

# Information Theory and Coding - Prof. Emere Telatar

Jean-Baptiste Cordonnier, Sebastien Speierer, Thomas Batschelet

October 17, 2017

## 1 Data compression

**Definition 1.1** (Information). *Abstractly, **information** can be thought of as the resolution of uncertainty.*

Given an alphabet  $\mathcal{U}$  (e.g.  $\mathcal{U} = \{a, \dots, z, A, \dots, Z, \dots\}$ ), we want to assign binary sequences to elements of  $\mathcal{U}$ , i.e.

$$\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^* = \{\emptyset, 0, 1, 00, 01, \dots\}$$

For  $\mathcal{X}$  a set

$$\begin{aligned}\mathcal{X}^n &\equiv \{(x_0 \dots x_n), x_i \in \mathcal{X}\} \\ \mathcal{X}^* &\equiv \bigcup_{n \geq 0} \mathcal{X}^n\end{aligned}$$

**Definition 1.2.** A code  $\mathcal{C}$  is called **singular** if

$$\exists (u, v) \in \mathcal{U}^2, u \neq v \quad \text{s.t.} \quad C(u) = C(v)$$

Non singular code is defined as opposite

**Definition 1.3.** A code  $\mathcal{C}$  is called **uniquely decodable** if

$$\forall u_1, \dots, u_n, v_1, \dots, v_n \in \mathcal{U}^* \quad \text{s.t.} \quad u_1, \dots, u_n \neq v_1, \dots, v_n$$

we have

$$\mathcal{C}(u_1) \dots \mathcal{C}(u_n) \neq \mathcal{C}(v_1) \dots \mathcal{C}(v_n)$$

i.e.,  $\mathcal{C}$  is non-singular

**Definition 1.4.** Suppose  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$  and  $\mathcal{D} : \mathcal{V} \rightarrow \{0, 1\}^*$  we can define

$$\mathcal{C} \times \mathcal{D} : \mathcal{U} \times \mathcal{V} \rightarrow \{0, 1\}^* \quad \text{as} \quad (\mathcal{C} \times \mathcal{D})(u, v) \rightarrow \mathcal{C}(u)\mathcal{D}(v)$$

**Definition 1.5.** Given  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$ , define

$$\mathcal{C}^* : \mathcal{U}^* \rightarrow \{0, 1\}^* \quad \text{as} \quad \mathcal{C}^*(u_1, u_n) = \mathcal{C}(u_1) \dots \mathcal{C}(u_n)$$

**Definition 1.6.** A code  $\mathcal{U} \rightarrow \{0, 1\}^*$  is **prefix-free** if for no  $u \neq v$   $\mathcal{C}(u)$  is a prefix of  $\mathcal{C}(v)$ .

**Theorem 1.1.** If  $\mathcal{C}$  is prefix-free then  $\mathcal{C}$  is uniquely decodable.

**Definition 1.7.**  $l(\mathcal{C}(u))$  is the length of the code word  $\mathcal{C}(u)$  and  $l(\mathcal{C})$  is the expected length of the code:

$$l(\mathcal{C}) = \sum_u l(\mathcal{C}(u))p(u)$$

**Definition 1.8** (Kraft sum). Given  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$

$$\text{kraftsum}(\mathcal{C}) = \sum_u 2^{l(\mathcal{C}(u))}$$

**Lemma 1.2.** if  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$  and  $\mathcal{D} : \mathcal{V} \rightarrow \{0, 1\}^*$  then

$$\text{kraftsum}(\mathcal{C} \times \mathcal{D}) = \text{kraftsum}(\mathcal{C}) \times \text{kraftsum}(\mathcal{D})$$

*Proof.*

$$\begin{aligned} \text{kraftsum}(\mathcal{C} \times \mathcal{D}) &= \sum_{u,v} 2^{-(l(\mathcal{C}) * l(\mathcal{D}))} \\ &= \sum_u 2^{-l(\mathcal{C})} \sum_v 2^{-l(\mathcal{D})} \end{aligned}$$

□

**Corollary 1.2.1.**  $\text{kraftsum}(\mathcal{C}^n) = (\text{kraftsum}(\mathcal{C}))^n$

**Proposition 1.1.** if  $\mathcal{C}$  is non-singular, then

$$\text{kraftsum}(\mathcal{C}) \leq 1 + \max_n l(\mathcal{C}(u))$$

In coding theory, the **Kraft-McMillan inequality** gives a necessary and sufficient condition for the existence of a uniquely decodable code for a given set of codeword lengths.

**Theorem 1.3.** if  $\mathcal{C}$  is uniquely decodable, then  $\text{kraftsum}(\mathcal{C}) \leq 1$

*Proof.*  $\mathcal{C}$  is uniquely decodable  $\equiv \mathcal{C}^*$  is non singular

$$\begin{aligned} \Rightarrow \text{kraftsum}(\mathcal{C}^n) &\leq 1 + \max_{u_1, \dots, u_n} l(\mathcal{C}^n) \\ \Rightarrow \text{kraftsum}(\mathcal{C})^n &\leq 1 + nL, \quad L = \max l(\mathcal{C}(n)) \end{aligned}$$

A growing exp cannot be bounded by a linear function

$$\Rightarrow \text{kraftsum}(\mathcal{C}) \leq 1$$

□

**Theorem 1.4.** Suppose  $\mathcal{C} : \mathcal{U} \rightarrow \mathcal{N}$  is such that  $\sum_u i^{\mathcal{C}(u)} \leq 1$ , then, there exist a prefix-free code  $\mathcal{C}' : \mathcal{U} \rightarrow \{0, 1\}$  s.t.  $\forall l(\mathcal{C}(u)) = l(\mathcal{C}'(u))$

*Proof.* Let  $\mathcal{U} = \{u_1, \dots, u_n\}$  and  $\mathcal{C}(u_1) \leq \mathcal{C}(u_2) \leq \dots \leq \mathcal{C}(u_k) = \mathcal{C}_{max}$ . Consider the complete binary tree up to depth  $\mathcal{C}_{max}$  initially all nodes are available to be used as codewords. For  $i = 1, 2, \dots, n$ , place  $\mathcal{C}(u_i)$  at an available node at level  $\mathcal{C}(u_i)$  remove all descendant of  $\mathcal{C}(u_i)$  from the available list.

**Corollary 1.4.1.** Suppose  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$  is u.d., then there exist an  $\mathcal{C}' : \mathcal{U} \rightarrow \{0, 1\}^*$  which is prefix-free and  $l(\mathcal{C}'(n)) = l(\mathcal{C}(n))$

□

**Example 1.**  $\mathcal{U} = \{a, b, c, d\}$ ,  $\mathcal{C} : \{0, 01, 011, 111\}$  and  $\mathcal{C}' : \{0, 10, 110, 111\}$   
In this case, decoding  $\mathcal{C}$  may require delay, while decoding  $\mathcal{C}'$  is instanteneous.

## 2 Alphabet with statistics

Suppose we have an alphabet  $\mathcal{U}$ , and suppose we have a random variable  $\mathcal{U}$  taking values in  $\mathcal{U}$ . We denote by  $p(u) = \Pr(U = u)$ ,  $u \in \mathcal{U}$  with  $p(u) \geq 0$  and  $\sum_u p(u) = 1$ .

Suppose we have a code  $\mathcal{C} : \mathcal{U} \rightarrow \{0, 1\}^*$ . We then have  $\mathcal{C}(u)$  a random binary string and  $l(\mathcal{C}(u))$  a random integer.

**Example 2.**  $\mathcal{U} = \{a, b, c, d\}$   
 $p : \{0.5, 0.25, 0.125, 0.125\}$   
 $\mathcal{C} : \{0, 01, 110, 111\}$

then we have

$$l(\mathcal{C}(u)) = \begin{cases} 1, & p = 0.5 \\ 2, & p = 0.25 \\ 3, & p = 0.125 + 0.125 + 0.25 \end{cases}$$

We can measure how efficient  $\mathcal{C}$  represents  $\mathcal{U}$  by considering

$$E[l(\mathcal{C}(u))] = \sum_u p(u) \mathcal{C}(u) \quad \text{with} \quad \mathcal{C}(u) = l(\mathcal{C}(u))$$

**Theorem 2.1.** if  $\mathcal{C}$  is u.d., then

$$E[l(\mathcal{C}(u))] \geq \sum_u p(u) \log\left(\frac{1}{p(u)}\right)$$

*Proof.* let  $\mathcal{C}(u) = l(\mathcal{C}(u))$ , we know  $\sum_u 2^{-\mathcal{C}(u)} \leq 1$  because  $\mathcal{C}$  is u.d.

$$\begin{aligned} E[l(\mathcal{C}(u))] &= \sum_u p(u) \mathcal{C}(u) = \sum_u p(u) \log_2\left(\frac{1}{q(u)}\right) \\ &\equiv \sum_u p(u) \log\left(\frac{q(u)}{p(u)}\right) \leq 0 \\ &\equiv \sum_u p(u) \ln\left(\frac{q(u)}{p(u)}\right) \leq 0 \\ &\leq \sum_u p(u) \left[\frac{q(u)}{p(u)} - 1\right] = \underbrace{\sum_u q(u)}_{\leq 1} - \underbrace{\sum_u p(u)}_{=1} \leq 0 \end{aligned}$$

□

**Theorem 2.2.** For any  $\mathcal{U}$ , there exists a prefix-free code  $\mathcal{C}$  s.t.

$$E[l(\mathcal{C}(u))] < 1 + \sum_{u \in \mathcal{U}} p(u) \log\left(\frac{1}{p(u)}\right)$$

*Proof.* Given  $\mathcal{U}$ , let

$$\begin{aligned} \mathcal{C}(u) &= \lceil \log\left(\frac{1}{p(u)}\right) \rceil < 1 + \log\left(\frac{1}{p(u)}\right) \\ \Rightarrow \sum_u 2^{-\mathcal{C}(u)} &\leq \sum_u p(u) = 1 \\ \Rightarrow \sum_u p(u) \mathcal{C}(u) &< \sum_u p(u) \log\left(\frac{1}{p(u)}\right) + \underbrace{1}_{\sum p(u)} \end{aligned}$$

□

**Definition 2.1** (Entropy). *Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process.*

**Theorem 2.3.** *The entropy of a random variable  $U \in \mathcal{U}$  is*

$$H(U) = \sum_{u \in \mathcal{U}} p(u) \log\left(\frac{1}{p(u)}\right)$$

with  $p(u) = \Pr(U = u)$

Note that  $H(U)$  is a function of the distribution  $\mathcal{C}_u(\cdot)$  of the random variable  $U$ , it isn't a function of  $U$ .

$$H(U) = E[f(U)] \quad \text{where} \quad f(U) = \log\left(\frac{1}{p(u)}\right)$$

How to design optimal codes (in the sense of minimizing  $E[l(\mathcal{C}(u))]$ )?  
Formally, given a random variable  $U$ , find  $\mathcal{C}(u) \rightarrow \mathcal{N}$  s.t.

$$\sum_{u \in U} 2^{\mathcal{C}(u)} \leq 1 \quad \text{that minimizes} \quad \sum_{u \in U} p(u) \mathcal{C}(u)$$

Properties of optimal prefix-free codes

- if  $p(u) < p(v)$  then  $\mathcal{C}(u) \geq \mathcal{C}(v)$
- The two longest codewords have the same length
- There is an optimal code such that the two least probable letters are assigned codewords that differ in the last bit.

Observe that if  $\{\mathcal{C}(u_1), \dots, \mathcal{C}(u_{k-1}), \mathcal{C}(u_k)\}$  is a prefix-free collection of the property that

$$\begin{aligned} \mathcal{C}(u_{k-1}) &= \alpha 0 \\ \mathcal{C}(u_k) &= \alpha 1 \end{aligned} \quad \text{with} \quad \alpha \in \{0, 1\}^*$$

then  $\{\mathcal{C}(u_1), \dots, \mathcal{C}(u_{k-2}), \alpha\}$  is also a prefix-free collection.  
Also

$$\begin{aligned} \sum_{u \in \mathcal{U}} p(u) l(\mathcal{C}(u)) &= p(u_1) l(\mathcal{C}(u_1)) + \dots + p(u_{k-2}) l(\mathcal{C}(u_{k-2})) + [p(u_{k-1}) + p(u_k)] (l(\alpha) + 1) \\ &= (p(u_{k-1}) + p(u_k)) + \sum_{v \in \mathcal{V}} p(v) l(\mathcal{C}'(v)) \end{aligned}$$

So we have shown that with

$$E[l(\mathcal{C}(U))] = p(u_{k-1}) + p(u_k) + E[l(\mathcal{C}'(v))]$$

if  $\mathcal{C}$  is optimal for  $U$ , then  $\mathcal{C}'$  is optimal for  $V$

### 3 Entropy and mutual information

**Definition 3.1** (Joint entropy). Suppose  $U, V$  are random variables with  $p(u, v) = P(U = u, V = v)$ , the joint entropy is

$$H(UV) = \sum_{u,v} p(u, v) \log \frac{1}{p(u, v)}$$

**Theorem 3.1.**

$$H(UV) \leq H(U) + H(V)$$

with equality iff  $U$  and  $V$  are independants.

*Proof.* We want to show that

$$\sum_{u,v} p(u, v) \log \frac{1}{p(u, v)} \leq \sum_u p(u) \log \frac{1}{p(u)} + \sum_v p(v) \log \frac{1}{p(v)} \iff \sum_{u,v} p(u, v) \log \frac{p(u)p(v)}{p(u, v)} \leq 0$$

We use  $\ln z \leq z - 1 \forall z$  (with equality iff  $z = 1$ ):

$$\sum_{u,v} p(u, v) \log \frac{p(u)p(v)}{p(u, v)} \leq \sum_{u,v} p(u, v) \left[ \frac{p(u)p(v)}{p(u, v)} - 1 \right] = \sum_{u,v} p(u)p(v) - \sum_{u,v} p(u, v) = 1 - 1 = 0$$

□

Same definitions of entropy holds for  $n$  symbols.

**Definition 3.2** (Joint Entropy). Suppose  $U_1, U_2, \dots, U_n$  are RVs and we are given  $p(u_1 \dots u_n)$ , the joint entropy is

$$H(U_1, \dots, U_n) = \sum_{u_1 \dots u_n} p(u_1 \dots u_n) \log \frac{1}{p(u_1 \dots u_n)}$$

**Theorem 3.2.**

$$H(U_1, \dots, U_n) \leq \sum_{i=1}^n H(U_i)$$

with equality iff  $U$ s are independants

**Corollary 3.2.1.** if  $U_1, \dots, U_n$  are i.i.d. then  $H(U_1 \dots U_n) = nH(U_1)$

**Definition 3.3** (Conditional entropy).

$$H(U|V) = \sum_{u,v} p(u, v) \log \frac{1}{p(u|v)}$$

**Theorem 3.3.**

$$H(UV) = H(U) + H(V|U) = H(V) + H(U|V)$$

**Theorem 3.4.**

$$H(U) + H(V) \geq H(U, V) = H(V) + H(U|V)$$

**Definition 3.4** (Mutual information). Mutual information measures the amount of information that can be obtained about one random variable by observing another.

$$\begin{aligned} I(U; V) &= I(V; U) = H(U) - H(U|V) \\ &= H(V) - H(V|U) \\ &= H(U) + H(V) - H(UV) \end{aligned}$$

We can apply the chain rule on the entropy as follow

$$H(U_1, U_2, \dots, U_n) = H(U_1) + H(U_2|U_1) + \dots + H(U_n|U_1, U_2 \dots U_{n-1})$$

**Definition 3.5** (Conditional mutual information).

$$\begin{aligned} I(U; V|W) &= H(U|W) - H(U|VW) \\ &= H(V|W) - H(V|UW) \\ &= \mathbb{E}_{u,v,w} \left[ \log \frac{p(uv|w)}{p(u|w)p(v|w)} \right] \end{aligned}$$

**Theorem 3.5.**

$$I(V; U_1 \dots U_n) = I(V; U_1) + I(V; U_2|U_1) + \dots + I(V; U_n|U_1 \dots U_{n-1})$$

**Notation 1.**

$$U^n \triangleq (U_1, U_2, \dots, U_n)$$

**Theorem 3.6.**

$$I(U; V|W) \geq 0$$

equality iff conditioned on  $w$ ,  $u$  and  $v$  are independant, that is iff  $U - V - W$  is a Markov chain.

*Proof.*

$$\begin{aligned} I(U; V|W) &= \frac{1}{\ln 2} \sum_{u,v,w} p(u, v, w) \ln \frac{p(u|w)p(v|w)}{p(uv|w)} \\ &\geq \frac{1}{\ln 2} \sum_{u,v,w} p(u, v, w) \left[ \frac{p(u|w)p(v|w)}{p(uv|w)} - 1 \right] \\ &= \frac{1}{\ln 2} \sum_{u,v,w} (p(w)p(u|w)p(v|w) - p(uvw)) \\ &= \frac{1}{\ln 2} (1 - 1) \\ &= 0 \end{aligned}$$

□

## 4 Data processing

**Theorem 4.1.**  $U - V - W$  is a MC  $\iff I(U; W|V) = 0$

**Corollary 4.1.1.**  $I(U; V) \geq I(U; W)$  and by symmetry of MC  $I(W; V) \geq I(U; W)$

*Proof.*

$$I(U; VW) = I(U; V) + I(U; W|V) = I(U; V)$$

and

$$I(U; VW) = I(U; W) + I(U; V|W) \geq I(U; W)$$

□

**Theorem 4.2.** Given  $U$  a RV taking values in  $\mathcal{U}$  then  $0 \leq H(U) \leq \log |\mathcal{U}|$ .  $H(U) = 0$  iff  $U$  is constant,  $H(U) = \log |\mathcal{U}|$  iff  $U$  is  $p(u) = 1/|\mathcal{U}|$  for all  $u$ .

*Proof.* For the lower bound,

$$H(U) = \sum_u \underbrace{p(u)}_{\geq 0} \underbrace{\log \frac{1}{p(u)}}_{\geq 0} \geq 0$$

For the upper bound,

$$\begin{aligned} H(U) - \log |\mathcal{U}| &= \sum_u p(u) \log \frac{1}{p(u)} - \sum_u p(u) \log |\mathcal{U}| \\ &= \frac{1}{\ln 2} \sum_u p(u) \ln \frac{1}{|\mathcal{U}|p(u)} \\ &\leq \frac{1}{\ln 2} \sum_u p(u) \left( \frac{1}{|\mathcal{U}|p(u)} - 1 \right) \\ &= \frac{1}{\ln 2} \left[ \sum_u \frac{1}{|\mathcal{U}|} - \sum_u p(u) \right] \\ &= 0 \end{aligned}$$

□

**Theorem 4.3.**  $I(U; V) = 0 \iff U \perp V$

**Definition 4.1** (Entropy rate of a stochastic process).  $\lim_{n \rightarrow \infty} \frac{1}{n} H(U^n)$  if the limit exists.

**Theorem 4.4.** For stationary stochastic process  $U^n$ , the sequences

$$a_n = \frac{1}{n} H(U^n) \text{ and } b_n = H(U_n | U^{n-1})$$

are positive and non increasing. Then  $a = \lim_{n \rightarrow \infty} a_n$  and  $b = \lim_{n \rightarrow \infty} b_n$  exists and  $a = b$ .

*Proof.*

$$\begin{aligned} b_{n+1} &= H(U_{n+1} | U_1, U_2, \dots, U_n) \\ &\leq H(U_{n+1} | U_2, \dots, U_n) \\ &= H(U_n | U_1, U_2, \dots, U_{n-1}) \\ &= b_n, \text{ because } U_1 \dots U_n \sim U_2 \dots U_{n+1} \text{ (Stationarity).} \end{aligned}$$

Hence, it is non-increasing.

For the  $\{a_n\}$ , observe that

$$\begin{aligned} a_n &= \frac{1}{n} H(U^n) = \frac{1}{n} \left[ H(U_1) + H(U_2|U_1) + H(U_3|U^2) + \cdots + H(U_n|U^{n-1}) \right] \\ &= \frac{1}{n} \left[ b_1 + b_2 + \cdots + b_n \right] \end{aligned}$$

and by the "Lemma", whenever  $b_n \rightarrow b$ ,  $a_n \rightarrow b$  □

**Lemma 4.5** (Cesaro). *Suppose  $b_n \rightarrow b$ ,*

*then,*

$$a_n = \frac{1}{n} \left[ b_1 + b_2 + \cdots + b_n \right] \text{ also converges and to } 1.$$

*Proof.* Since  $b_n \rightarrow b$ ,  $\left( \equiv \forall \epsilon > 0, \exists n(\epsilon) \text{ s.t. } \forall n > n(\epsilon) |b_n - b| < \epsilon \right)$

$\exists B \text{ s.t. } |b_n| < B \text{ for all } n.$

Take  $n > n_1(\epsilon) \triangleq \dots$  then

$$\begin{aligned} |a_n - b| &\leq \frac{|b_1 - b| + |b_2 - b| + |b_3 - b| + \cdots + |b_n - b|}{n} \\ \text{so } |a_n - b| &\leq \frac{1}{n} \left[ \sum_{i=1}^{n_0(\epsilon)} \underbrace{|b_i - b|}_{2B} + \sum_{i=n_0(\epsilon)+1}^n \underbrace{|b_i - b|}_{\leq \epsilon} \right] \leq \frac{n_0(\epsilon)2B}{n} + \epsilon < 2\epsilon \\ &\text{for } n > n_1(\epsilon) \triangleq \max, \{n_0(\epsilon) \frac{1}{\epsilon} n_0(\epsilon) 2B\} \end{aligned}$$

□

**Theorem 4.6.** *Given a stationary process with entropy rate  $r$ :*

$$r = \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathcal{U}^n)$$

*then*

1. *for every source coding scheme*

$$\mathcal{C}_n : \mathcal{U}^n \rightarrow \{0, 1\}^*$$

*the expected number of bits / letter is given by*

$$\frac{1}{n} E[l(\mathcal{C}(\mathcal{U}^n))] \geq r$$

2. *for any  $\epsilon > 0$ , there exists a source coding scheme  $\mathcal{C}_n : \mathcal{U}^n \rightarrow \{0, 1\}^*$  s.t.*

$$\frac{1}{n} E[l(\mathcal{C}_n(\mathcal{U}^n))] < r + \epsilon$$

*Proof.* 1. we already know

$$\frac{1}{n} E[l(\mathcal{C}_n(\mathcal{U}^n))] \geq \frac{1}{n} H(\mathcal{U}_1 \dots \mathcal{U}_n)$$

and the right term is decreasing



2. we also know that for each  $n, \exists \mathcal{C}_n$  that is prefix-free s.t.

$$E[l(\mathcal{C}_n(U^n))] < \underbrace{\frac{1}{n}H(\mathcal{U}^n)}_r + \underbrace{\frac{1}{n}}_0$$

we can find  $n$  large enough s.t. the RHS  $< r + \epsilon$

□

## 5 Typicality and typical set

Suppose we have a sequence  $U_1, U_2, \dots$  of i.i.d. random variables taking values in an alphabet  $\mathcal{U}$ . Suppose we observe  $u_1, u_2, \dots, u_n$ . We will call it to be *typical*-( $\epsilon, p$ ) if

$$p(u)(1 - \epsilon) \leq \frac{\# \text{ of times } u \text{ appears in } u_1, \dots, u_n}{n} \leq p(u)(1 + \epsilon)$$

**Theorem 5.1.**  $u^n$  is  $(\epsilon, p)$ -typical then

$$2^{-nH(u)(1+\epsilon)} \leq Pr(U^n = u^n) \leq 2^{-nH(u)(1-\epsilon)}$$

*Proof.*

$$Pr(U^n = u^n) = \prod_{i=1}^n Pr(U_i = u_i) = \prod_{i=1}^n p(u_i) = \prod_{u \in \mathcal{U}} p(u)^{\#_u}$$

with  $\#_u$  the number of times  $u$  appears in  $u_1, \dots, u_n$  where

$$n(1 - \epsilon)p(u) \leq \#_u \leq n(1 + \epsilon)p(u)$$

consequently

$$p(u)^{np(u)(1-\epsilon)} \geq p(u)^{\#_u} \geq p(u)^{np(u)(1+\epsilon)}$$

then

$$\left(\prod_n p(u)^{p(u)}\right)^{(1-\epsilon)n} \geq Pr(U^n = u^n) \geq \left(\prod_n p(u)^{p(u)}\right)^{(1+\epsilon)n}$$

but

$$p(u)^{p(u)} = 2^{-p(u) \log(\frac{1}{p(u)})} \Rightarrow \prod p(u)^{p(u)} = 2^{-H(u)}$$

□

**Definition 5.1** (Typical set).

$$T(n, \epsilon, p) = \{u^n \in U^n : u^n \text{ is } (\epsilon, p)\text{-typical}\}$$

**Theorem 5.2.** 1. if  $u^n \in T(n, \epsilon, p)$  then

$$p(u^n) = Pr(U^n = u^n) = 2^{-nH(u)(1 \pm \epsilon)}$$

when  $U_i$  i.i.d.

2.

$$\lim_{n \rightarrow \infty} Pr(U^n \in T(n, \epsilon, p)) = 1$$

3.

$$|T(n, \epsilon, p)| \leq 2^{n(H(u)(1+\epsilon))}$$

4.

$$|T(n, \epsilon, p)| \geq (1 - \epsilon)2^{nH(u)(1-\epsilon)}$$

*Proof.* **TODO:**

□

**Definition 5.2** (Kullback-Leiber divergence (information gain)). *If we compress data in a manner that assumes  $q(u)$  is the distribution underlying some data, when, in reality,  $p(u)$  is the correct distribution, the Kullback-Leiber divergence is the number of average additional bits per datum necessary for compression.*

**Lemma 5.3.** *if  $U_1 \dots U_n$  are i.i.d. with distribution  $q$  and  $u_1 \dots u_n$  is  $(\epsilon, p)$ -typical, then*

$$\begin{aligned} \Pr \{U^n = u^n\} &= \left[ \prod_u q(u)^{p(u)} \right]^{n(1+\epsilon)} \\ &= 2^{-n(1+\epsilon)} \sum_u p(u) \log \frac{1}{q(u)} \end{aligned}$$

We know that  $\Pr \{U^n \in T(n, \epsilon, p)\} \rightarrow 1$  as  $n \rightarrow \infty$  and

$$(1 - \epsilon)2^{nH(U)(1-\epsilon)} \leq |T(n, \epsilon, p)| \leq 2^{nH(U)(1+\epsilon)}$$

**TODO:** Something wrong in the formula above

**Observation 5.1.** *Suppose  $U_1 \dots U_n$  are i.i.d. following  $q$  and  $u^n \in T(n, \epsilon, p)$*

$$\left[ \prod_u q(u)^{p(u)} \right]^{n(1+\epsilon)} \leq \Pr \{U^n = u^n\} \leq \left[ \prod_u q(u)^{p(u)} \right]^{n(1-\epsilon)}$$

and

$$\begin{aligned} \prod_u q(u)^{p(u)} &= 2^{-\sum p(u) \log \frac{1}{q(u)}} \\ \sum_u p(u) \log \frac{1}{q(u)} &= \underbrace{\sum_u p(u) \log \frac{1}{p(u)}}_{H(p)} + \underbrace{\sum_u p(u) \log \frac{p(u)}{q(u)}}_{D(p||q)} \end{aligned}$$

**Corollary 5.3.1.** *if  $U_1 \dots U_n$  are i.i.d. following distribution  $q$ , then*

$$2^{-n[(1+\epsilon)D(p||q)+2\epsilon H(p)]} \leq \Pr \{U^n \in T(n, \epsilon, p)\} \leq 2^{-n[(1-\epsilon)D(p||q)-2\epsilon H(p)]}$$

*Proof.*

$$\Pr \{U^n \in T(n, \epsilon, p)\} = \sum_{u^n \in T(n, \epsilon, p)} \Pr \{U^n = u^n\}$$

We have

$$\begin{aligned} 2^{-n[H(p)+D(p||q)](1+\epsilon)} &\leq \Pr \{U^n = u^n\} \leq 2^{-n[H(p)+D(p||q)](1-\epsilon)} \\ 2^{nH(p)(1-\epsilon)} &\leq |T(n, \epsilon, p)| \leq 2^{nH(p)(1+\epsilon)} \end{aligned}$$

□

**Observation 5.2.**

$$D(p||q) = \sum_u p(u) \log \left( \frac{p(u)}{q(u)} \right) \geq 0 \text{ with equality iff } p = q$$

**Example 3.**  $U \in \{0, 1\}$ ,  $p = \frac{1}{2}$ ,  $q = \frac{1}{2} - \delta$ ,  $\frac{1}{2} + \delta$

$$D(p||q) = \frac{1}{2} \log \frac{1}{1-2\delta} + \frac{1}{2} \log \frac{1}{1+2\delta} = \frac{1}{2} \log \frac{1}{1-4\delta^2} = -\frac{1}{2} \log(1-4\delta^2) \approx \frac{1}{2} 4\delta^2 + o(\delta^4)$$

So if we want  $2^{-nD(p||q)}$  small  $n = \Omega(1/\delta^2)$

**Example 4.** Suppose we are told that  $U$  is  $p$  distributed and  $p(u)$  are powers of 2. We design a prefix-free code  $\mathcal{C}$  to minimize  $\sum_u p(u)l(\mathcal{C}(u))$ . We have been misinformed and  $U \sim q$ , then:

$$\begin{aligned} E[l(\mathcal{C}(u))] &= \sum_u q(u) \log \frac{1}{p(u)} \\ &= \underbrace{H(q)}_{\text{length for optimal code}} + \underbrace{D(q||p)}_{\text{penalty for misbelief}} \end{aligned}$$

## 5.1 Universal data compression

Suppose we know that the distribution  $p$  of  $U$  is either  $p_1, p_2 \dots p_k$ , can we design a code  $\mathcal{C} : U \rightarrow \{0, 1\}^*$

$$E[l(\mathcal{C}(U))] \leq H(U) + \text{small for every } p$$

$$E\left[\frac{1}{n}l(\mathcal{C}(U))\right] \leq o(n) + E\left[h_2\left(\frac{K}{n}\right)\right]$$

with  $K = \sum_{i=1}^n u_i$

We have  $\frac{E[K]}{n} = \theta_1$  and  $E\left[h_2\left(\frac{K}{n}\right)\right] \leq h_2\left(E\left[\frac{K}{n}\right]\right) = h_2(\theta)$

**Design  $\mathcal{C}$**  Because the probability of a bit string is only dependant of the number of 1s (or 0s), it makes sense to encode two strings with the same numbers of 1 with code words of same lengths. Given  $u_1 \dots u_n \in \{0, 1\}^n$ , first count the number of 1, call it  $k$ .

$$\mathcal{C}(u_1 \dots u_n) = \underbrace{\text{describe } k}_{\lceil \log(n+1) \rceil} \underbrace{\text{describe } u_1 \dots u_n}_{\lceil \log \binom{n}{k} \rceil}$$

We now want to evaluate

$$\frac{1}{n} E[l(\mathcal{C}(U))]$$

when  $U_1 \dots U_n$  are i.i.d with  $p_1 = \theta$  and  $p_0 = 1 - p_1$

**Observation 5.3.** for any  $0 \leq \alpha \leq 1$

$$\begin{aligned} 1 = 1^n &= (\alpha + (1 - \alpha))^n = \sum_{i=0}^n \binom{n}{i} \alpha^i (1 - \alpha)^{n-i} \\ &\geq \binom{n}{k} \alpha^k (1 - \alpha)^{n-k} \end{aligned}$$

Then for all  $\alpha$

$$\binom{n}{k} \leq \alpha^{-k} (1 - \alpha)^{-(n-k)} = 2^{-n(\frac{k}{n} \log \frac{1}{\alpha} + (1 - \frac{k}{n}) \log \frac{1}{1-\alpha})}$$

We pick  $\alpha = \frac{k}{n}$ , and we get

$$\binom{n}{k} < 2^{nh_2(\frac{k}{n})}$$

Using this bound we have

$$\frac{1}{n}l(\mathcal{C}(u_1 \dots u_n)) \leq \frac{2}{n} + \frac{\log(n+1)}{n} + h_2\left(\frac{k}{n}\right)$$

$$E\left[\frac{1}{n}l(\mathcal{C}(U))\right] \leq \frac{2}{n} + \frac{\log(n+1)}{n} + h_2\left(\frac{k}{n}\right)$$

**Claim 5.1.** Suppose  $U_i$  are i.i.d. with  $Pr\{U_1 = 1\} = \theta$ . We have  $E\left[\frac{k}{n}\right] = \theta$  and  $E\left[h_2\left(\frac{k}{n}\right)\right] \leq h_2(E\left[\frac{k}{n}\right]) = h_2(\theta)$ . So

$$\lim_{n \rightarrow \infty} \frac{1}{n} E[l(\mathcal{C}(u_1 \dots u_n))] \leq h_2(\theta)$$

consequently this scheme is asymptotically optimal.

*Proof.* To prove the claim we need to show that if  $\beta_1 \dots \beta_k$  are in  $[0, 1]$  and  $q_1 \dots q_k$  are non negative numbers that sum to 1 then

$$\sum_{i=1}^k q_i h_2(\beta_i) \leq h_2\left(\sum_{i=1}^k q_i \beta_i\right)$$

Let  $U$  and  $V$  be random variables with  $U \in \{0, 1\}$  and  $V \in \{1, \dots, k\}$  with

$$\begin{aligned} Pr\{V = i\} &= q_i \\ Pr\{U = 1|V = i\} &= \beta_i \\ Pr\{U = 0|V = i\} &= 1 - \beta_i \end{aligned}$$

Then,

$$\begin{aligned} Pr\{U = 1\} &= \sum_i q_i \beta_i \\ H(U) &= h_2\left(\sum_i q_i \beta_i\right) \\ H(U|V) &= \sum_i q_i h_2(\beta_i) \end{aligned}$$

And we already know that  $H(U) \geq H(U|V)$  □

**TODO: THOMAS SCRIBE ICI**

Suppose we have an infinite string  $u_1 u_2 \dots, u_i \in U$ , and

$$u_1 u_2 \dots = v_1 v_2 \dots \text{ with } v_i \in U^*, v_i \neq v_j \text{ when } i \neq j$$

for any  $k$  we have

$$\lim_{m \rightarrow \infty} \frac{\text{length}(v_1 \dots v_m)}{m} \geq k \Rightarrow \lim_{m \rightarrow \infty} \frac{\text{length}(v_1 \dots v_m)}{m} = \infty$$

**Definition 5.3.** Given an infinite string  $u_1 u_2 \dots$  and a machine  $M$ , let

$$\rho_M(u_1 u_2 \dots) = \lim_{n \rightarrow \infty} \frac{\text{length of the output } M \text{ after reading } u_1 u_2 \dots}{n}$$

also given  $s > 0$ , define

- The compressibility of  $U^*$  by  $s$ -state machines

$$\rho_s(u_1 u_2 \dots) = \min_M \rho_M(u_1 u_2 \dots)$$

with  $M$  an  $s'$ -state machine with  $s' \leq s$

- Compressibility of  $U^*$  by finite state machines

$$\rho_{FSM}(u_1 u_2 \dots) = \lim_{s \rightarrow \infty} \rho_s(u_1 u_2 \dots)$$

**Definition 5.4.** Suppose  $u_1 u_2 \dots$  an infinite sequence, define  $m(n)$  as the largest  $m$  for which  $u_1 \dots u_n = v_1 \dots v_m$  with distinct  $v_1 \dots v_m$

**Example 5.**

$$u = aaaaaaaaaa, \quad \underbrace{\emptyset}_{v_1} \underbrace{a}_{v_2} \underbrace{aa}_{v_3} \underbrace{aaa}_{v_4} \underbrace{aaaa}_{v_5} \Rightarrow m(10) = 5$$

So far we know that

$$\frac{\text{length of the output of any } s\text{-state IL machine when it reads } u_1 u_2 \dots}{n} \geq \frac{m(n) \log(\frac{m(n)}{8s^2})}{n}$$

with

$$\frac{m(n) \log(\frac{m(n)}{8s^2})}{n} = \frac{m(n) \log(m(n))}{n} - \frac{m(n) \log(8s^2)}{\text{length}(v_1 \dots v_m)}$$

hence if  $M$  is a  $s$ -state machine

$$\rho_M(u_1 u_2 \dots) \geq \overline{\lim}_{n \rightarrow \infty} \frac{m(n) \log(m(n))}{n} \quad \text{then} \quad \rho_{FSM}(u_1 u_2 \dots) \geq \overline{\lim}_{n \rightarrow \infty} \frac{m(n) \log m(n)}{n}$$

## 6 Lemple-Ziv data compression method

Given some alphabet  $U$  to both encoder and decoder, they also agree an order on  $U$ :

1. Start with a dictionary  $D = U$
2. To each word  $w \in D$ , assign a  $\lceil \log |D| \rceil$ -bit binary description in the dictionary order
3. Parse the first word in  $u_1 u_2 \dots$  in the dictionary, output its binary description
4. replace  $w$  in  $D$  by  $\{wu, \forall u \in U\}$ .
5. Go to 2.

**Example 6.** Define an alphabet  $U = \{a, b, c\}$  with  $a \leq b \leq c$  and an input message

$$u = bbacac$$

- Create the dictionary  $D = \{a, b, c\}$  and its corresponding binary description  $D_{bin} = \{00, 01, 10\}$
- The first word in the message is 'a', output its binary description

$$output = 01$$

- Update the dictionary:

$$D = \{a, ba, bb, bc, c\} \quad D_{bin} = \{000, 001, 010, 011, 100\}$$

- Parse the next word 'ba' and output its binary description

$$output = 01001$$

- Update the dictionary

$$D = \{a, baa, bab, bac, bb, bc, c\} \quad D_{bin} = \{000, 001, \dots\}$$

- Continue until the end of the input data...

The decoder can proceed in a similar way to iteratively update the dictionary while decoding the message.

### 6.1 Analysis of LZ

Observe that LZ parses the string  $u_1 u_2 \dots$  into  $v_1 v_2 \dots$  with  $v_i \in U^*$  or  $v_i \in D_i$  where  $D_i$  is the dictionary at step  $i$ .

When going from iteration  $i \rightarrow i + 1$ ,  $v_i$  is removed from  $D$ , consequently  $v_1, v_2, v_3$  are distinct.

The length of the output of LZ after reading  $v_1 \dots v_m$  is given by

$$\text{LZ output's length} = \lceil \log |U| \rceil + \lceil \log(2|U| - 1) \rceil + \lceil \log(3|U| - 2) \rceil + \dots + \lceil \log(m|U| - m + 1) \rceil$$

we observe that

$$\text{LZ output's length} < m(\log(m|U|) + 1) = m \log(2m|U|)$$

Also we have

$$\begin{aligned} \# \text{ bits / letter} &< \frac{m \log(2m|U|)}{\text{length}(v_1 \dots v_m)} \\ &= \frac{m \log(m)}{\text{length}(v_1 \dots v_m)} + \frac{m \log(2|U|)}{\text{length}(v_1 \dots v_m)} \end{aligned}$$

therefore

$$\rho_{LZ}(u_1 u_2 \dots) = \lim_{m \rightarrow \infty} \frac{\# \text{ bits}}{\text{letter}} \leq \lim_{m \rightarrow \infty} \frac{m \log(m)}{\text{length}(v_1 \dots v_m)} \leq \lim_{n \rightarrow \infty} \frac{m(n) \log(m(n))}{n} \leq \rho_{FSM}(u_1 u_2 \dots)$$

So we have proved the following theorem:

**Theorem 6.1.** *for every  $u_1u_2\dots$*

$$\rho_{LZ}(u_1u_2\dots) \leq \rho_{FSM}(u_1u_2\dots)$$

**Corollary 6.1.1.** *if  $u_1u_2\dots$  is stationary*

$$\rho_{LZ}(u_1u_2\dots) = \text{entropy rate of } u_1u_2\dots$$

# Appendices

## A Markov chains

$U_1 - U_2 - \dots - U_n$  forms a Markov chain if the joint probability distribution of the RVs is

$$p(a, b, c, d) = p(a)p(b|a)p(c|b)p(d|c)$$

which is equivalent to  $(U_1, \dots, U_{k-1})$  are independant of  $(U_{k+1}, \dots, U_n)$  when conditioned on  $U_k$  for any  $k$ .

**Theorem A.1.** *The reverse of a MC is a MC*

## B Stochastic processes

A stochastic process is a collection  $U_1, U_2 \dots U_n$  of RVs each taking values in  $\mathcal{U}$ . It is described by its joint probability

$$p(u^n) = P(U_1 \dots U_n = u_1 \dots u_n) = P(U^n = u^n)$$

**Definition B.1** (Stationary stochastic process). *A process  $U_1, U_2, \dots$  is called stationary if for every  $n$  and  $k$  and  $u_1 \dots u_n$ , we have*

$$p(u^n) = p(U_1 \dots U_n = u_1 \dots u_n) = p(U_{1+k} \dots U_{n+k} = u_1 \dots u_n)$$

*In other words, the process is time shift invariant.*