

符号理論・暗号理論
- No.3 ハフマン符号化 -

渡辺 裕

Coding Theory / Cryptography
- No.3 Huffman Coding -

Hiroshi Watanabe

固定長符号と可変長符号(1)

- 固定長符号 (Fixed Length Code: FLC)
 - シンボルA,B,C,...を表現するバイナリ符号はすべて同じ長さ
 - シンボルA,B,C,...の生起確率が等しい場合に有効
 - 自然2進符号、Gray Code
- 可変長符号 (Variable Length Code: VLC)
 - バイナリ符号の長さがシンボルA,B,C,...ごとに異なる符号
 - シンボルA,B,C,...の生起確率が異なる場合に有効
 - ハフマン符号、算術符号
 - 可変長符号化を行うことを、エントロピー符号化 (Entropy Coding) と呼ぶこともある

Fixed Length Code and Variable Length Code(1)

- Fixed Length Code: FLC
 - Equal length binary code for Symbol A,B,C,...
 - Effective when occurrence probability for A,B,C,... is same
 - Natural binary code, Gray Code
- Variable Length Code: VLC
 - Unequal length binary code for Symbol A,B,C,...
 - Effective when occurrence probability for A,B,C,... is different
 - Huffman coding, Arithmetic coding
 - Variable length coding is sometimes called Entropy Coding

固定長符号と可変長符号(2)

シンボル	生起確率	固定長符号	ハフマン符号
A	0.4	000	0
B	0.2	001	100
C	0.15	010	101
D	0.1	011	110
E	0.05	100	11100
F	0.04	101	11101
G	0.03	110	11110
H	0.03	111	11111
エントロピー = 2.44 ビット		3 ビット/シンボル	2.5 ビット/シンボル

Fixed Length Code and Variable Length Code(2)

Symbol	Probabilit	FLC	Huffman Code
A	0.4	000	0
B	0.2	001	100
C	0.15	010	101
D	0.1	011	110
E	0.05	100	11100
F	0.04	101	11101
G	0.03	110	11110
H	0.03	111	11111
Entropy = 2.44 bit		3 bit/symbol	2.5 bit/symbol

符号の種類

- 非ブロック符号..... (算術符号)
- ブロック符号
 - 特異な符号
 - 非特異な符号
 - 一意復号不可能な符号
 - 一意復号可能な符号
 - 瞬時復号不可能な符号
 - 瞬時復号可能な符号..... (ハフマン符号)

Type of Codes

- Non-block Code..... (Arithmetic Code)
- Block Code
 - Singular Code
 - Non-singular Code
 - Uniquely undecodable Code
 - Uniquely Decodable Code
 - Non-instant decodable code
 - Instant decodable code..... (Huffman Code)

平均符号長

- 情報源 $s_i (i=1,2,\dots,N)$ のシンボルの出現確率を p_i とする。
- それぞれのシンボルに割り当てる符号語の長さを l_i とすれば、平均符号長 L は次式で与えられる。

$$L = \sum_{i=1}^N l_i p_i$$

Average Code Length

- Let the occurrence probabilities of source symbol $s_i (i=1,2,\dots,N)$ be p_i
- When code length to each symbol is noted as l_i , an average code length L is given by

$$L = \sum_{i=1}^N l_i p_i$$

平均符号長(2)

■ 計算例

シンボル	生起確率	ハフマン符号
A	0.4	0
B	0.2	100
C	0.15	101
D	0.1	110
E	0.05	11100
F	0.04	11101
G	0.03	11110
H	0.03	11111

$$\begin{aligned}
 L &= \sum_{i=1}^8 l_i p_i \\
 &= 1 \times 0.4 \\
 &\quad + 3 \times 0.2 + 3 \times 0.15 + 3 \times 0.1 \\
 &\quad + 5 \times 0.05 + 5 \times 0.04 \\
 &\quad + 5 \times 0.03 + 5 \times 0.03 \\
 &= 2.5
 \end{aligned}$$

Average Code Length(2)

■ Example

Symbol	Probability	Huffman code
A	0.4	0
B	0.2	100
C	0.15	101
D	0.1	110
E	0.05	11100
F	0.04	11101
G	0.03	11110
H	0.03	11111

$$\begin{aligned}
 L &= \sum_{i=1}^8 l_i p_i \\
 &= 1 \times 0.4 \\
 &\quad + 3 \times 0.2 + 3 \times 0.15 + 3 \times 0.1 \\
 &\quad + 5 \times 0.05 + 5 \times 0.04 \\
 &\quad + 5 \times 0.03 + 5 \times 0.03 \\
 &= 2.5
 \end{aligned}$$

コンパクト符号

- 情報源 S に対して、シンボルを一意復号可能な符号に符号化するとき、平均符号長 L を最小とする符号を、コンパクト符号と呼ぶ。
- ハフマンにより与えられたコンパクト符号の構成法を、ハフマン符号と呼ぶ。
- ところで、一意復号可能な符号とは？

Compact Code

- When symbol of the source S is encoded to uniquely decodable code, the code, which minimize the average code length L , is called "compact code"
- Compact code structure given by Huffman is called "Huffman code"
- By the way, what is "uniquely decodable code"?

一意復号可能な符号

- 一意復号可能な符号
 - 符号系列は、“1”が出現してから連続する“1”の個数を数えれば、B,C,Dの区別が可能
 - それ以外の場合は“0”が単独で存在するので常に一意に復号できる。
 - 符号系列“0110”は“0”+“110”(AC)と復号できる

シンボル	符号語
A	0
B	10
C	110
D	111

Uniquely Decodable Code

- Uniquely Decodable Code
 - B,C,D can be distinguished by counting the number of successive "1" after the appearance of "1" in the code sequence
 - Otherwise, "0" exists alone, it can be decoded uniquely
 - Code sequence "0110" can be decoded to "0"+"110" (AC)

Symbol	Code
A	0
B	10
C	110
D	111

一意復号可能な符号(2)

- 一意復号不可能な符号
 - 符号系列“0110”は“01”+“10”(DB)および“0”+“11”+“0”(ACA)の2通りの復号結果の可能性がある、一意に復号できない

シンボル	符号語
A	0
B	10
C	11
D	01

Uniquely Decodable Code(2)

- Uniquely un-decodable Code
 - Code sequence "0110" can be decoded to 2 ways. "01"+"10" (DB) and "0"+"11"+"0" (ACA). This sequence cannot be decoded uniquely.

Symbol	Code
A	0
B	10
C	11
D	01

非瞬時符号

- 一意復号可能な符号であっても、符号到着と同時に、一意に復号できない符号が存在する。これを非瞬時符号と呼ぶ。

Non-instant Code

- Even though a code is uniquely decodable, there is a case that a code cannot be decoded instantaneously. It is called non-instant code.

非瞬時符号(2)

■ 非瞬時符号の例

- 一意復号可能な符号であるが、非瞬時符号である。

シンボル	符号語
A	0
B	01
C	011
D	111

- 符号系列 "01111110..." は "0"+"111"+"111"+"0..." (ADD...) と一意に復号できるが、"01111..." の時点では "01"+"111" (BD) に解釈でき、"011111..." の時点では "011"+"111" (CD) に解釈できる。
- ADDに確定できるのは、2度目の"0"が到着してから。

Non-instant Code(2)

■ Example of Non-instant Decodable Code

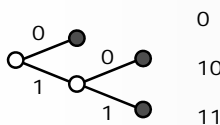
- Uniquely Decodable Code, but Non-instant Code

Symbol	Code
A	0
B	01
C	011
D	111

- Code sequence "01111110..." can be uniquely decoded as "0"+"111"+"111"+"0..." (ADD...) however, at the time of "01111..." it can be interpreted as "01"+"111" (BD), at the time of "011111...", it can be interpreted as "011"+"111" (CD)
- After the arrival of the second "0", ADD is determined.

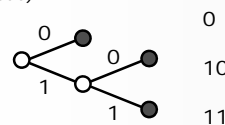
瞬時符号の条件

- 非瞬時符号
 - 符号語と同じパターンが他の符号語の頭の部分に現れている
- プレフィックス条件
 - どの符号も他の符号の語頭 (Prefix) にならない
- 符号の木
 - 各符号が枝の先(終端接点)にのみ割り当てられる



Condition for Instant Code

- Non-instant Code
 - Same pattern appears at the head of other code
- Prefix condition
 - Any code does not appear at other code's header
- Code Tree
 - Each code assigned to end of branches (terminated node)



ハフマン符号

■ 構成法

- 各シンボルに対する枝を作り出現確率を与える
- 確率の最も小さい2つの枝を節点で結び、符号“0”“1”を与える
- 統合した枝の確率の和を求め、ひとつの枝とみなす
- 再度、確率も最も小さい2つを枝を探す
- この処理を枝がひとつになるまで繰り返す

Huffman Code

■ Structure

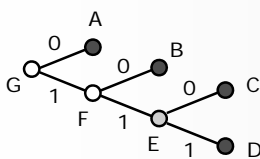
- Make branch to each symbol and assign probability
- Combine two branches giving least two probability and assign "0""1"
- Obtain probability of unified branch, and regarded as one branch
- Again, search branches giving least two probabilities
- Repeat this procedure until branches unified to one

ハフマン符号(2)

■ ハフマン符号の構成例

- CとDを統合しE(0.15)とする
- BとEを統合しF(0.4)とする
- FとAを統合しG(1.0)とする

シンボル	確率	符号語
A	0.6	0
B	0.25	10
C	0.1	110
D	0.05	111

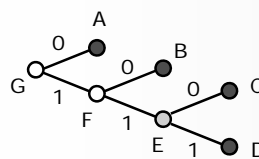


Huffman Code(2)

■ Example of constructing Huffman code

- Unify C and D, make E(0.15)
- Unify B and E, make F(0.4)
- Unify F and A, make G(1.0)

Symbol	Probability	Code
A	0.6	0
B	0.25	10
C	0.1	110
D	0.05	111

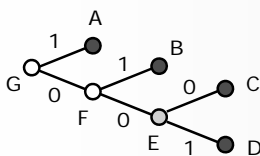


ハフマン符号(3)

■ 別のハフマン符号の構成例

- CとDを統合しE(0.15)とする
- BとEを統合しF(0.4)とする
- FとAを統合しG(1.0)とする

シンボル	確率	符号語
A	0.6	1
B	0.25	01
C	0.1	000
D	0.05	001



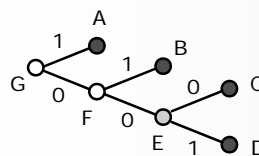
ハフマン符号の構成は1種類ではない

Huffman Code(3)

■ Another example of Huffman code

- Unify C and D, make E(0.15)
- Unify B and E, make F(0.4)
- Unify F and A, make G(1.0)

Symbol	Probability	Code
A	0.6	1
B	0.25	01
C	0.1	000
D	0.05	001



Structure of Huffman Code is not one way

ハフマン符号の応用

- 画像符号化におけるハフマン符号
 - 2値画像符号化におけるランレングス符号化
 - H.261におけるDCT係数の2次元VLC
 - 動きベクトルのVLC
 - Coded Block PatternのVLC
 - Macroblock typeのVLC

Application of Huffman Code

- Huffman Code at Image Coding
 - Run-length coding at binary image coding
 - 2-D VLC of DCT coefficient at H.261
 - VLC for motion vectors
 - VLC for Coded Block Pattern
 - VLC for Macroblock type

ランレングス符号化

- 2値画像では、白画素と黒画素の連続する長さをシンボルとみなし、ハフマン符号化を行う
- 白画素を0, 黒画素を1とすると、連続する0の個数 (Zero Run Length), 連続する1の個数に対してハフマン符号を設計

0 0 0 0 1 1 0 0 0 0 1 1 1 0 0
|← 4 →|←2→|← 5 →|← 3 →|

Run-length Coding

- Successive length of white and black pixels are regarded as symbol and Huffman coding is applied at binary image coding
- White pixel "0", black pixel "1", Huffman code is designed to successive number of "0" (Zero Run Length) and "1"

0 0 0 0 1 1 0 0 0 0 1 1 1 0 0
|← 4 →|←2→|← 5 →|← 3 →|

2次元VLC

- 2次元DCT係数をZigzag Scan, 1次元のデータ配列に変換
- 係数が0でないデータと0の部分に分離
- 有効係数(非0係数)の量子化レベル(量子化インデクス)と、その直前に連続する0の個数 (Zero Run Length) の組み合わせをシンボルとして扱う
- 生起確率の高い部分の (Run,Level) の組み合わせをハフマン符号化 (Variable Length Code: VLC)
- それ以外の部分は、Escape符号との組み合わせで固定長符号化 (Fixed Length Code: FLC)

2-dimensional VLC

- Arrange 2-D DCT coeffs. by zigzag scan to 1-D data
- Separate coeffs. to non-zero and zero ones parts
- Treat the combination of quantization levels of significant coeffs. (non-zero coeffs.) and successive "0"s (zero run-length) before it as a symbol
- High probable part of the combination of (Run,Level) is Huffman coded (Variable Length Code: VLC)
- Low probable part is fixed length coded by the combination of Escape code and FLC (Fixed Length Code)

2次元VLC(2)

■ 8x8 DCT係数の算出

入力画像ブロック	DCT 係数
40 41 40 41 42 72 73 74	699 -147-15 29 9 -4 -3 -5
41 40 40 44 78 110 110 112	-142 -68 13 17 -7 -4 3 -3
40 42 41 79 111 112 112 111	-1 63 33-13-22 4 10 -1
40 40 43 83 110 111 111 113	→ -31 18 4-10-5 5 5 -5
51 52 56 83 110 110 111 110	-31 7 2 -2 -1 2 -1 0
72 73 83 110 111 114 110 112	-1 20 -5 -7 6 2 -3 -4
110 111 110 112 112 111 110 113	-6 3-14 -4 6 5 -4 -4
112 111 112 110 110 112 114 111	0 10 0 1 1 -3 -4 3

2-D VLC(2)

■ Calculation of 8x8 DCT Coefficients

Input Image Block	DCT Coefficients
40 41 40 41 42 72 73 74	699 -147-15 29 9 -4 -3 -5
41 40 40 44 78 110 110 112	-142 -68 13 17 -7 -4 3 -3
40 42 41 79 111 112 112 111	-1 63 33-13-22 4 10 -1
40 40 43 83 110 111 111 113	→ -31 18 4-10-5 5 5 -5
51 52 56 83 110 110 111 110	-31 7 2 -2 -1 2 -1 0
72 73 83 110 111 114 110 112	-1 20 -5 -7 6 2 -3 -4
110 111 110 112 112 111 110 113	-6 3-14 -4 6 5 -4 -4
112 111 112 110 110 112 114 111	0 10 0 1 1 -3 -4 3

2次元VLC(3)

■ 量子化 (量子化ステップ16)

量子化 DCT 係数	IDCT 値 (復号画像)
699 -152-0-24 0 0 0 0	47 43 34 34 55 80 86 76
-136 -72-0 24 0 0 0 0	38 36 38 52 79 104 112 108
0 56 40 0-24 0 0 0	39 37 47 73 99 114 118 119
-24 24 0 0 0 0 0 0	→ 41 39 53 83 105 110 110 113
-24 0 0 0 0 0 0 0	47 47 62 90 108 110 111 116
0 24 0 0 0 0 0 0	81 81 88 100 108 110 111 114
0 0 0 0 0 0 0 0	111 113 110 105 103 108 112 111
0 0 0 0 0 0 0 0	110 115 111 99 99 112 122 122

Zigzag scanning

2-D VLC(3)

■ Quantization (Step=16)

Quantized DCT Coeffs.	IDCT Value (Decoded Image)
699 -152-0-24 0 0 0 0	47 43 34 34 55 80 86 76
-136 -72-0 24 0 0 0 0	38 36 38 52 79 104 112 108
0 56 40 0-24 0 0 0	39 37 47 73 99 114 118 119
-24 24 0 0 0 0 0 0	→ 41 39 53 83 105 110 110 113
-24 0 0 0 0 0 0 0	47 47 62 90 108 110 111 116
0 24 0 0 0 0 0 0	81 81 88 100 108 110 111 114
0 0 0 0 0 0 0 0	111 113 110 105 103 108 112 111
0 0 0 0 0 0 0 0	110 115 111 99 99 112 122 122

Zigzag scanning

2次元VLC(4)

■ Zigzag Scan による量子化値と0ランレングス

```
696 -152 -136 0 -72 0 24 0 56 -24 -24 24 40 24
00000000 24 00 -24 00000000...000
```

■ Zigzag Scan による量子化レベル(インデクス)と0ランレングス

```
43 -9 -8 0 -4 0 1 0 3 -1 -1 1 2 1
00000000 1 00 -1 00000000...000
```

2-D VLC(4)

■ Quantized Value and "0" Run-length by zigzag scan

```
696 -152 -136 0 -72 0 24 0 56 -24 -24 24 40 24
00000000 24 00 -24 00000000...000
```

■ Quantized Level (Index) and "0" Run-length by zigzag scan

```
43 -9 -8 0 -4 0 1 0 3 -1 -1 1 2 1
00000000 1 00 -1 00000000...000
```

2次元VLC(5)

■ Zero-RunとLevelの組み合わせ

Data #	Run	Level
1	(DC)	43
2	0	9
3	0	-8
4	1	-4
5	1	1
6	1	3
7	0	-1
8	0	-1

Data #	Run	Level
9	0	1
10	0	2
11	0	1
12	8	1
13	2	-1
14	EOB	

EOB: End of Block

2-D VLC(5)

■ Combination of Zero-Run and Level

Data #	Run	Level
1	(DC)	43
2	0	9
3	0	-8
4	1	-4
5	1	1
6	1	3
7	0	-1
8	0	-1

Data #	Run	Level
9	0	1
10	0	2
11	0	1
12	8	1
13	2	-1
14	EOB	

EOB: End of Block

2次元VLC(6)

■ DC成分: 8bit FLC

- $43 = 32+8+2+1 = "0010\ 1011"$

■ (Run, Level): Huffman Coding

- (0, 9) = "0000 0001 1000 0"
- (0, -8) = "0000 0001 1101 1"
- (1, -4) = "0000 0011 00 1"
- (1, 1) = "011 0"
- (1, 3) = "0010 0101 0"
- (0, -1) = "11 1"
- (0, -1) = "11 1"
- (0, 1) = "11 0"

2-D VLC(6)

■ DC Component: 8bit FLC

- $43 = 32+8+2+1 = "0010\ 1011"$

■ (Run, Level): Huffman Coding

- (0, 9) = "0000 0001 1000 0"
- (0, -8) = "0000 0001 1101 1"
- (1, -4) = "0000 0011 00 1"
- (1, 1) = "011 0"
- (1, 3) = "0010 0101 0"
- (0, -1) = "11 1"
- (0, -1) = "11 1"
- (0, 1) = "11 0"

2次元VLC(7)

■ (Run, Level): Huffman Coding

- (0, 2) = "0100 0"
- (0, 1) = "11 0"
- (0, 1) = "11 0"
- (8, 1) = "0000 111 0"
- (2, -1) = "0101 1"
- EOB = "10"

2-D VLC(7)

■ (Run, Level): Huffman Coding

- (0, 2) = "0100 0"
- (0, 1) = "11 0"
- (0, 1) = "11 0"
- (8, 1) = "0000 111 0"
- (2, -1) = "0101 1"
- EOB = "10"

2次元VLC(8)

■ H.261 2-D VLC の符号長

		Level															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	... 127
RUN	0	2	5	6	8	9	9	11	13	13	13	13	14	14	14	14	
	1	4	7	9	11	13	14	14									
	2	5	8	11	13	14											
	3	6	9	13	14												
	4	6	11	13													
	5	7	11	14													
	6	7	13														
	7	7	13														
	8	8	13														
	9	8	14														
	10	9	14														
	11	9															
	12	9															

2-D VLC(8)

■ Code length of H.261 2-D VLC

		Level															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	... 127
RUN	0	2	5	6	8	9	9	11	13	13	13	13	14	14	14	14	
	1	4	7	9	11	13	14	14									
	2	5	8	11	13	14											
	3	6	9	13	14												
	4	6	11	13													
	5	7	11	14													
	6	7	13														
	7	7	13														
	8	8	13														
	9	8	14														
	10	9	14														
	11	9															
	12	9															

2次元VLC(9)

		Level															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	... 127
RUN	13	9															
	14	11															
	15	11															
	16	11															
	17	13															
	:	:															
	21	13															
	22	14															
	:	:															
	26	14															
	:	:															
	63																

2-D VLC(9)

		Level															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	... 127
RUN	13	9															
	14	11															
	15	11															
	16	11															
	17	13															
	:	:															
	21	13															
	22	14															
	:	:															
	26	14															
	:	:															
	63																