Entropy and the Source Coding Theorem (§4)

Entropy of a random variable (RV) X with distribution P:

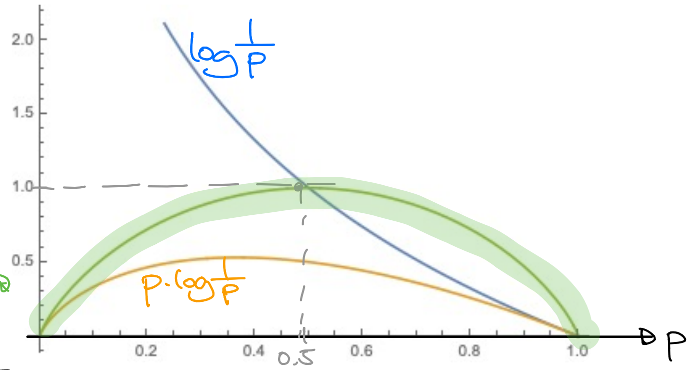
$$H(X) := H(P) := \sum_{x} P(x) \cdot \log \frac{1}{P(x)}$$

always base 2

Unit "bit"

$$0 \cdot \log \frac{1}{0} = 1$$

$$= E\left[\log \frac{1}{P(X)}\right]$$



e.g. $X \sim \mathrm{Bernoulli}(p)$: binary entropy

$$H(X) = -p \cdot \log(p) - (1-p)\log(1-p) \in [0,1]$$

Properties:

* $H(X) \geq 0$, $=$ iff constant $\quad p \cdot \log\frac{1}{p} \geq 0 \quad \forall p \in [0,1]$
* $H(X) \leq \log \#\{x : P(x) > 0\} \leq \log \#\mathcal{A}_X$

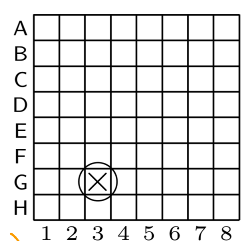
  $H(X) = \log \#\mathcal{A}_X \iff X$ uniform
* If X, Y independent: $H(XY) := H(X,Y) := H(X) + H(Y)$

  Pf: $\log \frac{1}{P(x,y)} = \log \frac{1}{P(x)} + \log \frac{1}{P(y)} \rightsquigarrow$ take expectation values. $\square$

Interpretation? $h(x) = h(X=x) = \log_2 \frac{1}{P(x)}$ is called information content of outcome $x \in \mathcal{A}_X$. Why? Three suggestive examples:

① Uniformly random number in $\{0, \dots, 255\}$: $H(X) = \log_2 256 = 8$ bit

② Poor man's submarine game: Single submarine hidden, other player asks if submarine in some square $\rightarrow$ hit/miss



1st move: $P(\text{hit}) = \frac{1}{64} \longrightarrow h(\text{hit}) = 6$ bit learned precise location (64 options)

$P(\text{miss}) = \frac{63}{64} \longrightarrow h(\text{miss}) \approx 0.0227$ bit learned little (63 remaining)

2nd move: $P(\text{miss}) = \frac{62}{63} \longrightarrow h(\text{miss}) \approx 0.0230$ bit
(if 1st missed)

after 32 misses: $\sum h(\text{miss}) = \log\frac{64}{63} + \dots + \log\frac{33}{32} = \log\frac{64}{32} = 1$ bit localized to 1/2 of squares

after 48 misses: $\sum h(\text{miss}) = \log\frac{64}{16} = 2$ bit localized to 1/4 of the squares

hit in 49th round: $h(\text{hit}) = \log\frac{1}{\frac{1}{16}} = 4$ bit $\longrightarrow \sum = 6$ bit $= H(\text{position})$

More generally: If we hit when n squares remaining

$$\sum h(\text{miss}) + h(\text{hit}) = \log \frac{64}{63} + \dots + \log \frac{n+1}{n} + \log \frac{n}{1} = \log 64 = 6 \text{ bit}$$

③ "Wenglish" has 2^{15} words in $\{A, \dots, Z\}^5$ s.th. frequence of single letters matches English. Let W be uniformly random word in this list.

$H(W) = 15$ bit, i.e. on average 3 bit/letter

but e.g. $p(W_1 = Z) = 0.1\% \Rightarrow h(W_1 = Z) \approx 10$ bit $\leftarrow$ no contradiction; we learn less info from the rest since few words start with Z

$\ll \frac{1}{26}$

Compression

Consider a data source modeled by a RV X. WANT:

$$X \longrightarrow \boxed{C} \longrightarrow \{0,1\}^\ell \longrightarrow \boxed{D} \longrightarrow \hat{X} = X$$

Raw bit content: $H_0(X) := H_0(P) := \log \#\{x : P(x) > 0\}$ $\leftarrow$ The book uses $\log \#\mathcal{A}_X$

* Can compress X into ℓ bits $\Leftrightarrow \ell \geq H_0(X)$

  Pf: Need one distinct bitstring for each possible outcome, i.e. $\#\{0,1\}^\ell \geq \#\{x : P(x) > 0\}$ $\square$

* $0 \leq H(X) \leq H_0(X) \leq \log \#\mathcal{A}_X$ (see above)

Lossy COMPRESSION: What if we allow small probability of error? $P_r(\hat{X} \neq X) \leq \delta$?

Need $S \subseteq \mathcal{A}_X$ s.th. $P_r(X \notin S) \leq \delta$.
Such an S is called δ-sufficient. Define:

δ-essential bit content: log

$$H_\delta(X) := H_\delta(P) := \min\{\#S : S \text{ is } \delta\text{-sufficient}\}$$

$\hookrightarrow$ Can compress X into ℓ bits with error probability $\leq \delta \Leftrightarrow \ell \geq H_\delta(X)$

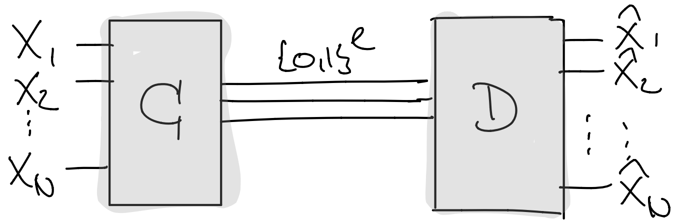
ex:

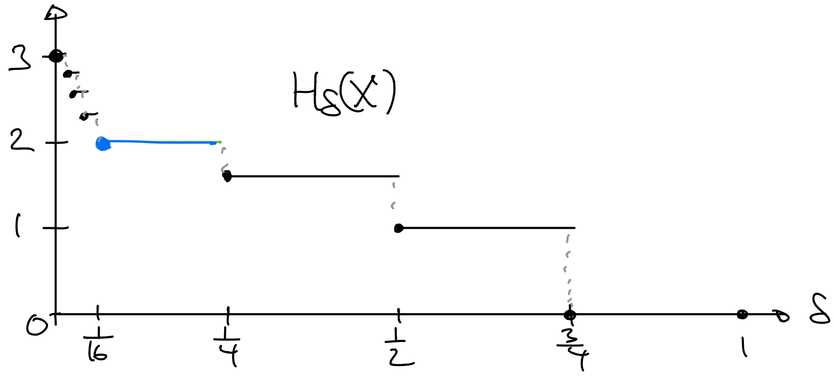
x	$P(x)$	$\delta = 0$	$\delta = 1/16$
a	1/4	000	00
b	1/4	001	01
c	1/4	010	10
d	3/16	011	11
e	1/64	100	—
f	1/64	101	—
g	1/64	110	—
h	1/64	111	—

$H_\delta(X)$ is in general quite a messy function... $\rightarrow$

Clear how to do it???

What if we compress blocks of symbols $X_1, X_2, \dots, X_N \overset{IID}{\sim} P$?





s.th. $\Pr(\hat{X}^N = X^N) \geq 1-\delta$?

NOTATION: $X^N = (X_1, \dots, X_N) = X_1 \cdots X_N$

We "know" the answer: $H_\delta(X^N)$! But how to compute...?

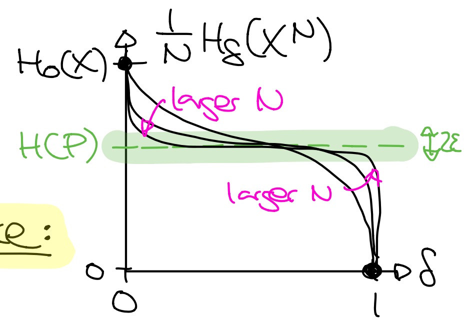
Shannon's Source Coding Theorem: Let $X_1, X_2, X_3, \dots \overset{IID}{\sim} P$ and $0<\delta<1$:

$$\lim_{N\to\infty} \frac{H_\delta(X^N)}{N} = H(P)$$

← RHS is independent of δ!!!

#bits/symbol = COMPRESSION RATE for error δ

(i.e. $\forall \delta \in (0,1), \varepsilon > 0\ \exists N_0\ \forall N \geq N_0 : \left| \frac{H_\delta(X^N)}{N} - H(P) \right| \leq \varepsilon$)



Thus: H(P) is "optimal" compression rate for an IID source:
(independent of $0<\delta<1$!!!)

* If $R > H(P)$: $\exists N_0\ \forall N \geq N_0$: CAN compress at rate R (= into $\ell \leq RN$ bits)
* If $R < H(P)$: $\exists N_0\ \forall N \geq N_0$: CANNOT compress at rate R

Why should this be true? For "typical" samples $x^N = x_1 \cdots x_N$:

$$\#\{k : x_k = x\} \sim N \cdot P(x) \Longrightarrow P(x^N) = P(x_1) \cdots P(x_N) \sim \prod_x P(x)^{N \cdot P(x)}$$

$$\Longrightarrow \frac{1}{N} \log \frac{1}{P(x^N)} = \frac{1}{N} \sum_{k=1}^{N} \log \frac{1}{P(x_k)} \approx H(P)$$

Let's try to formalize this:

Typical set: $T_{N,\varepsilon} = \left\{ x^N \in \mathcal{A}_X^N : \left| \frac{1}{N} \log \frac{1}{P(x^N)} - H(P) \right| \leq \varepsilon \right\}$

... to be continued ...