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.

January 18, 2007

Example of FVLC

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

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

<table>
<thead>
<tr>
<th>first-order code</th>
<th>second-order code</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(X=x) \times x )</td>
<td>code word</td>
</tr>
<tr>
<td>1/3 1</td>
<td>0</td>
</tr>
<tr>
<td>1/3 2</td>
<td>10</td>
</tr>
<tr>
<td>1/3 3</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>rate</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 \( A = \{a_1, \ldots\} = \{1, 2, \ldots\} \), 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:

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

If \( A \) has \( M \) symbols, then \( A^k \) has \( M^k \) sequences.

Alternate notation: \( A^k = \{a_1, a_2, \ldots\} \), where each \( a_i \) is some sequence of length \( k \) from \( A \).

Probabilities: \( (P_1, P_2, \ldots) \), where \( P_i = \Pr((X_1,\ldots,X_k) = a_i) \) or \( p(x) = p_X(x) = \Pr(X=x) \)

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:

\[
k = \text{blocklength},
\]

\[
C = \{v_1, v_2, \ldots\} = \text{codebook, uniquely decodeable (usually prefix),}
\]

with one codeword for each sequence \( x \) in \( A^k \),

Codeword lengths \( l_1, l_2, \ldots \), where \( l_i = l(v_i) = l(i) = l(a_i) \)

Each codeword has the form \( v_i = (v_{i,1},\ldots,v_{i,k}) \in \{0,1\}^* \),

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

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

Specifically, \( z = e(x) = v_i \) when \( x = a_i \).

decoding procedure depends on the codebook.
The operation of an FVL code with blocklength $k = 3$ is illustrated below:

The performance of the code is determined by its average length, which is given by:

\[ -L = \sum_{x \in A^k} l(x) p(x) = \sum_{i} l_i p_i \]

and rate, which is defined as:

\[ R = \frac{-L}{k} \text{ bits/source symbol} \]

**Key Questions**

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

   Use the 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 \( \{p(x) : x \in A^k\} \).

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

3. How small can be the rate of any FVL code with any blocklength?

**Answers**

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

   Use the 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 \( \{p(x) : x \in A^k\} \).
2. How small can be the rate of an FVL code with blocklength $k$?

Define:

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

$$R^*_k = \frac{L^*_k}{k} = \text{smallest rate among all FVL codes with blocklength } k$$

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

**FVL Coding Theorem 1:**

$$H(X_1,\ldots,X_k) \leq \frac{L^*_k}{k} < H(X_1,\ldots,X_k) + 1$$

$$H_k(X) \leq R^*_k < H_k(X) + \frac{1}{k}$$

where

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

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

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,

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

$H_k$'s converge to a limit.

$$\lim_{k \to \infty} H_k = \inf \limits_k H_k$$

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:**

$$H_\infty = \lim_{k \to \infty} H_k = \text{entropy rate of the source } X$$
Coding Theorem 1 and Fact 1 imply that the 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 $H_\infty$.
Moreover, for any small number $\varepsilon$, we can choose $k$ large enough that $H_k(X) \leq H_\infty + \frac{\varepsilon}{2}$ and $\frac{1}{k} \leq \frac{\varepsilon}{2}$.
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 $\varepsilon$ can be arbitrarily small, this shows there are codes with rate arbitrarily close to $H_\infty$. In some cases, there may actually be a code with rate $H_\infty$, 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
\[ R_{FVL}^* \Delta \inf_k R_k^* = H_\infty. \]
Equivalently, no FVL code can have rate less than $H_\infty$, 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 $p(0)=.7$, $p(1) = .3$
   
   \[
   R_1^* = 1, \quad \{0, 1\} \text{ is an optimal code with blocklength 1}
   \]
   
   \[
   R_2^* = \frac{1.81}{2} = .905, \quad \{0, 10, 110, 111\} \text{ is an optimal code with blocklength 2}
   \]
   
   \[
   R_3^* = \frac{2.726}{3} = .9087, \quad \{00, 10, 010, 011, 1100, 1101, 1110, 1111\} \text{ is an optimal code with blocklength 3}
   \]

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

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

   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:
   
   \[ R_1^* \geq R_k^* \geq R_{mk}^* \geq R^* \text{ for all positive integers } k \text{ and } m. \]
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

\[
\sum \frac{L(C_k)}{k} + \sum \frac{L(C_l)}{l}.
\]

Since \( R^*_k \leq R(C) \), we have from the above that \( R_k \leq \frac{R^*_k}{k} + \frac{R^*_l}{l} \).

In most cases, there is no code with rate exactly equal to \( H_\infty \), and there is no code with blocklength \( k \) and rate exactly equal to \( H_k \).

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

\[
H_k = H_{k-1} + \frac{1}{k} H(X_{n+1}, X_n) - \frac{1}{k+1} H(X_{n+1} | X_n).
\]

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.

Example: A stationary Markov source \( X \) alphabet \( A = \{1, 2, 3\} \), \( P(X_0 = i) = \frac{1}{3} \), \( P(X_{n+1} = j | X_n = i) = \begin{cases} \frac{1}{2}, & j = i \\ \frac{1}{4}, & j \neq i \end{cases} \)

<table>
<thead>
<tr>
<th>( k )</th>
<th>( H_k )</th>
<th>( R_k )</th>
<th>( H_k + \frac{1}{k} R_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.585</td>
<td>1.583</td>
<td>1.556</td>
</tr>
<tr>
<td>2</td>
<td>2.043</td>
<td>1.556</td>
<td>1.500</td>
</tr>
<tr>
<td>3</td>
<td>2.567</td>
<td>1.567</td>
<td>1.500</td>
</tr>
<tr>
<td>4</td>
<td>3.153</td>
<td>1.556</td>
<td>1.500</td>
</tr>
<tr>
<td>5</td>
<td>3.786</td>
<td>1.556</td>
<td>1.500</td>
</tr>
</tbody>
</table>
(6) "Redundancy of a code" \( \Delta = \text{Rate} - H_\infty \)

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')

It can be shown that

\[
\hat{R}_k^* \leq \begin{cases} 
H(X_1, \ldots, X_k) + P_{\max} & \text{if } P_{\max} < 1/2 \\
H(X_1, \ldots, X_k) + P_{\max} + 0.086 & \text{if } P_{\max} \geq 1/2 
\end{cases}
\]

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

\[
\hat{R}_k \leq \begin{cases} 
H_k(X) + P_{\max}/k & \text{if } P_{\max} < 1/2 \\
H_k(X) + P_{\max}/k + 0.086/k & \text{if } P_{\max} \geq 1/2 
\end{cases}
\]

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

---

Estimates of $H_k$ for Various Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>$M$</th>
<th>$\log_2 M$</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>$H_4$</th>
<th>$H_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>26</td>
<td>4.70</td>
<td>4.14</td>
<td>3.85</td>
<td>3.67</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>27</td>
<td>4.75</td>
<td>4.03</td>
<td>3.68</td>
<td>3.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>26</td>
<td>4.70</td>
<td>4.27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>27</td>
<td>4.75</td>
<td>4.09</td>
<td>3.66</td>
<td>3.39</td>
<td>3.21</td>
<td></td>
</tr>
<tr>
<td>French</td>
<td>26</td>
<td>4.70</td>
<td>3.98</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>26</td>
<td>4.70</td>
<td>4.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German</td>
<td>26</td>
<td>4.70</td>
<td>4.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portuguese</td>
<td>26?</td>
<td>4.70?</td>
<td>3.92</td>
<td>3.72</td>
<td>3.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Russian</td>
<td>36</td>
<td>5.17</td>
<td>4.55</td>
<td>4.00</td>
<td>3.65</td>
<td>3.42</td>
<td></td>
</tr>
<tr>
<td>Samoan</td>
<td>17</td>
<td>4.09</td>
<td>3.40</td>
<td>3.04</td>
<td>2.83</td>
<td>2.69</td>
<td></td>
</tr>
<tr>
<td>Tamil</td>
<td>30</td>
<td>4.91</td>
<td>4.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arabic</td>
<td>32</td>
<td>5.00</td>
<td>4.21</td>
<td>3.99</td>
<td>3.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>4700</td>
<td>12.20</td>
<td>9.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>$x$</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3</td>
<td>1</td>
</tr>
<tr>
<td>1/3</td>
<td>2</td>
</tr>
<tr>
<td>1/3</td>
<td>3</td>
</tr>
<tr>
<td>Rate</td>
<td>1.67</td>
</tr>
</tbody>
</table>

**second-order code**

<table>
<thead>
<tr>
<th>$x_1, x_2$</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/6 11</td>
<td>000</td>
</tr>
<tr>
<td>1/12 12</td>
<td>0110</td>
</tr>
<tr>
<td>1/12 13</td>
<td>0111</td>
</tr>
<tr>
<td>1/12 21</td>
<td>100</td>
</tr>
<tr>
<td>1/6 22</td>
<td>001</td>
</tr>
<tr>
<td>1/12 23</td>
<td>101</td>
</tr>
<tr>
<td>1/12 31</td>
<td>110</td>
</tr>
<tr>
<td>1/12 32</td>
<td>111</td>
</tr>
<tr>
<td>1/6 33</td>
<td>010</td>
</tr>
<tr>
<td>Avg length</td>
<td>19/6</td>
</tr>
<tr>
<td>Rate</td>
<td>1.583</td>
</tr>
</tbody>
</table>

Rate = 1.5 bits/symbol

(= $H_\infty$ 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

$$C_a = \{v_{a,1}, v_{a,2}, \ldots \}$$

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 $X_{n-1} = a$,

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

Overall Rate:

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

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

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

where

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

and the overall rate is

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

where

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

= conditional entropy of $X_2$ 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 $X$,

$$H(X_2|X_1) \leq R^*_c < H(X_2|X_1) + 1,$$

where $R^*_c$ denotes the least rate of conditional codes applied to source $X$. 

January 18, 2007

LH-18
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 \( R_c^* \leq H(X_2|X_1) + 1 \) can be tightened by using the bound \( R < H(X_2|X_1) + p_{c,\text{max}} + 0.0836 \), where \( p_{c,\text{max}} \) 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 | 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:

\( R \equiv H(U) \)

Often, \( H(U) \equiv H(X_2|X_1) \)

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

Improvement:

Replace "delay" with a predictor

\( \hat{X}_n = g(X_{n-1}, X_{n-2}, \ldots) \)
Improved Versions of Conditional Lossless Coding

There are two ways to improve conditional coding.

(1) $m$th-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) $m$th-order conditional coding

In $m$th-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 $\mathbf{x} = (x_1, \ldots, x_m)$ in $A^m$, there is a prefix codebook $C_{\mathbf{x}} = \{ v_{\mathbf{x},1}, v_{\mathbf{x},2}, \ldots \}$

with lengths

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

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

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

\begin{align*}
\text{source sequence} & \quad X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6 \quad \ldots \\
\text{encoded binary sequence} & \quad e(X_3 | X_2 \ X_1) \quad e(X_5 | X_4 \ X_3) \quad e(X_6 | X_5 \ X_4)
\end{align*}

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

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

\[
R_x = \bar{L}_x = \sum_i l_{x,i} p(i|x)
\]

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

\[
R = \sum_x p(x) R_x = \sum_x p(x) \bar{L}_x
\]

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

If for each \(x\), \(C_x\) is optimal for \((p(1|x), p(2|x), p(3|x), ... )\), then

\[
H(X_n|x) \leq R_x < H(X_n|x) + 1
\]

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

= conditional entropy of \(X_2\) given \(X_{n-m},...,X_{n-1}=x\).

The overall rate is \(R_{1|m}^*\) and from the above

\[
H_{1|m} \leq R_{1|m}^* < H_{1|m} + 1
\]

where \(H_{1|m} = H(X_n|X_{n-m},...,X_{n-1}) = \sum_x p(x) H(X_n|X_{n-m},...,X_{n-1}=x)\)

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

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

---

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

\[
H_{1|m} = \Delta (H(X_n|X_{n-m},...,X_{n-1}))
\]

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

Reference: Text Compression, Bell, Cleary, Witten, p. 95.
(2) **Conditional Block 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 \(m\)th-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 \(x = (x_1, \ldots, x_m)\) in \(A^m\), there is a prefix codebook \(C_x = \{v_{x,1}, v_{x,2}, \ldots\}\) with lengths \(\{l_{x,1}, l_{x,2}, \ldots\}\) and one distinct codeword for every sequence in \(A^k\). When \((X_{n-m}, \ldots, X_{n-1}) = x\), then \((X_n, \ldots, X_{n+k-1})\) is encoded with codebook \(C_x\).

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

![Source sequence](source_sequence.png)

**Definition:**

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

**Analysis**

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

Given that \(X^{n-1}_{n-m} = x\), the average rate of the code is

\[
R_x = \frac{1}{k} \bar{L}_x = \frac{1}{k} \sum l_{x,i} \cdot p(a_i|x)
\]

where \(p(a_i|x) = Pr(X^{n+k-1}_n = a_i|X^{n-1}_{n-m} = x)\), and the overall rate is

\[
R = \frac{1}{k} \sum x p(x) R_x = \frac{1}{k} \sum x p(x) \bar{L}_x
\]

where \(p(x) = Pr(X^{n-1}_{n-m} = x) = Pr(X^{m}_1 = x)\), (the latter by stationarity).

If \(C_x\) is optimal for \((p(a_1|x), p(a_2|x), p(a_3|x), \ldots)\), then

\[
\frac{1}{k} H(X^{n+k-1}_n|x) \leq R_x < \frac{1}{k} H(X^{n+k-1}_n|x) + \frac{1}{k}
\]

where

\[
H(X^{n+k-1}_n|x) = H(X^{n+k-1}_n|X^{n-1}_{n-m} = x) = -\sum p(a_i|x) \log_2 p(a_i|x)
\]

and from the above, the overall rate is

\[
H_{k|m} \leq R^*_{k|m} < H_{k|m} + \frac{1}{k}
\]
where
\[ H_{k|m} = \frac{1}{k} H(X_{n+k-1}^n | X_{n-m}^n) = (k|m)\text{-th order entropy} \]
and
\[ H(X_{n+k-1}^n | X_{n-m}^n) = \sum_x p(x) H(X_{n+k-1}^n | x) = -\sum_x \sum_i p(x) p(a_i|x) \log_2 p(a_i|x) \]
\[ = \text{conditional entropy of } X_{n+k-1}^n \text{ given } X_{n-m}^{n-1} \]
Again, note how similar this name is to that of \( H(X_{n+k-1}^n | X_{n-m}^{n}=x) \).

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

Moreover,

(b) \[ \lim_{k \to \infty} R_{k|m}^* = H_\infty, \text{ for any } m \]

(c) \[ H_\infty \leq \lim_{m \to \infty} R_{k|m}^* < H_\infty + \frac{1}{k}, \text{ for any } k. \]

When \( m=0 \), we interpret a \((k|m)\) conditional block code to mean an unconditioned 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 established on previous pages. (b) and (c) follow from (a) and Properties 15 and 16, given later, which show

\[ \lim_{k \to \infty} H_{k|m} = H_\infty \text{ for any } m, \text{ and } \lim_{m \to \infty} H_{k|m} = H_\infty \text{ for any } k. \]

This theorem demonstrates that one can approach the least possible rate, \( H_\infty \), 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 $C_x$. 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 $H_1 = 10$, $H_\infty = 2$
The values on a diagonal, such as those printed in bold, have the same complexity.

Table of $H_{k|m}$

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

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

where \( p(x,y) \) denotes \( \Pr(X=x \text{ and } Y=y) \).

Think of \((X,Y)\) as one 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 \( Y \)

\[
H(X|Y=y) = - \sum_x p(x|y) \log_2 p(x|y),
\]

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 \( Y \).

\[
H(X) - H(X|Y) = - \sum_{x,y} p(x,y) \log_2 p(x) + \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}
\]

\[
= - \sum_{x,y} p(x,y) \log_2 \frac{p(x)p(y)}{p(x,y)} - \sum_{x,y} p(x,y) \log_2 \frac{1}{p(x,y)} + \sum_{x,y} p(x,y) \log_2 \frac{1}{p(x)p(y)}
\]

\[
\geq - \sum_{x,y} p(x,y) \left( \frac{p(x)p(y)}{p(x,y)} - 1 \right) \log_2 \frac{1}{p(x,y)} + \sum_{x,y} p(x,y) \log_2 \frac{1}{p(x)p(y)}
\]

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")

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

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

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

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

\[
H(X_1,\ldots,X_k|Y_1,\ldots,Y_m) = \sum_{y} p(y) H(X_1,\ldots,X_k|Y_1,\ldots,Y_m=y_1,\ldots,y_m) \\
= -\sum_{x,y} p(x,y) \log_2 p(x|y)
\]

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. \( H(Y_1,\ldots,Y_n|X_1,\ldots,X_m) \leq H(Y_1,\ldots,Y_n|X_1,\ldots,X_{m'}) \), 0 \leq 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:  
\[
H(X_1,\ldots,X_k) = H(X_1) + H(X_2|X_1) + H(X_3|X_1,X_2) + H(X_4|X_1,X_2,X_3) + \ldots + H(X_k|X_1,X_2,\ldots,X_{k-1})
\]

Proof: Like that of Fact 3.

8. \( H(X_1,\ldots,X_k) \leq H(X_1) + H(X_2) + \ldots + H(X_k) \) with equality iff the \( X_i \)'s are independent  

Proof: Follows from the chain rule and the fact that \( H(X_i|X_1,\ldots,X_{i-1}) \leq H(X_i) \) 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) \( H_k \triangleq \frac{1}{k} H(X_1,\ldots,X_k) \)
(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. \( H_{1|m+1} \leq H_{1|m} \)
   Proof: Follows from Property 6 and stationarity.

10. \( H_k = \frac{1}{k} \left( H_1 + H_{1|1} + H_{1|2} + \ldots + H_{1|k-1} \right) \geq H_{1|k-1} \geq H_1 \)
    Proof: \( H_k = \frac{1}{k} H(X_1, \ldots, X_k) \)
        \( = \frac{1}{k} (H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \ldots + H(X_k|X_1, X_2, \ldots, X_{k-1})) \)
        by chain rule
        \( = \frac{1}{k} (H_1 + H_{1|1} + H_{1|2} + \ldots + H_{1|k-1}) \) by stationarity
        \( \geq \frac{1}{k} (H_{1|k-1} + H_{1|k-1} + H_{1|k-1} + \ldots + H_{1|k-1}) \) by Prop. 9.
        \( = H_{1|k-1} \geq H_1 \) by Prop. 9

11. \( H_{k+1} \leq H_k \)
    Since \( H_k \)'s are nonincreasing and bounded below by zero, they must have a limit. Hence, \( H_\infty \) 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 \( H(X_1), H_{1|1}, H_{1|2}, \ldots, H_{1|k-1}, H_1 \).
    Since the extra term in \( H_{k+1} \) is no larger than all other terms, \( H_{k+1} \leq H_k \).

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 \( H_\infty \).

13. \( H_{k|m} = \frac{1}{k} \left( H_{1|m} + H_{1|m+1} + \ldots + H_{1|m+k-1} \right) \)
    Proof: \( H_{k|m} = \frac{1}{k} H(X_{n+k-1}|X_{n-m}) \)
        \( = \frac{1}{k} \left( H(X_n|X_{n-m}^{n-1}) + H(X_{n+1}|X_{n-m}^{n-1}) + \ldots + H(X_{n+k-1}|X_{n-m}^{n+k-2}) \right) \)
        \( = \frac{1}{k} (H_{1|m} + H_{1|m+2} + \ldots + H_{1|m+k-1}) \) by stationarity

14. \( H_{1|k+m} \leq H_{k|m} \leq H_k \)
    Proof:
    (a) \( H_{k|m} = \frac{1}{k} H(X_n^{n+k-1}|X_{n-m}^{n-1}) \leq \frac{1}{k} H(X_n^{n+k-1}) = H_k \)
    (b) By Prop. 13, \( H_{k|m} \) is the average of terms, each of which is at least \( H_{1|k+m} \). Therefore, \( H_{k|m} \geq H_{1|k+m} \).
15 \( H_{k+1|m} \leq H_{k|m} \)
\( H_{k|m+1} \leq H_{k|m} \)

Therefore,
\[
H_1 \geq H_2 \geq H_3 \geq \ldots \text{ decreases with } k
\]
\[
H_{1|1} \geq H_{2|1} \geq H_{3|1} \geq \ldots
\]
\[
H_{1|2} \geq H_{3|2} \geq H_{3|2} \geq \ldots
\]
\[
\ldots \ldots \ldots \ldots \ldots
\]
\[
\text{decreases with } m
\]

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

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

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 \( H_{k|m} = \frac{1}{k} (H_{1|m} + H_{1|m+2} + \ldots + H_{1|k+m}) \)

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:
\[
H(X_1) = H_1 = H_k = H_\infty = H_{1|1} = H_{1|k} = H_{k|m}, \text{ for all } k,m
\]

B. (first-order) Markov source:

**Definition:** A stationary source is first-order Markov if
\[
p(x_m|x_1,\ldots,x_{m-1}) = p(x_m|x_{m-1}) \text{ for all } m \text{ and } x_1,\ldots,x_m
\]

Then for any \( m \geq 1, \)
\[
H_{1|m} = H(X_m|X_1,\ldots,X_{m-1}) = - \sum_{x_1\ldots x_m} p(x_1,\ldots,x_m) \log_2 p(x_m|x_1,\ldots,x_{m-1})
\]
\[
= - \sum_{x_1\ldots x_m} p(x_1,\ldots,x_m) \log_2 p(x_m|x_{m-1}) = - \sum_{x_{m-1},x_m} p(x_{m-1},x_m) \log_2 p(x_m|x_{m-1})
\]
\[
= H(X_m|X_{m-1}) = H(X_2|X_1) = H_{1|1} \text{ by stationarity}
\]

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 \( H_\infty \), as it should.
Similarly, one can show
\[ H_{k|m} = H_{1|1} = H_\infty \] 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. \textbf{nth-order Markov source:}

\textbf{Definition:} A stationary source is \textit{nth-order Markov} if
\[ p(x_k|x_1, \ldots, x_{k-1}) = p(x_k|x_{k-n}, \ldots, x_{k-1}) \] 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 \( k > n \),
\[ H_k = \frac{1}{k} (nH_n + H_{1|n} + H_{1|n} + \ldots + H_{1|k-1}) = \frac{1}{k} (nH_n + H_{1|n} + H_{1|n} + \ldots + H_{1|n}) \]
\[ = \frac{n}{k} H_n + \frac{k-n}{k} H_{1|n} \]
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.