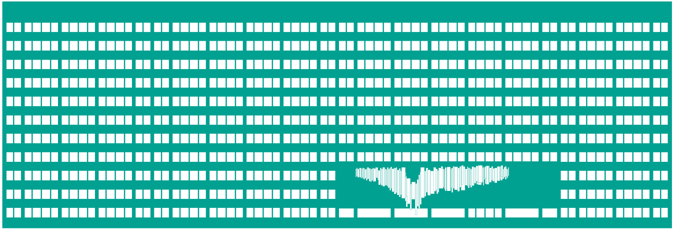
VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

VSB TECHNICAL
UNIVERSITY
OF OSTRAVA



www.vsb.cz

Analysis and Signal Compression

Information and Probability Theory

Michal Vašinek

VŠB – Technická univerzita Ostrava

FEI/EA404

michal.vasinek@vsb.cz

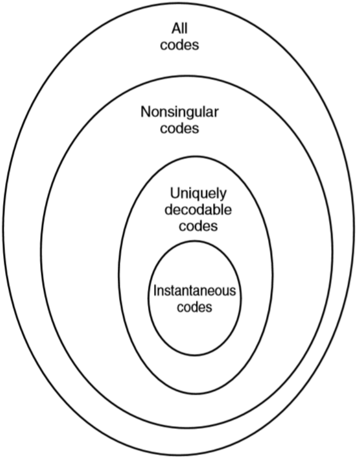
2020, February 12th

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Content

- Classes of Codes
- Variable-length coding:
  - Unary coding
  - Golomb coding
  - Elias Gamma and Delta coding
  - Fibonacci coding

Codes



Obrázek: Classes of codes, Cover and Thomas, Elements of Information Theory, p. 106.

Suppose we have an alphabet $\Sigma = \{a, b, c\}$ and we assigned to symbols of Σ these codes:

- $C(a) = 0$
- $C(b) = 00$
- $C(c) = 01$

Task (1 pt)

Try to decode sequence $s = 00001$.

Nonsigular code

- Let X be a range of random variable $\mathbf{X}$, for instance the alphabet of input data.
- Let D be d-ary alphabet of output, for instance binary alphabet $D = \{0, 1\}$.
- Let $C : X \to D^*$ be a mapping. Mapping C assigns a code from D^* to symbols from X.

Nonsingular Code

A code is said to be nonsingular if every element of the range of X maps into different string in D^*; that is:

$$x \neq x' \to C(x) \neq C(x')$$

Nonsingular code

Nonsingular Code

A code is said to be nonsingular if every element of the range of X maps into different string in D^*; that is:

$$x \neq x' \rightarrow C(x) \neq C(x')$$

- Let C('a') = 0, C('b')=00, C('c')=01 be codewords of code C.
- Encode the sequence $s = abc$, i.e. $C(s) = 0\ 00\ 01 = 00001$.
- We can decode in many ways: aaac, bac, abc.
- Can be solved by adding special separating symbol. For instance with code 11.

Motivation

Suppose we have an alphabet $\Sigma = \{a, b, c\}$ and we assigned to symbols of Σ these codes:

- $C(a) = 10$
- $C(b) = 00$
- $C(c) = 11$
- $C(d) = 110$

Task (1 pt)

Try to decode the sequence $s = 11110001100$.

Definition

The extension C^* of a code C is the mapping from finite length strings of X to finite-length strings of D, defined by:

$$C(x_1 x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n)$$

For instance, if $C(x_1) = 00$ and $C(x_2) = 11$ then $C(x_1 x_2) = 0011$.

Definition

A code is called uniquely decodable if its extension is nonsingular.

Any encoded string in a uniquely decodable code has only one possible source string producing it.

Instantaneous (Prefix) Codes

Definition

A code is called a prefix code or an instantaneous code if no codeword is a prefix of any other codeword.

- Symbol x_i can be decoded as soon as we come to the end of the codeword.
- Self-punctuating code. We don't need delimiters.

Codes - example

X	Singular	Nonsingular	Uniquely Decodable	Prefix
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

- Uniquely decodable: if the first two bits are 11, then we have to look at the next bit.
- Prefix: if the first two bits are 10, we know that there is no codeword with prefix 10—.

Unary Coding

- Simplest prefix-code: sequence of zeros delimited by one.

X	Codeword	X	Codeword
0	1	6	000 0001
1	01	7	0000 0001
2	001	8	0 0000 0001
3	0001	9	00 0000 0001
4	0 0001	10	000 0000 0001
5	00 0001	11	0000 0000 0001

- Example: $C(120) = C(1)C(2)C(0) = 010011$

Unary Coding - Properties

X	Codeword
0	1
1	01
2	001

- We can start decoding in any position in the compressed representation:

$$01001101 \rightarrow_3 01101$$

Start decoding from the third(indexed from zero) position:

1. Find the first 1 from the current position: 4th position
2. Decode the next codeword: $D(1) \rightarrow 0$
3. Decode the next codeword: $D(01) \rightarrow 1$

Bonus

Task (2 pts)

Say we encode numbers i, where $i \in \{0 \dots N\}$ by unary coding. Each number occurs with probability $p(i)$. What is the average code length?

Unary Coding - Properties

- Average code length:

$$\hat{C} = \sum_{i \in X} (i+1)p(i)$$

- Can be such as a simple code useful?
- Suppose a (time, value) tupples, for instance trading price records: (1, 1000), (1, 1002), (2, 1100), (4, 998)...
- The time coordinate never decreases and may be transformed into differences:

$$(1-0),(1-1),(2-1),(4-2),\cdots \rightarrow 1,0,1,2,\ldots$$

- Encode differences using unary code: $C(1012\ldots) = 01101001\ldots$
- Use other coding techniques to encode value.

Suppose we have the following probability distribution:

- $p(0) = 0.1$
- $p(1) = 0.2$
- $p(2) = 0.3$
- $p(3) = 0.4$

Using unary code we would obtain a code with average codeword length: $0.1 * 1 + 0.2 * 2 + 0.3 * 3 + 0.4 * 4 = 3$ bits.

Fastest win (1 pt)

Assuming you are forced to use unary codes. What would you do to improve this result?

Golomb Codes

- Invented in 1960s by Solomon W. Golomb.
- It is an optimal prefix code for symbols following geometric distributions.
- Applicable in situations when small values in the input stream are significantly more likely than large values.
- The code for number x is determined by adjustable parameter M.
  1. Apply integer division of x by M.
  2. Integer part q is encoded by unary coding and the remainder r by truncated binary coding.

Truncated binary coding

- Suppose the following example: we wish to encode numbers from $N = \{0, 1, 2, 3, 4\}$.
- To distinguish between these numbers we need $\log_2 5 = 2.32$ bits, we can work only with integers => $k = \lceil \log_2 5 \rceil = 3$ bits.
- Notice that binary codes 101, 110 and 111 are unused.
- Better solution: use $(k-1)$ bits to encode first $2^k - |N|$ numbers and to the rest add Offset and encode $(X + Offset)$ in binary using k bits.

X	Offset	Encoded Value	Binary	Truncated
0	0	0	000	00
1	0	1	001	01
2	0	2	010	10
3	3	6	011	110
4	3	7	100	111

Bonus

Fastest win (1pt)

What has to apply for the size of the input alphabet to have a binary code equal to a truncated binary code?

Golomb Codes - Example

- Let $M = 5$, the number we wish to encode be 13.
- The quotient $q = 13/5 = 2$ and remainder $r = 3$.
- The number of bits needed in truncated binary coding $k = \lceil \log_2 M \rceil = 3$.
- Encode 2 by unary: $C(2) = 001$ and encode 3 by truncated binary coding: $C(3) = 110$.
- Decode by reversing the process:
  1. Start with the sequence 001110.
  2. Read sequence of zeros, stop if you obtain the first one, decode unary code for quotient.
  3. Use $c = \lceil \log_2 M \rceil - 1 = 2$ bits. Read the next c (in this case 2) bits, i.e. 11. If the value is smaller than $2^k - M$ then decode these $(k-1)$ bits as binary number otherwise decode as k bits binary number. Remainder $r = D(110) = 6 \rightarrow 3$.
  4. Decode the number x as $x = qM + r$.

Golomb Codes - Applications

- Historically developed for encoding of runs of zeros or ones.
- Rice codes - subset of Golomb codes when M is a power of two => simplification in remainder coding, fixed length binary code.
- Encoding of geometrically distributed signals.
- Audio codecs - Shorten, FLAC.
- JPEG - LS.

- Positive integer coding developed by Peter Elias.
- Assume we have a number x, and its binary representation $b(x)$.
- We encode the length $|b(x)| - 1$ using unary code and the number itself is stored in binary.
- Example: $x = 10$, $b(10) = 1010$ then the length $|b(10)| = 4$, encode $4 - 1$ in unary as $C(4 - 1) = 0001$.
- Note that each positive integer represented in binary starts with 1 => we can omit this 1.
- Elias gamma $\gamma(10) = 0001010$.

Elias Gamma Decoding

Decoding:

1. Read initial zeros and count them => we obtain number n.
2. Compute $n+1$ to obtain the length of binary representation.
3. Read next $(n+1)$ bits and convert them to decimals to obtain x.

Example:

- Let $\gamma(x) = 0001010$.
- $n = 3$ as there are three zeros in the beginning of γ-code.
- 4 bits are used to represent binary number.
- Binary code 1010 is easily converted to decimals: $1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 10$.

Bonus

Derive (3pts)

What is the length of the Elias gamma codeword $|\gamma(x)|$ for any positive number x?

Elias Gamma - Properties and Usage

- Number x is encoded using $2\lfloor\log_2 x\rfloor + 1$ bits.
- Elias gamma code is used in information retrieval systems to encode differences between docIDs related to particular term:
  - Let $docID$ be the index of document in collection of documents.
  - Construct a table of occurences of some term(word) in documents.
  - For each term construct a list of documentIDs in sorted order from smallest to largest.
  - Use Elias Gamma code to encode differences between two consecutive docIDs as: $\gamma(docID_i - docID_{i-1})$.

Elias Delta

- To represent number x, Elias delta uses: $\lfloor \log_2 x \rfloor + 2\lfloor \log_2(\lfloor \log_2(x) \rfloor + 1) \rfloor + 1$ bits.
- Uses Elias Gamma code instead of unary code:
  1. Separate x into the highest power of 2 it contains (2^N) and the remaining N binary digits.
  2. Encode $N + 1$ with Elias Gamma code.
  3. Append the remaining N binary digits to the representation of $N + 1$.

X	N	N+1	Elias δ
$1 = 2^0$	0	1	1
$2 = 2^1 + 0$	1	2	0 10 0
$3 = 2^1 + 1$	1	2	0 10 1
$4 = 2^2 + 0$	2	3	0 11 00
$5 = 2^2 + 1$	2	3	0 11 01
$6 = 2^2 + 2$	2	3	0 11 10
$7 = 2^2 + 3$	2	3	0 11 11

Elias Delta - decoding

1. Read and count zeros from the stream until you reach the first one. Call this count of zeros L.
2. Considering the one that was reached to be the first digit of an integer, with a value of 2^L, read the remaining L digits of the integer. Call this integer $N+1$, and subtract one to get N.
3. Put a one in the first place of our final output, representing the value 2^N.
4. Read and append the following N digits.

Elias Delta - decoding example

1. C(x) = 001010011
2. 2 leading zeros 00 1 010011 => $L = 2$
3. Read $L = 2$ bits following the one: 00 1 01
4. Decode $N + 1 = 00101 = 5$
5. $N = 5 - 1 = 4$, read N remaining bits to obtain 0011
6. Compute $2^N + dec(0011) = 16 + 3 = 19$

Fibonacci Code - Encoding

- Codewords contain no consecutive ones => use 11 as a codeword separating sequence.
- Fibonacci sequence: the next number is given as a sum of the two preceding Fibonacci numbers.

1. Find the largest Fibonacci number equal to or less than N; subtract this number from N, keeping track of the remainder.
2. If the number subtracted was the ith Fibonacci number $F(i)$, put a 1 in place $i-2$ in the code word (counting the left most digit as place 0).
3. Repeat the previous steps, substituting the remainder for N, until a remainder of 0 is reached.
4. Place an additional 1 after the rightmost digit in the code word.

Fibonacci Code - Encoding Example

1. Fib = 1, 2, 3, 5, 8, 13, 21
2. Encoding $x = 23$.
3. Find the largest Fibonacci number equal to or less than x => Fib = 21. 7th Fibonacci number -> our codeword will have 7+1 bits, next set bit 7 to 1.
4. Subtract 21 from x => $x_{next} = 2$.
5. The next Fib that can be subtracted is 2th Fibonacci number = 2 => set bit 2 to 1. $x_next = 2 - 2 = 0$, remainder equal to 0 so we stop. 0100001.
6. Append the final one: 01000011

Fibonacci Code - Decoding

1. Read bits until you see two consecutive ones.
2. Sum Fibonacci numbers corresponding to ones in binary string.

Example:

Fib	1	2	3	5	8	13	-
Code	1	0	1	0	0	1	1
Sum	1	0	3	0	0	13	17

DÄŻkuji za pozornost

Michal Vašinek

VŠB – Technická univerzita Ostrava

FEI/EA404

michal.vasinek@vsb.cz

2020, February 12th

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY