ORIE 4742 - Info Theory and Bayesian ML

Lecture 4: Source Coding

February 4, 2020

Sid Banerjee, ORIE, Cornell

entropy and information

rv X taking values $\mathcal{X} = \{a_1, a_2, \ldots, a_k\}$, with pmf $\mathbb{P}[X = a_i] = p_i$

Shannon's entropy function

- outcome $X = a_i$ has *information content*: $h(a_i) = \log_2 \left(\frac{1}{p_i}\right)$
- random variable X has *entropy*: $H(X) = \mathbb{E}[h(X)] = \sum_{i=1}^{k} p_i \log_2 \left(\frac{1}{p_i}\right)$

- only depends on distribution of X (i.e., $H(X) = H(p_1, p_2, \ldots, p_k)$)
- $H(X) \geq 0$ for all X
- if $X \perp\!\!\!\perp Y$, then $H(X, Y) = H(X) + H(Y)$
  where joint entropy $H(X, Y) \triangleq \sum_{(x,y)} p(x, y) \log_2 1/p(x, y)$
- if $X \sim$ uniform on $\mathcal{X}$, then $H(X) = \log_2 |\mathcal{X}|$; else, $H(X) \leq \log_2 |\mathcal{X}|$

source coding

the source coding problem

suppose we are given a database $D = (X_1 X_2 \dots X_N)$, where each X_i is a letter in an alphabet $\mathcal{X}$, generated iid according to $X_i \sim \{p_1, p_2, \dots, p_k\}$

the source coding problem

suppose we are given a database $D = (X_1 X_2 \dots X_N)$, where each X_i is a letter in an alphabet $\mathcal{X}$, generated iid according to $X_i \sim \{p_1, p_2, \dots, p_k\}$

lossless compression

compress every database D into a *codeword* $L = \phi(D)$ such that we can exactly recover $D = \phi^{-1}(L)$

δ-lossy compression $L = \phi(D)$ defined only for $D \in \mathcal{S}_\delta$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$

the source coding problem

suppose we are given a database $D = (X_1 X_2 \dots X_N)$, where each X_i is a letter in an alphabet $\mathcal{X}$, generated iid according to $X_i \sim \{p_1, p_2, \dots, p_k\}$

lossless compression

compress every database D into a *codeword* $L = \phi(D)$ such that we can exactly recover $D = \phi^{-1}(L)$

Shannon's source coding theorem

if X has entropy $H(X)$, then can compress $D = (X_1 X_2 \dots X_n)$ into a codeword $L = \phi(D)$ of expected size $|L| = n\ell$ bits, such that

$$H(X) \leq \ell < H(X) + \frac{1}{n}$$

moreover, no lossless encoder ϕ has expected codeword size $< nH(X)$

Mackay's bent coin lottery

A coin with $p_1 = 0.1$ will be tossed $N = 1000$ times.

The outcome is $\mathbf{x} = x_1 x_2 \ldots x_N$.

e.g., $\mathbf{x} =$ 000001001000100...00010

You can buy any of the 2^N possible tickets for £1 each, before the coin-tossing.

If you own ticket $\mathbf{x}$, you win £1,000,000,000.

Q To have a 99% chance of winning, at lowest possible cost, which tickets would you buy?

- And how many tickets is that?

Express your answer in the form $2^{(\cdots)}$.

Lottery tickets available

2^N

0000000000.....00000
0000000000.....00001
0000000000.....00010
0000000000.....00011
0000000000.....00100
0000000000.....00101
0000000000.....00110
0000000000.....00111
⋮
0010000001.....01000
⋮
1111111111.....11101
1111111111.....11110
1111111111.....11111

Mackay's bent coin lottery: warmup

what if you could buy only one ticket?

Mackay's bent coin lottery: warmup

what if you could buy k tickets?

recall: two useful facts

- counting via binary entropy for $N \in \mathbb{N}$, $k \leq N$: $\binom{N}{k} \approx 2^{NH_2(k/N)}$
- Chebyshev's inequality for any rv. X with mean $\mathbb{E}[X]$, finite variance $\sigma^2 > 0$, and any $k > 0$: $\mathbb{P}\left[|X - \mathbb{E}[X]| \geq k\sigma\right] \leq \frac{1}{k^2}$

Mackay's bent coin lottery: solution

(lossy) source coding theorem for binary sources

given $X^N = (X_1 X_2 \dots X_N)$, where each $X_i \sim$Bernoulli(p)

δ-lossy compression

$L = \phi(X^N)$ defined only for $X^N \in \mathcal{S}_\delta$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$

(lossy) source coding theorem for binary sources

given $X^N = (X_1 X_2 \ldots X_N)$, where each $X_i \sim$Bernoulli(p)

δ-lossy compression

$L = \phi(X^N)$ defined only for $X^N \in \mathcal{S}_\delta$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$

- δ-sufficient subset S_δ: smallest subset of $\{0,1\}^N$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$
- essential information content in X^N: $H_\delta(X^N) \triangleq \log_2 |S_\delta|$

(lossy) source coding theorem for binary sources

given $X^N = (X_1 X_2 \dots X_N)$, where each $X_i \sim$Bernoulli(p)

δ-lossy compression

$L = \phi(X^N)$ defined only for $X^N \in \mathcal{S}_\delta$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$

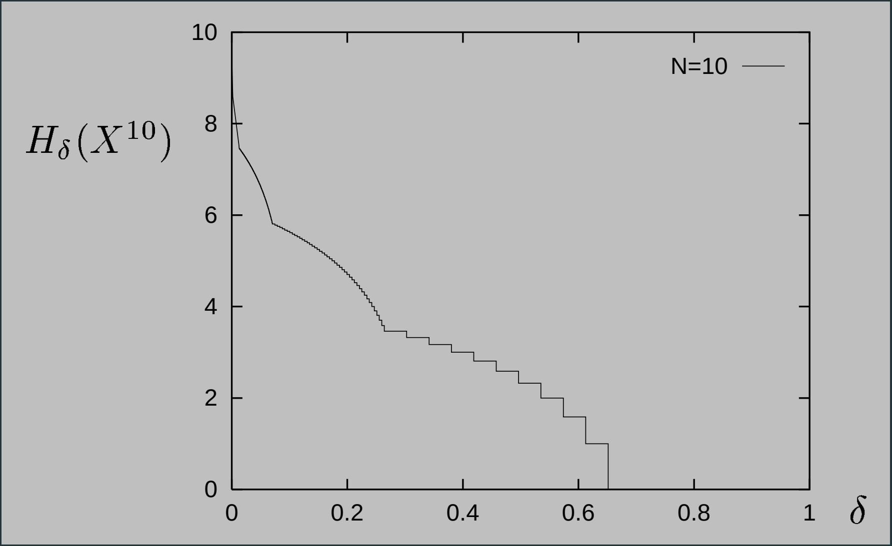
- δ-sufficient subset S_δ: smallest subset of $\{0,1\}^N$ s.t. $\mathbb{P}[\mathcal{S}_\delta] \geq 1 - \delta$
- essential information content in X^N: $H_\delta(X^N) \triangleq \log_2 |S_\delta|$

Shannon's source coding theorem (lossy version)

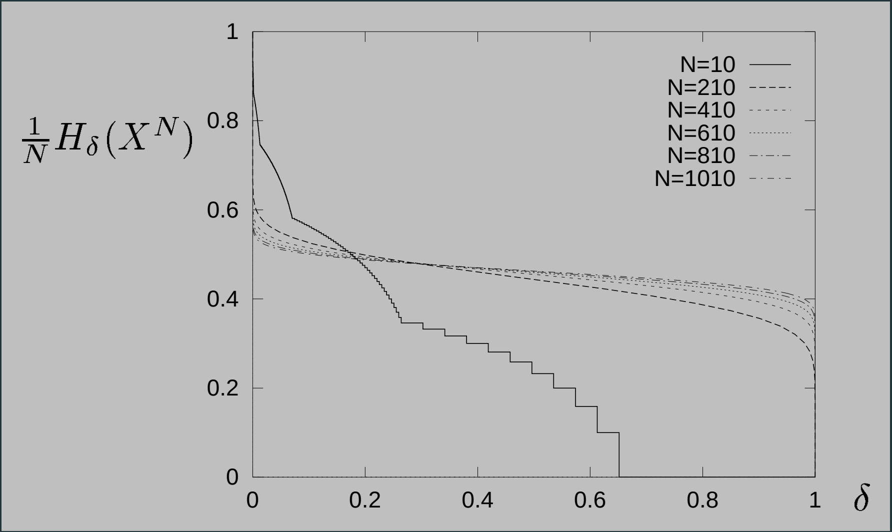
if X has entropy $H(X)$, then for any $\epsilon > 0$ and $0 < \delta < 1$, there exists N_0 s.t. for all $N > N_0$, we have

$$\left| \frac{H_\delta(X^N)}{N} - H(X) \right| \leq \epsilon$$

(lossy) source coding for binary sources: intuition



(lossy) source coding for binary sources: intuition



lossless source coding

from lossy to lossless compression

given $X^N = (X_1 X_2 \ldots X_N)$, where each $X_i \sim$Bernoulli(p)

from lossy to lossless compression

given $X^N = (X_1 X_2 \dots X_N)$, where each $X_i \sim$Bernoulli(p)

Shannon's source coding theorem

if X has entropy $H(X)$, then for any $\epsilon > 0$ and $0 < \delta < 1$, there exists N_0 s.t. for all $N > N_0$, we have a lossless code $L = \phi(X^N)$ s.t.

$$\left| \frac{\mathbb{E}[L]}{N} - H(X) \right| \leq \epsilon$$

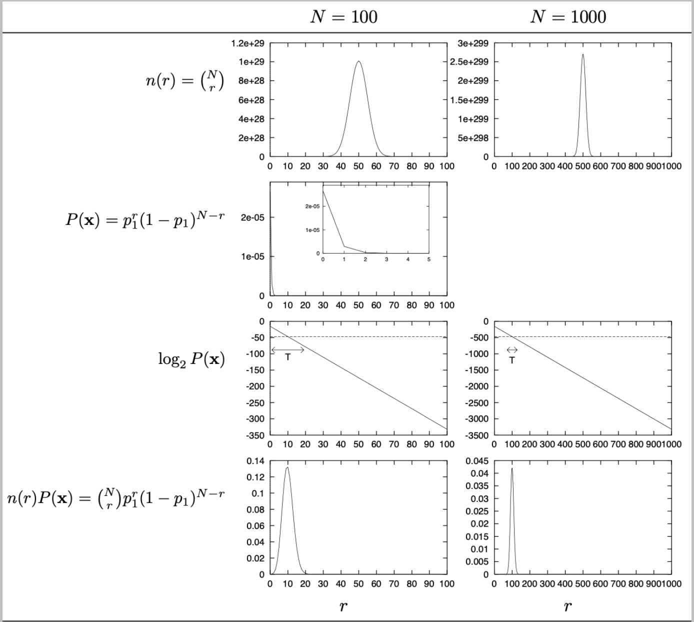
lossless compression via typical set encoding

typical set

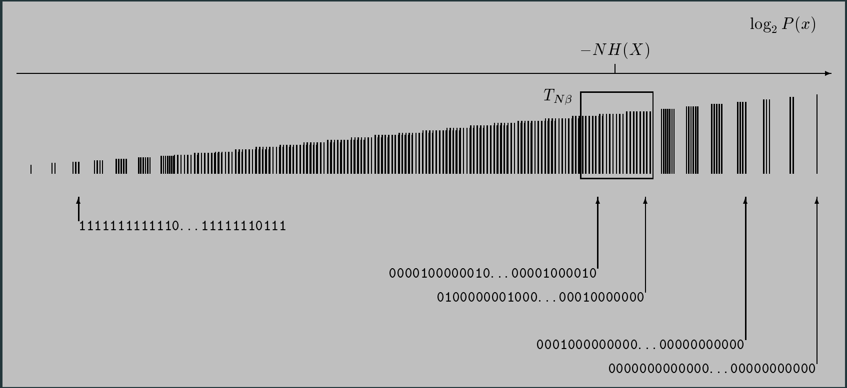
iid source produces $X^N = (X_1 X_2 \ldots X_n)$; each $X_i \in \mathcal{X}$ has entropy $H(X)$

then X^N is very likely to be one of $\approx 2^{NH(X)}$ typical strings,
all of which have probability $\approx 2^{NH(X)}$

visualizing the typical set



visualizing ‘asymptotic equipartition’



practical source coding solutions

symbol codes

$$X_1 X_2 \ldots X_n \quad \rightarrow \quad \phi(X_1)\phi(X_2)\ldots\phi(X_n)$$

stream codes

$$X_1 X_2 \ldots X_n \quad \rightarrow \quad \phi(X_1)\phi(X_2|X_1)\phi(X_3|X_1X_2)\ldots\phi(X_n|X_1X_2\ldots X_{n-1})$$

symbol codes (Mackay chapter 5)

symbol codes

expected length of symbol code

let $X \sim \{p(x)\}_{x \in \mathcal{X}}$, and consider code $C(\cdot)$, and let $\ell(x) = |C(x)|$

the expected length of C is $\mathbb{E}[L(C, X)] = \sum_x p(x)\ell(x)$

what we want from symbol code C:

- unique decodability: $\forall\, x_1 x_2 \dots x_n \neq y_1 y_2 \dots y_n$, we have $C(x_1)C(x_2)\dots C(x_n) \neq C(y_1)C(y_2)\dots C(y_n)$
- easy to decode
- small $\mathbb{E}[L(C, X)]$

types of symbol codes

consider source producing $X \sim \{a, b, c, d\}$ with prob $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$

prefix codes

the limits of unique decodability

Kraft-McMillan inequality

for any $C \equiv$ uniquely decodable binary code over $\mathcal{X}$, with $\ell(x) = |C(x)|$

$$\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$$

moreover, for any $\{\ell(x)\}$ satisfying this, we can find a prefix code

the limits of unique decodability

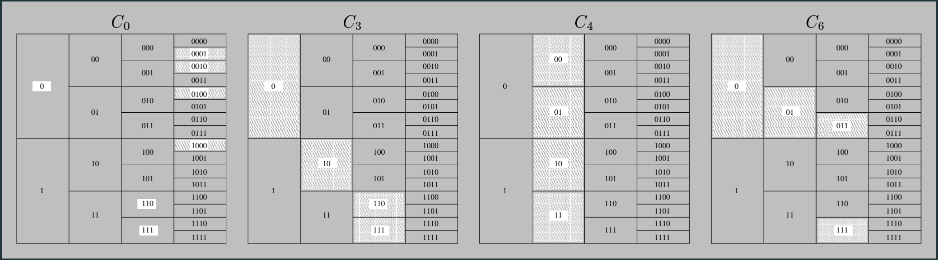
Kraft's inequality: for prefix codes $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$

Kraft's symbol-code supermarket

Kraft's inequality: for prefix codes $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</td><td rowspan="16">The total symbol code budget</td></tr>
<tr><td>0001</td></tr>
<tr><td rowspan="2">001</td><td>0010</td></tr>
<tr><td>0011</td></tr>
<tr><td rowspan="4">01</td><td rowspan="2">010</td><td>0100</td></tr>
<tr><td>0101</td></tr>
<tr><td rowspan="2">011</td><td>0110</td></tr>
<tr><td>0111</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">10</td><td rowspan="2">100</td><td>1000</td></tr>
<tr><td>1001</td></tr>
<tr><td rowspan="2">101</td><td>1010</td></tr>
<tr><td>1011</td></tr>
<tr><td rowspan="4">11</td><td rowspan="2">110</td><td>1100</td></tr>
<tr><td>1101</td></tr>
<tr><td rowspan="2">111</td><td>1110</td></tr>
<tr><td>1111</td></tr>
</table>

Kraft's symbol-code supermarket



optimizing expected code length

- entropy of X: $H(X) = \sum_{i \in \mathcal{X}} p_i \log_2 \left(\frac{1}{p_i}\right)$
- Kraft-McMillan inequality: UD code $\{\ell_i\}_{\{i \in \mathcal{X}\}}$ satisfies $\sum_{i \in \mathcal{X}} 2^{-\ell_i} \leq 1$

optimizing expected code length

let $X \sim \{p(x)\}_{x \in \mathcal{X}}$, and consider code $C(\cdot)$, and let $\ell(x) = |C(x)|$

the expected length of C is $\mathbb{E}[L(C, X)] = \sum_x p(x)\ell(x)$

relative entropy and Gibb's inequality

relative entropy (or Kullback-Leibler (KL) divergence)

the relative entropy $D_{KL}(p||q)$ between two distributions $p(x)$ and $q(x)$ defined over alphabet $\mathcal{X}$ is

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} p(x) \ln \left(\frac{p(x)}{q(x)} \right)$$

the function $\phi(x) = x \ln x$

relative entropy and Gibb's inequality

the relative entropy $D_{KL}(p||q) = \sum_{x \in \mathcal{X}} p(x) \ln \left(\frac{p(x)}{q(x)} \right) \geq 0$ for all p, q

optimizing expected code length

optimizing expected code length

aside: cross entropy

the cross entropy of p given q: $H_p(q) = \sum_{x \in \mathcal{X}} p(x) \ln\left(\frac{1}{q(x)}\right)$
– avg length of message from if 'p mis-estimated as q'

how good is the best symbol code?

Huffman code

consider $X \sim \{a, b, c, d\}$ with prob $\{0.5, 0.25, 0.125, 0.125\}$

Huffman code

consider $X \sim \{a, b, c, d, e, f\}$ with prob $\{0.4, 0.14, 0.13, 0.12, 0.11, 0.10\}$

aside: information content in a perfect code

let C be a perfect code for X, and given database $X_1 X_2 \ldots X_n$, suppose we pick one bit at random from the encoded sequence $C(X_1)C(X_2)\ldots C(X_n)$. what is the probability this bit is a 1?

aside: information content in a perfect code

stream codes (Mackay chapter 6)

problems with Huffman codes

changing ensembles

the extra bit: we know Huffman gives $H(X) \leq \mathbb{E}[L_C(X)] \leq H(X) + 1$

a	0.001	00000
b	0.001	00001
c	0.990	1
d	0.001	00010
e	0.001	00011
f	0.001	0100
g	0.001	0101
h	0.001	0110
i	0.001	0111
j	0.001	0010
k	0.001	0011

$\mathbb{E}[\text{length}] = 1.034$

$H(X) = 0.114$

$\mathbb{E}[L]/H(X) = 9$

the guessing game

_ _ _ _ _ _ _ _   _ _   _ _ _ _ _ _ _ _ _ _ -

_ _ _ _   _ _   _ _ _   -   _ _ _ _   _ _ _ _

how to model data sources

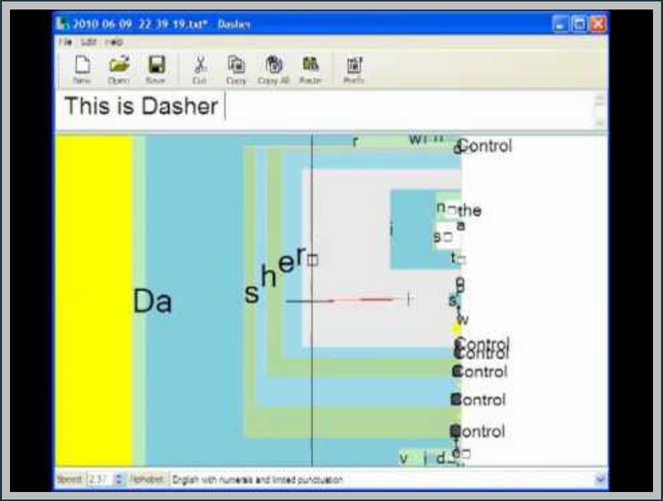
two approaches to stream coding

arithmetic coding

arithmetic coding

arithmetic coding

application of arithmetic coding beyond compression



https://www.youtube.com/watch?v=nr3s4613DX8

Lempel-Ziv codes (dictionary codes)

Lempel-Ziv-Welch coding

source substrings	λ	1	0	11	01	010	00	10
$s(n)$	0	1	2	3	4	5	6	7
$s(n)_{\text{binary}}$	000	001	010	011	100	101	110	111
(pointer, bit)		$(,1)$	$(0,0)$	$(01,1)$	$(10,1)$	$(100,0)$	$(010,0)$	$(001,0)$

aside: from coin-flips to distributions

we are given a fair coin (i.e., $X_i \sim \text{Bernoulli}(p)$, and want to use it to generate a rv. $Y \sim \{a, b, c, d, e, f\}$ with prob $\{0.5, 0.25, 0.125, 0.125\}$

aside: from coin-flips to distributions

we are given a fair coin (i.e., $X_i \sim \text{Bernoulli}(p)$, and want to use it to generate a rv. $Y \sim \{a, b\}$ with prob $\{5/8, 3/8\}$

aside: from coin-flips to distributions

we are given a fair coin (i.e., $X_i \sim \text{Bernoulli}(p)$, and want to use it to generate a rv $Y \sim \{a, b\}$ with prob $\{1/3, 2/3\}$

puzzle: generating a fair coin

we are given a coin with some unknown bias p

how can we use it to generate a Bernoulli(1/2) random variable

blank

blank