

# Information Theory and Networks

## Lecture 12: Coding Language

Matthew Roughan

`<matthew.roughan@adelaide.edu.au>`

`http://www.maths.adelaide.edu.au/matthew.roughan/  
Lecture\_notes/InformationTheory/`

School of Mathematical Sciences,  
University of Adelaide

September 18, 2013

# Part I

## Coding Language

# Redundancy

This parrot is no more. It has ceased to be. It's expired and gone to meet its maker. This is a late parrot. It's a stiff. Bereft of life, it rests in peace. If you hadn't nailed it to the perch, it would be pushing up the daisies. It's rung down the curtain and joined the choir invisible. This is an ex-parrot.

*Monty Python, The Dead Parrot Sketch*

# Section 1

## Entropy Rate

# Entropy Rate [CT91, p.63]

## Definition (Entropy Rate)

The **entropy rate** of a stochastic process  $\{X_i\}$  is defined by

$$H(\mathcal{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n)$$

when the limit exists.

# Entropy Rate: Example 1

IID symbols

Remember that entropy is additive for independent RVs, so

$$\begin{aligned} H(\mathcal{X}) &= \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \\ &= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(X_i) \\ &= \lim_{n \rightarrow \infty} \frac{nH(X_1)}{n} \\ &= H(X_1) \end{aligned}$$

as you would hope!

## Entropy Rate: Example 2

independent, but not identical

As before

$$\begin{aligned} H(\mathcal{X}) &= \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \\ &= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(X_i) \end{aligned}$$

but this time, the  $H(X_i)$  are not all equal. When the limit exists, this just looks like the expected entropy, but it doesn't have to exist, e.g.,  $X \in \{0, 1\}$  with  $p_i = P(X_i = 1)$ , where

$$p_i = \begin{cases} 0.5, & \text{if } 2k < \log \log i \leq 2k + 1 \\ 0, & \text{if } 2k + 1 < \log \log i \leq 2k + 2 \end{cases}$$

for integer  $k$ . The process has arbitrarily long stretches where  $H(X_i) = 1$ , followed by exponentially longer stretches where  $H(X_i) = 0$ , so the running average oscillates (and hence has no limit).

## Entropy Rate: Alt Def

There is an alternative definition for entropy rate:

$$H'(\mathcal{X}) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1}, \dots, X_1)$$

- $H(\mathcal{X})$  is the long term rate at which entropy grows per symbol
- $H'(\mathcal{X})$  is the conditional entropy of the last symbol given the long-term history of a process.

It turns out they are the same thing (for cases we care about).

### Theorem

*For a stationary stochastic process, the two entropy rates exist (the limits are defined), and they are equal*

$$H'(\mathcal{X}) = H(\mathcal{X})$$



# Entropy Rate: Example 3

## Markov Chain

The nice thing about the second definition is it gives us an approach for calculating the entropy rate for a Markov Chain. The Markov property immediately tells us that

$$H(X_n | X_{n-1}, \dots, X_1) = H(X_n | X_{n-1})$$

So the entropy rate for a Markov Chain is just

$$H'(\mathcal{X}) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1}, \dots, X_1) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1})$$

And we can calculate that directly in terms of the transition matrix giving the following result.

# Entropy Rate: Example 3

## Markov Chain

### Theorem

*The entropy rate of a stationary Markov Chain with transition matrix  $P$  and stationary distribution  $\pi$  is just*

$$H(\mathcal{X}) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1}) = - \sum_{i,j} \pi_i p_{ij} \log p_{ij}$$

## Entropy Rate: Example 3

Proof.

From the definition of conditional entropy

$$\begin{aligned} \lim_{n \rightarrow \infty} H(X_n | X_{n-1}) \\ = - \lim_{n \rightarrow \infty} \sum_{i,j} p(X_{n-1} = i) p(X_n = j | X_{n-1} = i) \log p(X_n = j | X_{n-1} = i) \end{aligned}$$

For a stationary (homogeneous) Markov Chain  $p(X_n = j | X_{n-1} = i) = p_{ij}$  independent of  $n$ , and we consider (here) only finite state cases, so the limit can be taken inside the summation to give:

$$\begin{aligned} H(\mathcal{X}) &= - \sum_i \lim_{n \rightarrow \infty} p(X_{n-1} = i) \sum_j p_{ij} \log p_{ij} \\ &= - \sum_i \pi_i \sum_j p_{ij} \log p_{ij} \end{aligned}$$

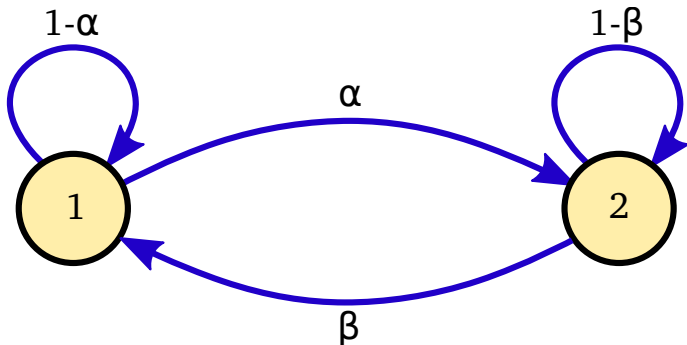
□

# Entropy Rate: Example 3

## Markov Chain

Example: Two state process, with probability transition matrix

$$P = \begin{bmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{bmatrix}$$



# Entropy Rate: Example 3

## Markov Chain

Stationary distribution:

$$\pi = \left( \frac{\beta}{\alpha + \beta}, \frac{\alpha}{\alpha + \beta} \right)$$

Entropy:

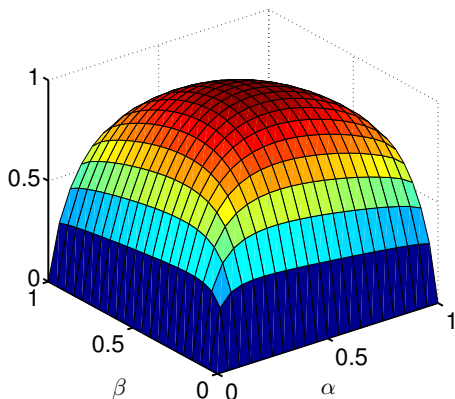
$$\begin{aligned} H(\mathcal{X}) &= - \sum_{i,j} \pi_i p_{ij} \log p_{ij} \\ &= \frac{\beta}{\alpha + \beta} (\alpha \log \alpha + (1 - \alpha) \log(1 - \alpha)) \\ &\quad + \frac{\alpha}{\alpha + \beta} (\beta \log \beta + (1 - \beta) \log(1 - \beta)) \\ &= \frac{\beta H(\alpha)}{\alpha + \beta} + \frac{\alpha H(\beta)}{\alpha + \beta} \end{aligned}$$

# Entropy Rate: Example 3

Markov Chain

Entropy:

$$H(\mathcal{X}) = \frac{\beta H(\alpha)}{\alpha + \beta} + \frac{\alpha H(\beta)}{\alpha + \beta}$$



## Section 2

# Block-Based Compression

## Block encoding

Encode blocks of  $n$  symbols, e.g.,  $(X_1, X_2, \dots, X_n)$ , then the expected code length for the entire block will be

$$H(X_1, X_2, \dots, X_n) \leq E[\ell(X_1, X_2, \dots, X_n)] < H(X_1, X_2, \dots, X_n) + 1$$

If the  $X_i$  are IID, then

$$H(X_1, X_2, \dots, X_n) = nH(X)$$

so the length of code per input symbol satisfies

$$H(X) \leq L_n < H(X) + 1/n$$

If we use large blocks, we can achieve very close to the best possible efficiency, but the assumption that the symbols are IID is a little too strong.

- We need to deal with more general stochastic processes
- We need to incorporate correlations



# Block encoding of correlated data

## Theorem

The minimum expected codeword length per symbol for coding  $n$  symbol blocks,  $L_n^* = E[\ell(X_1, X_2, \dots, X_n)]/n$ , satisfies

$$H(X_1, X_2, \dots, X_n) \leq L_n^* < H(X_1, X_2, \dots, X_n) + 1,$$

and for a stationary stochastic process

$$L_n^* \rightarrow H(\mathcal{X})$$

where  $H(\mathcal{X})$  is the entropy rate of the process.

In essence this says we can make the code as close to optimum as we like by increasing the block size.

# Obvious Solution

Code blocks:

- Take equal length blocks and code them
- Problems:
  - ▶ Huffman coding needs probabilities
    - ★ do you estimate them from the file – two passes?
    - ★ or use generic probabilities – not quite accurate for a particular file?
  - ▶ blocks of length  $n$  have  $d^n$  possible “symbols”
    - ★ estimating small probabilities is hard
    - ★ do you include the (large) dictionary in the compressed file?
    - ★ Huffman needs complete recalculation to change the block size
  - ▶ block coding introduces **delay**
    - ★ more on that later

# Problem 1

We saw Huffman coding was good but there are some problems

- How do you know the probabilities?
  - ▶ if we want to do this for a particular file it takes two passes, and then we need to copy the dictionary
  - ▶ easy enough to measure letter frequencies in English, in general
  - ▶ how useful is the general case for a specific document?
  - ▶ how do errors in probability estimates (for small probabilities these could be large) affect efficiency

# Problem 1

What is the cost incurred if we have an incorrect estimate of the probabilities  $p_i$  [CT91, pp.89-90].

## Theorem

*The expected length of codewords under  $p(x)$  of the code assignment*

$$\ell(x) = \lceil \log(1/q(x)) \rceil$$

*satisfies*

$$H(p) + D(p\|q) \leq E_p[\ell(X)] < H(p) + D(p\|q) + 1.$$

Effectively, the cost of using the wrong distribution  $q$  is the relative entropy between  $q$  and  $p$ , i.e.,  $D(p\|q)$ .

# Problem 1

Proof.

$$\ell(x) = \lceil \log(1/q(x)) \rceil$$

so

$$\begin{aligned} E[\ell(X)] &= \sum_x p(x) \lceil \log(1/q(x)) \rceil \\ &< \sum_x p(x) (\log(1/q(x)) + 1) \\ &= \sum_x p(x) (\log(p(x)/[q(x)p(x)])) + 1 \\ &= \sum_x p(x) \log(p(x)/q(x)) + \sum_x p(x) \log(1/p(x)) + 1 \\ &= D(p||q) + H(p) + 1 \end{aligned}$$

And similarly for the lower bound. □

# Huffman coding with estimates: Example 1

$X$	probability $p$	optimal codewords	probability estimate $q$	actual codewords
a	0.25	01	0.28	01
b	0.25	10	0.22	10
c	0.2	11	0.16	000
d	0.15	000	0.16	001
e	0.15	001	0.18	11
$H(X)$	2.286		2.286	
$E_{p\ell}$		2.3		2.35

$$D(p\|q) = 0.016$$

# Assignment

Create block Huffman codes for English:

- Analyse text again, this time looking not just at frequencies, but also at the Markov modes or order 1-5.
  - ▶ you may simplify by only using lower case, and ignoring punctuation
  - ▶ so you should have  $27^n$  symbols to code, for block size  $n$
- Compare the efficiency of each code, both theoretically by calculating appropriate entropies, and in practice, by coding your text.
- Again, generate a short report on the results (include your tree for the block length 1 model).

# Further reading I



Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, John Wiley and Sons, 1991.



Gjerrit Meinsma, *Data compression & information theory*, Mathematisch cafe, 2003, [wwwhome.math.utwente.nl/~meinsmag/onzin/shannon.pdf](http://wwwhome.math.utwente.nl/~meinsmag/onzin/shannon.pdf).