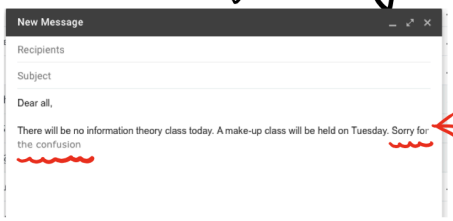
2: Uncertainty, compression, and entropy ①

[A] Source Model

With probability as our means for modeling uncertainty, we can now return to our goal of answering "how much information is revealed when a question is answered."

Recall the following example:



for the confusion
for the inconvenience
for the confusion

Gmail sever can simply assume that this unknown phrase was generated by a probability distribution. We call the generator of this phrase (the user) a source which is captured by a pmf P on $\mathcal{X}$.

- We may treat this entire phrase as only symbol $X \sim P$ on $\mathcal{X}$.
- Or, we can treat each word as a symbol and the source generates a sequence of symbols.

* Memoryless Source: The sequence $\{X_t\}_{t=1}^{\infty}$ is output of a memoryless source if it is independent & identically distributed

* Markov source of order m: Given $(X_{t-m}, \ldots, X_{t-1})$, the random variable X_t is independent of $\{X_{t-j}\}_{j>m}$.

This implies $p(x_1, \ldots, x_n) = \prod_{t=1}^{n} p(x_t | x_1, \ldots, x_{t-1})$

$= \prod_{t=1}^{n} p^{(m)}(x_t | x_{t-m}, \ldots, x_{t-1})$

which is a fixed conditional prob called the transition matrix

→ Now that we know how to model uncertainty, how do we measure it?

Thesis: Uncertainty of a source equals the number of bits used to represent its output

avg. no. of bits or "likely" no. of bits ←

(both are roughly the same by estimates we saw in the previous lecture)

B An Example

For our running example, suppose the gmail server goes over all my composed emails and finds that the first phrase follows similar expression $1/2$ the times, the next $1/4$ times, the next $1/8$, $\frac{1}{16}$, $\frac{1}{32}$ times. Overall gmail found 256 possible phrases.

$P = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \underbrace{\ldots}_{\text{all the remaining possibilities are equally likely}}\right)$

Then, how do we store these phrases? (3)

* Make a look-up table and just store the entry number:

Record #	Phrase
1	phrase 1
2	phrase 2
3	phrase 3

- Uses 1 byte to represent the phrase.

* Why just one table? Why not two?

→ First use 1 bit to represent which table and then represent the entry in the table:

Table 1 ← $\{\frac{1}{2}, \frac{1}{4}\}$; Table 2 ← $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \dots\}$

1 bit 8 bits

Worst-case requirement is now ____

But the average # of bits = $1 + \frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 8$

$= 3.75$ bits

- Maybe we can try 2 bits for the first table:

Table 1 ← $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$

⇒ avg. no. of bits $= 1 + \frac{15}{16} \cdot 2 + \frac{1}{16} \cdot 8 = \frac{39}{16} \approx 2.44$ bits

- Maybe 3 bits for table 1: avg. no. of bits ≈ 4.15 bits

(we can convince ourselves that 2.44 bits is the best)

* Why just two tables ?

Lets make multiple tables. We should make at least 4 tables.

00 :	Table 1 ← $\{1/2, 1/4\}$	1 bit	
01 :	Table 2 ← $\{1/8, 1/16\}$	1 bit	$\Rightarrow$ avg. no. of bits $= 2 + \frac{3}{4}.1 + \frac{3}{16}.1 + \frac{1}{32}.8$
10 :	Table 3 ← $\{\frac{1}{32}\}$	0 bit	
11 :	Table 4 ← {251 symbols}	8 bits	≈ 3.25

Easy to improve : put $1/2$ in Table 1 alone.

$$\Rightarrow \text{avg. no. of bits} = 2 + \frac{1}{2}.0 + \frac{3}{8}.1 + \frac{3}{32}.1 + \frac{8}{32}$$

$$\approx 2.72$$

Another view: Tables are "super-symbols" with prob. equal to sum of prob. of all symbols inside it. This view leads to Huffman code

Table 1 ← $\{\frac{1}{2}$, Table 2 $\}$ $\{\frac{1}{4}$, Table 3 $\}$ $\{\frac{1}{8}$, Table 4 $\}$

$\{\frac{1}{32}$, Table 6 $\}$ $\{\frac{1}{16}$, Table 5 $\}$

$$\Rightarrow \text{avg. no. of bits} = \frac{1}{2}.1 + \frac{1}{4}.2 + \frac{1}{8}.3 + \frac{1}{16}.4 + \frac{1}{32}.5 + \frac{1}{32}.8$$

$$\approx 2.03$$

Yet another view:

? ← 0, ? ← 1, ? ← 00, ? ← 01, ? ← 11, ← 8 bit sequences

$1/2$ $1/4$ $1/8$ $1/16$ $1/32$ remaining symbols

$$\Rightarrow \text{avg. no. of bits} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8}.2 + \frac{1}{16}.2 + \frac{1}{32}.2 + \frac{1}{32}.8$$

$$\approx 1.44 \text{ bits}$$

Where does this improvement end ?

A few lessons

(a) The two-table example is instructive: In some applications we can treat the no. of bits used for the first table as the effective length of the final representation, if symbols in Table 2 appear with very small prob.

(b) The final scheme we presented is not practical, but can serve as a tool for characterizing a measure of uncertainty.

C A Basic Compression Problem

The first remark above motivates the following quantity:

$$L_\varepsilon(P) = \min \{ \lceil \log_2 |A| \rceil : A \subset \mathcal{X} \text{ s.t. } P(A) \geq 1-\varepsilon \}$$

"fixed-length code" $\equiv$ min no. of bits required to represent a set of probability $\geq 1-\varepsilon$

$\equiv$ min no. of bits required to represent Table 1

The second remark motivates the next definition:

$$\bar{L}(P) = \min \{ \sum_x p(x) |e(x)| : e: \mathcal{X} \to \{0,1\}^* \text{ is } 1\text{-}1 \}$$

"variable-length code" $\equiv$ min avg length of a code which represents $\mathcal{X}$ as bits

Note that both these quantities have been defined as mathematical entities, motivated by our search for a measure of uncertainty, and are devoid of any practical constraints. When we revisit compression later, we will consider practical schemes for compression.

Building on our scheme for the example we saw, we can identify the optimal schemes for both $L_\varepsilon(P)$ and $\bar{L}(P)$.

→ Simply arrange the probabilities in a decreasing order and assign sequences of increasing length to the symbols.

Specifically, let $p_{(1)} \geq p_{(2)} \geq \dots \geq p_{(k)}$ denote the probabilities of the pmf P arranged in a decreasing order, with (i) denoting a symbol with the i^{th} highest probability.

Theorem

(a) $$L_\varepsilon(P) = \lceil \log N_\varepsilon(P) \rceil$$

where $$N_\varepsilon(P) = \min\left\{ \ell : \sum_{i=1}^{\ell} p_{(i)} \geq 1-\varepsilon \right\}.$$

(b) $$\bar{L}(P) = \sum_{i=1}^{k} p_{(i)} \underbrace{\left\lceil \log\left(\frac{i}{2}+1\right)\right\rceil}_{= \min\left\{ j \in \mathbb{N} : \sum_{\ell=1}^{j} 2^\ell \geq i \right\}}$$

Proof: (a) Clearly, $L_\varepsilon(P) \leq \lceil \log N_\varepsilon(P) \rceil$. For the other direction, consider a set $A \subset \mathcal{X}$ with $P(A) \geq 1-\varepsilon$. Note that

$$1-\varepsilon \leq \sum_{i=1}^{|A|} p_i \leq \sum_{i=1}^{|A|} p_{(i)} \Rightarrow |A| \geq N_\varepsilon(P),$$

which implies $L_\varepsilon(P) \geq \lceil \log N_\varepsilon(P) \rceil$.

(b) Again, $\bar{L}(P) \leq \sum_{i=1}^{k} p_{(i)} \lceil \log\left(\frac{i}{2}+1\right)\rceil$ follows by using the scheme which simply assigns the shortest available binary sequence to symbols in the decreasing order of their probabilities.

For the other direction, for any other scheme that assigns different binary sequences to each symbol, let $\ell(i)$ denote the length of the sequence assigned to $p_{(i)}$. We show that

$$\sum_{i=1}^{t} p_{(i)}\,\ell(i) \geq \sum_{i=1}^{t} p_{(i)} \left\lceil \log\left(\frac{i}{2}+1\right)\right\rceil \quad \text{for all} \quad 1 \leq t \leq k.$$

Clearly, the claim holds for $t=1$. Suppose it holds for a $t \in \mathbb{N}$.

Then,
$$\sum_{i=1}^{t+1} p_{(i)}\,\ell(i) = \sum_{i=1}^{t} p_{(i)}\,\ell(i) + p_{(t+1)}\,\ell(t+1)$$
$$\geq \sum_{i=1}^{t} p_{(i)} \left\lceil \log\left(\frac{i}{2}+1\right)\right\rceil + p_{(t+1)}\,\ell(t+1)$$

[by induction hypothesis]

Namely, the induction hypothesis claims that it is optimal to have the first t symbols represented by $\{0, 1, 00, 01, 11, 10, \ldots\}$. We restrict to such schemes now. But then, if we represent $(t+1)$ with the $(t+1)^{\text{th}}$ shortest binary sequence, we will only improve the performance. Therefore,

$$\sum_{i=1}^{t+1} p_{(i)}\,\ell(i) \geq \sum_{i=1}^{t+1} p_{(i)} \left\lceil \log\left(\frac{i}{2}+1\right)\right\rceil.$$

(Alternatively, we can use induction to show that the optimal scheme with represent the symbols $(1), \ldots, (t)$ using the t-shortest binary sequences. ∎

Exercise: Show that

$$\frac{L(P)}{1-\varepsilon} - \frac{\varepsilon \lceil \log |\mathcal{X}| \rceil}{1-\varepsilon} \leq L_\varepsilon(P) \leq \frac{L(P)}{\varepsilon} + 1.$$

D Enter Shannon Entropy

The bounds we develop in the exercise above suggest (roughly) that the difference between $L_\varepsilon(P)$ and $\bar{L}(P)$ depends on ε, but both are fundamentally the same quantity.

So, we will simply focus on $L_\varepsilon(P)$ and treat it as our measure of uncertainty.

The expression we have identified for $L_\varepsilon(P)$ doesn't look very useful. For instance, it has an ε in-built in it but we seek a fundamental notion of uncertainty that does not depend on such external parameters.

We now find an approximate upper bound that is not as accurate as the previous formula, but brings-out a new feature of P that measures its uncertainty.

→ Instead of selecting the most likely symbols, we simply retain the symbols that are more likely than a threshold.

Let $A_\lambda = \{x : P(x) \geq 2^{\lambda}\}$

$= \{x : -\log p(x) \leq \lambda\}$

another interpretation: Keep symbols that are not too rare, with rarity measured as $-\log p(x)$

Exercise: $|A_\lambda| \leq \lambda$

Theorem. If for $\lambda > 0$ we have

$$P\left(\{x : -\log p(x) \leq \lambda\}\right) \geq 1-\varepsilon.$$

Then,

$$\boxed{L_\varepsilon(P) \leq \lambda}$$

In words: A $(1-\varepsilon)$-prob. upper bound on the random variable $Z = -\log p(X)$ is an upper bound for $L_\varepsilon(P)$

And what is our estimate for a large prob. upper bound for a random variable? Markov ineq. gives an estimate of $\mathbb{E}[Z]$ since $P\left(Z \leq \frac{\mathbb{E}[Z]}{\varepsilon}\right) \geq 1-\varepsilon$.

Thus, by the Theorem above,

$$L_\varepsilon(P) \leq \frac{\mathbb{E}[-\log p(X)]}{\varepsilon}$$

We have discovered our fundamental measure of uncertainty!

$$H(P) := \sum_x p(x) \log \frac{1}{p(x)} \quad : \text{Shannon Entropy}$$

Exercise (i) Compute entropy for P given by

(a) $\left(\frac{1}{1024}, \ldots, \frac{1}{1024}\right)$ (b) $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{1024}, \ldots, \frac{1}{1024}\right)$ (c) $\left(\frac{3}{4}, \frac{1}{1024}, \ldots, \frac{1}{1024}\right)$

(ii) Plot entropy of $\text{Ber}(p)$ as a function of p.

E Random hash

Upto now, we have seen that using P we can form a list of size $\approx 2^{H(P)}$ which contains the unknown $X \sim P$ with high probability.

In our question/answer analogy, the person knowing the answer can reveal the answer to the guesser using $\approx H(P)$ bits, if he knows the list. We used this observation to justify that the information revealed by the answerer is $\approx H(P)$. This may not be satisfactory since there seems to be some hidden information (in form of the guesslist) shared between the guesser and the answerer.

We now show that the same result holds even when the answerer does not know the list. Specifically, we show:

The answerer can send $\ell \approx H(P)$ bits about the answer, without knowing P, which will allow the guesser to get the answer with high probability.

→ The tool we use is a random hash.

Definition (ℓ-bit random hash) Consider a random function $F: \mathcal{X} \to \{0,1\}^\ell$ chosen uniformly over the set of all functions

from $\mathcal{X}$ to $\{0,1\}^\ell$. The output $F(x)$ is called the ℓ-bit random hash of the input x.

Lemma (Collision-resistance property) For a random function F as above and every distinct $x, x' \in \mathcal{X}$, we have.

$$\mathbb{P}\left(F(x) = F(x')\right) = 2^{-\ell}.$$

Proof. A simple exercise.

Thus, a random hash is an "almost" one-one mapping.

Suppose now that the guesser has a list of A such that $X \in A$. More generally, assume

$$\mathbb{P}(X \in A) \geq 1 - \varepsilon.$$

The answerer can simply send $F(X)$ and the guesser can declare the answer $\hat{X}$ as that $x \in \mathcal{L}$ for which $F(x) = F(X)$. (If there are multiple, use any one)

Then,

$$\begin{aligned}\mathbb{P}(\hat{X} \neq X) &= \mathbb{P}(\hat{X} \neq X,\ X \in A) + \mathbb{P}(\hat{X} \neq X,\ X \in A^c) \\ &\leq \mathbb{P}(\hat{X} \neq X \mid X \in A) + \mathbb{P}(X \in A^c) \\ &\leq \mathbb{P}(\hat{X} \neq X \mid X \in A) + \varepsilon\end{aligned}$$

Also,

$$\begin{aligned}\mathbb{P}(\hat{X} \neq X \mid X \in A) &= \sum_{x \in A} \frac{P(x)}{P(A)} \cdot \sum_{x' \in A:\, x' \neq x} \mathbb{P}(F(x) = F(x')) \\ &\leq |A| \cdot 2^{-\ell}.\end{aligned}$$

Thus,

$$\mathbb{P}(\hat{X} \neq X) \leq |A| 2^{-\ell} + \varepsilon$$

If we choose s.t. $|A| \leq 2^{L_\varepsilon(P)}$, namely the small A s.t. $P(A) \geq 1-\varepsilon$, then

$$\mathbb{P}(\hat{X} \neq X) \leq 2^{L_\varepsilon(P)-\ell} + \varepsilon \leq \eta + \varepsilon$$

if $\ell \geq L_\varepsilon(P) + \frac{1}{\eta}$.

In particular, we can choose $\ell \gtrsim H(P)$ to get a small prob. of error in view of the results of the last section.