

One Part of Shannon's Source Coding Theorem

Quentin Hibon

September 26, 2023

Abstract

This document contains a proof of the necessary condition on the code rate of a source code, namely that this code rate is bounded by the entropy of the source. This represents one half of Shannon's source coding theorem, which is itself an equivalence.

This proof is taken directly from the textbook [1], and transcribed rather literally into Isabelle. It is thus easier to keep the textbook proof in mind to understand this formal proof.

Contents

1 Basic types	1
2 Locale for the source coding theorem	2
3 Source coding theorem, direct: the entropy is a lower bound of the code rate	2
3.1 The letter set	2
3.2 Codes and words	2
3.3 Related to the Kraft theorem	3
3.4 Inequality of the kraft sum (source coding theorem, direct) .	4
3.4.1 Sum manipulation lemmas and McMillan theorem . .	4
3.4.2 Technical lemmas about the logarithm	6
3.4.3 KL divergence and properties	6

```
theory Source-Coding-Theorem  
imports HOL-Probability.Information  
begin
```

1 Basic types

```
type-synonym bit = bool  
type-synonym bword = bit list  
type-synonym letter = nat
```

type-synonym 'b word = 'b list

type-synonym 'b encoder = 'b word \Rightarrow bword

type-synonym 'b decoder = bword \Rightarrow 'b word option

2 Locale for the source coding theorem

locale source-code = information-space +

fixes fi :: 'b \Rightarrow real

fixes X :: 'a \Rightarrow 'b

assumes distr-i: simple-distributed M X fi

assumes b-val: b = 2

fixes enc::'b encoder

fixes dec::'b decoder

assumes real-code:

dec (enc x) = Some x

enc w = [] \longleftrightarrow w = []

x \neq [] \longrightarrow enc x = enc [hd x] @ enc (tl x)

3 Source coding theorem, direct: the entropy is a lower bound of the code rate

context source-code

begin

3.1 The letter set

definition L :: 'b set where

L \equiv X ' space M

lemma fin-L: finite L

<proof>

lemma emp-L: L \neq {}

<proof>

3.2 Codes and words

abbreviation real-word :: 'b word \Rightarrow bool where

real-word w \equiv (set w \subseteq L)

abbreviation k-words :: nat \Rightarrow ('b word) set where

k-words k \equiv {w. length w = k \wedge real-word w}

lemma rw-tail:

assumes real-word w

shows $w = [] \vee \text{real-word } (tl\ w)$
 ⟨proof⟩

definition $\text{code-word-length} :: 'e\ \text{encoder} \Rightarrow 'e \Rightarrow \text{nat}$ **where**
 $\text{code-word-length } e\ l = \text{length } (e\ [l])$

abbreviation $\text{cw-len} :: 'b \Rightarrow \text{nat}$ **where**
 $\text{cw-len } l \equiv \text{code-word-length } \text{enc } l$

definition $\text{code-rate} :: 'e\ \text{encoder} \Rightarrow ('a \Rightarrow 'e) \Rightarrow \text{real}$ **where**
 $\text{code-rate } e\ Xo = \text{expectation } (\lambda a. (\text{code-word-length } e\ ((Xo)\ a)))$

lemma $\text{fi-pos}: i \in L \Longrightarrow 0 \leq \text{fi } i$
 ⟨proof⟩

lemma (**in** prob-space) simp-exp-composed :
assumes X : $\text{simple-distributed } M\ X\ Px$
shows $\text{expectation } (\lambda a. f\ (X\ a)) = (\sum x \in X.\ \text{space } M. f\ x * Px\ x)$
 ⟨proof⟩

lemma cr-rw :
 $\text{code-rate } \text{enc } X = (\sum i \in X.\ \text{space } M. \text{fi } i * \text{cw-len } i)$
 ⟨proof⟩

abbreviation $\text{cw-len-concat} :: 'b\ \text{word} \Rightarrow \text{nat}$ **where**
 $\text{cw-len-concat } w \equiv \text{foldr } (\lambda x\ s. (\text{cw-len } x) + s)\ w\ 0$

lemma cw-len-length : $\text{cw-len-concat } w = \text{length } (\text{enc } w)$
 ⟨proof⟩

lemma maj-fold :
assumes $\bigwedge l. l \in L \Longrightarrow f\ l \leq \text{bound}$
assumes $\text{real-word } w$
shows $\text{foldr } (\lambda x\ s. f\ x + s)\ w\ 0 \leq \text{length } w * \text{bound}$
 ⟨proof⟩

definition $\text{max-len} :: \text{nat}$ **where**
 $\text{max-len} = \text{Max } ((\lambda x. \text{cw-len } x)\ 'L)$

lemma max-cw :
 $l \in L \Longrightarrow \text{cw-len } l \leq \text{max-len}$
 ⟨proof⟩

3.3 Related to the Kraft theorem

definition $\mathcal{K} :: \text{real}$ **where**
 $\mathcal{K} = (\sum i \in L. 1 / b \wedge (\text{cw-len } i))$

lemma pos-cw-len : $0 < 1 / b \wedge \text{cw-len } i$ ⟨proof⟩

lemma \mathcal{K} -pos: $0 < \mathcal{K}$

\langle proof \rangle

lemma \mathcal{K} -pow: $\mathcal{K} = (\sum i \in L. 1 / b^{\text{powr } cw\text{-len } i})$

\langle proof \rangle

lemma k -words-rel:

$k\text{-words } (Suc\ k) = \{w. (hd\ w \in L \wedge tl\ w \in k\text{-words } k \wedge w \neq [])\}$

\langle proof \rangle

lemma bij - k -words:

shows $bij\text{-betw } (\lambda wi. Cons\ (fst\ wi)\ (snd\ wi))\ (L \times k\text{-words } k)\ (k\text{-words } (Suc\ k))$

\langle proof \rangle

lemma $finite$ - k -words: $finite\ (k\text{-words } k)$

\langle proof \rangle

lemma $cartesian$ -product:

fixes $f::('c \Rightarrow real)$

fixes $g::('d \Rightarrow real)$

assumes $finite\ A$

assumes $finite\ B$

shows $(\sum b \in B. g\ b) * (\sum a \in A. f\ a) = (\sum ab \in A \times B. f\ (fst\ ab) * g\ (snd\ ab))$

\langle proof \rangle

lemma \mathcal{K} -power:

shows $\mathcal{K}^k = (\sum w \in (k\text{-words } k). 1 / b^{\wedge(cw\text{-len-concat } w)})$

\langle proof \rangle

lemma $bound$ - len - $concat$:

shows $w \in k\text{-words } k \implies cw\text{-len-concat } w \leq k * max\text{-len}$

\langle proof \rangle

3.4 Inequality of the kraft sum (source coding theorem, direct)

3.4.1 Sum manipulation lemmas and McMillan theorem

lemma sum - $vimage$ - $proof$:

fixes $g::nat \Rightarrow real$

assumes $\bigwedge w. f\ w < bd$

shows $finite\ S \implies (\sum w \in S. g\ (f\ w)) = (\sum m=0..<bd. (card\ ((f^{-1}\{m\}) \cap S)) * g\ m)$

(is $- \implies - = (\sum m=0..<bd. ?ff\ m\ S))$

\langle proof \rangle

lemma sum - $vimage$:

fixes $g::nat \Rightarrow real$

assumes $bounded: \bigwedge w. w \in S \implies f\ w < bd$ **and** $0 < bd$

assumes *finite: finite S*
shows $(\sum w \in S. g (f w)) = (\sum m=0..<bd. (card ((f - \{m\}) \cap S)) * g m)$
(is ?s1 = ?s2)
 $\langle proof \rangle$

lemma *K-rw*:
 $(\sum w \in (k\text{-words } k). 1 / b^{\wedge}(cw\text{-len-concat } w)) = (\sum m=0..<Suc (k * max-len). card (k\text{-words } k \cap ((cw\text{-len-concat}) - \{m\})) * (1 / b^{\wedge}m))$ **(is ?L = ?R)**
 $\langle proof \rangle$

definition *set-of-k-words-length-m* :: *nat* \Rightarrow *nat* \Rightarrow 'b word set **where**
set-of-k-words-length-m k m = {xk. xk \in k-words k} \cap (cw-len-concat) - {m}

lemma *am-inj-code*: *inj-on enc ((cw-len-concat) - {m})* **(is inj-on - ?s)**
 $\langle proof \rangle$

lemma *img-inc*: *enc 'cw-len-concat - {m} \subseteq {bl. length bl = m}* $\langle proof \rangle$

lemma *bool-lists-card*: *card {bl::bool list. length bl = m} = b^{\wedge}m*
 $\langle proof \rangle$

lemma *bool-list-fin*: *finite {bl::bool list. length bl = m}*
 $\langle proof \rangle$

lemma *set-of-k-words-bound*:
shows *card (set-of-k-words-length-m k m) \leq b^{\wedge}m* **(is ?c \leq ?b)**
 $\langle proof \rangle$

lemma *empty-set-k-words*:
assumes *0 < k*
shows *set-of-k-words-length-m k 0 = {}*
 $\langle proof \rangle$

lemma *K-rw2*:
assumes *0 < k*
shows $(\sum m=0..<Suc (k * max-len). card (set-of-k-words-length-m k m) / b^{\wedge}m) \leq (k * max-len)$
 $\langle proof \rangle$

lemma *K-power-bound* :
assumes *0 < k*
shows $K^{\wedge}k \leq k * max-len$
 $\langle proof \rangle$

theorem *McMillan* :
shows $K \leq 1$
 $\langle proof \rangle$

lemma *entropy-rw*: $\mathcal{H}(X) = -(\sum_{i \in L} f_i \log_b(f_i))$
 ⟨proof⟩

3.4.2 Technical lemmas about the logarithm

lemma *log-mult-ext3*:

$0 \leq x \implies 0 < y \implies 0 < z \implies x * \log_b(x*y*z) = x * \log_b(x*y) + x * \log_b z$
 ⟨proof⟩

lemma *log-mult-ext2*:

$0 \leq x \implies 0 < y \implies x * \log_b(x*y) = x * \log_b x + x * \log_b y$
 ⟨proof⟩

3.4.3 KL divergence and properties

definition *KL-div* :: 'b set \Rightarrow ('b \Rightarrow real) \Rightarrow ('b \Rightarrow real) \Rightarrow real **where**
 $KL\text{-div } S \ a \ d = (\sum_{i \in S} a_i * \log_b(a_i / d_i))$

lemma *KL-div-mul*:

assumes $0 < d \ d \leq 1$
assumes $\bigwedge i. i \in S \implies 0 \leq a_i$
assumes $\bigwedge i. i \in S \implies 0 < e_i$

shows $KL\text{-div } S \ a \ e \geq KL\text{-div } S \ a \ (\lambda i. e_i / d)$
 ⟨proof⟩

lemma *KL-div-pos*:

fixes $a \ e :: 'b \Rightarrow$ real
assumes *fin*: finite S
assumes *nemp*: $S \neq \{\}$
assumes *non-null*: $\bigwedge i. i \in S \implies 0 < a_i \ \bigwedge i. i \in S \implies 0 < e_i$
assumes *sum-a-one*: $(\sum_{i \in S} a_i) = 1$
assumes *sum-c-one*: $(\sum_{i \in S} e_i) = 1$

shows $0 \leq KL\text{-div } S \ a \ e$
 ⟨proof⟩

lemma *KL-div-pos-emp*:

$0 \leq KL\text{-div } \{\} \ a \ e$ ⟨proof⟩

lemma *KL-div-pos-gen*:

fixes $a \ d :: 'b \Rightarrow$ real
assumes *fin*: finite S
assumes *non-null*: $\bigwedge i. i \in S \implies 0 < a_i \ \bigwedge i. i \in S \implies 0 < d_i$
assumes *sum-a-one*: $(\sum_{i \in S} a_i) = 1$
assumes *sum-d-one*: $(\sum_{i \in S} d_i) = 1$

shows $0 \leq KL\text{-div } S \ a \ d$
 ⟨proof⟩

theorem *KL-div-pos2*:

fixes $a \ d :: 'b \Rightarrow$ real

assumes *fin*: *finite S*
assumes *non-null*: $\bigwedge i. i \in S \implies 0 \leq a \ i \ \bigwedge i. i \in S \implies 0 < d \ i$
assumes *sum-a-one*: $(\sum i \in S. a \ i) = 1$
assumes *sum-c-one*: $(\sum i \in S. d \ i) = 1$
shows $0 \leq KL\text{-div } S \ a \ d$
 <proof>

lemma *sum-div-1*:
fixes *f*::'b \Rightarrow 'c::field
assumes $(\sum i \in A. f \ i) \neq 0$
shows $(\sum i \in A. f \ i / (\sum j \in A. f \ j)) = 1$
 <proof>

theorem *rate-lower-bound*:
shows $\mathcal{H}(X) \leq \text{code-rate enc } X$
 <proof>

end

end

References

- [1] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.