Higher-Order Lossless Source Codes

(1) Block codes (aka dictionary-based codes)

- (a) Fixed-to-fixed length codes (FFLC)
- (b) Fixed-to-variable length codes (FVLC)
  (including the first-order codes discussed earlier)
- (c) Variable-to-fixed-length (VFLC)
- (d) Variable-to-variable-length codes (VVLC)
- (e) Run-length codes (a type VFLC or VVLC)

(2) Conditional codes

(3) Arithmetic codes

(4) Universal/adaptive codes

Forward (two pass)
or Backward/one pass

Dictionary based (principally, Lempel-Ziv type) or
Statistical (principally, adaptive arithmetic coding)

We will discuss (1b) and (2) here. Will discuss others later as time permits.

Example of FVLC

Source: stationary rand. process $\{X_n\}$, alphabet $A = \{1,2,3\}$, $H(X) = \log_2 3 = 1.585$

$$P(X_n=i) = \frac{1}{3}, \quad P(X_{n+1}=j|X_n=i) = \begin{cases} \frac{1}{2}, & j=i \\ \frac{1}{4}, & j\neq i \end{cases}$$

first-order code

P(X=x)	x	codeword
1/3	1	0
1/3	2	10
1/3	3	11
rate		1.67

second-order code

$P(X_n=x_1,X_{n+1}=x_2)$	x_1x_2	code word
1/6	11	000
1/12	12	0110
1/12	13	0111
1/12	21	100
1/6	22	001
1/12	23	101
1/12	31	110
1/12	32	111
1/6	33	010
avg length	19/6	
rate	1.583	

How is it that we have beaten the entropy?

As previously defined, the entropy is a lower bound only to the performance of first-order coding.

Formalities of Fixed-to-Variable-Length Block Codes

Source

Discrete-time, discrete-valued with alphabet $A = \{a_1,...\} = \{1, 2, ...\}$, finite or countably infinite

Modeled as a discrete-time random process

Notation: X or $\{X_n\}$ or $\{X_n\}_{n=-\infty}^{\infty}$ or $\{X_n\}_{n=1}^{\infty}$

Assume probability distribution is known, i.e. $\Pr(X_1=x_1,\ldots,X_k=x_k)$ is known for all choices of k and $(x_1,\ldots,x_k)$

Assume X is stationary, i.e.

$$\Pr(X_1=x_1,\ldots,X_k=x_k) = \Pr(X_{n+1}=x_1,\ldots,X_{n+k}=x_k) \quad \text{for any } n \text{ and } x_1,\ldots,x_k$$

Sequence notation:

A^k = all sequences $\underline{x} = (x_1,\ldots,x_k)$ of length k from alphabet A,

If A has M symbols, then A^k has M^k sequences.

Alternate notation: $A^k = \{\underline{a}_1, \underline{a}_2, \ldots\}$, where each $\underline{a}_i$ is some sequence of length k from A.

Probabilities: $(P_1, P_2,\ldots)$, where $P_i = \Pr((X_1,\ldots,X_k) = \underline{a}_i)$ or $p(\underline{x}) = p_{\underline{X}}(\underline{x}) = \Pr(\underline{X}=\underline{x})$

Fixed-to-variable length (FVL) Codes

An FVL code with (input) blocklength (aka "order" or "dimension") k

divides the source sequence into blocks of length k

encodes each block with a uniquely decodable, variable-length code having one codeword for every possible source sequence of length k.

Thus, it is essentially the same as a first-order variable-length code, except that a larger codebook is needed because one codeword is needed for each block of length k, as opposed to a smaller codebook with one codeword for each individual source symbols.

The key characteristics of such a code are:

k = blocklength,

$C = \{\underline{v}_1, \underline{v}_2, \ldots\}$ = codebook, uniquely decodeable (usually prefix), with one codeword for each sequence $\underline{x}$ in A^k,

Codeword lengths $l_1, l_2, \ldots$, where $l_i = l(\underline{v}_i) = l(i) = l(\underline{a}_i)$

Each codeword has the form $\underline{v}_i = (v_{i,1},\ldots,v_{i,l_i}) \in \{0,1\}^*$,

where $\{0,1\}^*$ is the set of all finite-length binary sequences.

e = encoding rule, which assigns a codeword in C to each $\underline{x}$ in A^k.

Specifically, $\underline{z} = e(\underline{x}) = \underline{v}_i$ when $\underline{x} = \underline{a}_i$.

decoding procedure depends on the codebook.

The operation of an FVL code with blocklength $k = 3$ is illustrated below:

$$\begin{array}{l} \text{source sequence} \\ \\ \text{encoded binary sequence} \end{array} \quad \begin{array}{c} \underbrace{X_1\ X_2\ X_3}\ \ \underbrace{X_4\ X_5\ X_6}\ \ \underbrace{X_7\ X_8\ X_9}\ \ \underbrace{X_{10}\cdots} \\ \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \\ e(X_1 X_2 X_3)\ \ e(X_4 X_5 X_6)\ \ e(X_7 X_8 X_9)\ \ e(X_{10}\cdots) \end{array}$$

The performance of the code is determined by its

Average length

$$\bar{L} = E\, l(\underline{X}) = E\, l(e(\underline{X})) = \sum_{\underline{x} \in A^k} l(\underline{x})\, p(\underline{x}) = \sum_i l_i P_i$$

Rate

$$R = \frac{\bar{L}}{k} \quad \text{bits/source symbol}$$

Key Questions

1. How to design blocklength FVL codes with input blocklength k small average length or rate?

2. How small can be the rate of an FVL code with blockength k?

3. How small can be the rate of any FVL code with any blocklength whatsoever? What value of k is best?

Answers

1. How to design FVL codes with small average length or rate?

Use same approaches as before, i.e. apply Shannon or Huffman design strategies to the set containing probabilities of $(P_1, P_2, \ldots)$, equivalently to $\{p(\underline{x}):\ \underline{x} \in A^k\}$

2. How small can be the rate of an FVL code with blockength k?

Define:

$\bar{L}_k^*$ = smallest avg length among all prefix (or UD) codes with blocklength k

$R_k^* = \frac{\bar{L}_k^*}{k}$ = smallest rate among all FVL: codes with blocklength k

From the Coding Theorem for first-order codes, we deduce

FVL Coding Theorem 1:

$$H(X_1,\ldots,X_k) \le \bar{L}_k^* < H(X_1,\ldots,X_k) + 1$$

$$H_k(X) \le R_k^* < H_k(X) + \frac{1}{k}$$

where

$$H(X_1,\ldots,X_k) = -\sum_i P_i \log_2 P_i = -\sum_{\underline{x}\in A^k} p(\underline{x}) \log_2 p(\underline{x}) = \text{entropy of } X_1,\ldots,X_k$$

$$H_k = H_k(X) = \frac{1}{k} H(X_1,\ldots,X_k) = \textit{kth-order entropy of } X$$

3. How small can be the rate of any FVL code with any blocklength whatsoever? What value of k is best?

Increasing k decreases the $\frac{1}{k}$ term, which is a good thing. But we also need to know how H_k varies with k?

Fact 1: For a stationary random process,

$$H_{k+1} \le H_k \le H_1$$

H_k's converge to a limit.

$$\lim_{k\to\infty} H_k = \inf_k H_k$$

Proof: The inequalities will be proved later when we discuss conditional coding. Since the H_k's are nonincreasing and bounded from below by zero, they must approach a limit. (This uses a basic from analysis (e.g. Math 451).) Since the H_k's are nonincreasing the limit is also the inf.

Define:

$$H_\infty = \lim_{k\to\infty} H_k = \textit{entropy rate of the source } X$$

Coding Theorem 1 and Fact 1 imply that that rate R of any code with blocklength k satisfies

$$R \geq H_k \geq H_\infty$$

That is, no FVL code whatsoever can have rate less than H_∞.

Moreover, for any small number ε, we can choose k large enough that

$$H_k(X) \leq H_\infty + \frac{\varepsilon}{2} \quad \text{and} \quad \frac{1}{k} \leq \frac{\varepsilon}{2}.$$

Then Coding Theorem 1 implies that for this k, there is a code with rate

$$R = R_k^* < H_k(X) + \frac{1}{k} \leq H_\infty + \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = H_\infty + \varepsilon.$$

Since ε can be arbitrarily small, this shows there are codes with rate arbitrarily close to H_∞. In some cases, there may actually be a code with rate H_∞, but in others there is not. In any event, we have proved the following:

FVL Coding Theorem 2: For a stationary source X, the "least" average rate of FVL codes of any blocklength is

$$R_{FVL}^* \triangleq \inf_k R_k^* = H_\infty.$$

Equivalently, no FVL code can have rate less than H_∞, and for any $\varepsilon > 0$, there is a code with rate less than $H_\infty + \varepsilon$.

Notes and Examples

(1) The R_k^*'s need not decrease monotonically.

Example: Consider a binary IID source with $p(0)=.7$, $p(1) = .3$

$R_1^* = 1$, $\{0, 1\}$ is an optimal code with blocklength 1

$R_2^* = \frac{1.81}{2} = .905$, $\{0, 10, 110, 111\}$ is an optimal code with blocklength 2

$R_3^* = \frac{2.726}{3} = .9087$ $\{00, 10, 010, 011, 1100, 1101, 1110, 1111\}$ is an optimal code with blocklength 3

(2) R_k^* is subadditive, i.e. for any positive integers k and l,

$$R_{k+l}^* \leq \frac{k}{k+l} R_k^* + \frac{l}{l+k} R_l^*$$

It can be shown that a subadditive sequence has a limit that equals its inf. Therefore, subadditivity implies

$$R^* = \inf_k R_k^* = \lim_{k\to\infty} R_k^*$$

Subadditivity implies that R_k^* is nonincreasing when k increases by an integer multiple:

$$R_1^* \geq R_k^* \geq R_{mk}^* \geq R^* \quad \text{for all positive integers } k \text{ and } m$$

Proof of subadditivity:

Let C_k and C_l be codes with blocklengths k and l, respectively, that have rates R_k^* and R_l^*, respectively. Let C be the blocklength $k+l$ code obtained by using C_k on the first k source symbols and C_l on the next l source symbols. That is, $C = C_k \times C_l$. Then, the average of length of C is sum of the average lengths of C_k and C_l, i.e. $\bar{L}(C) = \bar{L}(C_k) + \bar{L}(C_l)$,

and the rate of C

$$R(C) = \frac{\bar{L}(C)}{k+l} = \frac{k}{k+l}\,\frac{\bar{L}(C_k)}{k} + \frac{l}{k+l}\,\frac{\bar{L}(C_l)}{l}$$

$$= \frac{k}{k+l} R(C_k) + \frac{l}{k+l} R(C_l) = \frac{k}{k+l} R_k^* + \frac{l}{k+l} R_l^*$$

Since $R_k^* \le R(C)$, we have from the above that $R_k^* \le \frac{k}{k+l} R_k^* + \frac{l}{k+l} R_l^*$

(3) In most cases, there is no code with rate exactly equal to H_∞,

and there is no code with blocklength k and rate exactly equal to H_k.

(4) We will show later that for an IID source (i.e. stationary and memoryless)

$$H_1 = H_2 = \ldots = H_k = H_\infty$$

This happens only when the source is memoryless. So there is less benefit to increasing k for an IID source than for a source with memory.

(5) We will show later that for a (first-order) Markov source

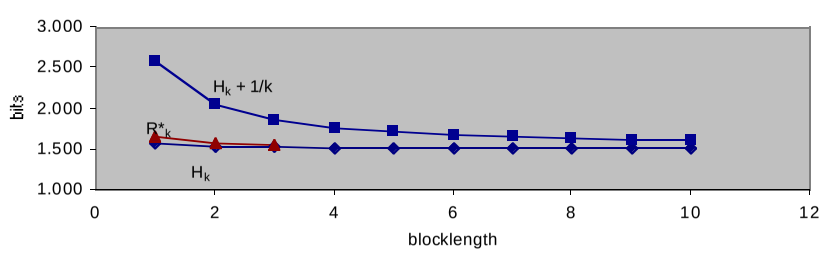
$$H_k = \frac{k-1}{k} H(X_1,X_2) - \frac{k-2}{k} H(X_1) = \frac{1}{k} H(X_1) + \frac{k-1}{k} H(X_2|X_1),$$

where $H(X_2|X_1)$ is the conditional entropy to be defined later.

<u>Example:</u> A stationary Markov source X

alphabet $A = \{1,2,3\}$, $P(X_n=i) = \frac{1}{3}$, $P(X_{n+1}=j|X_n=i) = \begin{cases} \frac{1}{2}, & j=i \\ \frac{1}{4}, & j\neq i \end{cases}$

k	1	2	3	∞
H_k	$\log_2 3$	$\frac{1}{2}(\log_2 3 + \frac{3}{2})$	$\frac{1}{3}(\log_2 3 + 2\,\frac{3}{2})$	$\frac{3}{2}$
	1.585	1.542	1.528	1.500
R_k^*	1.667	1.583	1.556	1.500
$H_k + \frac{1}{k}$	2.585	2.043	1.861	1.500
$H_k + \frac{P_{max}}{k}$	1.918	1.626	1.556	1.500



(6) "Redundancy of a code" $\stackrel{\Delta}{=}$ Rate - H_∞

By Coding Theorem 1:

redundancy of the best FVL code with blocklength k is at most 1/k.

(7) Tighter upper bound on R_k^* (Gallager[1])

It can be shown that

$$\bar{l}_k^* \le \begin{cases} H(X_1,\ldots,X_k) + P_{max} & \text{if } P_{max} < 1/2 \\ H(X_1,\ldots,X_k) + P_{max} + 0.086 & \text{if } P_{max} \ge 1/2 \end{cases}$$

where $P_{max} = \max\{P_1, P_2, \ldots\}$. Hence,

$$R_k^* \le \begin{cases} H_k(X) + P_{max}/k & \text{if } P_{max} < 1/2 \\ H_k(X) + P_{max}/k + 0.086/k & \text{if } P_{max} \ge 1/2 \end{cases}$$

Since P_{max} decreases with k (usually quite rapidly), this indicates that redundancy decreases more rapidly than 1/k.

[1] R.G. Gallager, "Variations on a them by Huffman," *IEEE Trans. Inform. Thy.*, pp. 668-674, Nov. 1978.

(8) Although there are other types of lossless codes besides FVL block codes, it turns out that for a stationary source, no code can have rate less than H_∞. Thus FVL codes perform as well as any codes.

(9) Complexity: An FVL code with blocklength k for a source with an M-symbol alphabet has M^k codewords. Thus, complexity of this type of coding grows rapidly (exponentially) with k.

Estimates of H_k for Various Languages

	$M=\|A\|$	$\log_2 M$	H_1	H_2	H_3	H_4	H_∞
English	26	4.70	4.14	3.85	3.67		1.3
English	27	4.75	4.03	3.68	3.48		
English	26	4.70	4.12				
English	27	4.75	4.09	3.66	3.39	3.21	
French	26	4.70	3.98				
Spanish	26	4.70	4.02				
German	26	4.70	4.10				
Portuguese	26?	4.70?	3.92	3.72	3.53		
Russian	36	5.17	4.55	4.00	3.65	3.42	
Samoan	17	4.09	3.40	3.04	2.83	2.69	
Tamil	30	4.91	4.34				
Arabic	32	5.00	4.21	3.99	3.49		
Chinese	4700	12.20	9.63				

Reference: *Text Compression*, Bell, Cleary, Witten, p. 95.
(Some "conversion" was needed to obtain H_2, H_3, H_4 from the table given there.)

Conditional Lossless Codes

Example: X, stationary, alphabet: $A = \{1,2,3\}$, $P(X_n=i) = \frac{1}{3}$, $P(X_{n+1}=j|X_n=i) = \begin{cases} \frac{1}{2}, & j=i \\ \frac{1}{4}, & j\neq i \end{cases}$

first-order code

$P(X=x)$	x	codeword
1/3	1	0
1/3	2	10
1/3	3	11
Rate		1.67

second-order code

$P(X_n=x_1,X_{n+1}=x_2)$	x_1x_2	codeword
1/6	11	000
1/12	12	0110
1/12	13	0111
1/12	21	100
1/6	22	001
1/12	23	101
1/12	31	110
1/12	32	111
1/6	33	010
avg length	19/6	
Rate	1.583	

conditional code

	previous symbol X_{n-1}					
X_n	1		2		3	
1	1/2	0	1/4	11	1/4	10
2	1/4	10	1/2	0	1/4	11
3	1/4	11	1/4	10	1/2	0
$\bar{l}$		1.5		1.5		1.5

Rate = 1.5 bits/symbol

(= H_∞ if source is Markov)

Is code uniquely decodable?

Yes, knowing the previous symbol, the decoder knows which prefix code was used to encode the current symbol.

It is said to be "backward adaptive".

Conditional Lossless Coding: Formal Introduction

Assume discrete-valued, stationary source X, as usual.

Basic idea:

When encoding X_n, use a code designed for the conditional probability distribution of X_n given the value of the previous symbol X_{n-1}.
That is, when $X_{n-1}=a$, use a prefix code

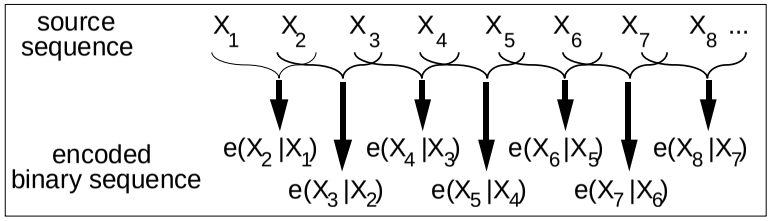
$C_a = \{\underline{v}_{a,1}, \underline{v}_{a,2}, \ldots\}$ with lengths $\{l_{a,1}, l_{a,2}, \ldots\}$

designed for

$p(1|a), p(2|a), p(3|a), \ldots$

where

$p(i|a) = \Pr(X_n=i \,|\, X_{n-1}=a)$



Conditional Rate: given $X_{n-1}=a$,

$$R_a = \bar{L}_a = \sum_i l_{a,i}\, p(i|a)$$

Overall Rate:

$$R = \sum_a p(a)\, \bar{L}_a = \sum_a \sum_i l_{a,i}\, p(a)\, p(i|a)$$

If for each a, C_a is optimal for $(p(1|a), p(2|a), p(3|a), \ldots)$, then

$$H(X|a) \le R_a < H(X|a) + 1$$

where

$$H(X|a) = H(X_2|X_1=a) = -\sum_i p(i|a) \log_2 p(i|a) = \text{cond'l entr'y of } X_2 \text{ given } X_1=a$$

and the overall rate is

$$H(X_2|X_1) \le R < H(X_2|X_1) + 1$$

where

$$H(X_2|X_1) = \sum_a p(a)\, H(X_2|X_1=a) = -\sum_a \sum_i p(a)\, p(i|a) \log_2 p(i|a)$$

$$= \text{conditional entropy of } X_2 \text{ given } X_1$$

Note how similar this notation is to that of $H(X_2|X_1=a)$. They are not the same.

Conditional Lossless Coding Theorem:

For a stationary source X,

$$H(X_2|X_1) \le R_c^* < H(X_2|X_1) + 1 ,$$

where R_c^* denotes the least rate of conditional codes applied to source X.

Notes:

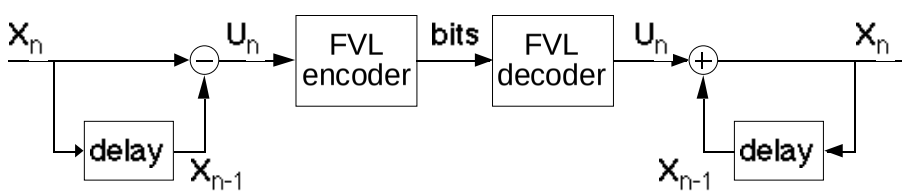
(1) This is a kind of backward adaptive coding. The encoder and decoder adapt the code based on the previous symbol.

(2) It is not absolutely essential that the C_a's be prefix codes. However, if not, one would have to require a kind of "mutual" unique decodability, namely, that any finite sequence of codewords from any of the codes be decodable in only one way.

(3) The upper bound $R_c^* < H(X_2|X_1)+1$ can be tightened by using the bound $R < H(X_2|X_1)+p_{c,max}+0.0836$, where $p_{c,max}$ is the largest value of $p(i|a)$ over all choices of i and a.

Predictive/Differential Coding

A special kind of conditional code



Example:

Alphabet of X: $A_X = \{0,1,2\}$

Alphabet of U: $A_U = \{-2,-1,0,1,2\}$

Codebook for U: $C_U = \{110, 01, 00, 10, 111\}$

Equivalent conditional encoding table

X_n	previous symbol X_{n-1} 0	1	2
0	00	01	110
1	10	00	01
2	111	10	00

(Rate is not so good in this example.)

Complexity:

Compute $U_n = X_n - X_{n-1}$

Store just the one codebook C_U with $2|A|-1$ codewords

Rate:

$R \cong H(U)$

Often, $H(U) \cong H(X_2|X_1)$

Improvement:

Replace "delay" with a predictor

$\tilde{X}_n = g(X_{n-1}, X_{n-2}, \ldots)$

Widely used in lossless image coding, e.g. "lossless JPEG".

Improved Versions of Conditional Lossless Coding

There are two ways to improve conditional coding.

(1) mth-order conditional encoding: condition on m past symbols (rather than on one)

(2) (m,k) conditional block coding: conditionally encode k symbols at a time (rather than one).

(1) mth-order conditional coding

In mth-order conditional coding, when encoding symbol X_n one uses a prefix codebook determined by the previous m symbols $X_{n-m},\ldots,X_{n-1}$.

That is, for every m-tuple $\underline{x} = (x_1,\ldots,x_m)$ in A^m, there is a prefix codebook

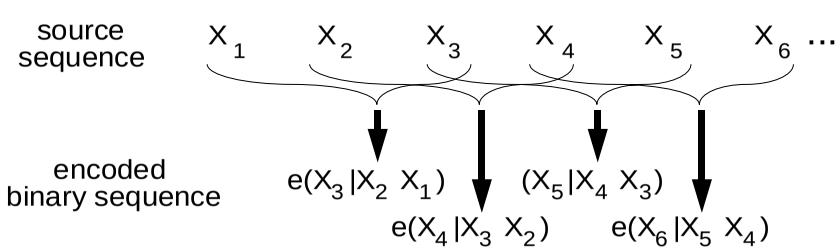
$$C_{\underline{x}} = \{\underline{v}_{\underline{x},1}, \underline{v}_{\underline{x},2}, \ldots\}$$

with lengths

$$\{l_{\underline{x},1}, l_{\underline{x},2}, \ldots\}$$

and a distinct codeword for every symbol in A. When $(X_{n-m},\ldots,X_{n-1}) = \underline{x}$, then X_n is encoded with codebook $C_{\underline{x}}$.

The operation of a 2nd-order conditional code is illustrated below:



Definition: $R^*_{1|m}$ = least rate of mth-order conditional coding for source X

Analysis

Given that $(X_{n-m},\ldots,X_{n-1}) = \underline{x}$, the conditional average rate of the code is

$$R_{\underline{x}} = \bar{L}_{\underline{x}} = \sum_i l_{\underline{x},i}\, p(i|\underline{x})$$

where $p(i|\underline{x}) = \Pr(X_n=i|X_{n-m},\ldots,X_{n-1}=\underline{x})$, and the overall rate is

$$R = \sum_{\underline{x}} p(\underline{x})\, R_{\underline{x}} = \sum_{\underline{x}} p(\underline{x})\, \bar{L}_{\underline{x}}$$

where $p(\underline{x}) = \Pr(X_{n-m}^{n-1}=\underline{x}) = \Pr(X_1^m=\underline{x})$, (the latter by stationarity).

If for each $\underline{x}$, $C_{\underline{x}}$ is optimal for $(p(1|\underline{x}), p(2|\underline{x}), p(3|\underline{x}), \ldots)$, then

$$H(X_n|\underline{x}) \le R_{\underline{x}} < H(X_n|\underline{x}) + 1$$

where $H(X_n|\underline{x}) = H(X_n|X_{n-m},\ldots,X_{n-1}=\underline{x}) = -\sum_i p(i|\underline{x}) \log_2 p(i|\underline{x})$

$= $ conditional entropy of X_2 given $X_{n-m},\ldots,X_{n-1}=\underline{x}$.

The overall rate is $R^*_{1|m}$ and from the above

$$H_{1|m} \le R^*_{1|m} < H_{1|m} + 1$$

where $H_{1|m} = H(X_n|X_{n-m},\ldots,X_{n-1}) = \sum_a p(\underline{x})\, H(X_n|X_{n-m},\ldots,X_{n-1}=\underline{x})$

$= $ conditional entropy of $H(X_n|X_{n-m},\ldots,X_{n-1})$

Again, note how similar this name is to that of $H(X_n|X_{n-m},\ldots,X_{n-1}=\underline{x})$.

Estimates of $H(X_n|X_{n-m},\ldots,X_{n-1})$ for various languages

$$H_{1|m} \triangleq H(X_n|X_{n-m},\ldots,X_{n-1})$$

	$M=\|A\|$	$\log_2 M$	$H(X_1)$	$H_{1\|1}$	$H_{1\|2}$	$H_{1\|3}$	$H_{1\|7}$	$H_{1\|11}$	$H_{1\|\infty}$
English	26	4.70	4.14	3.56	3.3				1.3
English	27	4.75	4.03	3.32	3.1				
English	26	4.70	4.12						
English	27	4.75	4.09	3.23	2.85	2.66	2.43	2.40	
French	26	4.70	3.98						
Spanish	26	4.70	4.02						
German	26	4.70	4.10						
Portuguese	26?	4.70?	3.92	3.51	3.15				
Russian	36	5.17	4.55	3.44	2.95	2.72	2.45	2.40	
Samoan	17	4.09	3.40	2.68	2.40	2.28	2.16	2.14	
Tamil	30	4.91	4.34						
Arabic	32	5.00	4.21	3.77	2.49				
Chinese	4700	12.20	9.63						

Reference: *Text Compression*, Bell, Cleary, Witten, p. 95.

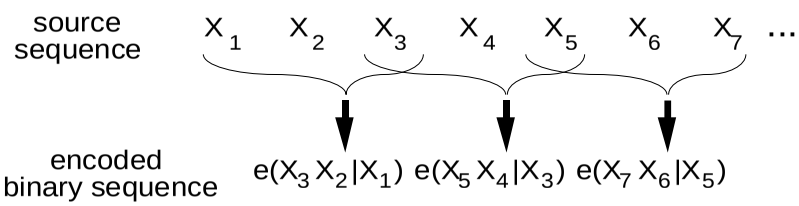
(2) Conditional <u>Block</u> Coding.

In (k|m)-conditional-block encoding, just as with FVL encoding with blocklength k, one divides the source sequence into blocks of length k.

Then as with mth-order conditional coding, when encoding a block beginning with symbol X_n, i.e. the block $(X_n,\ldots,X_{n+k-1})$ one uses a prefix codebook determined by the previous m symbols $X_{n-m},\ldots,X_{n-1}$.

That is, for every m-tuple $\underline{x} = (x_1,\ldots,x_m)$ in A^m, there is a prefix codebook $C_{\underline{x}} = \{\underline{v}_{\underline{x},1}, \underline{v}_{\underline{x},2}, \ldots\}$ with lengths $\{l_{\underline{x},1}, l_{\underline{x},2}, \ldots\}$ and one distinct codeword for every sequence in A^k. When $(X_{n-m},\ldots,X_{n-1}) = \underline{x}$, then $(X_n,\ldots,X_{n+k-1})$ is encoded with codebook $C_{\underline{x}}$.

The operation of a (1,2)-cond'l block code is illustrated below:



Definition:

$R^*_{k|m}$ = least rate of any (k|m)-conditional code for X

Analysis

To allow more compact expressions, let X_u^v denote the sequence $(X_u,\ldots,X_v)$.

Given that $X_{n-m}^{n-1} = \underline{x}$, the average rate of the code is

$$R_{\underline{x}} = \frac{1}{k}\bar{L}_{\underline{x}} = \frac{1}{k}\sum_i l_{\underline{x},i}\, p(\underline{a}_i|\underline{x})$$

where $p(\underline{a}_i|\underline{x}) = \Pr(X_n^{n+k-1} = \underline{a}_i | X_{n-m}^{n-1} = \underline{x})$, and the overall rate is

$$R = \frac{1}{k}\sum_{\underline{x}} p(\underline{x})\, R_{\underline{x}} = \frac{1}{k}\sum_{\underline{x}} p(\underline{x})\, \bar{L}_{\underline{x}}$$

where $p(\underline{x}) = \Pr(X_{n-m}^{n-1}=\underline{x}) = \Pr(X_1^m=\underline{x})$, (the latter by stationarity).
If $C_{\underline{x}}$ is optimal for $(p(\underline{a}_1|\underline{x}), p(\underline{a}_2|\underline{x}), p(\underline{a}_3|\underline{x}), \ldots)$, then

$$\frac{1}{k} H(X_n^{n+k-1}|\underline{x}) \le R_{\underline{x}} < \frac{1}{k} H(X_n^{n+k-1}|\underline{x}) + \frac{1}{k}$$

where

$$H(X_n^{n+k-1}|\underline{x}) = H(X_n^{n+k-1}|X_{n-m}^{n-1}=\underline{x}) = -\sum_i p(\underline{a}_i|\underline{x}) \log_2 p(\underline{a}_i|\underline{x})$$

= conditional entropy of X^{n+k-1} given $X_{n-m}^{n-1}=\underline{x}$.

and from the above, the overall rate is

$$H_{k|m} \le R^*_{k|m} < H_{k|m} + \frac{1}{k}$$

where

$$H_{k|m} = \frac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1}) = (k|m)\text{-th order entropy}$$

and

$$H(X_n^{n+k-1}|X_{n-m}^{n-1}) = \sum_{\underline{x}} p(\underline{x})\, H(X_n^{n+k-1}|\underline{x}) = -\sum_{\underline{x}} \sum_{i} p(\underline{x})\, p(\underline{a}_i|\underline{x}) \log_2 p(\underline{a}_i|\underline{x})$$

$$= \text{conditional entropy of } X^{n+k-1} \text{ given } X_{n-m}^{n-1}$$

Again, note how similar this name is to that of $H(X_n^{n+k-1}|X_{n-m}^{n-1}=\underline{x})$.

We now summarize and take the limit.

Coding Theorem for Conditional Block Codes:

For a stationary source X and integers $k \geq 1$, $m \geq 0$

(a) $H_{k|m} \leq R^*_{k|m} < H_{k|m} + \frac{1}{k}$,

where

$R^*_{k|m}$ denotes the least rate of any $(k|m)$ conditional block code

$H_{k|m} = \frac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1})$ is the $(k|m)$-th order entropy.

Moreover,

(b) $\lim_{k\to\infty} R^*_{k|m} = H_\infty$, for any m

(c) $H_\infty \leq \lim_{m\to\infty} R^*_{k|m} < H_\infty + \frac{1}{k}$, for any k.

When $m=0$, we interpret a $(k|m)$ conditional block code to mean an unconditional block code. This theorem generalizes those on pp. 7,9.

Proof: For $m=0$, (a) and (b) were established in the theorems on pp. 7,9.

For $m \geq 1$, (a) was establiished on previous pages. (b) and (c) follow from (a) and Properties 15 and 16, given later, which show

$$\lim_{k\to\infty} H_{k|m} = H_\infty \text{ for any } m, \quad \text{and} \quad \lim_{m\to\infty} H_{k|m} = H_\infty \text{ for any } k$$

This theorem demonstrates that one can approach the least possible rate, H_∞, in a number of ways.

Complexity of conditional and conditional block codes

Suppose the source alphabet contains M symbols.

A direct implementation of an (k|m) conditional block code requires storing the M^k codewords of each of the M^m codes $C_{\underline{x}}$. That is, the total storage required for encoding or decoding is proportional to M^{k+m}.

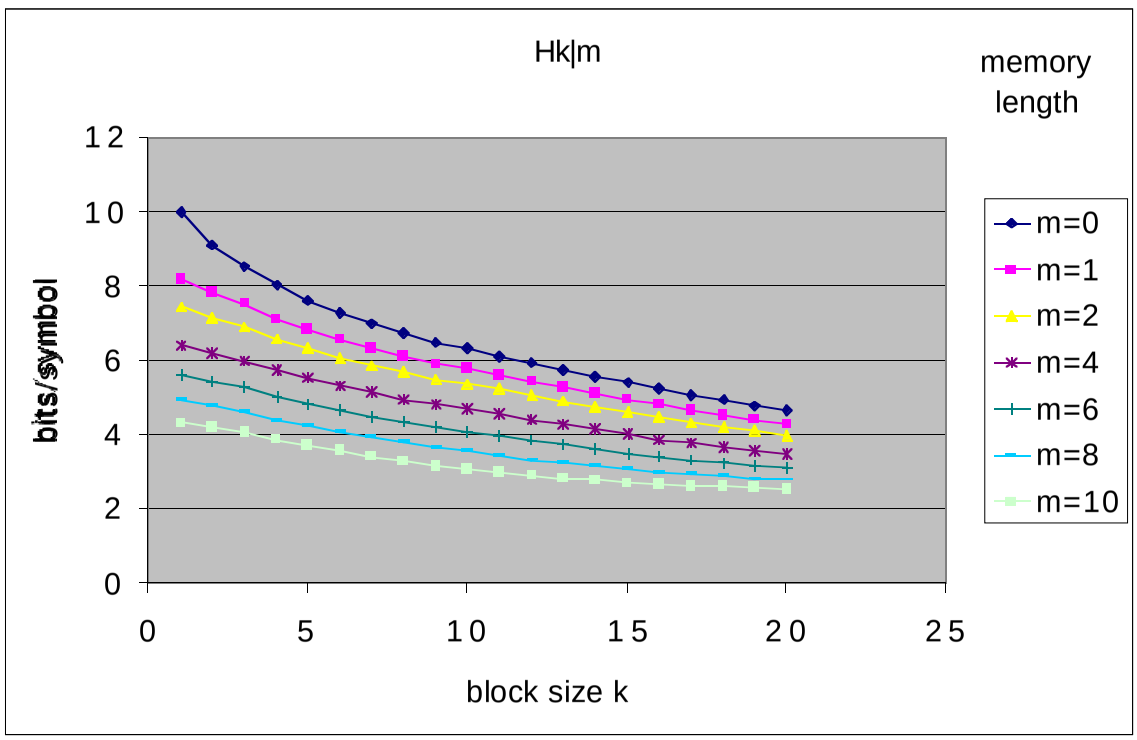
Thus complexity grows exponentially with m and k.

Suppose we fix a value K and require that $k+m = K$, thereby fixing the complexity, at least approximately. What values of k and m lead to the lowest rate?

The answer depends on the source, so there is no universally best choice. One can use $H_{k|m} + 1/k$ as a guide to the choice of k and m.

The facts given later show that among choices of k,m such that $k+m=K$, $H_{k|m}$ is smallest when $k=1$ and $m=K-1$, i.e. $H_{1|K-1}$ is smallest.

Example: A hypothetical 20th-order Markov source with H_1= 10, H_∞ = 2



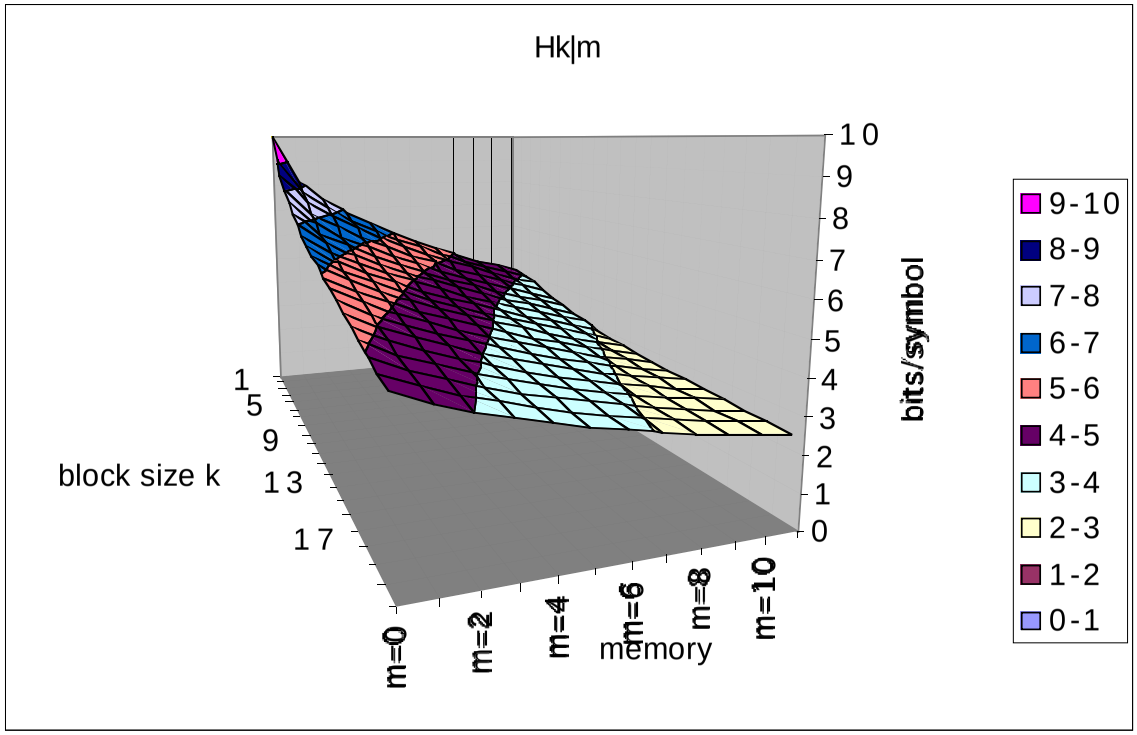


Table of $H_{k|m}$

k = \ m =	0	1	2	3	4	5	6	7	8	9	10	11	12
1	10	8.21	7.47	6.9	6.42	**6**	5.62	5.27	4.94	**4.6**	4.34	4.07	3.8
2	9.11	7.84	7.19	6.66	**6.2**	5.81	5.44	5.1	**4.8**	4.49	4.21	3.94	3.68
3	8.56	7.53	6.93	**6.4**	6.01	5.63	5.28	**4.9**	4.64	4.35	4.07	3.81	**3.6**
4	8.03	7.15	**6.6**	6.15	5.75	5.38	**5**	4.73	4.43	4.15	3.88	**3.6**	3.38
5	7.62	**6.8**	6.34	5.91	5.52	**5.2**	4.85	4.54	4.25	3.98	**3.7**	3.47	3.23
6	**7.3**	6.58	6.1	5.69	**5.3**	4.99	4.67	4.38	4.1	**3.8**	3.57	3.33	3.09
7	7	6.34	5.89	**5.5**	5.15	4.82	4.51	4.22	**3.9**	3.69	3.44	3.2	2.96
8	6.74	6.13	**5.7**	5.32	4.98	4.66	4.36	**4.1**	3.81	3.56	3.31	3.07	2.84
9	6.51	**5.9**	5.52	5.15	4.82	4.51	**4.2**	3.94	3.68	3.43	3.19	2.95	2.75
10	**6.3**	5.77	5.37	5.02	4.69	**4.4**	4.1	3.83	3.57	3.33	3.09	2.86	2.67
11	6.14	5.62	5.23	4.88	**4.6**	4.27	3.99	3.72	3.47	3.22	2.99	2.78	2.61
12	5.95	5.45	5.07	**4.7**	4.42	4.13	3.86	3.6	3.35	3.12	2.91	2.71	2.56
13	5.76	5.28	**4.9**	4.59	4.28	4	3.73	3.47	3.24	3.04	2.84	2.66	2.52
14	5.59	**5.1**	4.77	4.45	4.15	3.87	3.61	3.37	3.15	2.96	2.78	2.61	2.48
15	**5.4**	4.97	4.63	4.31	4.02	3.75	3.5	3.28	3.08	2.9	2.73	2.57	2.45
16	5.26	4.82	4.49	4.18	3.89	3.64	3.41	3.2	3.01	2.84	2.68	2.54	2.42
17	5.1	4.68	4.35	4.05	3.78	3.54	3.32	3.13	2.95	2.79	2.64	2.5	2.4
18	4.95	4.54	4.22	3.94	3.68	3.46	3.25	3.06	2.9	2.75	2.6	2.48	2.37
19	4.81	4.41	4.1	3.84	3.6	3.38	3.18	3.01	2.85	2.71	2.57	2.45	2.35
20	4.67	4.29	4	3.74	3.52	3.31	3.12	2.96	2.81	2.67	2.54	2.43	2.34

The values on a diagonal, such as those printed in bold, have the same complexity.

More Definitions and Properties of Entropy

For Two Random Variables

Definitions:

(a) The entropy of a pair of random variables X and Y (sometimes called their "joint" entropy) is

$$H(X,Y) = -\sum_{x,y} p(x,y) \log_2 p(x,y), \qquad \text{where } p(x,y) \triangleq \Pr(X=x \text{ and } Y=y).$$

Think of (X,Y) as <u>one</u> random variable with an alphabet consisting of pairs. Then the above is just the usual definition of entropy, and has the usual properties.

(b) The conditional entropy of random variable X given a particular value of random variable Y

$$H(X|Y=y) = -\sum_{x} p(x|y) \log_2 p(x|y), \qquad \text{where } p(x|y) = \Pr(X=x|Y=y).$$

This is an ordinary entropy -- namely the entropy of X when a specific value of Y is given -- and as such it has all the usual properties of entropy.

(c) Conditional entropy of random variable X given a random variable Y

$$H(X|Y) = \sum_{y} p(y)\, H(X|Y=y) = -\sum_{x,y} p(x,y) \log_2 p(x|y)$$

This is an average of the previous kind of conditional entropy.

Properties: (elementary in information theory)

1. $H(X|Y) \geq 0$ with equality iff X is a function of Y

   Proof: $H(X|Y=y) \geq 0$ for all y because it is an ordinary entropy. Since $H(X|Y)$ is the average of such, it too is nonnegative. $H(X|Y) = 0$ if and only if $H(X|Y=y)=0$ for each y with positive probability. This happens if and only for each y with positive probability, there is a value x such that $\Pr(X=x|Y=y)=1$. In other words, Y determines X; i.e. X is a function of Y.

2. $H(X|Y) \leq H(X)$ with equality iff X and Y are independent.

   Proof: We'll show $H(X) - H(X|Y) \geq 0$ with equality iff X indep of Y.

$$\begin{aligned}
H(X) - H(X|Y) &= -\sum_{x,y} p(x,y) \log_2 p(x) + \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(y)} \\
&= -\sum_{x,y} p(x,y) \ln \frac{p(x)\,p(y)}{p(x,y)} \frac{1}{\ln 2} \\
&\geq -\sum_{x,y} p(x,y) \left(\frac{p(x)\,p(y)}{p(x,y)} - 1 \right) \frac{1}{\ln 2} \qquad \text{using } \ln z \leq z-1 \\
&= -\sum_{x,y} p(x)\,p(y) \frac{1}{\ln 2} + \sum_{x,y} p(x,y) \frac{1}{\ln 2} = 0.
\end{aligned}$$

   Equality holds if and only if $p(x)\,p(y) = p(x,y)$ for all x,y; i.e. if and only if X and Y are independent.

3. Chain rule: $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

Proof:

$$H(X,Y) = -\sum_{x,y} p(x,y) \log_2 p(x)p(y|x)$$

$$= -\sum_{x,y} p(x,y) \log_2 p(x) - \sum_{x,y} p(x,y) \log_2 p(y|x)$$

$$= H(X) + H(Y|X).$$

Interchanging X and Y in this argument shows $H(X,Y) = H(Y)+H(X|Y)$.

4. $H(X,Y) \leq H(X) + H(Y)$

with equality iff X and Y are independent

Proof: Using Facts 3 and 2,

$$H(X,Y) = H(Y) + H(X|Y) \leq H(Y) + H(X)$$

with equality iff X and Y are independent.

5. $H(X) \leq H(X,Y)$ with equality iff Y is a function of X

Proof: By Fact 3, $H(X) = H(X,Y) - H(Y|X) \leq H(X,Y)$, where the inequality is from Fact 1, and equality holds iff $H(X|Y) = 0$, which by Fact 1 happens iff X is a function of Y.

More Than Two Random Variables

Definitions:

(d) Entropy of $X_1,\ldots,X_k$ (sometimes called their "joint entropy")

$$H(X_1,\ldots,X_k) = -\sum_{\underline{x}} p(\underline{x}) \log_2 p(\underline{x}) \quad \text{where } \underline{x} = (x_1,\ldots,x_k)$$

(e) Conditional entropy of random variables $X_1,\ldots,X_k$ given particular values of random variables $Y_1,\ldots,Y_m$

$$H(X_1,\ldots,X_k|Y_1,\ldots,Y_m=y_1,\ldots,y_m) = -\sum_{\underline{x}} p(\underline{x}|\underline{y}) \log_2 p(\underline{x}|\underline{y}), \quad \text{where } \underline{y} = (y_1,\ldots,y_m)$$

(f) Conditional entropy of random variables $X_1,\ldots,X_k$ given random variables $Y_1,\ldots,Y_m$

$$H(X_1,\ldots,X_k|Y_1,\ldots,Y_m) = \sum_{\underline{y}} p(\underline{y}) H(X_1,\ldots,X_k|Y_1,\ldots,Y_m=y_1,\ldots,y_m)$$

$$= -\sum_{\underline{x},\underline{y}} p(\underline{x},\underline{y}) \log_2 p(\underline{x}|\underline{y})$$

Note: If one thinks of $(X_1,\ldots,X_k)$ and $(Y_1,\ldots,Y_m)$ each as a single random variable with a vector-valued alphabet, then the above formulas are the same as the corresponding "two-random variable" formulas.

Properties:

6. $H(Y_1,...,Y_n|X_1,\ldots,X_m) \le H(Y_1,\ldots,Y_n|X_1,\ldots,X_{m'})$, $0 \le m' < m$,

   with equality iff $Y_1,\ldots,Y_n$ is conditionally independent of $X_{m'+1},\ldots,X_m$ given $X_1,\ldots,X_{m'}$.

   Proof: Like that of Fact 2.

7. Chain rule:

$$H(X_1,...,X_k) = H(X_1) + H(X_2|X_1) + H(X_3|X_1,X_2) + H(X_4|X_1,X_2,X_3) + \ldots + H(X_k|X_1,X_2,\ldots,X_{k-1})$$

   Proof: Like that of Fact 3.

8. $H(X_1,...,X_k) \le H(X_1) + H(X_2) + ... + H(X_k)$

   with equality iff the X_i's are independent

   Proof: Follows from the chain rule and the fact that $H(X_i|X_1,\ldots,X_{i-1}) \le H(X_i)$ with equality if and only if X_i is independent of $X_1,\ldots,X_{i-1}$.

For Stationary Random Processes

Let $\{X_k\}$ be stationary random process.

Definitions:

(g) $H_k \triangleq \frac{1}{k} H(X_1,...,X_k)$

(h) $H_{1|m} \triangleq H(X_n|X_{n-m},X_2,\ldots,X_{n-1})$ (same for all n) ($H_{1|0} \triangleq H(X_1) = H_1$)

(i) $H_{k|m} \triangleq \frac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1})$ $(H_{1|m} = H_{1|m})$

(j) $H_\infty \triangleq \lim_{k\to\infty} H_k$ = entropy rate of X

Facts to be established:

$H_{k|m}$ decreases or stays the same as k or m increases

$H_{k|m} \to H_\infty$ as k and/or m go to infinity

Properties:

9. $H_{1|m+1} \le H_{1|m}$

   Proof: Follows from Property 6 and stationarity.

10. $H_k = \frac{1}{k}\left(H_1 + H_{1|1} + H_{1|2} + \ldots + H_{1|k-1}\right) \ge H_{1|k-1} \ge H_{1|k}$

   Proof:
$$\begin{aligned} H_k &= \tfrac{1}{k} H(X_1,\ldots,X_k) \\ &= \tfrac{1}{k}\left(H(X_1) + H(X_2|X_1) + H(X_3|X_1,X_2) + \ldots + H(X_k|X_1,X_2,\ldots,X_{k-1})\right) \\ &\qquad \text{by chain rule} \\ &= \tfrac{1}{k}\left(H_1 + H_{1|1} + H_{1|2} + \ldots + H_{1|k-1}\right) \quad \text{by stationarity} \\ &\ge \tfrac{1}{k}\left(H_{1|k-1} + H_{1|k-1} + H_{1|k-1} + \ldots + H_{1|k-1}\right) \quad \text{by Prop. 9.} \\ &= H_{1|k-1} \ge H_{1|k} \quad \text{by Prop. 9} \end{aligned}$$

11. $H_{k+1} \le H_k$

   Since H_k's are nonincreasing and bounded below by zero, they must have a limit. Hence, H_∞ is a well-defined quantity.

   Proof: By Prop. 10, H_k is the average of k terms $H_1, H_{1|1}, H_{1|2}, \ldots, H_{1|k-1}$.

   Similarly, H_{k+1} is average of k+1 terms $H(X_1), H_{1|1}, H_{1|2}, \ldots, H_{1|k-1}, H_{1|k}$.

   Since the extra term in H_{k+1} is no larger than all other terms, $H_{k+1} \le H_k$.

12. $\lim_{k\to\infty} H_{1|k} = \lim_{k\to\infty} H_k \triangleq H_\infty$

   Proof: Since the $H_{1|k}$'s are nonincreasing with k and bounded below by zero, they must have a limit. Since by Prop. 10, H_k is the average of the k terms $H_1, H_{1|1}, H_{1|2}, \ldots, H_{1|k-1}$, the limit of the $H_{1|k}$'s equals the limit of the H_k's, which by definition is H_∞.

13. $H_{k|m} = \frac{1}{k}\left(H_{1|m} + H_{1|m+1} + \ldots + H_{1|m+k-1}\right)$

   Proof:
$$\begin{aligned} H_{k|m} &= \tfrac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1}) \\ &= \tfrac{1}{k}\left(H(X_n|X_{n-m}^{n-1}) + H(X_{n+1}|X_{n-m}^{n}) + \ldots + H(X_{n+k-1}|X_{n-m}^{n+k-2})\right) \\ &= \tfrac{1}{k}\left(H_{1|m} + H_{1|m+2} + \ldots + H_{1|m+k-1}\right) \quad \text{by stationarity} \end{aligned}$$

14. $H_{1|k+m} \le H_{k|m} \le H_k$

   Proof:

   (a) $H_{k|m} = \frac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1}) \le \frac{1}{k} H(X_n^{n+k-1}) = H_k$

   (b) By Prop. 13, $H_{k|m}$ is the average of terms, each of which is at least $H_{1|k+m}$. Therefore, $H_{k|m} \ge H_{1|k+m}$.

15 $H_{k+1|m} \le H_{k|m}$

$H_{k|m+1} \le H_{k|m}$

Therefore,

$$\begin{array}{cccccc} H_1 & \ge & H_2 & \ge & H_3 & \ge \text{ ... decreases with } k \\ H_{1|1} & \ge & H_{2|1} & \ge & H_{3|1} & \ge \\ H_{1|2} & \ge & H_{3|2} & \ge & H_{3|2} & \ge \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \text{decreases with } m & & & & & \end{array}$$

16. $\lim_{k\to\infty} H_{k|m} = H_\infty$ for any m

Proof: Recall from Prop. 13 that $H_{k|m} = \frac{1}{k}\left(H_{1|m} + H_{1|m+1} + \ldots + H_{1|k+m}\right)$

It follows that the limit as $k\to\infty$ of $H_{k|m}$ equals $\lim_{k\to\infty} H_{1|k+m} = H_\infty$

17. $\lim_{m\to\infty} H_{k|m} = H_\infty$ for any k

Proof: Recall from Prop. 13 that $H_{k|m} = \frac{1}{k}\left(H_{1|m} + H_{1|m+2} + \ldots + H_{1|k+m}\right)$

It follows that the limit as $m\to\infty$ of $H_{k|m}$ equals $\lim_{m\to\infty} H_{1|k+m} = H_\infty$

Examples:

A. IID source:

$H(X_1) = H_1 = H_k = H_\infty = H_{1|1} = H_{1|k} = H_{k|m}$, for all k,m

B. (first-order) Markov source:

Definition: A stationary source is first-order Markov if

$p(x_m|x_1,\ldots,x_{m-1}) = p(x_m|x_{m-1})$ for all m and $x_1,\ldots,x_m$

Then for any $m \ge 1$,

$$H_{1|m} = H(X_m|X_1,\ldots,X_{m-1}) = -\sum_{x_1\ldots x_m} p(x_1,\ldots,x_m)\log_2 p(x_m|x_1,\ldots,x_{m-1})$$

$$= -\sum_{x_1\ldots x_m} p(x_1,\ldots,x_m)\log_2 p(x_m|x_{m-1}) = -\sum_{x_{m-1},x_m} p(x_{m-1},x_m)\log_2 p(x_m|x_{m-1})$$

$$= H(X_m|X_{m-1}) = H(X_2|X_1) = H_{1|1} \quad \text{by stationarity}$$

Since all the $H_{1|m}$'s equal $H_{1|1}$,

$$H_\infty = \lim_{m\to\infty} H_{1|m} = H_{1|1}$$

By Prop. 10 and the above

$$H_k = \frac{1}{k}(H_1+H_{1|1}+H_{1|2}+\ldots+H_{1|k-1}) = \frac{1}{k}(H_1+(k-1)H_{1|1}) = \frac{1}{k}H_1+\frac{k-1}{k}H_{1|1}$$

from which we see that H_k converges down to H_∞, as it should.

Similarly, one can show

$$H_{k|m} = H_{1|1} = H_\infty \quad \text{for all } k,m$$

Because $H_{1|1} = H_\infty$, we see that first-order conditional coding can come within one bit of optimal for first-order Markov sources. In other words, there is no reason to use 2nd or higher order conditional coding, and the only reason to use block coding is to reduce the possibly one bit of redundancy.

C. nth-order Markov source:

Definition: A stationary source is nth-order Markov if

$$p(x_k|x_1,\ldots,x_{k-1}) = p(x_k|x_{k-n},\ldots,x_{k-1}) \quad \text{for all } k>n \;\&\; x_1,\ldots,x_k$$

Then as in the case of 1st-order Markov sources, it can be shown that if $m \geq n$,

$$H_{1|m} = H_{k|m} = H_{1|n} = H_\infty, \quad \text{for every } k,$$

and that if $k > n$,

$$H_k = \frac{1}{k}(nH_n + H_{1|n} + H_{1|n+1} + \ldots + H_{1|k-1}) = \frac{1}{k}(nH_n + H_{1|n} + H_{1|n} + \ldots + H_{1|n})$$

$$= \frac{n}{k} H_n + \frac{k-n}{k} H_{1|n}$$

which converges down to $H_\infty = H_{1|n}$ as $k \to \infty$

Since $H_{1|n} = H_\infty$, nth-order conditional coding comes within 1 bit of opt'l for nth-order Markov sources. Thus there is no reason to use higher order conditional coding, and the only reason to use block coding is to reduce the at most 1 bit redundancy.