TN3125
Information and Computation

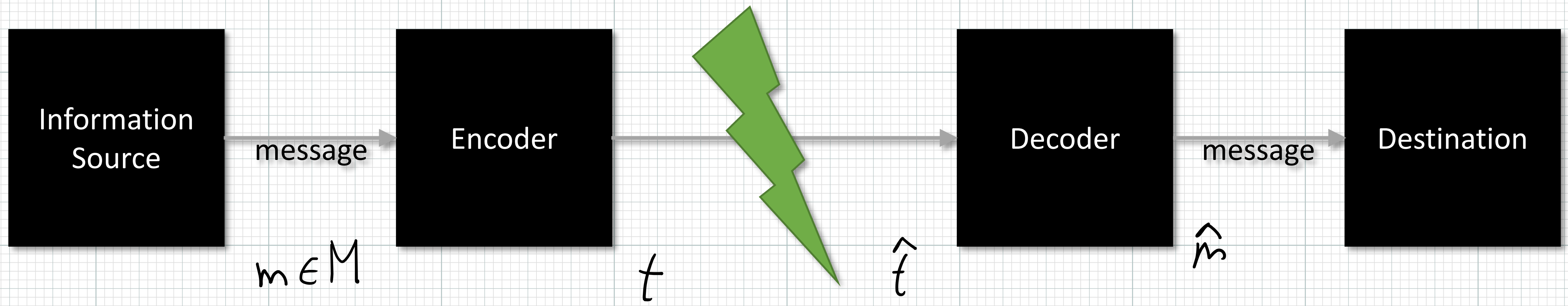
Lecture 2

1- *Introduction*

Exam

- Time
- Difficulty
- Estimated grade

The abstract communications model



$$m = \hat{m}$$

Summary of week 1

- We derived an information measure from basic axioms
- We proved basic properties of entropy
- We did not show the operational value of entropy

Problem 1: *Measuring information* (50 points total)

(a) (5 points) Let X be an ensemble with the uniform distribution over the set $\{-2, -1, 0, 1, 2\}$. What is the value of $p_X(-2)$?

Answer 1a

(b) (5 points) Let $f = x^2$, and $Y = f(X)$. What is the value of $p_Y(1)$?

Answer 1b

Let us imagine that you want to transmit a word from the foreign language Icish to a friend. In this language words have always only two letters, first a consonant then a vowel. The consonants in the language are $\mathcal{C} = \{b, c\}$ and they occur respectively with probabilities $\{1/2, 1/2\}$. The vowels in the language are $\mathcal{V} = \{a, e\}$. The probability of having a vowel in a word depends on the consonant as follows:

	a	e
b	$1/2$	$1/2$
c	$1/4$	$3/4$

Let CV be the joint ensemble that represents the occurrence of the different words and C, V be the ensembles representing the occurrence of consonants and vowels respectively.

(c) What is the entropy of C? (10 points)

Answer 1c

Entropy of V :

$$H(V) = -\sum_i p_V(x_i) \log p_V(x_i)$$

$$p_V(a) = \sum_c p_{VC}(a, c)$$

$$= \sum_c p_{VC}(a \mid c)\ p_C(c)$$

$$= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} = \frac{3}{8}$$

$$p_V(e) = \frac{5}{8}$$

	a	e
b	$1/2$	$1/2$
c	$1/4$	$3/4$

We have focused our study on information measures on the entropy function. However, there is a whole zoo of entropic measures with different properties and operational interpretations. Given an ensemble X, we define its collision entropy as follows:

$$H_c(X) = -\log \sum_{x \in \mathcal{X}} (p_X(x))^2 \ .$$

This is also a very useful quantity with applications in cryptography (you will learn about it if you take the master's course Quantum communications and cryptography).

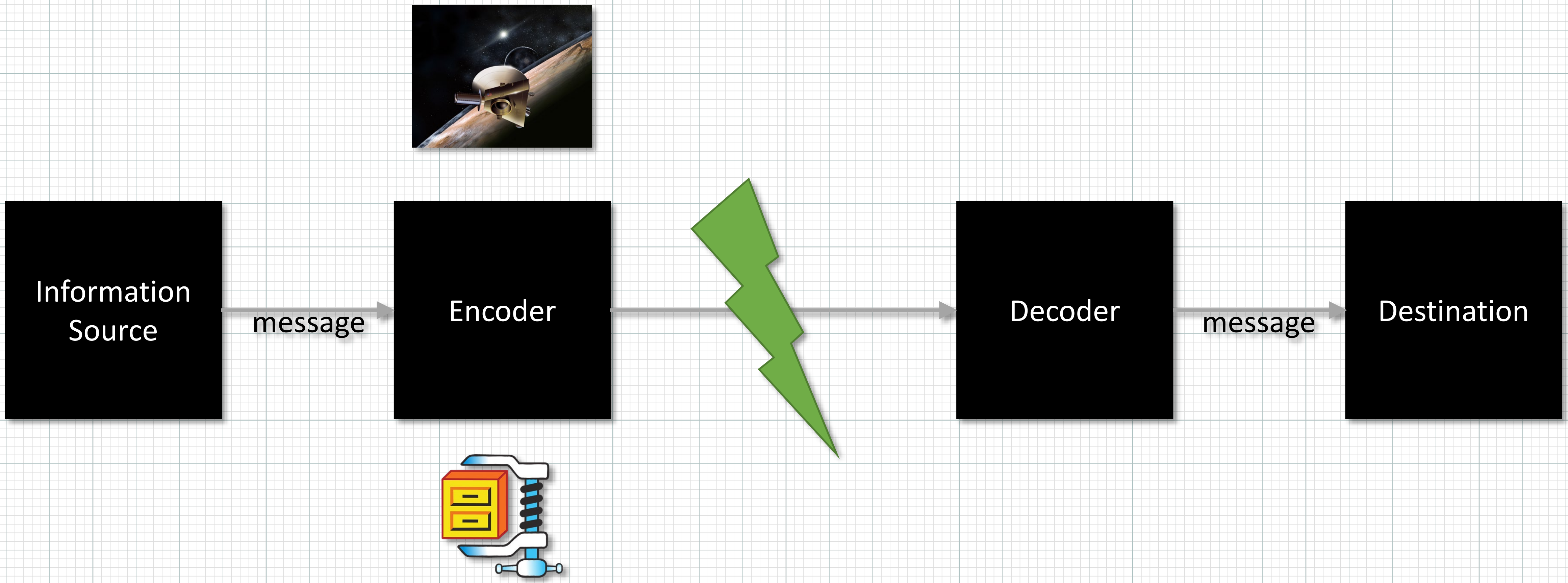
(f) (10 points) Prove that the collision entropy can not be greater than entropy. That is prove: $H(X) \geq H_c(X)$. (Hint: Jensen's inequality can make this proof very simple)

Answer 1f

$-H(x)$ $\quad\quad\quad\quad$ $-H_c(x)$

$$\sum_x p_x(x) \log p_x(x) \leq \log \left(\sum_x p_x(x)^2\right)$$

The abstract communications model



Learning goals for week 2

- Understand what is a code and what are its basic properties
- Given a code decide whether or not it is uniquely decodable
- Given a random variable, construct its Huffman code
- Encode and decode with arithmetic codes

- Not on exam. Map the different codes to formats: (gif, png, jpg, zip, bzip, etc.)

TN3125
Information and Computation

Lecture 2

Codes

Our old friend the weather forecast

	Days with no rain	Days with rain
Rotterdam	212	153
Atacama desert	360	5

- How do we compare codes?
- What is the best code to send one day of weather forecast? Is it the same in Rotterdam and in the Atacama desert?
- What is the best code if we want to send the forecast for several days?

Codes

- **Definition.** A D-ary code for an ensemble X is a function C that takes elements from A_X to D^* the set of finite length words in an alphabet with D symbols.
- **Example.**

$$A_x = \{\text{rain}, \text{no rain}\}, \quad D = \{0, 1\}$$

$$c(\text{rain}) = \underline{00}$$

$$c(\text{no rain}) = \underline{111}$$

Codewords

Uniquely decodable codes

- **Definition**. A code C is non-singular if for all $w_1, w_2 \in D$, $c(w_1) \neq c(w_2)$ unless $w_1 = w_2$.
- **Examples.** $C_1 = \{0, 00, 1\}, \quad C_2 = \{0, 10, 110\}$

- **Definition.** Let $w \in D^*, w = (w_1, \ldots, w_n)$. We denote by $c(w) = c(w_1) \ldots c(w_n)$

- **Definition**. A code C is uniquely decodable if for all $w_1, w_2 \in D^*$, $c(w_1) \neq c(w_2)$ unless $w_1 = w_2$.
- **Examples.**

Instantaneous codes

- **Definition**. A code is called instantaneous if it can be decoded symbol by symbol from left to right without regarding future symbols.

Prefix codes

- **Definition.** Let $w_1, w_2 \in D^*$, w_1 is a prefix of w_2 if there exist $t \in D^*$ such that $(w_1, t) = w_2$.
- **Examples.** $w_1 = 001$, $w_2 = 00100$, $t = 00$

- **Definition.** A prefix code is a code where no codeword is the prefix of any other codeword
- **Examples.**

Prefix codes are uniquely decodable

- **Lemma.** Prefix codes are uniquely decodable

$$\underbrace{0\ 1\ 0\ \ldots}_{w_1}\ \underbrace{\ldots\ 0\ \ldots}_{w_2}\ \ldots\ 1$$

Prefix = instantaneous

- **Lemma**. A code is instantaneous if and only if it is prefix

$\Leftarrow \quad x_1 x_2 \ldots x_n \longrightarrow c(x_1) | c(x_2) | \ldots c(x_n)$

$\Rightarrow \left(\neg \text{prefix} \Rightarrow \neg \text{instantaneous} \right)$

Suppose $\exists x, y : c(x) = c(y) t$

$\exists t$

If we receive $c(y)$ we need to wait to know whether we have x or y

Exercise

- **Definition.** Let $w_1, w_2 \in D^*$, w_1 is a suffix of w_2 if there exist $t \in D^*$ such that $(t, w_1) = w_2$.
- **Definition.** A suffix code is a code where no codeword is the suffix of any other codeword
- **Question.** Are all suffix codes also prefix codes? If yes, prove it. If not, give a counterexample.
- **Question.** Are suffix codes uniquely decodable?
- **Question.** Are suffix codes instantaneous?

Average length of a code

- Let $w \in D^*$, we denote the length of the sequence by $|w|$ or by $l(w)$
- **Example** $w = 010$ $|w| = 3$

- **Definition**. Let C be a code for ensemble X. The average length of the code is given by

$$L(C) = \sum_{x \in X} p_X(x)|C(x)|$$

- **Example** $X = \{ \text{coins}, \text{tails} \}$ with uniform distribution
$c(\text{coins}) = 00$, $c(\text{tails}) = 1$ $L(c) = \frac{1}{2} 1 + \frac{1}{2} 2 = 1.5$

Kraft-MacMillan’s inequality

- **Theorem**. The length of a uniquely decodable source code C for the random variable X taking values in a d-ary alphabet verifies:

$$\sum_{x} \frac{1}{d^{l(C(x))}} \leq 1$$

moreover (converse), given a set of lengths satisfying the inequality, it is possible to construct a uniquely decodable code.

$$\ell(x) \equiv \ell(c(x))$$

Proof

Let $S = \sum_{x \in X} 2^{-\ell(x)}$,

$$S^n = \left(\sum_{x \in X} 2^{-\ell(x)}\right)^n = \sum_{x_1 \in X} 2^{-\ell(x_1)} \cdot \sum_{x_2 \in X} 2^{-\ell(x_2)} \cdots \sum_{x_n \in X} 2^{-\ell(x_n)}$$

$$= \sum_{(x_1, \dots x_n)} 2^{-\ell(x_1)} \cdots 2^{-\ell(x_n)} = \sum_{x_1, \dots x_n \in X^n} 2^{-(\ell(x_1) + \dots + \ell(x_n))}$$

$$= \sum_{\ell=1}^{n \cdot \ell_{max}} W_\ell \cdot 2^{-\ell} \leq \sum_{\ell=1}^{n \cdot \ell_{max}} 2^{\ell} \cdot 2^{-\ell} = n \cdot \ell_{max}$$

Entropy limits the length of codes

- **Theorem**. The length of a uniquely decodable binary code C for a random variable X satisfies:

$$L(C) \geq H(X)$$

Proof

Consider $H(X) - L(C) = -\sum_i p_i \log_2 p_i - \sum_i p_i \, l(x_i)$

$$= -\sum_i p_i \log_2 p_i - \sum_i p_i \log_2 2^{l(x_i)}$$

$$= -\sum_i p_i \log_2 p_i \cdot 2^{l(x_i)}$$

$$= \sum_i p_i \log_2 \frac{1}{p_i \cdot 2^{l(x_i)}}$$

$$\leq \log_2 \sum_i p_i \frac{1}{\cancel{p_i} \, 2^{l(x_i)}}$$

$$\leq \log_2 1 = 0$$

Recap

- Several types of codes: non-singular, uniquely decodable, prefix and instantaneous. The latter two coincide.
- Uniquely decodable codes satisfy Kraft-MacMillan inequality
- If a code that satisfies the Kraft-MacMillan inequality we can find another code that also satisfies it and is a prefix code.
- The average length of a uniquely decodable code is bounded from below by the entropy of the ensemble.

TN3125
Information and Computation

Lecture 2

2 – Towards optimal codes

Exercise

- Find whether or not the code $C = \{01,100,1101,10111,01011\}$ is uniquely decodable

Sardinas-Patterson algorithm

- $C_0 = \{c_1, c_2, \dots, c_n\}$ are the codewords.
- $C_1 = \{w : uw = v; u, v \in C\}$
- For $n \geq 2$
  - $C_n = \{w : uw = v; u \in C, v \in C_{n-1} \text{ or } u \in C_{n-1}, v \in C\}$
  - If C_n contains a codeword, the code is not uniquely decodable
  - If C_n is empty, the code is uniquely decodable
  - If there is m such that $C_n = C_m$, the code is uniquely decodable
  - Else continue

- $C = \{01{,}100{,}1101{,}10111{,}01011\}$

Example

- $C = \{01, 100, 1101, 0111\}$

$C_0 = \{01, 100, 1101, 0111\}$

$C_1 = \{w : uw = v, u, v \in C_0\} = \{11\}$

$C_2 = \{w : uw = v, u \in C_0, v \in C_{n-1}$ or $u \in C_{n-1}, v \in C_0\} = \{01\}$

Exercise

- $C = \{01, 100, 0101, 1111\}$

Exercise

- $C_0 = \{01,100,1101,10111,01011\}$

$C_1 = \{011\}$

$C_2 = \{1\}$

$C_3 = \{00, 0111, 101\}$

$C_4 = \{11\}$

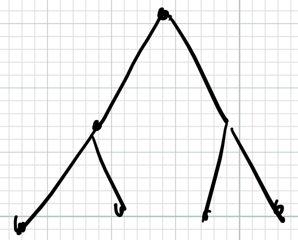
$C_5 = \{01$

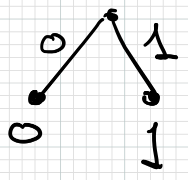
Remarks on Sardinas-Patterson's algorithm

- The algorithm will finish since all the elements in C_n are suffixes of C
- The algorithm is correct

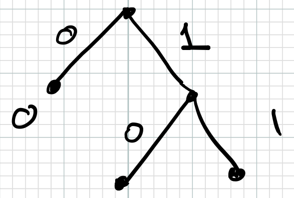
Prefix codes can be represented by (binary) trees

- $C = \{0,1\}$





- $C = \{0,10,11\}$



Exercise. Find the tree representation of:

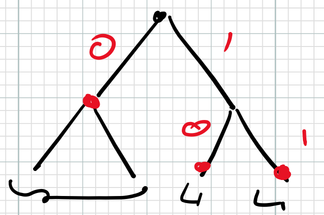
- $C = \{0001,0010,0100,1000\}$
- $C = \{00,01,100,101\}$

There is a prefix code for a set of lengths verifying KM

- Let $l_1 \leq l_2 \leq \cdots \leq l_n$, the lengths of codewords, $w_1, w_2, \dots, w_n$
- We are going to construct a binary tree with depth l_n
- To codeword w_i we assign a set of $2^{l_n - l_i}$ leaves of the tree
- The number of leaves assigned is below the total number of leaves of the tree 2^{l_n}. Why?
- We place the leaves from left to right, from those of the shortest codeword w_1 to the longest codeword w_n.
- We prune the tree removing all the leaves and edges until the root of each set of leaves.

Examples

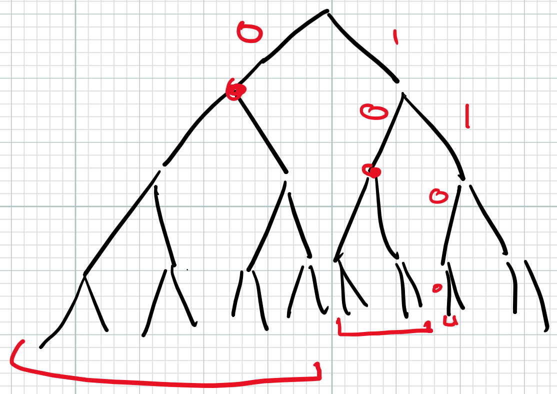
- $L = \{1,2,2\}$



$\{0, 10, 11\}$

- $L = \{2,4,4\}$

- $L = \{1,2,4\}$



$\{0, 10, 1100\}$

Exercise

- Construct a prefix code for lengths $L = \{2,4,4\}$

The optimal code is at most one bit from entropy

- **Exercise.** Let X be an ensemble and consider a binary code C with lengths $l\big(C(x)\big) = \lceil \log 1/p_X(x) \rceil$ for all symbols $x \in X$. Show that there exist a code C with the indicated lengths that satisfies:

$$H(X) \leq L(C) \leq H(X) + 1$$

Proof

Consider $l_i = \lceil \log 1/p_i \rceil$,

$$\sum_i 2^{-p_i} =$$

$$\sum_i 2^{\lfloor \log p_i \rfloor} \leq \sum_i 2^{\log p_i} = \sum_i p_i = 1$$

$$-\lceil \log 1/p_i \rceil = \lfloor \log p_i \rfloor$$

$$l(C) = \sum_i p_i \lceil \log 1/p_i \rceil$$

$$\leq \sum_i p_i \left(\log 1/p_i + 1 \right)$$

$$= H(x) + 1 \qquad \square$$

$$2^{\lfloor \log p \rfloor} \leq 2^{\log p}$$

TN3125
Information and Computation

Lecture 2

3 – Huffman codes

Huffman codes

- Given an ensemble X the Huffman code $C(X)$
  - Is a prefix code
  - Can be (reasonably) efficiently constructed
  - Its length is at most one bit from entropy!
- Huffman codes were invented by David Huffman while he was a master student.

Construction algorithm

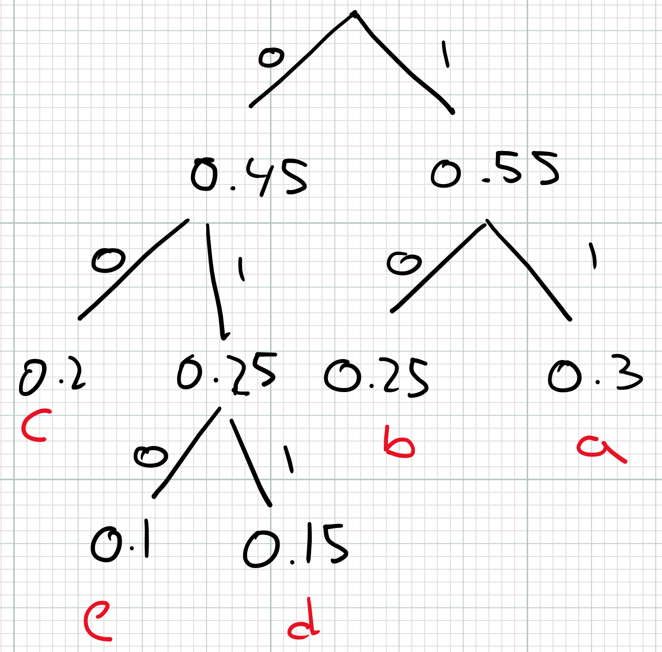
- Huffman(X)
  - Order the probabilities such that $p_X(x_1) \geq p_X(x_2) \geq \cdots \geq p_X(x_n)$
  - Construct ensemble X' with $n-1$ elements:
$$\begin{aligned} p_{X'}(x_i) &= p_X(x_i) && \text{if } 1 \leq i \leq n-2 \\ p_{X'}(x_{n-1}) &= p_X(x_{n-1}) + p_X(x_n) && \text{else} \end{aligned}$$
  - Define the code C_X as an extension of the code $C_{X'}$
$$\begin{aligned} C_{X'}(x_i) &= C_X(x_i) && \text{if } 1 \leq i \leq n-2 \\ C_X(x_{n-1}) &= C_{X'}(x_{n-1})0 && , \\ C_X(x_n) &= C_{X'}(x_{n-1})1 && . \end{aligned}$$
  - If the size of X' is one end, otherwise get the Huffman code of X'

Example 1

$\{0.3, 0.25, 0.2, 0.15, 0.1\}$

a	0.3	0.3	0.3	0.55
b	0.25	0.25	0.25	0.45
c	0.2	0.2	0.45	
d	0.15	0.25		
e	0.1			

$$C_x(e) = C_{x'}(d)\,0$$
$$C_x(d) = C_{x'}(d)\,1$$



a	11
b	10
c	00
d	011
e	010

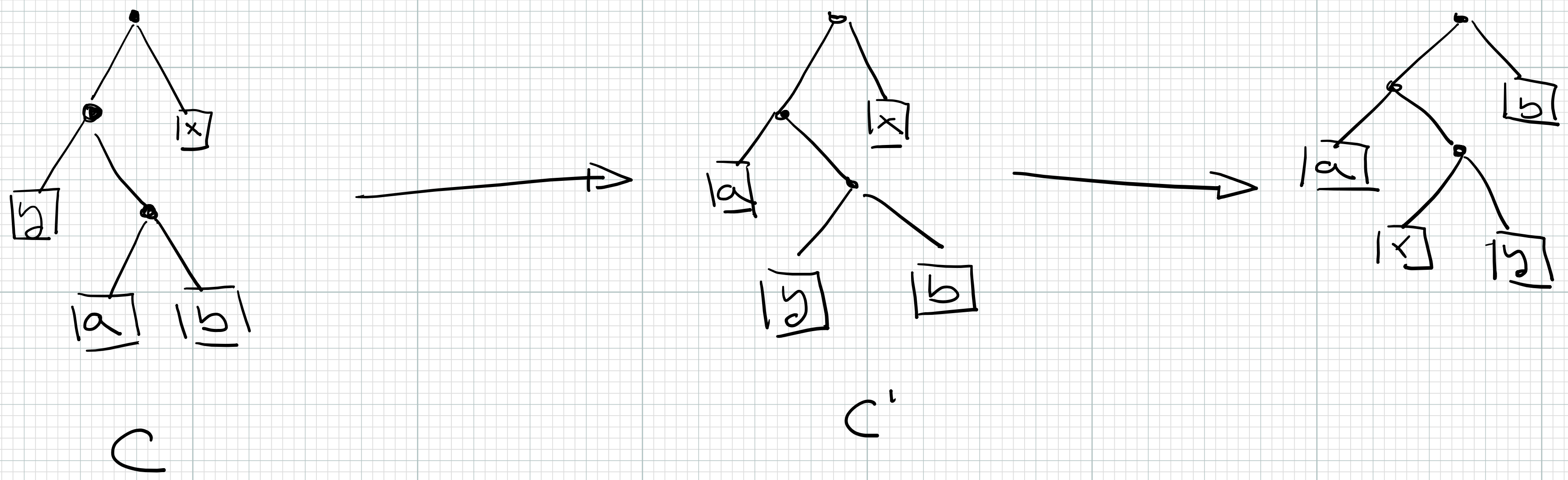
Example 2

$\{0.6, 0.3, 0.05, 0.05\}$

Example 3

Huffman codes are optimal.

- Claim 1. If x, y are the two symbols with smallest probability, there exists an optimal code C where $C(x), C(y)$ are the longest codewords and differ only in the last bit.



Let's see that $L(c) \geq L(c')$

$$L(c) - L(c') = p_a \cdot l_a + p_x l_x - p_a l_x - p_x l_a$$

$p_x \leq p_a$, $l_a > l_x$

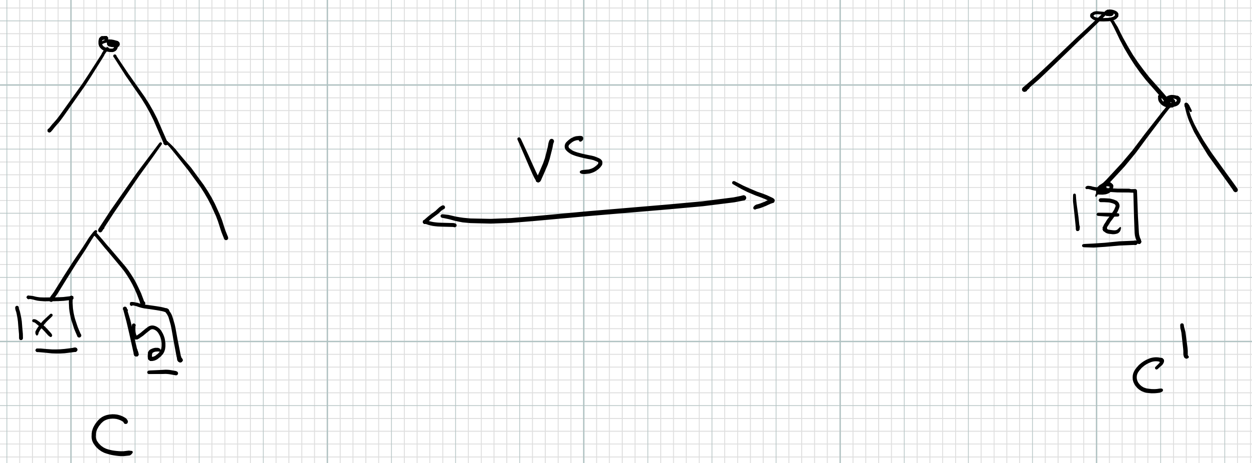
$$= p_a (l_a - l_x) - p_x (l_a - l_x)$$

$$= (p_a - p_x)(l_a - l_x) > 0$$

Huffman codes are optimal

- Claim 2.
  - For any code C satisfying that the two symbols with smallest probability have codewords of equal length differing only in the last bit (Claim 1)
  - and the code C' that results from removing symbols x, y and adding symbol z with $p(z) = p(x) + p(y)$.

$$L(C) = L(C') + p(z)$$



$$L(c) - L(c') = p_x l_x + p_y l_y - p_z l_z = d(p_x + p_y) - p_z(d-1)$$

$$p_z = p_x + p_y$$

$$d = l_x = l_y = l_z + 1$$

$$= d\, p_z - p_z d + p_z$$

$$= p_z$$

Huffman codes are optimal

- Claim 3. The Huffman algorithm produces **an** optimal code.

Huffman code extension

- If one bit from entropy is not enough, we can do Huffman coding for X^n

Exercise

- Consider a binary ensemble X with probabilities $\{0.9,0.1\}$ construct the Huffman code for X, X^2, X^3 and find the average length of each code.

Example

$p_x(0) = 0.1$ $\quad$ $p_x(1) = 0.9$

0.9	0.81	0.79
0.1	0.09	0.081
	0.09	0.081
	0.01	0.081
		0.009
		0.009
		0.009
		0.001

Recap

- Huffman is optimal for an ensemble
- It is within one bit of entropy
- It can get arbitrarily close to entropy by doing extensions of the ensemble
- Costly!
- Good if probabilities known a priori

TN3125
Information and Computation

Lecture 2

4 – Beyond symbol codes

Recap

- There are different types of codes: non-singular, uniquely decodable, prefix and instantaneous
- The average length of a uniquely decodable code is bounded by entropy
- We can decide whether or not a code is uniquely decodable
- Huffman gets us within one bit of entropy
- Arithmetic codes allow to get arbitrarily close to entropy without code extension

You will do great in the exam if

- You can decide if a code is uniquely decodable
- Find a prefix code given a set of lengths
- Construct a Huffman code and an extended Huffman code
- Find the interval for a word in arithmetic coding and recover a word from a point in the real line
- Prove basic properties of these codes

Resources

- Lecture notes
- Slides
- Cover and Thomas chapter 5
- MacKay chapters 5