac{4t}{3} + t\frac{1}{4t} \log 4t \\ &= \frac{3}{4} \log \frac{4}{3} t + \frac{1}{4} \log 4t \end{aligned}$$

Propriety 7.

$$l^{OPT}(X, \delta) \leq \min\{l \in \mathbb{N}_+ : \mathbb{P}(h_X(X) > l) \leq \delta\}$$

"achievability"*: There is a compressor with length l and error probability $\leq \mathbb{P}(h_X(X) > l)$*
Moreover, for any $\tau > 0$

$$l^{OPT}(X, \delta) \geq \min\{l \in \mathbb{N}_+ : \mathbb{P}(h_X(X) > l + \tau) - 2^{-\tau} \leq \delta\}$$

"converse"*: For any compressor and any $\tau > 0$, the probability of error is at least $P(h_X(X) > l + \tau) - 2^{-\tau}$*

Proof. • Let l satisfy $\mathbb{P}(h_X(X) > l) \leq \delta$. Take $S = \{x \in \mathcal{X} : P_X(x) \geq 2^{-l})\}$.
Note that $|S| \leq 2^l$
Moreover

$$\begin{aligned}
\mathbb{P}(X \in S) &= \mathbb{P}(P_X(X) \geq 2^{-l}) \\
&= \mathbb{P}(-\log_2 P_X(X) \leq l) \\
&= 1 - \mathbb{P}(h_X(X) > l) \\
&\geq 1 - \delta
\end{aligned}$$

• Converse

Given C with length l and error probability $\leq \delta$.

$$\begin{aligned}
S &= \{x \in \mathcal{X} : D(C(X)) = x\} \\
S &\subset D(\{0,1\}^l) \text{ so } |S| \leq 2^l \\
\text{We have } \mathbb{P}(X \in S) &= \mathbb{P}(D(C(X)) = X) \\
&\geq 1 - \delta
\end{aligned}$$

$$\begin{aligned}
1 - \delta &\leq \sum_{x \in S} P_X(x) \\
&= \sum_{x \in S : \underbrace{P_X(x) \geq 2^{-l-\tau}}_{-\log_2 P_X(x) \leq l+\tau}} P_X(x) + \sum_{x \in S : P_X(x) < 2^{-l-\tau}} P_X(x) \\
&\leq \sum_{x : -\log_2 P_X(x) \leq l+\tau} P_X(x) + 2^l . 2^{-l-\tau} \\
&= \underbrace{\mathbb{P}(h_X(X) \leq l + \tau)}_{=1-\mathbb{P}(h_X(X) > l+\tau)} + 2^{-\tau}
\end{aligned}$$

So l satisfies

$$\mathbb{P}(h_X(X) > l + \tau) - 2^{-\tau} \leq \delta$$

□

Important special case

$$\begin{aligned}
&X^n = X_1 X_2 ... X_n \\
&\quad \text{with } X_i \text{ independent and identically distributed} \\
&\quad \text{with same distribution as } X
\end{aligned}$$

Theorem 4 (Shanon's source coding theorem)**.** *For any* $\delta \in (0,1), 0 < \delta < 1$

$$\lim_{n \to \infty} \frac{l^{OPT}(X^n, \delta)}{n} = H(X)$$

Proof. Need to get a handle on $\mathbb{P}(h_{X^n} > l)$

$$
\begin{aligned}
h_{X^n}(X^n) = -\log_2 P_{X^n}(X^n) &= -\log_2 P_X(X_1)P_X(X_2)...P_X(X_n)\\
&= \sum_{i=1}^{n} \log_2 P_X(X_i)\\
\mathbb{E}\big(h_{X^n}\big) &= n\mathbb{E}\big(h_X(X_i)\big)\\
&= nH(X)\\
\mathbb{P}(|h_{X^n}(X^n) - nH(X)| \geq t) &\leq \frac{\mathbb{V}\big(h_{X^n}(X^n)\big)}{t^2}\\
\mathbb{V}(h_{X^n}(X^n)) &= \mathbb{V}\left(-\sum_{i=1}^{n} \log_2 P_X(X_i)\right)\\
&= n\mathbb{V}(h_X(X))\\
\text{So } \mathbb{P}(|h_{X^n}(X^n) - nH(X)| \geq t) &= \frac{n\mathbb{V}(h_X(X))}{t^2}
\end{aligned}
$$

Set $t = \sqrt{\frac{n\mathbb{V}(h_X(X))}{\delta}}$, we get:

$$
\begin{aligned}
\mathbb{P}(h_{X^n}(X^n)) > nH(X) + \sqrt{\frac{n\mathbb{V}(h_X(X))}{\delta}} &\leq \delta\\
\text{So } l^{OPT}(X^n, \delta) \leq nH(X) + \sqrt{\frac{n\mathbb{V}(h_X(X))}{\delta}}&\\
\text{So } \lim_{n\to\infty} \frac{l^{OPT}(X^n, \delta)}{n} \leq H(X)&
\end{aligned}
$$

For the lower bound, let $\alpha > 0$ take $t = \sqrt{\frac{n\mathbb{V}(h_X(X))}{\alpha}}$

$$
\mathbb{P}(h_{X^n}(X^n)) \leq \underbrace{nH(X) - \sqrt{\frac{n\mathbb{V}(h_X(X))}{\alpha}}}_{l+\tau} \leq \alpha
$$

Now take $l = nH(X) - 2\sqrt{\frac{n\mathbb{V}(h_X(X))}{\alpha}}$ and $\tau = \sqrt{\frac{n\mathbb{V}(h_X(X))}{\alpha}}$.
Choose α small enough such that $1 - \delta > \alpha + 2^{-\tau}$

$$
\begin{aligned}
\mathbb{P}(h_{X^n}(X^n) > l + \tau) \leq 1 - \alpha > \delta + 2^{-\tau}&\\
\text{So } l^{OPT}(X, \delta) \geq l = nH(X) - 2\sqrt{\frac{n\mathbb{V}(h_X(X))}{\alpha}}&
\end{aligned}
$$

$\square$

In tutorial, we showed that S, chosen here as $\{x \in \mathcal{X} : P_X(X) \geq 2^{-l}\}$ can be picked at random, but gives therefore a good code (the bound is almost the same). In practice:

- Stream of symbols
- But not independent (ex: informati$\bullet\bullet\rightarrow$ information) so the rest of the word depends on the context
  Hoffman with larger blocks may be inefficient
- Do not even know usually the distribution $\rightarrow$ universal compressor

3.4 Universal compression

Consider a stream X^n with n symbols in $\mathcal{X}$. We do not have access to P_{X^n}.

3.4.1 Arithmetic code

Idea: learn a model for data.

Ex $P_1(a) = \frac{1}{|\mathcal{X}|}$. Then, when we see $x_1, ..., x_{i-1}$,

$$P_i(a|x_1, ..., x_{i-1}) = \frac{1 + |\{j \in \{1, ..., i-1\}, x_j = a\}|}{|\mathcal{X}| + (i-1)}$$

Remark Simple to compute, only need to keep $|\mathcal{X}|$ counters.

It is useful to interpretate a bitstring as an interval in $[0, 1]$

$$\begin{aligned} 01 &\mapsto [0.01; 0.1] \\ y &\mapsto [0.y; 0.y + 0.\underbrace{0...01}_{|y|}] \end{aligned}$$

Idea encode stream as intervals. Each new symbol: choose a subinterval of current interval with length proportional to the probability given by model.

$\mathcal{X} = \{a_1, a_2, ..., a_n\}$

Algorithm For a new symbol $x_i = a_k$, chose subinterval given by

$$\left[\underbrace{\sum_{p=1}^{k-1} P_i(a_p|x_1...x_{i-1})}_{\alpha}, \underbrace{\sum_{p=1}^{k} P_i(a_p|x_1...x_{i-1})}_{\beta}\right]$$

In absolute terms: if current interval is $[u_{i-1}, v_{i-1}]$:

$$\begin{aligned} u_i &= u_{i-1} + (v_{i-1} - u_{i-1})\alpha \\ v_i &= u_{i-1} + (v_{i-1} - u_{i-1})\beta \end{aligned}$$

Problem From interval to bitstrings?

Solution Find largest dyadic interval included in it.

Overall,

$$x_1, ..., x_n \to I_{x_1,...,x_n} \to \text{Find } \underbrace{I_y}_{\text{dyadic interval}} \subset I_{x_1,...,x_n} \to \text{output } \underbrace{y}_{\text{bitstring}}$$

Remark Decoding is easy if agree on model P_i.

3.4.2 Lempel-Ziv coding

No probabilistic model, based on dictionary of words that appeared. Read sequence into words:

$$m_0, m_1, ..., m_L$$

- $m_0 = \emptyset$
- $m_i = m_j.x$ for some $x \in \mathcal{X}$
- m_i are distinct for $0 \le i < L$
- $m_L = m_j$ for some $j < L$

$$\underbrace{\quad}_{m_0}\underbrace{a}_{\substack{m_1\\ j=0\\ x=a}}\underbrace{b}_{\substack{m_2\\ j=0\\ x=b}}\underbrace{aa}_{\substack{m_3\\ j=1\\ x=a}}\underbrace{aaa}_{\substack{m_4\\ j=3\\ x=a}}\underbrace{b}_{\substack{m_1\\ j=2}}$$

Each word encoded in a pair (pointer to j, additional letter x). Encoding word m_i cost $\lceil \log i \rceil + 1$.

Lossy compression *Will not talk about it.*
For images, audio, we don't need exact recovery $\to$ rate *distortion*.

4 Noisy channel coding

4.1 Setting

Channel

- Input alphabet $\mathcal{X}$
- Output alphabet $\mathcal{Y}$

$$W_{Y|X}(y|x) = \text{ probability of outputting } y \text{ when the input is } x$$

Our task Find E, D to send messages with small error probability.

$$s \in \{1, ..., M\} \;—\boxed{E}—\quad x \quad —\boxed{W}—\quad y \quad —\boxed{D}—\; \hat{s} \in \{1, ..., M\}$$

Definition 11. *An M-code for W is a pair of functions*

$$E : [M] \to \mathcal{X}$$
$$D : \mathcal{Y} \to [M]$$

- $E(s)$ is called codeword and $\{E(1), ..., E(M)\}$ is codebook.
- Decoding region $D_S = D^{-1}(\{s\}) = \{y : D(y) = s\}$

We will talk about random variables

$$\underbrace{S}_{\substack{\text{original}\\ \text{message}}},\ \underbrace{X}_{\substack{\text{encoded}\\ \text{message}\\ \text{or}\\ \text{channel input}}},\ \underbrace{Y}_{\substack{\text{channel}\\ \text{output}}},\ \underbrace{\hat{S}}_{\substack{\text{decoded}\\ \text{message}}}$$

For the rest of the section, we assume that the distribution on messages is uniform, i.e.:

$$P_{SXY\hat{S}}(s,x,y,\hat{s}) = \frac{1}{M}\mathbb{1}_{x=E(s)}.W(y|x).\mathbb{1}_{\hat{s}=D(y)}$$

Definition 12 (Error Probability)**.** *In that case, we define the error probability as follow:*

$$\begin{aligned} P_{err} &= \mathbb{P}(S \neq \hat{S}) \\ &= 1 - \frac{1}{M}\sum_{s=1}^{M}\sum_{y\in\mathcal{Y}} W(y|E(s)).\mathbb{1}_{D(y)=s} \end{aligned}$$

Note P_{err} was called *block error probability* in the first lectures, this is the *average* error probability.

Two others

- $P_{err,max} = \max_{s\in[M]} \mathbb{P}(\hat{S} \neq s|S=s)$
  Maximum error probability
- If $M = 2^k$, we can see $[M] = \{0,1\}^k$, and thus define $P_{bit} = \frac{1}{k}\sum_{i=1}^{k}\mathbb{P}(S_i \neq \hat{S}_i)$

Question Trade-off between M and P_{err}.

Definition 13.

$$M^{OPT} = \max\{M : \textit{there is an } M\textit{-code with } P_{err} \leq \delta\}$$

Remark $\log_2 M^{OPT}(W,\delta) =$ number of bit used

Important thing to keep in mind Let W^n be n independent copies of W.

$$W^n(y_1...y_n|x_1...x_n) = W(y_1|x_1)...W(y_n|x_n)$$

With $n\to\infty$ and want $\delta \to 0$. So we are interested in $\frac{\log M^{OPT}(W^n,\delta)}{n}$, which is the number of bits that can be send per channel use.

Examples

1. $\mathcal{X} = \{1,...,N\}$
   $\mathcal{Y} = \{1,...,N\}$
   $W(y|x) = \mathbb{1}_{x=y}$
   - If $M \leq N$, $E = id, D = id$ so $P_{err} = 0$.

   If $M > N$, take E, D and M-code for $\mathbb{N}$.

$$\begin{aligned} P_{err} &= 1 - \frac{1}{M}\sum_{s=1}^{M}\sum_{y\in\mathcal{Y}} W(y|E(s)).\mathbb{1}_{D(y)=s} \\ &= 1 - \frac{1}{M}\sum_{y\in\mathcal{Y}}\underbrace{\sum_{s=1}^{M} W(y|E(s)).\mathbb{1}_{D(y)=s}}_{\underbrace{W(y|E(D(Y)))}_{\leq 1}} \\ &\geq 1 - \frac{|y|}{M} = 1 - \frac{N}{M} \end{aligned}$$

2. Binary symmetric channel. $f < 1/2$
   $BSC_f^{\times n}$: n independent copies of BSC_f.

   Repetition code Message set $= \{0,1\}^{n/3}$

   Encoder: $E(s_1 ... s_{\frac{n}{3}}) = s_1 s_1 s_1 ... s_{\frac{n}{3}} s_{\frac{n}{3}} s_{\frac{n}{3}}$

   Decoder: $D(y_1 ... y_n) = maj(y_1 y_2 y_3) maj(y_4 y_5 y_6) ... maj(y_{n-2} y_{n-1} y_n)$
   $P_{bit} = 3f^2 - 2f^3, P_{err} = 1 - (1 - 3f^2 + 2f^3)^{n/3}$

Definition 14. *The information capacity of a channel W with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$ is*

$$C(W) = \max_{\substack{P_X \\ \text{distribution over } \mathcal{X}}} I(X:Y) \qquad \text{with } P_{XY}(x,y) = P_X(x) W_{Y|X}(y|x)$$

Ex

1. $W(y|x) = \mathbb{1}_{y=x} \qquad \mathcal{X} = \{1, ..., N\}, \mathcal{Y} = \{1, ..., N\}$. For any P_X:
$$\begin{aligned} I(X:Y) &= H(X) - \underbrace{H(X|Y)}_{=0} \\ &= H(X) \end{aligned}$$
   So $C(W) = \max_{P_X} I(X:Y) = \max_{P_X} H(X) = \log N$
2. $P_X = (1/3, 1/3, 1/3)$
   $I(X:Y) = \underbrace{H(X)}_{\log 3} - H(X|Y)$
   $H(X|Y) = ?$
$$P_{X|Y=0}(x) = \frac{P_X(x) W(0|x)}{1/2} = \begin{cases} 2/3 & \text{if } x = a \\ 1/3 & \text{if } x = b \end{cases}$$
   $H(X|Y) = h_2(1/3)$
   $I(X:Y) = 2/3$
   A better distribution is: $P_X = (1/2, 0, 1/2)$
   $I(X:Y) = \underbrace{H(X)}_{=1} - \underbrace{H(X|Y)}_{=0} = 1$
   We cannot do better:
   $I(X:Y) \leq H(Y) \leq \log |\mathcal{Y}| = 1$
   $C(W) = 1$
3. BSC_f $P_X(0) = 1 - p$ and $P_X(1) = p$
$$\begin{aligned} I(X:Y) &= H(Y) - H(Y|X) \\ H(Y|X) &= P_X(0) H(Y)_{P_{Y|X=0}} \\ &\quad + P_X(1) H(Y)_{P_{Y|X=0}} \\ &= h_2(f) \end{aligned}$$

$$Y = \begin{cases} 0 & \text{wp } (1-p)(1-f) + pf \\ 1 & \text{wp } p(1-f) + (1-p)f \end{cases}$$
   $H(Y) = h_2\left((1-p)(1-f) + pf\right)$
   $I(X:Y) = h_2\left((1-p)(1-f) + pf\right) - h_2(f)$
   This is maximized for $p = 1/2$. So $C(BSC_F) = 1 - h_2(f)$

4.2 Converse bounds

Theorem 5. *Any M-code for W satisfies*

$$\log M \leq \frac{C(W) + h_2(P_{err})}{1 - P_{err}}$$

Proof. We start with the case $P_{err} = 0$. Take an M-code.

$$P_{SXY\hat{S}}$$

$$\begin{aligned}
\log M = H(S) &= I(S : \hat{S}) + \underbrace{H(S|\hat{S})}_{=0 \text{ as } P_{err}=0} \\
&= I(S : \hat{S}) \\
S \to X \to Y &\to \hat{S} \text{ is a markov chain} \\
I(S : \hat{S}) &\leq I(X : Y)
\end{aligned}$$

$$\text{(See data processing inequality in tutorial: } X \to Y \to Z \text{ Markov chain } I(X : Y) \geq I(X : Z))$$

$$\log M \leq I(X : Y) \leq C(W)$$

Now general P_{err}:

Lemma 2 (Fano's inequality)**.** *If $S \in \{1, ..., M\}$ and $\hat{S}$ such that $\mathbb{P}(S \neq \hat{S}) \leq \epsilon$, then*

$$H(S|\hat{S}) \leq h_2(\epsilon) + \epsilon \log M$$

Proof. Introduce

$$E = \begin{cases} 1 & \text{if } S = \hat{S} \\ 0 & \text{if } S \neq \hat{S} \end{cases}$$

$$\begin{aligned}
H(S|\hat{S}) &= H(E, S|\hat{S}) - \underbrace{H(E|S\hat{S})}_{=0} \\
&= H(E|\hat{S}) + H(S|E\hat{S}) \\
H(E|\hat{S}) &\leq H(E) \leq h_2(\epsilon) \qquad \text{assuming } \epsilon \leq 1/2 \\
H(S|E\hat{S}) &= \underbrace{P_E(0)}_{\leq \epsilon} \underbrace{H(S|\hat{S})_{P_{S\hat{S}|E=0}}}_{\leq \log M} \\
&+ P_E(1) \underbrace{H(S|\hat{S})_{P_{S\hat{S}|E=0}}}_{\leq \log M} \\
&\leq \log M
\end{aligned}$$

□

Back to the theorem:

$$\begin{aligned}\log M &\leq I(X:Y) + H(S|\hat{S}) \\ &\leq \underbrace{I(X:Y)}_{\leq C(W)} + h_2(P_{err}) + P_{err}\log M \\ \log M &\leq \frac{C(W) + h_2(P_{err})}{1 - P_{err}}\end{aligned}$$

$\square$

Example

1. Identity channel on $\{1, ..., N\}$. For any M-code:

$$\log M \leq \frac{\log N + h_2(P_{err})}{1 - P_{err}}$$

   We have already seen that

$$\begin{aligned}P_{err} \geq 1 - \frac{N}{M} \Rightarrow \frac{M}{N} &\leq \frac{1}{1 - P_{err}} \\ \Rightarrow \log M &\leq \underbrace{\log N + \log\left(\frac{1}{1 - P_{err}}\right)}_{\substack{\text{This was better} \\ \text{but very specific}}}\end{aligned}$$

2. Binary symmetric channel $BSC_f^{\times n}$

$$\log M \leq \frac{C(BSC_f^{\times n}) + h_2(P_{err})}{1 - P_{err}}$$

   We know that

$$\begin{aligned}C(BSC_f) &= 1 - h_2(f) \\ C(BSC_p^{\times n}) &= \max_{P_{X^n}} I(X^n : Y^n)\end{aligned}$$

   Easy to find a lower bound:

$$\begin{aligned}P_{X_1 \ldots X_n} &= P_{X_1} \times ... \times P_{X_n} \\ C(BSC_f^{\times n}) &\geq I(X^n : Y^n) \\ \text{But there } X_i, Y_i & \text{ are mutually independant} \\ &= \sum_{i=1}^{n} I(X_i : Y_i) \\ \text{If take } P_{X_i} = \text{unif, then } & I(X_i : Y_i) = C(BSC_f) \\ &= nC(BSC_f)\end{aligned}$$

Theorem 6. *Given two channels*

$$W^1_{Y_1|X_1} \qquad W^2_{Y_2|X_2}$$

Define:

$$W^{12}_{Y_1Y_2|X_1X_2}(y_1y_2|x_1x_2) = W^1_{Y_1|X_1}(y_1|x_1).W^1_{Y_2|X_2}(y_2|x_2)$$

Then:

$$C(W^{12}) = C(W^1) + C(W^2)$$

Proof. • Easy direction: $C(W^{12}) \geq C(W^1) + C(W^2)$. Choose $P_{X_1X_2} = P_{X_1} \times P_{X_2}$
Then $P_{X_1X_2Y_1Y_2} = P_{X_1Y_1} \times P_{X_2Y_2}$ using the definition of W^2. So

$$I(X_1X_2 : Y_1Y_2) = \underbrace{I(X_1 : Y_1)}_{\substack{\text{mutual information between input} \\ \text{and output of } W^1}} + I(X_2 : Y_2)$$

By taking the sup over P_{X_1} and P_{X_2}

$$C(W^{12}) \geq C(W^1) + C(W^2)$$

- More difficult direction: $\leq$ Take a general $P_{X_1X_2}$, X_1 and X_2 *not* independent.

$$\begin{aligned}
I(X_1X_2 : Y_1Y_2) &= H(Y_1Y_2) - H(Y_1Y_2|X_1X_2) \\
&\leq H(Y_1) + H(Y_2) - \sum_{x_1x_2} P_{X_1X_2}(x_1x_2)H(Y_1Y_2)_{P_{Y_1Y_2|X_1X_2=x_1x_2}} \\
P_{X_1X_2}(y_1y_2) &= W^1(y_1|x_1)W^2(y_2|x_2) \quad \text{by definition of } W^{12} \\
H(Y_1Y_2)_{P_{Y_1Y_2|X_1X_2=x_1x_2}} &= H(Y_1)_{P_{Y_1|X_1X_2=x_1x_2}} + H(Y_2)_{P_{Y_2|X_1X_2=x_1x_2}} \\
&= H(Y_1)_{P_{Y_1|X_1=x_1}} + H(Y_2)_{P_{Y_2|X_2=x_2}}
\end{aligned}$$

We get:

$$\begin{aligned}
I(X_1X_2 : Y_1Y_2) &\leq H(X_1) + H(Y_1|X_1) + H(Y_2) - H(Y_2|X_2) \\
&= I(X_1 : Y_1) + I(X_2 : Y_2) \leq C(W^1) + C(W^2)
\end{aligned}$$

□

4.3 Achievability bound

We had define $h_X(X) = \log_2 \frac{1}{P_X(X)}$ with $\mathbb{E}h_X(X) = H(X)$.

Definition 15. *For $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, we define the mutual information density:*

$$\text{for } x, y \in \mathcal{X} \times \mathcal{Y}\text{: } i_{XY}(x : y) = \log_2 \frac{P_{X|Y}(X|Y)}{P_Y(Y)} = \log_2 \left(\frac{P_{XY}(x, y)}{P_X(x)P_Y(y)} \right)$$

If $P_{XY}(x, y) = 0$ but $P_X(x) > 0, P_Y(y) > 0$, let $i_{XY}(x : y) = -\infty$; if $P_X(x) = 0$ or $P_Y(y) = 0$, then $i_{XY}(x : y) = +\infty$

Observe that

$$\begin{aligned}
\mathop{\mathbb{E}}_{(X,Y)\sim P_{XY}}(i_{XY}(X : Y)) &= \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_{XY}(x, y)i_{XY}(x : y) \\
&= I(X : Y)
\end{aligned}$$

Example $X^n = X_1X_2...X_n \in \{0, 1\}^n$ uniform; $Y^n = Y_1...Y_n \in \{0, 1\}^n$ where
$Y_i = \begin{cases} X_i & \text{wp } 1 - f \\ 1 \oplus X_i & \text{wp } f \end{cases}$. We assume $f < 1/2$.

$$
\begin{aligned}
i_{X^nY^n}(x^ny^n) &= \log_2 \frac{P_{Y^n|X^n}(y^n|x^n)}{P_{Y^n}y^n} \\
&= \log_2 \frac{\prod_{i=1}^{f}(1-f)^{x_i\oplus y_i \oplus 1} f^{x_i \oplus y_i}}{2^{-n}} \\
&= n + \log_2 (1-f)^{n-d_H(x^n,y^n} f^{d_H(x^n,y^n)} \qquad \text{with } d_H(x^n,y^n) = |\{i \in [n] : x_i \neq y_i\}| \\
&= n + (n - d_H(x^n,y^n))\log_2(1-f) + d_H(x^n,y^n)\log_2 f \\
i_{X^nY^n}(X^n : Y^n) &= n + (n - d_H(X^n : Y^n))\log_2(1-f) + d_H(X^n : Y^n)\log_2 f \\
\mathbb{E}(i_{X^nY^n}(X^n : Y^n) &= n(1-h_2(f))
\end{aligned}
$$

Theorem 7. *Let W be a channel input $\mathcal{X}$ and output $\mathcal{Y}$. For any P_X on $\mathcal{X}$, define $P_{XY}(xy) = P_X(x)W(y|x)$ and any $\tau > 0$, there exists an M-code with*

$$
P_{err} \leq \mathbb{P}\big(i_{XY}(X:Y) < \log M + \tau\big) + 2^{-\tau}
$$

This means that we can send $\log M$ bits with error probability lower than δ provided $\mathbb{P}(i_{XY}(X : Y) \leq \log M) \lessapprox \delta$

Proof. We need to construct a (E, D).

$$
\begin{aligned}
P_{err} &= 1 - \frac{1}{M}\sum_{s=1}^{M}\sum_{y\in\mathcal{Y}} W(y|E(s))\mathbb{1}_{D(y)=s} \\
&= \frac{1}{M}\sum_{y\in\mathcal{Y}} W\big(y|E(D(y))\big)
\end{aligned}
$$

If we choose $D^*(y) = \operatorname*{argmax}_{s\in[M]} W(y|E(s))$, then for this D^*,

$$
P_{err} = 1 - \frac{1}{M}\sum_{y\in\mathcal{Y}} \max_{s\in[M]} W(y|E(s))
$$

This is optimal but not so easy to analyse. Instead we define a threshold and let $D(y)$ be the only s above the threshold.

Recall that we had a distribution P_X over $\mathcal{X}$. Define $P_{XY}(x,y) = P_X(x)W(y|x)$ and $P_Y(y) = \sum_x P_X(x)W(y|x)$. The threshold will be, if $W(y|E(s)) \lessapprox MP_Y(y)$

$$
D(y) = \begin{cases} s & \text{if there is a unique } s \text{ such that } i_{XY}(E(s):y) \geq \log M + \tau \\ x_0 & \text{otherwise} \end{cases}
$$

For this D, we analyse the eror probability $P_{err} = \frac{1}{M}\sum_{s=1} M \underbrace{P_{err,s}}_{\substack{\text{error prob} \\ \text{for msg } s}}$

$$
\begin{aligned}
\underbrace{P_{err,s}}_{=\mathbb{P}(\hat(S)\neq S|S=s)} &= \sum_{y\in\mathcal{Y}} W(y|E(s))\mathbb{1}_{D(y)\neq s} \\
&\leq \sum_{y\in\mathcal{Y}} W(y|E(s))\mathbb{1}_{(i_{XY}(E(s):y)<\log M+\tau) \text{ or } (\exists s'\neq s: i_{XY}(E(s'):y)\geq \log M+\tau)} \\
&\leq \underbrace{\sum_{y\in\mathcal{Y}} W(y|E(s))\mathbb{1}_{i_{XY}(E(s):y)<\log M+\tau}}_{\mathbb{P}_{Y\sim W(\cdot|E(s))}(i_{XY}(E(s):Y)<\log M+\tau|S=s)} + \sum_{s'\neq s}\underbrace{\sum_{y\in\mathcal{Y}} W(y|E(s))\mathbb{1}_{i_{XY}(E(s):y)\geq \log M+\tau}}_{\mathbb{P}_{Y\sim W(\cdot|E(s))}(i_{XY}(E(s'):y)\geq\log M+\tau)}
\end{aligned}
$$

Aside on $BSC_f \quad f < 1/2$
In this case $i_{X^nY^n}(E(s) : Y^n) = n + (n - d_H(E(s), Y^n)) \log(1-f) + d_H(E(s), Y^n) \log_2 f$. The optimal decoder is

$$D^*(y^n) = \underset{s \in [M]}{d_H}(E(s), y^n$$

And the "threshold" decoder is

$$D(y^n) = \begin{cases} s & \text{if there is a unique } s \text{ s.t. } D_H(E(s), y) \leq \text{---} \\ * & \text{otherwise} \end{cases}$$

Now we choose E. $E(1), ..., E(M)$ random according to P_X and independent, and we compute the expectation of the error probability.

1.

$$\begin{aligned} \mathop{\mathbb{E}}_{E(s) \sim P_X} \Big(\sum_{y \in \mathcal{Y}} W(y|E(s)) \mathbb{1}_{i_{XY}(E(s):y) < \log M + \tau} \Big) &= \sum_{x \in \mathcal{X}} P_X(x) \sum_{y \in \mathcal{Y}} W(y|x) \mathbb{1}_{i_{XY}(x:y) < \log M + \tau} \\ &= \mathbb{P}(i_{XY}(x : y) < \log M + \tau) \end{aligned}$$

2.

$$\begin{aligned} \mathop{\mathbb{E}}_{\substack{E(s) \sim P_X \\ E(s') \sim P_y}} \Big(\sum_{y \in \mathcal{Y}} W(y|E(s)) \mathbb{1}_{i_{XY}(E(s'):y) \geq \log M + \tau} \Big) &= \sum_{x, x', y} P_X(x) P_X(x') W(y|x) \mathbb{1}_{\underbrace{i_{XY} x':y) \geq \log M + \tau}_{\frac{W(y|x')}{P_Y(y)} \geq M.2^{-\tau}}} \\ &= \sum_{x', y} P_Y(y).P_X(x') \mathbb{1}_{W(y|x') \geq P_Y(y) M.2^{\tau}} \\ &\leq \sum_{x, y} W(y|x'). \frac{2^{-\tau}}{M}.P_X(x') \\ &= \frac{2^{-\tau}}{M} \end{aligned}$$

So overall, we have

$$\mathbb{E}(P_{err}) = \frac{1}{M} \sum_{s=1}^{M} \mathbb{E}(P_{err,s}) \leq \mathbb{P}(i_{XY}(X : Y) < \log M + \tau) + \underbrace{(M-1)}_{\text{sum for } s \neq s'} \frac{2^{-\tau}}{M}$$

This implies that there exists an M-code with

$$P_{err} \leq \mathbb{P}(i_{XY} < \log M + \tau) + 2^{-\tau}$$

□

Important special case Memoryless channel $W^{\times n}$
Look at rate: $\frac{\log_2 M}{n}$.

Theorem 8 (Shannon's noisy coding theorem)**.** *Let W be a channel. For any $\delta \in (0, 1)$*

$$C(W) \leq \lim_{n \to \infty} \frac{\log M^{OPT}(W^{\times n})}{n} \leq \frac{C(W)}{1 - \delta}$$

Proof. • For upper bound: Follows directly from converse and $C(W^{\times n}) = nC(W)$

- For lower bound, let P_X be a distribution on $\mathcal{X}$ achieving $\max_{P_X} I(X:Y)$ (for channel W). We define $X_1, ..., X_n$ n independent random variables with distribution P_X, let $Y_1, ..., Y_n$ the corresponding outputs.

$$i_{X^n:Y^n}(X^n:Y^n) = \sum_{i=1}^{n} i_{XY}(X_i:Y_i) \quad \text{i.i.d.}$$
$$\mathbb{P}(i_{X^nY^n}(X^n:Y^n) \leq n(I(X:Y)-\epsilon) \underset{n\to\infty}{\to} 0 \qquad \text{WLLN}$$

Take $M = \lceil 2^{n(I(X:Y)-2\epsilon)} \rceil$ and $\tau = n\epsilon$, then $\mathbb{P}(i_{X^nY^n}(X^n:Y^n) \leq \log M + \tau) + 2^{-\tau} \leq 2^{-n\epsilon} + \frac{\delta}{2} \leq \delta$ for large enough n.

$\square$

Comments:

- It turns out that for any $\delta \in \{0,1\}$,

$$\lim_{n\to\infty} \frac{\log M^{OPT}(W^{\times n}, \delta)}{n} = C(W)$$

One can obtain good finite n bounds:

$$\frac{\log_2 M^{OPT}(W^{\times n}, \delta)}{n} = C(W) + \frac{Q\delta}{\sqrt{n}} + O(\frac{\log n}{n})$$

Zero error coding For the binary symmetric channel:

$$M^{OPT}(BSC^{\times n}, 0) = 1$$

For $W(1|1) = 1/2, W(2|1) = 1/2, W(2|2) = 1/2, W(3|2) = 1/2, W(3|3) = 1$, with the codebook $\{1,3\}$, we can decode with zero error.

For zero error, the relevant description of W is the confusability graph.

$G(W) =$
- vertices are channel input $\mathcal{X}$
- (u,v) is on an edge if $\exists y \in \mathcal{Y}, W(y|x) > 0$ and $W(y|v) > 0$

Then,

$$M^{OPT}(W,0) = |\text{MaxIndSet}(G(W))|$$

Where an independent set of G is a subset of vertices with no edge between them.

We can ask the question for a memoryless channel:

$$\lim_{n\to\infty} \frac{\log M^{OPT}(W^{\times n}, 0)}{n}$$

$G(W^{\times n}) =$Vertices indexed by $\mathcal{X}^n$

Edges: $(x_1, ..., x_n) \sim (x'_1, ..., x'_n)$ (There is an edge between $(x_1, ..., x_n)$ and $(x'_1, ..., x'_n)$)

$$\begin{aligned}
&\exists y_1 ... y_n W^{\times n}(y_1...y_n|x_1...x_n) > 0 \text{ and } W(y_1...y_n|x_1'...x_n') > 0 \\
\Leftrightarrow &\exists y_1 ... y_n W(y_1|x_1)...W(y_n|x_n) > 0 \text{ and } W(y_1|x1')...W(y_n|x_n') > 0 \\
\Leftrightarrow &x_1 \sim x_1' \text{ and } x_2 \sim x_2' ... \text{ and } x_n \sim x_n' \text{ in graph } G(W)
\end{aligned}$$

The question is then: given a graph G, how does

$$\text{MaxIndSet}(G^{\times n})$$

grow with n?

Given an independent set I for G, then I^n is an independent set of $G^{\times n}$.

$$I^n = \{(x_1, ..., x_n) \text{ such that } x_i \in I\}$$

So $\text{MaxIndSet}(G^{\times n}) \geq \text{MIS}(G)^n$

Famous example Let C_5 be the connected graph with 5 edges and 5 vertices. $\text{MIS}(C_5) = 2$ but $\text{MIS}(C_5^{\times 2}) = 5$. So $MIS(C_5^{\times 2n}) \geq 5^n$.

$$\lim_{n\to\infty} \frac{\log M^{OPT}(C_5^{\times n}, 0)}{n} \geq \frac{1}{2}\log 5$$

It is possible to show that

$$\lim_{n\to\infty} \frac{\log M^{OPT}(C_5^{\times n}, 0)}{n} = \frac{1}{2}\log 5 \qquad \text{Hard}$$

5 Information and combinatorics

Simple inequality

Lemma 3 (Shearer's lemma)**.** $(X_1, ..., X_n), S_1, ..., S_m \subseteq [n] = \{1, ..., n\}$ *Suppose that for all $i \in [n]$, i appears in more ($\geq$) than k sets, then:*

$$H(X_1, ..., X_n) \leq \frac{1}{k}\sum_{j=1}^{m} H(X_{S_j})$$

Where

$$H(X_S) = H(X_{e(1)}...X_{e(|S|)}) \qquad \textit{with } S = \{e_1, ..., e_n\}$$

Proof.

$$\begin{aligned}
H(X_1, .., X_n) &= H(X_1) + H(X_2|X_1) + ... + H(X_n|X_1, ..., X_{n-1}) \\
S_j &= \{e_j(1), ..., e_j(|S_j|)\} \qquad \text{with } e_j(1) \leq e_j(2) \leq ... \\
H(X_{S_j}) &= H(X_{e_j(1)}) + H(X_{e_j(2)}|X_{e_j(1)}) + ... \geq H(X_{e_j(1)}|X_1...X_{e_j(1)-1}) + H(X_{e_j(2)}|X_1...X_{e_j(2)})
\end{aligned}$$

For each $i \in [n]$, the term $H(X_i|X_1...X_{i-1})$ appears k times int h lower bound on

$$\sum_{j=1}^{m} H(X_{S_j})$$

So we get the bound. $\square$

Application 1 Projection of points sets.

S set of m points in $\mathbb{R}^3, S = \{a(1), ..., a(m)\}, a(i) = \{a_i(1), a_i(2), a_i(3)\}$. Define
$\Pi_{XY} = \{(a_1(i), a_2(i)\ i \in [m]\}$; $\Pi_{XZ} = \{(a_1(i), a_3(i)\ i \in [m]\}$, $\Pi_{YZ} = \{(a_2(i), a_3(i)\ i \in [m]\}$

Suppose $|\Pi_{XY}|, |\Pi_{XZ}|, |\Pi_{YZ}| \leq n$. How large can m be?

Claim If S has m points with projections of size $\leq n$, then $m \leq n^{2/3}$

Proof.

$$P_{A_1A_2A_3}(a_1a_2a_3) = \begin{cases} \frac{1}{m} & \text{if } (a_1, a_2, a_3) \in S \\ 0 & \text{otherwise} \end{cases}$$

$$H(A_1A_2A_3) = \log m$$

The condition $|\Pi_{XY}| \leq n$ says $|H(A_1A_2) \leq \log n$. Using Shannon's lemma:

$$\begin{aligned} \log m = H(A_1A_2A_3) &\leq \frac{1}{2}(H(A_1A_2) + H(A_1A_3) + H(A_2A_3)) \\ &= \frac{3}{2}\log n \end{aligned}$$

$$m \leq n^{3/2}$$

□

Application 2 Number of independent sets in a graph

Let n be the number of vertices of the graph. We look at d-regular graphs.

Theorem 9. *If G is a bipartite d-regular graph with n vertices, then*

$$|\underbrace{I(G)}_{\textit{Set of indep. sets of G}}| \leq (2^{d+1} - 1)^{\frac{n}{2d}}$$

This bound is achieved by taking copies of bipartite complete graph.

Proof. $[n] = \{1, ..., n\}$ labels of vertices, $[n] = A \cup B$ with edges only on A and B, $|A| \geq |B|$. Let I be a uniformly random independent set in $I(G)$. Let $X_i = \mathbb{1}_{i \in I}$

$$\begin{aligned} H(X_1...X_n) &= \log |I(G)| \\ &= H(X_A) + H(X_B|X_A) \\ H(X_B|X_A) &\leq \sum_{b \in B} H(X_b|X_A) \\ &\leq \sum_{b \in B} H(X_b|X_{N(b)}) \\ &\text{with } N(b) = \{a \in A : (a, b) \in E\} \end{aligned}$$

Define

$$Q_b = \begin{cases} 1 & \text{if } |I \cap N(b)| = 0 \\ 0 & \text{otherwise} \end{cases}$$
$$\leq \sum_b H(X_b|Q_b)$$

$$\begin{aligned} H(X_b|Q_b) &= P_{Q_b}(0)H(P_{X_b|Q_b=0}) \\ &\quad + P_{Q_b}(1)H(P_{X_b|Q_b=1}) \\ &\leq P_{Q_b}(1) = q_b \\ H(X_B|X_A) &\leq \sum_b q_b \\ H(X_B|X_A) &\leq \frac{1}{d}\sum_{b\in B} H(X_{N(b)}) \text{ using Shearer's lemma and degree } d \end{aligned}$$

Note that $H(X_{N(b)}Q_b) = H(X_{N(b)})$.

$$\begin{aligned} H(X_{N(b)}Q_b) &= H(Q_b) + H(X_{N(b)}|Q_b) \\ &= h_2(q_b) + P_{Q_b(0)}H(P_{X_{N(b)}|Q_b=0}) \\ &\quad + \underbrace{P_{Q_b(1)}H(P_{X_{N(b)}|Q_b=1})}_{=0} \\ &\leq h_2(q_b) + (1-q_b)\log(2^d - 1) \\ H(X_1...X_n) &\leq \frac{1}{d}\sum_b h_2(q_b) + (1-q_b)\log(2^d-1) + \sum_b q_b \end{aligned}$$

It turns out that this is at most

$$\frac{n}{2d}\log(2^{d+1}-1)$$

For any $q_b \in [0,1]$ using fact that $|B| \leq \frac{n}{2}$ $\square$

6 Error correcting code

Shannon's theorem says that for any nontrivial channels there are M-codes with $M \approx 2^{nC(W)}$ codewords that can be decoded with very small error probability given the output of the channel W. It even said that provided we pick the codewords at random with a good distribution, then most codes are good. Our objective now is to explicitly construct good codes. The notion of a good code depends on the channel being studied and involves both the construction of an encoder and a decoder. To simplify the study it is useful to consider a different error model than the one we considered so far and in this model the existence of a decoder is directly related to a simple property of the codebook. Recall that in the Shannon model, an encoder is good if there exists a decoder that can decode with a small error probability. In the Hamming model, a good encoder is one for which there is a decoder that can correct any error of weight at most t. The models are not exactly the same but they are related and we will see that it is possible to construct good codes in the Shannon sense using good codes in the Hamming sense.

6.1 General error-correcting codes

Definition 16. *A code C of blocklength n over an alphabet Σ is a subset of Σ^n. We usually write $q = |\Sigma|$. The dimension of a code is defined as $k = \log_q |C|$.*

Remark Note that a way to specify a code is as an injective encoding function C: $\Sigma^k \to \Sigma^n$ and the code corresponds to the image of the encoding function C. Even though they are not the same objects, we will be using the word "code" for both of these. As mentioned before, we consider the Hamming error model where our objective is to be able to correct all errors of weight at most t. Note that if you want to think it terms of channels, you should see $\mathcal{X} = \mathcal{Y} = \Sigma$ and then taking n copies of the channel for example.

Definition 17. *C is t-error correcting if there exists a decoding map $D : \Sigma^n \to C$ such that for any $c \in C$ and any error pattern e with at most t errors $D(c + e) = c$.*

Let us look at simple examples

1. The repetition code $C_{rep} = \{000, 111\}$. This code has $q = 2$, $n = 3$, $k = 1$. It is 1-error correcting. In fact, my decoding function can map to 000 inputs of weight at most 1 and map to 111 inputs of weight ≥ 2.
2. The binary code defined by $C_\oplus(x_1x_2) = x_1x_2(x_1 \oplus x_2)$ has $q = 2, n = 3, k = 2$. It is not 1-error correcting. In fact $C_\oplus(00) = 000$ and $C_\oplus(01) = 011$. If I apply a weight 1 error to the first codewords I can get 010, but I can also get to 010 by applying a weight 1 error to the second codeword. So I can detect that there is an error but I cannot correct for it.

From this example, one sees that the relevant parameter that governs how many errors a code can correct is the Hamming distance between the codewords.

Definition 18 (Minimum distance of a code)**.** *The Hamming distance between $u, v \in \Sigma^n$ is defined by $\Delta(u, v) = |\{i \in [n] : u_i \neq v_i\}|$.*

The minimum distance (or just distance) of a code C is defined as

$$d = \min_{c,c' \in C, c \neq c'} \Delta(c, c')$$

Note that in the Hamming distance, we do not have a notion of distance between two symbols in Σ they are either the same or different. For example, if we think of $\Sigma = \{0, 1\}$ and consider the bitstrings $u = 0010$ and $v = 1110$, their Hamming distance is 2. However, if we consider $\Sigma = \{0, 1\}$ and consider $u, v \in \Sigma 2$, then their Hamming distance is 1.

Let us look at the examples we considered before

1. The repetition code C_{rep} has a minimum distance of 3
2. The code $C_\oplus$ has a minimum distance of 2. In fact, take two different codewords $c = C_\oplus(x_1x_2)$ and $c' = C_\oplus(y_1y_2)$. Then if $\Delta(x_1x_2, y_1y_2) = 2$, then $\Delta(c, c') \geq 2$. Otherwise, if $\Delta(x_1x_2, y_1y_2) = 1$, then $\Delta(c, c') = 2$.

We now see that minimum distance is directly related to the number of errors that can be corrected. We only do here the special case of d odd, the even case will be done in the tutorial.

Propriety 8. *Assume $d \geq 3$ is odd. Then the following are equivalent.*

- *C has minimum distance d*
- *C can correct $\frac{d-1}{2}$ errors*

Proof. Suppose C has minimum distance d. Then define the function $D : \Sigma^n \to C$ by $D(y) = \operatorname*{argmin}_{c \in C} \Delta(c, y)$. Then suppose c_1 is transmitted and $\Delta(c_1, y) \leq t$. Then let $D(y) = c$. We have $\Delta(c_1, c) \leq \Delta(c_1, y) + \Delta(y, c) \leq t + t$. This is equal to $2d$ provided $t = \frac{d-1}{2}$. As such $c = c_1$.

Now suppose C has distance $\leq d - 1$. Then there exists $c_1, c_2 \in C$ with $\Delta(c_1, c_2) \leq d - 1$. Consider y such that $\Delta(y, c_1), \Delta(y, c_2) \leq \frac{d-1}{2}$. This y could be received for either c_1 or c_2 so C cannot correct $\frac{d-1}{2}$ errors. $\square$

Notation We use the notation $(n,k,d)_q$-code when blocklength n, dimension k, minimum distance d and the alphabet Σ has size q. Let us see another less trivial code that we have already encountered in the first lecture. This is the Hamming code. It is also a binary code with $q=2$. We may define it by

$$C_H(x_1x_2x_3x_4) = (x_1, x_2, x_3, x_4, x1 \oplus x_2 \oplus x_4, x_1 \oplus x_3 \oplus x_4, x_2 \oplus x_3 \oplus x_4)$$

This is a $(7,4,d)_2$ code where we still have to determine d. I claim that the minimum distance is 3. First $0000000 \in C_H$ and $1000110 \in C_H$ and they are at distance 3. Moreover, for two different codewords $C_H(x)$ and $C_H(y)$, we can write

$$\begin{aligned}\Delta(C_H(x), C_H(y)) &= |\{i \in [7] : C_H(x)_i \neq C_H(y)_i\}| \\ &= |\{i \in [7] : C_H(x)_i + C_H(y)_i \neq 0\}| \\ &= |C_H(x) + C_H(y)| \\ &= |C_H(x+y)|\end{aligned}$$

as the mapping C_H is a *linear* map. So it suffices to determine $\min_{x\neq 0} |C_H(x)|$. We do this by considering the different cases for the Hamming weight of x. If $|x|=1$, then two or three of the following bits evaluate to 1: $x_1 \oplus x_2 \oplus x_4, x_1 \oplus x_3 \oplus x_4; x_2 \oplus x_3 \oplus x_4$. If $|x|=2$, then at least one of these bits evaluates to 1 and if $|x|=3$, we already have $|C_H(x)| \geq 3$. We conclude that C_H is a $(7,4,3)_2$ code.

Note that this code has a very nice property that we will be exploiting further. The encoding function is a linear function. In fact, we can see messages as elements of $\mathbb{F}_2^4$ and codewords as elements of $\mathbb{F}_2^7$ and the transformation is given by a matrix

$$G_H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and $C_H(x) = xG_H$ where we see x as a row vector in $\mathbb{F}_2^4$. One can in general define linear codes whenever Σ has a field structure so that Σ^n is a vector space over the field Σ and $C \subseteq \Sigma^n$ is a subspace. Before getting into the detailed study of linear codes, let us determine some simple bounds on the best parameters one can achieve for codes.

6.1.1 General bounds on the best codes

For a fixed n and q, we would like k and d to be as large as possible. For example, the Hamming code is a $(7,4,3)_2$ code, is it possible to improve it to a $(7,5,3)_2$ code for example? The answer is no by the following simple packing bound. Again, we only state here a simplified for with $q=2$ and $d=3$ but it is easy to generalize (see tutorial).

Theorem 10 (Hamming bound (special case))**.** *Every binary code with blocklength n, dimension k and distance $d=3$ satisfies*

$$k \leq n - log_2(n+1)$$

For $n=7$ and $d=3$, this gives $k \leq 4$, which means the Hamming code is optimal in this sense.

Proof. Let C be such a code and c_1, c_2 be two codewords. For $u \in \{0,1\}^n$, let $B(u,1) = \{v \in \{0,1\}^n : \Delta(u,v) \leq 1\}$. We have $B(c_1,1) \cap B(c_2,1) = \emptyset$. In addition$|B(u,1)| = 1+n$. As a result,

$$|\bigcup_{c\in C} B(c,1)| = (n+1)2^k$$

But clearly this number is at most the size of the whole space which is 2^n. So

$$k \leq n - log_2(n+1):$$

□

Note that having equality in this bound means that we have perfect packing, i.e., $\bigcup_{c \in C} B(c,1) = \{0,1\}^n$ Such codes are called perfect codes.

Theorem 11. *Let $q \leq 2$, $1 \leq d \leq n$. There exists a $(n,k,d)_q$-code with $k \geq n - \log_q Vol_q(d-1,n)$*

Proof. Greedily construct C.

```
C = ∅
while There is x ∈ Σ^n with Δ(x,c) ≥ d for all c ∈ C do
    C ← C ∪ {x}
```

Clearly at any time C has minimum distance $\geq d$.

When the algorithm terminates :

$$\begin{aligned} \forall x \in \Sigma^n, \exists c \in C : \Delta(x,c) \leq d-1 \\ \Sigma^n \subseteq \bigcup_{c \in C} B(c, d-1) \\ q^n \leq |\bigcup_{c \in C} B(c,d-1)| \leq \sum_{c \in C} |B(c,d-1)| = |C| Vol_q(d-1,n) \\ = q^k Vol_q(d-1,n) \end{aligned}$$

So $k \geq n - \log_q Vol_q(d-1,n)$ □

6.2 Linear error correcting code

Theorem 12. *The size of any finite field*[1] *is $q = p^s$ for some prime p and integer $s \geq 1$. Moreover, there is a unique field of size q denoted $\mathbb{F}_q$.*

- For $p = q$, $\mathbb{F}_q$ can be seen as integers mod p with the usual addition and multiplication.
- For $q = p^s$, elements of $\mathbb{F}_q$ are polynomials in $\mathbb{F}_p[X]$ modulo an irreducible polynomial $Q \in \mathbb{F}_p[X]$ of degree s.

Definition 19. *Let q be a prime power. $C \subset \mathbb{F}_q^n$ is a linear code if it is a linear subspace of ${\mathbb{F}_q}^n$, i.e., if $x, y \in C$, $x + y \in C$ and $a.x \in C$ for $a \in \mathbb{F}_q$.*

Notation $[n,k,d]_q$, *with k the dimension and d the distance.*

[1] fr : corps

Example Repetition code $C = \{000, 111\}$ is a linear code over $\mathbb{F}_2$. This forms a $[3, 1, 3]_2$ code.

Propriety 9. *Let S be a linear subspace of $\mathbf{F}_q^n$.*

1. *$|S| = q^k$ for k integer*
2. *There exists a basis $v_1, v_2, ..., v_k$ such that for any $x \in S$, there is unique $(a_1, ..a_k) \in \mathbf{F}_q^k$ such that $x = \sum_{i=1}^{k} a_i \vec{v_i}$*
   Then the $k \times n$ matrix

   $$G = \begin{pmatrix} \leftarrow & v_1 & \rightarrow \\ & \vdots & \\ \leftarrow & v_k & \rightarrow \end{pmatrix} \qquad x = (a_1, ..., a_k)$$

   is called a generator matrix. Note that rows of G are linearly independent and so G has full rank.
3. *There exists a full rank $(n-k) \times n$ matrix called parity check matrix such that for all $x \in S$, $Hx^T = 0_{n-k}$*

Example for repetition code

$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ and $H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1 + x_3 \end{pmatrix}$$

Sketch of proof. To construct a basis, can do it in a greedy way. Take $v_1 \in S$ non-zero. Then, at step t, $v_t \notin \left\{\sum_{i=1}^{t-1} a_i v_i : a_i \in \mathbb{F}_q\right\}$.

We obtain $v_1, ..., v_k$: It is clear that $v_1, ..., v_k$ generates S. Also by induction, it is simple to show that $\left\{\sum_{i=1}^{t-1} a_i v_i\right\}$ contains exactly q^t elements.

$$N = \left\{ y \in \mathbb{F}_q^n : \sum_{i=1}^{n} x_i y_i = 0 \; \forall x \in S \right\}$$

N is a linear subspace of $\mathbb{F}_q^n$. To obtain a parity check matrix, take a basis of N. □

Minimum distance of a linear code

Propriety 10. *The minimum distance of a linear code C is given by $d = \min_{c \in C | c \neq 0} |c|$ where* $|c| = |\{i \in [n] : c_i \neq 0\}$

Proof. $0 \in C$ and $\Delta(0, c) = |c|$ so the minimum distance is at most $\min_{c \in C | c \neq 0} |c|$.

For $c_1 \neq c_2$ $c_1, c_2 \in C$

$$\Delta(c_1, c_2) = |c_1 - c_2| \geq \min_{\substack{c \in C \\ c \neq 0}}$$

□

Propriety 11. *Let C be an $[n,k,d]_q$ code with parity check matrix $H = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ H^1 & H^2 & \dots & H^n \\ \downarrow & \downarrow & \cdots & \downarrow \end{pmatrix}$.*

Let t = minimum number of linearly dependent columns.

Then

$$d = t$$

Proof.

- Begin with $t \leq d$.
  Let c be a codeword with $|c| = d$. Then $Hc^T = 0$. But $Hc^T = \sum_{i=1}^n c_i H^i$.
  The support of c gives d linearly dependent columns of H.
- For $t \geq d$, let $H^{i_1}, ..., H^{i_t}$ be linearly dependent. There exits $C_{i_1}, ..., C_{i_t}$ such that $\sum_{j=1}^t c_i H^{i_j} = 0$. Define $x\mathbb{F}_q^n$ with $x_{i_j} = c_{i_j}$ for all j and $x_i = 0$ otherwise.
  Then $x \in C$ as $Hx^T = 0$ and $|x| = t$ so $d \leq t$

□

Example Generalized Hamming codes.

$q = 2$. For $n \geq 3$,

$H = \begin{pmatrix} H_r^1 & \dots & H_r^{2^r-1} \end{pmatrix}$ where H_r^i is the binary representation of i of length r.

$$H_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

H_r has rank r because $\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, ..., \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ are linearly independent.

H_r defined a $[2^r - 1, 2^r - 1 - r, ?]_2$-code.

For $r = 3$, we know the min distance is 3.

Claim H_r defines a $[2^r - 1, 2^r - 1 - r, 3]_2$-code.

Proof.

- H_r^1, H_r^2 and H_r^3 satisfy $H_r^1 + H_r^2 + H_r^3 = 0$. So $d \leq 3$.
- In addition, a distance of 2 would mean that there is a pair $i \neq j$ with $H_r^i + H_r^j = 0$. But this would implies that $i = j$. So $d \geq 3$.

□

Rate of this code $= \frac{2^r - r - 1}{2^r - 1}$ very close to 1 but min distance 3 is poor.

Dual code of a linear code

Definition 20. *Let C be a linear code with parity check matrix H. The code with generator matrix H is called $C^\perp$ dual code.*

If C is an $[n,k]_q$-code, $C^\perp$ is an $[n,n-k]_q$-code.

Dual code of Hamming code $C_{Ham,r}$

$$\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Let's call $C_{Sim,r} = C^\perp_{Ham,r}$. One encoding function f $C_{Sim,r}$ is given by $C_{Sim,r}(x) = xH_r$. We define $C_{Had,r}(x) = x \begin{pmatrix} 0 & & & \\ \vdots & h_r^1 & \dots & H_r^{2^r-1} \\ 0 & & & \end{pmatrix}$ Hadamard code.

Propriety 12. *The minimum distance of codes $C_{Sim,r}$ and $C_{Had,r}$ is 2^{r-1}*

Proof. Sufficient to prove it for C_Had, r.

Claim For any $c \in C_{Had,r}, c \neq 0, |c| = 2^{r-1}$. For any $c \in C_{Had,r}, c \neq 0$, there exists $x \neq 0 \in \mathbb{F}_q^r$ such that $c = (xH_r^0, ..., xH_r^{2^r-1})$. We can write $c = (< x, u >)_{u \in \{0,1\}^n}$ As $x \neq 0$, $\exists i, x_i = 1$. Let $e_i = (0...0 \underbrace{1}_{i} 0...0) \in \mathbb{F}_q^r$.
$v = u + e_i$
$< x, v > = < x, u > + < x, e_i > = < x, u > +1$
So components $< x, v >$ and $< x, u >$ are distinct. □

Encoding and decoding a linear code

Encoding To an $[n,k,d]$-code C, we can associate a neutral encoding function.

Take G a generator matrix for C. Let the set of messages be $\mathbb{F}_q^k$.

The encoding function is

$$\begin{aligned} C : &\mathbb{F}_q^k \to \mathbb{F}_q^n \\ &a \mapsto aG \end{aligned}$$

Which can be computed in $n.k$ operations in $\mathbb{F}_q$

Decoding

- Error detection: with parity check matrix H, costs $n.(n-k)$ operation in general
- Detection:
  Start with $x \in C$
  Error $e \in \mathbb{F}_q^n$
  Receive: $y = x + e \in \mathbb{F}_q^n$
  But

$$\underbrace{Hy^T}_{\text{syndrome}} = \underbrace{Hx^T}_{=0} + He^T = He^T$$

input : $y \in \mathbb{F}_q^k$
output: $x \in C$
for $i = 0$ *to* t **do**
  for $e \in \mathbb{F}_q^k$ *of weight* i **do**
    if $He^T = Hy^T$ **then**
      return $y - e$

Algorithm Generic decoding linear code

$$\text{Number of steps} = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$$

Polynomial for t constant, but exponential in t.

Ex of Hamming code $[2^r - 1, 2^r - 1 - r, 3]_2$-code, Parity check matrix H_r.

- Start by computing syndrome: $s = H_r y^T$
- Want to find e of weight ≤ 1 such that $s = H_r e^T$

If we have an error in position i, $e_i = (0...0 \underbrace{1}_{i} 0...0)$

$$\begin{aligned} H_r e_i^T &= H_r^i \qquad i\text{-th column of } H_r \\ &= i \text{written in binary} \end{aligned}$$

$$H_r = \underbrace{\begin{pmatrix} \quad & \quad \\ & \end{pmatrix}}_{2^r-1} \left.\vphantom{\begin{pmatrix} a \\ b \end{pmatrix}}\right\} r$$

Decoding Interpret $s \in \{0,1\}^r$ as a number between 1 and $2^r - 1$ and flip corresponding bit.

In general, the problem of decoding is : Find $e \in \mathbb{F}_q^k$ of smallest weight s.t.

$$He^T = s$$

6.3 Reed-Solomon codes

Based on univariate polynomials.

$$\begin{aligned} &f_m(X) = \sum_{i=0}^{d} m_i X^i \in \mathbb{F}_q[X] \qquad m_i \in \mathbb{F}_q \\ &\deg f_m = d \text{ if } m_d \neq 0 \end{aligned}$$

Definition 21. *We assume $1 \leq k \leq n \leq q$. Let $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{F}_q$ distinct.*
The Reed-Solomon code is :

$$RS : \mathbb{F}_q^k \to \mathbb{F}_q^m$$
$$RS(\underbrace{m}_{m=(m_0,...,m_{k-1})}) = (f_m(\alpha_1), ..., f_m(\alpha_n))$$

For $n^0, n^1 \in \mathbb{F}_q^k$:

$$f_{n^0}(X) + f_{n^1}(X) = f_{n^0+n^1}(X)$$

So

$$RS(m^0) + RS(m^1) = RS(m^0 + m^1)$$
$$\text{For } a \in \mathbb{F}_q \ RS(am) = aRS(m)$$

RS is a linear code.

Propriety 13. *The minimum distance of RS is*

$$n - k + 1$$

Proof.

$$\begin{aligned} RS(m) &= \big(f_m(\alpha_1), ..., f_m(\alpha_n)\big) \\ \text{Weight: } |RS(m)| &= |\{i \in [n] : f_n(\alpha_i) \neq 0\}| \\ &= n - |\{i \in [n] : f_m(\alpha_i) = 0\}| \end{aligned}$$

But if $m \neq 0$ then f_m is a non-zero polynomial of degree $\leq k-1$. So $|\{i \in [n] : f_m(\alpha_i) = 0\}| \leq k-1$. So $|\{i \in [n] : f_m(\alpha_i) = 0\}| \leq k-1$

Important fact A nonzero polynomial of degree $k-1$ has at most $k-1$ roots.
So $|RS(m)| \geq n-k+1$: RS are $[n, k, n-k+1]_q$-codes. □

This minimum distance is optimal as it achieves the Singleton bound (see tutorial).
Ex of generator matrix for RS: Take basis: $1, X, ..., X^{k-1}$

$$G = \begin{pmatrix} 1 & 1 & ... & 1 \\ \alpha_1 & \alpha_2 & ... & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & ... & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & ... & \alpha_n^{k-1} \end{pmatrix}$$

Efficient decoding of RS-codes Message to be send: P. Given y, We want to find P such that:

$$\Delta\left(\left(P(\alpha_1),...,P(\alpha_n)\right),y\right)\leq t$$

Where $t=\left\lfloor\frac{d-1}{2}\right\rfloor$ where $d=n-k+1=$ minimum distance.

This is a polynomial interpolation problem with errors.

Introduce

$$E(X)=\prod_{\substack{i=1\\ y_i\neq P(\alpha_i)}}^{n}(X-\alpha_i) \qquad \text{(error locator poly)}$$

$$\deg E\leq t$$

Claim We have for all $i\in[n]$

$$y_iE(\alpha_i)=P(\alpha_i)E(\alpha_i)$$

Reason: If $E(\alpha_i)=0$, clearly satisfied, if $E(\alpha_i)\neq 0$, there is no error at position i and so $P(\alpha i)=y_i$.

We have n equations, and the number of variable is:

$$\begin{aligned}\#\text{ variables } &= \text{ at most } t \text{ for } E \text{ and at most } k \text{ for } P\\ &\leq t+k\\ &\leq \frac{(n-k+1)-1}{2}+k\\ &=\frac{n+k}{2}\end{aligned}$$

But these equations are *not* linear in the variables.

Idea Relax the equation to

$$y_iE(\alpha_i)=N(\alpha_i) \text{ with } N \text{ polynomial of degree} \leq k-1+t$$

input : $(y_1,...,y_n)\in\mathbb{F}_q^n$ with promise $\min_m\Delta(y,RS(m))\leq t$
output: P polynomial of degree $\leq k-1$

1 Solve $y_iE(\alpha_i)=N(\alpha_i)$ $(\star)$ where variables are $e_0,...,e_{t-1}$ and $E(X)=e_0+e_1X+...+e_{t-1}X^{t-1}+X^t$ and $n_0,...,n_{t+k-1}$ and $N(X)=n_0+n_1X+...+n_{t+k-1}X^{t-k+1}$.
2 **if** *no solution or E does not divides N* **then**
3 | Fail
4 Return $P(X)=\frac{N(X)}{E(X)}$

Algorithm 1: Decoding of RS

Running time Solving the linear system can be done in $O(n^3)$

Proof. Correctness: First, we show that $(\star)$ has a valid solution.

$$RS(m) = \Big(f_m(\alpha_1), ..., f_m(\alpha_n)\Big)$$
$$\Delta(RS(m), y) \leq t$$

Define

$$E^*(X) = \prod_{i:y_i \neq f_m(\alpha_i)} (X - \alpha_i).X^{t-\Delta(9y,RS(m))}$$
$$\text{and } N^*(X) = f_m(X)E^*(X)$$
$$\text{We have } y_i.E^*(\alpha_i) = N^*(\alpha_i)$$

For this solution E^* divides N^* and we output f_m.
Let's show that this solution is unique. Let (N_1, E_1) and (N_2, E_2) be solutions of $(\star)$.

$$R(X) = N_1(X)E_2(X) - N_2(X)E_1(X)$$
$$\deg R \leq (k + t - 1) + t = 2t + k - 1$$
$$\text{recall } t = \left\lfloor \frac{n - k + 1 - 1}{2} \right\rfloor$$
$$2t + k - 1 = n - k + k - 1 = n - 1 \text{ (if } n - k \text{ even)}$$

In all cases

$$2t + k - 1 < n$$

On the other hand

$$N_2(\alpha_i) = y_i.E_2(\alpha_i)$$
$$N_1(\alpha_i) = y_i.E_1(\alpha_i)$$

So $E_1(\alpha_i)N_2(\alpha_i) = E_2(\alpha_i)N_1(\alpha_i)$, so $R_1(\alpha_i) = 0$ for all i, $\Rightarrow$ R has n distinct roots, so $R = 0$.
So $\frac{N_2(X)}{E_2(X)} = \frac{N_1(X)}{E_1(X)}$

□

Objective "Good" binary codes: $k = \Omega(n)$, $d = \Omega(n)$, explicit and efficient encoding/decoding.
As the Reed-Solomon codes, $[n, k, n - k + 1]_q$, they are optimal (achieving the singleton bound).

Issue $q \geq n$, the alphabet size should be large.

6.4 Concatenation of codes

Code C on alphabet $[q]=\{1,...,q\}$ with blocklength $(x_1,...,x_n\in C)$. Assume $q=2^t$. We can interpret $(x_1,...,x_n)$ as $(x_{11},x_{12},...,x_{1t},x_{21},...,x_{n,1},...,x_{nt})\in\{0,1\}^{nt}$.

This procedure gives a binary code with blocklength nt and dimension kt (2^{kt} codewords).

Consider $\left[n,\frac{n}{2},\frac{n}{2}+1\right]_n$ RS code, we will obtain a $\left(n\log_2 n,\frac{n}{2}\log_2 n,?\right)_2$ code.

Let $k=\log n$

$$\begin{aligned}\Delta(x_{1,1},...,x_{1,t},...,x_{n,1},...,x_{n,t},y_{1,1},...,y_{n,t}) &= |\{(i,j)\in[n]\times[t]:x_{i,j}\neq y_{i,j}\}|\\ &\geq |\{i\in[n]:x_i\neq y_i\}|\\ &\geq \frac{n}{2}+1 \qquad \text{(Distance of our original code)}\end{aligned}$$

We obtained a $\left(n\log_2 n,\frac{n}{2}\log_2 n,\frac{n}{2}+1\right)_2$.

Idea Instead of trivial representation: $x_i\to x_{i,1},...,x_{i,t}$ (binary representation), we will use a *code*.

Definition 22 (Concatenation Code)**.** *Let $C_{out}:[Q]^K\to[Q]^N$ a $(N,K,D)_Q$ code and $C_{in}:[q]^k\to[q]^n$ be a $(n,k,d)_q$ code with $Q=q^k$.*

Then the concatenation $C_{out}\circ C_{in}$ is a code on alphabet $[q]$ blocklength nN, dimension kK defined by

$$\begin{aligned}C:[Q]^K&\to[q]^{nN}\\ C(m)&=\Big(C_{in}\big(C_{out}(m)_1\big),C_{in}\big(C_{out}(m)_2\big),...,C_{in}\big(C_{out}(m)_N\big)\Big)\end{aligned}$$

Where $C_{out}(m)_i$ is the i-th symbol of $C_{out}(m)$

In the example: C_{out}: RS $\left[N,\frac{N}{2},\frac{N}{2}+1\right]_N$.
C_{in} is $(n=\log N,k=\log N,d=1)_2$, and $C_{in}(x)=x$ the trivial code.

Remark We have identified $[Q]$ with $[q]^k$. For that, we can take any bijection between the sets.

When C_{in} and C_{out} are *linear* codes, we can take this bijection so that $C_{out}\circ C_{in}$ is also a linear code.

In this, we use $[Q]=\mathbb{F}_{q^k}$ ($Q=q^k$) and $[q]^k=\left(\mathbb{F}_q\right)^k$. $\mathbb{F}_{q^k}$ can be seen as a vector space over $\mathbb{F}_q$.

Let $\sigma:\mathbb{F}_{q^k}\to\left(\mathbb{F}_q\right)^k$ be an isomorphism, G_{in} and G_{out} generator matrices for C_{in} and C_{out}.

$$G_{out\circ in}=$$
$$\underset{(\mathbb{F}_q)^{kK}}{\uparrow}\begin{pmatrix}\sigma^{-1} & & \\ 0 & \ddots & \\ & & \sigma^{-1}\end{pmatrix}\underset{(\mathbb{F}_{q^k})^{K}}{\uparrow}\; G_{out}\;\underset{(\mathbb{F}_{q^k})^{N}}{\uparrow}\begin{pmatrix}\sigma & & \\ 0 & \ddots & 0\\ & & \sigma\end{pmatrix}\underset{(\mathbb{F}_q)^{kN}}{\uparrow}\begin{pmatrix}G_{in} & & \\ 0 & \ddots & 0\\ & & G_{in}\end{pmatrix}\underset{(\mathbb{F}_q)^{nN}}{\uparrow}$$

Propriety 14. *If C_{out} is $(N,J,D)_{q^k}$ and C_{in} is $(n,k,d)_q$, then $C_{out\circ in}$ is a $(Nn,Kk,Dd)_q$ code.*

Proof. Let $m \neq m' \in [q^k]^K$ with $\Delta\Big(C_{out}(m), C_{out}(m')\Big) \geq D$.

If $C_{out}(m)_i \neq C_{out}(m')_i$, then $\Delta\Big(C_{in}\big(C_{out}(m)_i\big), C_{in}\big(C_{out}(m')_i\big)\Big) \geq d$, so

$$\begin{aligned}\Delta\big(C_{in\,\circ\,out}(m), C_{in\,\circ\,out}(m')\big) &= \underbrace{\sum_{i\,:\,C_{out}(m)\neq C_{out}(m')}}_{D} \Delta\Big(C_{in}\big(C_{out}(m)_i\big), C_{in}\big(C_{out}(m')_i\big)\Big)\\ &\geq Dd\end{aligned}$$

□

To construct a good code it remains to find a good *inner* code. What have we gained ? → Inner code is “small”, so we can more easily find a good one.

Explicit construction Explicit here means can construct code in time polynomial in the blocklength.

We construct G_{in} and G_{out}:

- For G_{in}: RS code $[\big[N, \frac{N}{2}, \frac{N}{2}+1\big]_N$ with $N = 2^k$. G_{out} is a Vandermonde matrix, so we can construct G_{out} in $O(N^2)$ steps.
- For G_{in}: Should have dimension k. We construct a code achieving Gilbert-Varshamov bound (See homework).

For example, cam construct a parity check matrix for a code with parameters $[n = 2k, k, d = 0.1n]_2$. This algorithm takes $O(2^{2k}poly(k)$ steps $= O\big(N^2 poly(\log N)\big)$ So we can get $G_{out\,\circ\,in}$ in time $poly(N)$.

$C_{out\,\circ\,in}$ is a $\big[N \cdot 2\log N, \frac{N}{2}\log N, \left(\frac{N}{2}+1\right)(0.2\log N\big]_2$

Decoding a concatenated code $D_{C_{in}}$, $D_{C_{out}}$ for $(y_1, ..., y_n) \in (\mathbb{F}_q)^N$

$$D_{C_{out\,\circ\,in}}(y_1, ..., y_N) = D_{C_{out}}(D_{C_{in}}(y_1), ..., D_{C_{in}}(y_n))$$

Running time

$$N \cdot \underbrace{Cost(D_{C_{in}})}_{\substack{\text{generic decoder runs in}\\ O(2^{2k}poly(k))=O(N^2 poly(\log(N)))}} + \underbrace{Cost(D_{C_{out}})}_{\text{For RS, } O(N^3)}$$

Propriety 15. *The algorithm $D_{C_{out\,\circ\,in}}$ can correct $< \frac{Dd}{4}$ errors.*

Proof. Let m be such that

$$\Delta(C_{out\,\circ\,in(m)}, y) < \frac{dD}{4}$$

We want to show that we return m.

We define $B = \{i \in [N]\ D_{C_{in}}(y_i) \neq C_{out}(m)_i\}$.

- If $|B| < \frac{D}{2}$ then $D_{C_{out}}$ can correct the errors and returns m
- Otherwise, if $|B| \geq \frac{D}{2}$, if $i \in B$, $\Delta(y_i, C_{in}(C_{out}(m)_i)) \geq \frac{d}{2}$
  So $\Delta\Big((y_1, ..., y_N), C_{in}(C_{out}(m)_1)...C_{in}(C_{out}(m)_N)\Big) \geq \frac{Dd}{4}$ → contradiction

□

6.5 An application of ECC

Communication complexity

$$\underset{x\in\{0,1\}^n}{\text{Alice}} \qquad\qquad \underset{y\in\{0,1\}^n}{\text{Bob}}$$

Objective Compute $f(x, y)$ (the communication) with minimum communication

Example

- $PAR(x, y) = \sum_{i=1}^{n}(x_i + y_i) \mod 2$
  Alice sends parity $\sum_i x_i \mod 2$ to Bob, and bob computes $\left(\sum_i x_i\right) + \left(\sum_i y_i\right)$
- $EQ(x, y) = \begin{cases} 0 & \text{if } x \neq y \\ 1 & \text{if } x = y \end{cases}$
  Alice sends x to Bob, Bob compute $EQ(x, y)$ and sends back the result to Alice.

Definition 23 (Communication cost)**.**

$$\begin{aligned} Cost(\mathcal{P}) &= \textit{total number of bits communicated} \\ &= |a_1| + |b_1| + ... + |a_k| + |b_k| \qquad\qquad \text{(Worst case over all input } x, y) \end{aligned}$$

Definition 24 (Protocol (rough definition, not easy to formalized))**.** *A labelled tree where each node is either a Alice node or a Bob node, and each label of a node can only depend on the node of the latest communication and the other nodes of the same person.*

Definition 25 (Communication complexity of f)**.**

$$D(f) = \min_{\mathcal{P} \text{ protocol compting } f} Cost(\mathcal{P})$$

$Cost(\mathcal{P})$ is the maximum over all input x_{ij} of the number of bits communicated by applying $\mathcal{P}$ on inputs (x, y).

We have seen $D(PAR) \leq 2$ and $D(EQ) \leq n + 1$

Propriety 16.

$$D(EQ) \geq n + 1$$

Proof. We will only prove $D(EQ) \geq n$

Assume by contradiction that $\mathcal{P}$ computes EQ and $Cost(\mathcal{P}) \leq n - 1$. At most 2^{n-1} possible transcripts. So there exists two inputs (x, x) and (x', x') $(x \neq x')$ that lead to the same communication. Let us call the transcript

$$\Big(a_1\big(= A_1(x) = A_1(x')\big), b_1\big(= B_1(x_1, a_1) = B_1(x'_1, a_1)\big), a_2, b_2, ..., a_k, b_k\Big)$$

(for input (x, x) and (x', x')). Analyse the protocol on (x, x'): Alice communicates $A_1(x) = a_1$
Bob communicates $B_1(x', a_1) = b_1$

So the communication transcript is the same for (x, x'), so $\mathcal{P}$ outputs 1 on input (x, x'), but as $x \neq x'$, $EQ(x, x') = 0 \neq \mathcal{P}(x, x')$. □

Randomized protocol

Require For all inputs $\mathbb{P}(\mathcal{P}(x, y) \neq f(x, y)) \leq \epsilon$.

Warning The randomness is inside the protocol, not over the input (as for randomized algorithms)

Definition 26.

$$R_\epsilon(f) = \min_{\mathcal{P}:\mathbb{P}\{\mathcal{P}(x,y)\neq f(x_{ij}\}\leq\epsilon} Cost(\mathcal{P})$$

Propriety 17.

$$R_{1/3}(EQ) = O(\log n)$$

Remark Can reduce the error probability $\frac{1}{3}$ to very small by repeating

Proof.

- Alice choose $i \in [n]$ and sends (i, x_i) to Bob
- Bob:
  - if $x_i = y_i$ outputs 1
  - if $x_i \neq y_i$ outputs 0

If $x = y$, $\mathcal{P}(x, y) = 1$ (the protocol is always correct).

If $x \neq y$, $\mathbb{P}(\underbrace{\mathcal{P}(x,y) = 0}_{\text{of beeing correct}}) = \frac{\Delta(x,y)}{n}$

Idea: use error correcting code, e.g. take a $[3n, n, 2n+1]_q$-RS code C with $q = O(n)$.

- Alice choose $i \in \{1, ..., 3n\}$ at random and sends $(i, C(X)_i)$ where $C(X)_i$ is the i-th element of $C(X)$, which lives in $\mathbb{F}_q$
- Bob checks if $C(X)_i = C(y)_i$:
  - if $C(X)_i = C(Y)_i$ outputs 1
  - if $C(X)_i \neq C(Y)_i$ outputs 0

Communication cost:

$$\lceil \log 3n \rceil + \lceil \log q \rceil + 1 = O(\log n)$$

Correctness If $x = y$: outputs always 1 If $x \neq y$:

$$\mathbb{P}(\mathcal{P}(x,y) = 0) = \frac{|\{i : C(X)_i \neq C(y)_i\}|}{3n} \geq \frac{2n+1}{3n} \geq \frac{2}{3}$$

□

To go further: Polar Codes and Quantum Codes

BEC_α: binary erasure channel, $C(BEC_\alpha) = 1 - \alpha$ (the bits erase, i.e. become uninterpretable, with probability α).

Shanon's theorem: There exists a family $C_N : \{0,1,\}^K \to \{0,1\}^N$ of codes with blocklength N and dim K with $\frac{K}{N} \to 1-\alpha$ and decoders $Dec_N : \{0,1,?\}^N \to \{0,1\}^K$ with

$$\underset{s\in\{0,1\}^K}{\mathbb{P}}(Dec_n \circ BEC_\alpha^{\otimes N} \circ C_n(s) \neq s) \underset{N\to\infty}{\to} 0$$

Objective C_N and Dec_N polynomial-time (or even linear-time) in N

Concatenation approach Cut N into blocks of length $n = c\log N$. This gives a good code for blocklength n, exponential in $n \to$ polynomial in N.

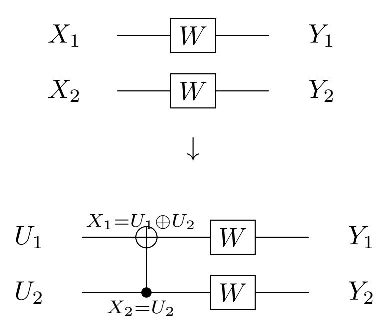
Only "in principle" construction

Polar codes (2007 Arilean)

Simple observation:
Two trivial cases:

- perfect channel $Y = X$
- useless channel Y independent of X

Polar code Transform 2 copies of W into W^- (worse channel) and W^+ (better channel). By repeated applications of this process, with N copies of W, we get N channels that are either perfect or useless.



Channel transformation W with input $\mathcal{X} = \{0,1\}$ and arbitrary output $\mathcal{Y}$. Define W^+ and W^-:

$$W^- : \{0,1\} \to \mathcal{Y}^2$$
$$W^+ : \{0,1\} \to \mathcal{Y}^2 \times \{0,1\}$$
$$W^-(y_1 y_2 | u_1) = \frac{1}{2} \sum_{u_2 \in \{0,1\}} W(y_1 \mid u_1 \oplus u_2) W(y_2 \mid u_2)$$
$$W^+(y_1 y_2 u_1 \mid u_2) = \frac{1}{2} W(y_1 \mid u_1 \oplus u_2).W(y_2 \mid u_2)$$
$$\left(\frac{1}{2} \to \text{Proba of } u_1\right)$$

Ex BEC_α

- W^-:
  - no erasures
    $$(Y_1, Y_2) = (U_1 \oplus U_2), U_2) \qquad \text{(With proba } (1-\alpha)^2)$$
    We can recover U_1
  - erasure on the first BEC
    $$(Y_1, Y_2) = (?, U_2) \qquad \text{(With proba } \alpha(1-\alpha))$$
    Independent of U_1
  - erasure on the second BEC
    $$(Y_1, Y_2) = (U_1 \oplus U_2, ?) \qquad \text{(With proba } (1-\alpha)\alpha)$$
    Independent of U_1
  - erasure on both
    $$(Y_1, Y_2) = (?, ?) \qquad \text{(With prob } \alpha^2)$$

  W^- is like an erasure channel with erasure probability $1 - (1-\alpha)^2$
- W^+:
  - no erasure
    $$(Y_1, Y_2, U_1) = (U_1 \oplus U_2, U_2, U_1)$$
    We can get U_2
  - erasure on the first bit
    $$(Y_1, Y_2, U_1) = (?, U_2, U_1$$
    We can get U_2
  - erasure on the second bit
    $$(Y_1, Y_2, U_1) = (U_1 \oplus U_2, ?, U_1)$$
    We can get U_2
  - erasure on both
    $$(Y_1, Y_2, U_1) = (?, ?, U_1)$$
    Independent of U_2

  W^+ is like an erasure channel with erasure probability α^2.

To formalize how good a channel is

$$I(W) = I(X:Y) \qquad \text{with } P_{XY}(x,y) = \frac{1}{2}W(y \mid x) \quad X \in \{0,1\}$$

Theorem 13. *Let W be a channel with binary input. Define (W^-, W^+), then*

$$\bullet\ \frac{1}{2}I(W^-) + \frac{1}{2}I(W^+) = I(W)$$
$$\bullet\ I(W^+) \geq I(W) \geq I(W^-)$$

with equality if and only if $I(W) \in \{0,1\}$

Proof.

$$\begin{aligned} I(W^-) &= I(U_1 : Y_1Y_2) \\ I(W^+) &= I(U_2 : Y_1Y_2U_1) \\ I(U_1 : Y_1Y_2) + I(U_2 : Y_1Y_1U_1) &= I(U_1 : Y_1Y_2) + \underbrace{I(U_2 : U_1)}_{=0} + I(U_2 : Y_1Y_2|U_1) \\ &= I(U_1U_2 : Y_1Y_2) \end{aligned}$$

But X_1X_2 is obtain from U_1U_2 by a bijection

$$\begin{aligned} I(U_1U_2 : Y_1Y_2) &= I(X_1X_2 : Y_1Y_2) \\ &= I(X_1 : Y_1) + I(X_2 : Y_2) && \text{(As } X_1 \text{ and } X_2 \text{ are independent)} \\ &= 2I(W) \end{aligned}$$

$$\begin{aligned} \bullet\ I(W^+) &= I(U_2 : Y_1Y_2U_1) \\ &\geq I(U_2 : Y_2) \\ &= I(X_2 : Y_2) \\ &= I(N) \end{aligned}$$

We have equality if and only if $I(U_2 : U_1Y_1|Y_2) = 0$, so $U_2 - Y_2 - U_1Y_1$ forms a Markov Chain ($P_{U_1Y_1|Y_2U_2} = P_{U_1Y_1|Y_2}$)
By a simple analysis, we can show that this happens if and only if $I(W) \in \{0,1\}$ □

Recursive step 2^n copies of $W \to W^{z_1 \dots z_n}$ for $(z_1, ..., z_n) \in \{+,-\}^n$
To use this for encoding

- Show polarization.

$$\frac{1}{2^n} \sum_{z_1 \dots z_n} I(W^{z_1 \dots z_n}) = I(W)$$

$$\mathcal{B} = \{(z_1, ..., z_n) : I(W^{z_1, \dots, z_n} = 0\}$$
$$\mathcal{G} = \{(z_1, ..., z_n) : I(W^{z_1, \dots, z_n}) \qquad = 0\} |G| = 2^n I(W)$$

Encoding msg $s \in \{0,1\}^{2^n C(W)}$

7 Introduction to Quantum Information

Important property assumed of information carrier: a bit has a finite number of distinguishable states.

The set of states is thus: Σ

In this lecture, $\Sigma = \{0,1\}$, 0 stands for "off", 1 for "on", as a physical carrier switch.

Assumption: A complete way to describe the switch is $s \in \Sigma$

Quantum theory: Superposition of different $s \in \Sigma$ is also allowed.

Representation of a quantum system: $\Sigma = \{0,1\}$

Warm up: probabilistic model

State of knowledge $= v = \begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix} \begin{array}{l} \to \text{proba of } 0 \\ \to \text{proba of } 1 \end{array}$

Remark The action of looking at the switch affects its state.

$$v = \begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix} \to \text{look at the system} \left.\begin{array}{l} \overset{\text{see } 0}{\to} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \overset{\text{see } 1}{\to} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{array}\right\} \text{updated state}$$

Quantum setting State: $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. It defines a *qubit*.

Two kinds of operations:

1. "Look at the switch" $\to$ measurement

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \overset{\text{measure}}{\to} \begin{array}{l} \text{See 0 with probability } |\alpha|^2 \to \begin{pmatrix} 1 \\ 0 \end{pmatrix} = e_0 \\ \text{See 1 with probability } |\beta|^2 \to \begin{pmatrix} 0 \\ 1 \end{pmatrix} = e_1 \end{array}$$

   An other way of seeing it is to interpret the state as a vector v. Then the probability of seeing 0 is $| < e_0, v > |^2$ and $| < e_1, v > |^2$ for 1.

2. "Transform without looking" $\to$ perform a unary $U \in \mathbb{C}^{2\times 2}$ with $U * U = I$, $U*$ being the conjugate transpose.

Example $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ $H = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$

Start in state "off" apply H, measure

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \to H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

We observe 0 and 1 with equal probability 1/2.

Example System in one of two states

$$v_0 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \qquad v_1 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

The objective is to determine whether the state is v_0 or v_1.

We apply H and then measure: $Hv_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $Hv_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

In general, if v and w are orthogonal, we can perfectly distinguish them.

How many bits can I store in a single qubit? We cannot store more than 2 bits, because v_0, v_1, v_2 would live in $\mathbb{C}^2$, so they cannot be orthogonal $\to$ cannot measure them perfectly.

Random access code

Idea Store 2 bits in 1 qubit with some error and only be interested in one of the two bits.

Definition 27. *A quantum random memory access encoding of 2 bits into 1 qubit with success probability p*

$$2 \overset{p}{\mapsto} 1$$

is a function

$$f : 0, 1^2 \to \mathbb{C}^2$$

such that for every $i \in \{1, 2\}$, there exists a U_i such that for all $x, y \in \{0, 1\}$

$$| < e_x, U_1 f(x, y) > |^2 \geq p$$

$$| < e_y, U_2 f(x, y) > |^2 \geq p$$

First limits of classical setting
$f : \{0, 1\}^2 \to \{0, 1\}$ (randomized)
$dec_i : \{0, 1\} \to \{0, 1\}$

One encoding $f(x, y) = x$
$\Rightarrow$ success probability $= 1/2$

Lemma 4. *No $2 \overset{p}{\mapsto} 1$ classical random access code exists with $p > 1/2$.*

Proof. $f : \{0, 1\} \to \{0, 1\}$
$dec_i : \{0, 1\} \to \{0, 1\}$ for $i \in \{1, 2\}$

Let $P^0 = (\mathbb{P}(dec_1(0) = 1), \mathbb{P}(dec_2(0) = 1)$ and $P^1 = (\mathbb{P}(dec_1(1) = 1), \mathbb{P}(dec_2(1) = 1)$

Fact: The line P^0P^1 cannot pass strictly inside every quadrant (division of $[0,1] \times [0,1]$ in four parts).

We take the quadrant that is missing, for example $(x, y) = (0, 0)$. If it is encoded as 0, then for the first bit, it recovers with good probability, but the second bit is output as 0 with a low probability (calculus). If (0,0) is encoded as 1, the reverse situation occurs.

More generally, for any mixture, decoding probabilities will be given by a point on the line P^0P^1, so one of the success probability is lower than 1/2. $\square$

Theorem 14. *There is a* $2 \overset{0.85}{\mapsto} 1$ *quantum random access code.*

Proof. We use $\frac{\pi}{8}$:

$$f(0,0) = \begin{pmatrix} \cos\frac{\pi}{8} \\ \sin\frac{\pi}{8} \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \cos\frac{\pi}{8} \\ -\sin\frac{\pi}{8} \end{pmatrix}$$

$$f(1,0) = \begin{pmatrix} \sin\frac{\pi}{8} \\ \cos\frac{\pi}{8} \end{pmatrix} \qquad f(1,1) = \begin{pmatrix} -\sin\frac{\pi}{8} \\ \cos\frac{\pi}{8} \end{pmatrix}$$

If $i = 1, U_1 = i$, the probability of decoding correctly is $\cos^2\frac{\pi}{8}$.

$\square$