Higher-Order Lossless Source Coding

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

(a) Fixed-to-variable length codes (FVLC)
(including the first-order codes discussed earlier)

(b) Variable-to-fixed-length (VFLC)

(c) Variable-to-variable-length codes (VVLC)

(d) 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 (1a) and (2) here. Will discuss (1d), (3) and (4) later as time permits.

Example of FVLC

Source: stationary random 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 X is stationary with known probability distribution, i.e.

$$Pr(X_1=x_1,...,X_k=x_k)$$

is assumed known for all choices of k and $x_1,...,x_k$ and

$Pr(X_1=x_1,...,X_k=x_k) = Pr(X_{n+1}=x_1,...,X_{n+k}=x_k)$ for any n, (this is stationarity)

Source 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 $p(\underline{x}) = p_{\underline{X}}(\underline{x}) = Pr(\underline{X}=\underline{x})$

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 first-order variable-length codes, except that a larger codebook is applied to blocks of length k, as opposed to a smaller codebook being applied to 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,

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

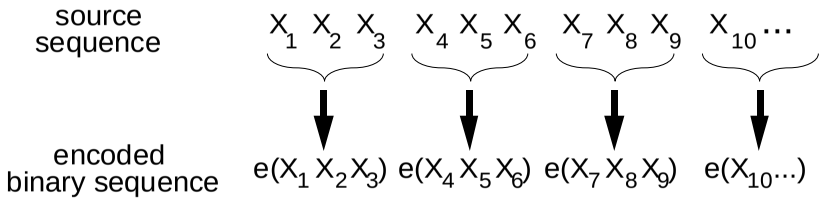
with each codeword of the form $\underline{v}_i = (v_{i,1},\ldots,v_{i,l_i}) \in \{0,1\}^*$, where $\{0,1\}^*$ is the set of all finite-length binary sequences.

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 rule 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

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

Rate

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

Key Questions

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

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

Define:

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

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

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

FVL Coding Theorem 1:

$$H(X_1,...,X_k) \le \overline{l}_k^* < H(X_1,...,X_k) + 1$$

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

where

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

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

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 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. (Result from analysis (Math 451).) Since the H_k's are nonincreasing the limit is also the inf.

Define:

$$H_\infty = \lim_{k\to\infty} H_{k+1} = \textit{entropy rate of the source } X$$

Coding Theorem 1 and Fact 1 imply that a code of blocklength k has rate

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

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

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

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, an IID source with three equiprobable letters.

$R_1^* = \frac{5}{3} = 1.67$ ($\{0, 10, 11\}$ is an optimal code with blocklenght 1)

$R_1^* = \frac{31}{18} = 1.72$ (5 codewords of length 3, 4 codewords of length 4)

$R_1^* = \frac{130}{81} = 1.60$ (5 codewords of length 4, 22 codewords of length 4)

(2) 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.

(3) 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.

(4) 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.

Example: Stationary Markov source X with

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

(5) "Redundancy of a code" $\triangleq$ Rate - H_∞

By Coding Theorem 1:

redundancy of the best FVL code with blocklength $k \leq 1/k$.

(6) Tighter upper bound on R_k^*:

It can be shown that

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

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

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

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

(7) Although there are types of lossless codes not covered by FVL Coding Theorem 2, it turns out that for a stationary source, no code can has rate less than the H_∞.

(8) 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.
(Conversion was needed to obtain H_2, H_3, H_4 from the table given there.)