Information Theory
Data compression

CoCoNut, 2016

Emmanuela Orsini

	Compression "SOURCE CODING"	Error correction "CHANNEL CODING"
Information theory	Source coding theorem Kraft-McMillan ineq.	Channel coding theorem Channel capacity
Coding methods	Symbol codes Huffman codes	Hamming codes Reed Solomon codes LDPC codes

- The Shannon information content of an outcome x_i:

$$h(x_i) = \log_2 \frac{1}{p(x_i)}$$

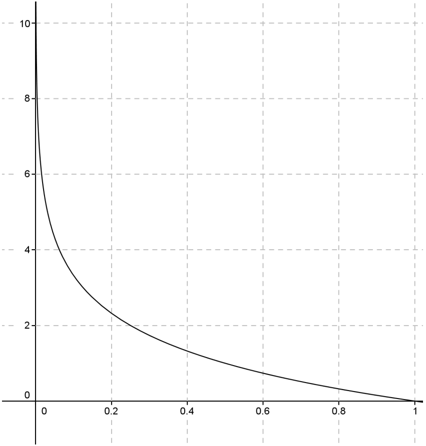
It is measured in bits.

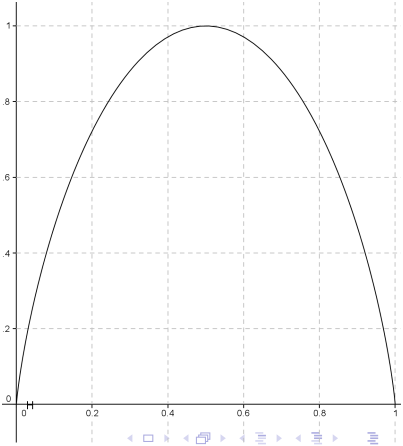
- The entropy $H(X)$ is a sensible measure of the average amount of information contained in each outcome we obtain:

$$H(X) = \sum_{i=1}^{m} p(x_i) \log_2 \frac{1}{p(x_i)}$$

If X is a r.v. such that $\Pr(X = 0) = p$ and $\Pr(X = 1) = 1 - p$, then the entropy of X is

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p)$$ **binary entropy function**





$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

The **capacity** of a DMC is

$$C = max_{p(x)} I(X;Y).$$

Theorem (Channel coding theorem (Informal))

The maximum rate R of information over a channel with arbitrarily low error probability is given by its channel capacity C.

- The capacity measures the rate at which block of data can be communicated over the channel with arbitrarily small probability of error.

- ⋆ Lossless codes which achieve **compression** (and decompression) without errors
- ⋆ Variable-length codes which encode one symbol at time

- ◇ **Theory**: How well these source codes perform?
- ◇ **Practice**: How can we construct a good source code?

- $\mathcal{A}^N$ the set of all strings of length N over an alphabet $\mathcal{A}$
- $\mathcal{A}^+$ the set of strings of all finite length over $\mathcal{A}$

DEFINITION: Let X be an m-ary source with $\mathcal{A} = \{a_1, \ldots, a_m\}$ and $p(a_1), \ldots, p(a_m)$. A **binary source code** C for X is an encoding map:

$$\begin{aligned} \mathcal{A} &\longrightarrow \{0,1\}^+ \\ a_i &\longmapsto c(a_i). \end{aligned}$$

The **extended code** C^+ is a mapping:

$$\begin{aligned} \mathcal{A}^N &\longrightarrow \{0,1\}^+ \\ \mathbf{a} = (a_1 a_2 \ldots a_N) &\longmapsto c(a_1)c(a_2)\ldots c(a_N) = c^+(\mathbf{a}), \end{aligned}$$

obtained from C by concatenation.

$\star$ Let $c(\mathbf{a})$, we denote its length by $l(\mathbf{a})$ and $l_i = l(a_i)$, for $a_i \in \mathcal{A}$.

⋆ **INTUITION**: to achieve compression, on average, we assign shorter encodings to the more probable outcomes and longer encodings to the less probable.

⋆ Let X be an m-ary source and $\{p_1, \ldots, p_m\}$ its probability distribution. The **expected length** of a code C for X is

$$L(C, X) = \sum_{i=1}^{m} p_i \cdot l_i,$$

where $\{l_1, \ldots, l_m\}$ is the set of codewords lengths.

The three properties that a source code has to achieve are:

1. **Correct decoding**: any encoded string must have a unique decoding
2. **Easy decoding** : the decoding procedure has to be easy/efficient
3. **Small expected length**: the code should achieve as much compression as possible

Example

Given three different symbols a, b, c, such that

$$\Pr(a) = 0.5 \qquad \Pr(b) = 0.25 \qquad \Pr(c) = 0.25.$$

With encoding:

$$\begin{aligned} a &\mapsto 0 \\ b &\mapsto 01 \\ c &\mapsto 10 \end{aligned}$$

- A code C is said to be **uniquely decodable** if $\forall\, \mathbf{x}, \mathbf{y} \in \mathcal{A}^+$, such that $\mathbf{x} \neq \mathbf{y}$, we have $c^+(\mathbf{x}) \neq c^+(\mathbf{y})$.
  - A source code is said to be **nonsingular** if every element of $\mathcal{A}$ is mapped into a different string of $\{0, 1\}^+$, i.e.

$$\text{if } a_i \neq a_j \text{ then } c(a_i) \neq c(a_j).$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$1$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$11$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$1100$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$11000$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110000$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$1100001$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$11000010$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110000100$$

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110000100$$

- A code C is called **instantaneous** if, for each transmitted codeword c, c can be interpreted as a codeword as soon as it is received.
  - Note that an instantaneous code is also uniquely decodable, but not the other way around.

Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110000100$$

- A code C is called **instantaneous** if, for each transmitted codeword c, c can be interpreted as a codeword as soon it is received.
  - Note that an instantaneous code is also uniquely decodable, but not the other way around.
- A code C is called **prefix code** if no codeword is a prefix of any other codeword.

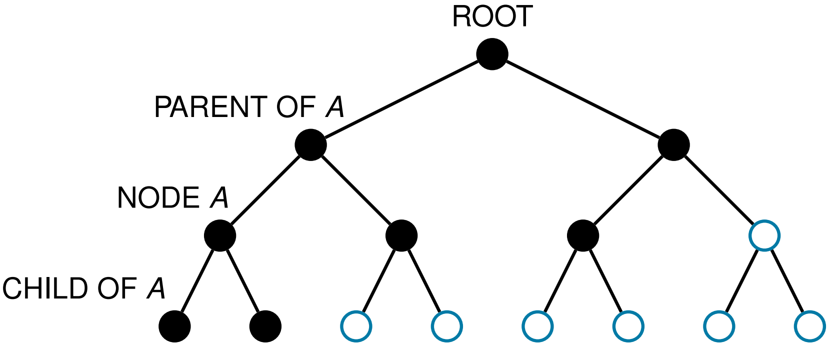
Example

Consider the u.d. code $C = \{1, 10, 000, 100\}$, enconding the alphabet symbols a, b, c, d respectively. Suppose to receive

$$110000100$$

- A code C is called **instantaneous** if, for each transmitted codeword c, c can be interpreted as a codeword as soon it is received.
  - Note that an instantaneous code is also uniquely decodable, but not the other way around.
- A code C is called **prefix code** if no codeword is a prefix of any other codeword.

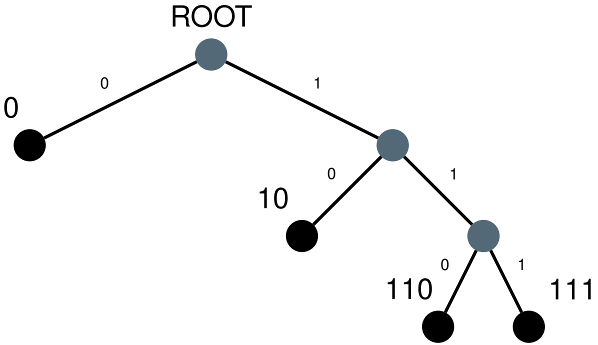
A code is instantaneous if and only if it is a prefix code.


ROOT
PARENT OF A
NODE A
CHILD OF A

- For every binary prefix code, there exists at least one binary tree such that each codeword corresponds to the sequence of labels of an unique path from the root to a leaf.
- Conversely, every binary tree defines a prefix code. The codewords of this prefix code are defined as the sequences of labels of each path from the root to each leaf of the coding tree.

Let C be the code $\{0, 10, 110, 111\}$

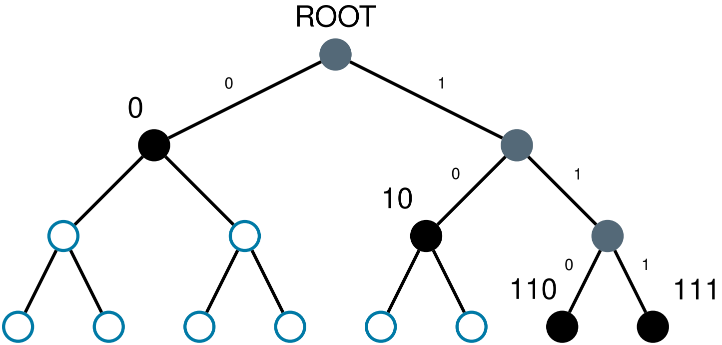
Let C be the code $\{0, 10, 110, 111\}$



C is a prefix code.

- A binary code is **complete** if there is no unused leaf in the corresponding binary tree.

Let C be the code $\{0, 10, 110, 111\}$



C is a prefix code.

- A binary code is **complete** if there is no unused leaf in the corresponding binary tree.

HOW MUCH CAN WE COMPRESS?

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code
4. $C_4 = \{00, 01, 10, 11\}$

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code
4. $C_4 = \{00, 01, 10, 11\}$
   In this case $L(X, C) = 2$ bits

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code
4. $C_4 = \{00, 01, 10, 11\}$
   In this case $L(X, C) = 2$ bits
5. $C_5 = \{0, 10, 110, 111\}$

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code
4. $C_4 = \{00, 01, 10, 11\}$
   In this case $L(X, C) = 2$ bits
5. $C_5 = \{0, 10, 110, 111\}$
   In this case $L(X, C) = 1.75$ and C is a prefix code

HOW MUCH CAN WE COMPRESS?

Consider a source X over $\mathcal{A} = \{a, b, c, d\}$ with probabilities $\{1/2, 1/4, 1/8, 1/8\}$.

1. $C_1 = \{1000, 0100, 0010, 0001\}$
   The entropy of X is 1.75 bits and $L(X, C) = 4$.
2. $C_2 = \{1, 01, 001, 0001\}$
   In this case $L(X, C) = 1.875$ and C is a **comma code**
3. $C_3 = \{1, 00, 010, 10\}$
   In this case $L(X, C) = 1.625$ and C is a not u.d. code
4. $C_4 = \{00, 01, 10, 11\}$
   In this case $L(X, C) = 2$ bits
5. $C_5 = \{0, 10, 110, 111\}$
   In this case $L(X, C) = 1.75$ and C is a prefix code
   Note that $l_i = \log_2 \frac{1}{p_i}$ $(p_i = 2^{-l_i})$

Kraft-McMillan

1. For each uniquely decodable binary code $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$, the codeword lengths must satisfy

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1.$$

This inequality is usually called **Kraft inequality.**

Kraft-McMillan

1. For each uniquely decodable binary code $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$, the codeword lengths must satisfy

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1.$$

This inequality is usually called **Kraft inequality.**

2. Given a set of codeword lengths $\{l_1, \ldots, l_m\}$, there exits a binary prefix code with these codeword lengths if and only if $l_i, i = 1, \ldots, m$, satisfy the Kraft inequality

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1.$$

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</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>

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</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>

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</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>

0	00	000	0000
			0001
		001	0010
			0011
	01	010	0100
			0101
		011	0110
			0111
1	10	100	1000
			1001
		101	1010
			1011
	11	110	1100
			1101
		111	1110
			1111

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</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>

- We want to minimize the expected length code $L(C, X)$
- The entropy is a lower bound:

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

- **Optimal source codelengths**: $L(C, X)$ is minimized and is equal to $H(X)$ only if the codelengths are equal to the *Shannon information content*:

$$l_i = \log_2(1/p_i)$$

- **Source coding theorem**: For a source (random variable) X, there exists a prefix code C with expected length satisfying:

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