UNIVERSITÄT
LEIPZIG

Asymmetric Numeral System

December 30, 2024

Huo Jiang

Fakultät für Mathematik und Informatik
Universität Leipzig

Outline

1. Introduction
2. Entropy Coding
3. Asymmetric Numeral System
4. rANS
5. Streaming-rANS
6. Summary
7. Reference

Introduction

Introduction

- Hi there, this is a self inspired latex beamer template for Leipzig university.

Entropy Coding

Entropy Coding

- The theoretical limit of compression is given by Shannon entropy, which measures the minimum average number of bits needed to encode a symbol from a data source.
- Lossless compression algorithms aim to approach this limit but cannot go below it.
- The entropy depends on the probability distribution of the symbols, with more skewed distributions being more compressible.
- Shannon Entropy [1]

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

- Shannon's Source Coding Theorem [1]

$$N \cdot H(X)$$

Asymmetric Numeral System

Motivation

ANS combines the compression ratio of arithmetic coding (which uses a nearly accurate probability distribution), with a processing cost similar to that of Huffman coding [4].

Basic Idea

- Symbol spread function

Basic Idea

- Symbol spread function
- Representation of information
  (represents an entire message as a single number, similar to Arithmetic Coding) –> example: representation of {2,0,2,5,1,8}

Basic Idea

- Symbol spread function
- Representation of information
  (represents an entire message as a single number, similar to Arithmetic Coding) –> example: representation of {2,0,2,5,1,8}
- Symmetric or Asymmetric

Basic Idea

- Symbol spread function
- Representation of information
  (represents an entire message as a single number, similar to Arithmetic Coding) –> example: representation of {2,0,2,5,1,8}
- Symmetric or Asymmetric
- rANS and tANS

Symbol spread function

- the function that determines the split of the $\mathbb{N}$ subsets corresponding to different symbols [1]

Symbol spread function

- the function that determines the split of the $\mathbb{N}$ subsets corresponding to different symbols [1]
- $\bar{s} : \mathbb{N} \rightarrow \mathcal{A}, \bar{s}(x) = \text{mod}(x, b) = s$

Symbol spread function

- the function that determines the split of the $\mathbb{N}$ subsets corresponding to different symbols [1]
- $\bar{s} : \mathbb{N} \rightarrow \mathcal{A}, \bar{s}(x) = \text{mod}(x, b) = s$
- In the standard binary system example: $\bar{s}(x) = \text{mod}(x, 2)$
  e.g. splits natural numbers into subsets of even/odd numbers [2]

Symbol spread function (Exa. 1)

$S = \{0, 1\}$

Exa. 1: Standard Binary System

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
s=0																			
s=1																			

Fig.1

Symbol spread function (Exa. 2)

$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$x' = 10x + s$

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		202		2025
$s = 0$																									
$s = 1$																									
$s = 2$																									
$s = 3$																									
$s = 4$																						...		...	
$s = 5$																									
$s = 6$																									
$s = 7$																									
$s = 8$																									
$s = 9$																									

Fig.2

Representation of information (Exa. 2)

- a finite sequence of symbols/digits:

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Representation of information (Exa. 2)

- a finite sequence of symbols/digits:

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- the sequence of symbols/digits to be encoded:

$$S_{in} = (2, 0, 2, 5)$$

Representation of information (Exa. 2)

- a finite sequence of symbols/digits:

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- the sequence of symbols/digits to be encoded:

$$S_{in} = (2, 0, 2, 5)$$

- How can we represent this sequence as a single number?

Representation of information (Exa. 2)

- a finite sequence of symbols/digits:

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- the sequence of symbols/digits to be encoded:

$$S_{in} = (2, 0, 2, 5)$$

- How can we represent this sequence as a single number?
- The easiest way is to concatenate the symbols/digits:

$$x_{out} = 2025$$

Representation of Information (Exa. 2)

- encoding the sequence of digits (2,0,2,5,1,8)

$$
\begin{aligned}
x_{\text{init}} &= 0 \\
x_1 &= x_{\text{init}} \times 10 + 2 = 2 \\
x_2 &= x_1 \times 10 + 0 = 20 \\
x_3 &= x_2 \times 10 + 2 = 202 \\
x_{\text{out}} &= x_3 \times 10 + 5 = 2025
\end{aligned}
$$

Representation of Information (Exa. 2)

– What is the encoding function?

Representation of Information (Exa. 2)

- What is the encoding function?
- Encoding function

$$x_{i+1} = x_i \times 10 + s_i$$

Representation of Information (Exa. 2)

- What is the encoding function?
- Encoding function

$$x_{i+1} = x_i \times 10 + s_i$$

- x' is the x-th appearance of symbol s

$$x' = C(s, x) = x \cdot 10 + s$$

Representation of Information (Exa. 2)

$x' = 10x + s$

$s = 0$, $s = 2$, $s = 2$, $s = 5$

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		202		2025
$s = 0$	0										1										2				
$s = 1$		0										1													
$s = 2$			0										1												
$s = 3$				0										1											
$s = 4$					0										1							...		...	
$s = 5$						0										1									
$s = 6$							0										1								
$s = 7$								0										1							
$s = 8$									0										1						
$s = 9$										0										1					

Fig.3

Representation of Information (Exa. 1)

Exa. 1: Standard Binary System

X' = 2X + s

s=1 s=0 s=0 s=1

X'	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
X s=0	0		1		2		3		4		5		6		7		8		9
X s=1		0		1		2		3		4		5		6		7		8	

Fig.4

Asymmetric Numeral System | Representation of information |

Exa. 1: Standard Binary System

X' = 2X + s

s=1 s=0 s=0 s=1

X'	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
x s=0	0		1		2		3		4		5		6		7		8		9
x s=1		0		1		2		3		4		5		6		7		8	

Exa. 2: Standard Decimal System

X' = 10X + s

s=2 s=0 s=2 s=5

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		202		2025
s=0	0										1										2				
s=1		0										1													
s=2			0										1												
s=3				0										1											
s=4					0										1							...		...	
s=5						0										1									
s=6							0										1								
s=7								0										1							
s=8									0										1						
s=9										0										1					

Fig.5

Asymmetric Numeral System | Symmetric or Asymmetric |

Exa. 1: Standard Binary System

X' = 2X + s

s=1 s=0 s=0 s=1

X'	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
X s=0	0		1		2		3		4		5		6		7		8		9
X s=1		0		1		2		3		4		5		6		7		8	

Exa. 2: Standard Decimal System

X' = 10X + s

s=2 s=0 s=2 s=5

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		202		2025
s=0	0										1										2				
s=1		0										1													
s=2			0										1												
s=3				0										1											
s=4					0										1							...		...	
s=5						0										1									
s=6							0										1								
s=7								0										1							
s=8									0										1						
s=9										0										1					

Fig.6

Exa. 3: Asymmetric Binary System

- Let $\mathcal{A} = \{0, 1\}$
- $Pr(0) = 1/4, Pr(1) = 3/4$

Exa. 3: Asymmetric Binary System

- Let $\mathcal{A} = \{0, 1\}$
- $Pr(0) = 1/4, Pr(1) = 3/4$
- a cycle of length 4, which contains 1 zero and 3 ones

Exa. 3: Asymmetric Binary System

- Let $\mathcal{A} = \{0, 1\}$
- $Pr(0) = 1/4, Pr(1) = 3/4$
- a cycle of length 4, which contains 1 zero and 3 ones
- $x' = C(s, x) \approx x/p_s$

Exa. 3: Asymmetric Binary System

Exa. 3: Asymmetric Binary System

$x' \approx x/Pr(s)$ | $Pr(0) = 1/4, Pr(1) = 3/4$

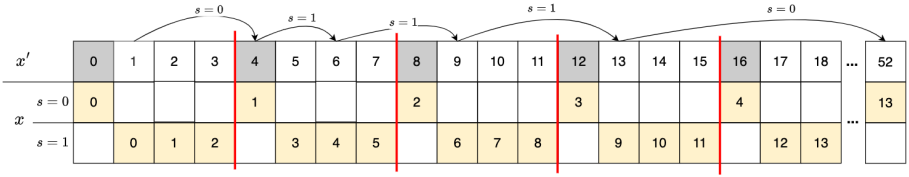


Fig.7

We refine even/odd number by changing their densities

Symmetric or Asymmetric

- Symmetric: uniform distribution
- Asymmetric: non-uniform distribution

tANS

- d

Performance

- f

rANS

rANS

- ”r” = ”range”: similar to range coding (RC)
- symbol spread function
- representation of information
- asymmetric

Exa.3

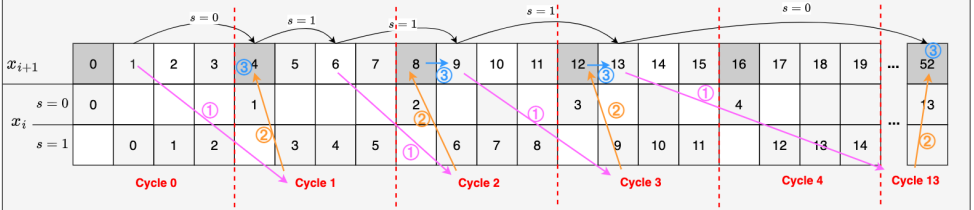
How to encode 101110? What is the encoding function $C(s, x)$?

$x' = C(s, x) \approx x/p_s$

Some notations

- $\mathcal{A} = \{s_1, s_2, \ldots, s_k\}$ is the alphabet set
- s is the symbol, if it is used in subscript: s is the current symbol, $s-1$ is the previous symbol, $s+1$ is the next symbol
- F_s is the absolute frequency of symbol s
- $m = \sum_{s \in \mathcal{A}} F_s$ is the total frequency
- $c_s = \sum_{s \in \mathcal{A}} F_{s-1}$ is the cumulative frequency (c_{s+1} is the cumulative frequency of the next symbol)
- $p_s = F_s / m$ is the probability of symbol s
- x_i is the current state (next state: x_{i+1}, previous state: x_{i-1})

$x_{i+1} \approx x_i / Pr(s)$ | $Pr(0) = 1/4, Pr(1) = 3/4$



Given current state x_i and next symbol s. How to find next state x_{i+1}, which is the x_i-th appearance of symbol s ?

STEPS / EXAMPLE	$Pr(0) = 1/4, Pr(1) = 3/4$		General
	$s = 0$	$s = 1$	
Step ①: Find cycle number k	$k = \lfloor x_i/1 \rfloor$	$k = \lfloor x_i/3 \rfloor$	$k = \lfloor x_i/F_s \rfloor$
Step ②: Navigate to the beginning (first number) of that cycle	$4 \cdot k$	$4 \cdot k$	$m \cdot k$
Step ③: Find the position of the symbol s in that cycle	part 1: first of one + 0	part 1: + mod$(x_i, 3)$	part 1: + mod(x_i, F_s)
- part 1: position of the symbol s in the sequence of the same symbol - part 2: how many other symbols before the symbol s	part 2: no other symbols + 0	part 2: only one other symbol (0) and with one of that symbol in one cycle + 1	part 2: only one other symbol (0) and with one of that symbol in one cycle $+c_s$
	$x_{i+1} = m \cdot k$	$x_{i+1} = 4 \cdot \lfloor x_i/3 \rfloor + \text{mod}(x_i, 3) + 1$	$x_{i+1} = m \cdot \lfloor x/F_s \rfloor + \text{mod}(x, F_s) + c_s$

$$x_{i+1} = C(s, x_i) = m \cdot \lfloor x_i/F_s \rfloor + \text{mod}(x_i, F_s) + c_s$$

Fig.8

Encoding Function [2] [5]:

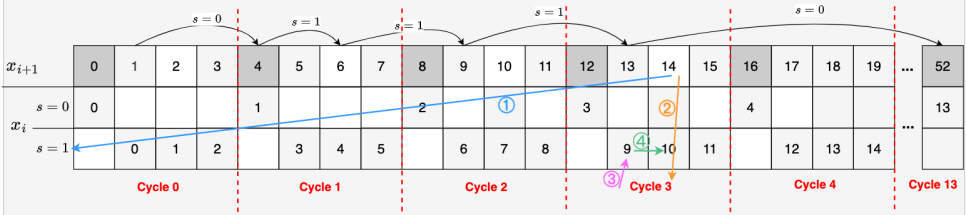
$$x_{i+1} = m \cdot \lfloor x_i / F_s \rfloor + \mathrm{mod}(x_i, F_s) + c_s$$

```python
def rANS_encoder(symbol_counts, s_input):
    # compute cumulative frequencies
    cumul_counts = []
    sum_counts = 0
    for count in symbol_counts:
        cumul_counts.append(sum_counts)
        sum_counts += count
    state = 0
    output = []
    for s in s_input:
        Fs = symbol_counts[s]
        Cs = cumul_counts[s]
        state = (state // Fs) * sum_counts + Cs + (state % Fs)
        output.append((s, state))

    L_avg = math.ceil(math.log(state) / math.log(2.0)) / len(s_input)
    return output, state, L_avg, entropy(symbol_counts)
```

Implemenation in Dash app with interactive visualization

$x_{i+1} \approx x_i / Pr(s)$ $\boxed{Pr(0) = 1/4, Pr(1) = 3/4}$



Given current state x_i. How to find previous state x_{i-1}, where x_i is the x_{i-1}-th appearance of symbol s ?

STEPS / EXAMPLE	$Pr(0) = 1/4, Pr(1) = 3/4$		General
	$s = 0$	$s = 1$	
Step ① : Find the symbol s for the state x_i	if $x_i \bmod 4 = 0$	otherwise or if $x_i \bmod 4 = 1, 2, 3$	s such that $c_s \leq \mathrm{mod}(x_i, m) < c_{s+1}$
Step ②: Find the cycle k of the current state x_i	$k = x_i/4$	$k = \lfloor x_I/4 \rfloor$	$k = \lfloor x_i/m \rfloor$
Step ③: Find the first number of that symbol s in that cycle k	$x_i/4 \cdot 1$	$\lfloor x_i/4 \rfloor \cdot 3$	$\lfloor x_i/m \rfloor \cdot F_s$
Step ④: Allocate to the exact position	$x_{i-1} = x_i/4 \cdot 1 + 0 - 0$	$x_{i-1} = \lfloor x_i/4 \rfloor \cdot 3 + 2 - 1$	$x_{i+1} = \lfloor x_i/m \rfloor \cdot F_s + \mathrm{mod}(x_i, F_s) - c_s$

1. s such that $c_s \leq \mathrm{mod}(x_1, m) < c_{s+1}$
2. $x_{i-1} = D(x_i) = \lfloor x_i/m \rfloor \cdot F_s + \mathrm{mod}(x_i, F_s) - c_s$

Fig.9

Decoding Function [2] [5]:

- s

$$s \text{ such that } c_s \leq \text{mod}(x_i, m) < c_{s+1}$$

- x_{i-1}

$$x_{i-1} = D(x_i) = \lfloor x_i / m \rfloor \cdot F_s + \text{mod}(x_i, F_s) - c_s$$

```python
def rANS_decoder(symbol_counts, num_symbols, state):
    cumul_counts = [0]  # Start with 0 as the first cumulative frequency
    total_frequency = sum(symbol_counts)
    for count in symbol_counts:
        cumul_counts.append(cumul_counts[-1] + count)
    def c_inv(y):
        return bisect_right(cumul_counts, y) - 1
    output = []  # List to store decoded symbols
    for _ in range(num_symbols):
        slot = state % total_frequency  # Find the slot in the cumulative frequency range
        s = c_inv(slot)                 # Get the symbol corresponding to the slot
        Fs = symbol_counts[s]           # Frequency of the symbol
        Cs = cumul_counts[s]            # Cumulative count of the symbol
        output.append((s,state))                # Append decoded symbol to the output
        state = (state // total_frequency) * Fs + (slot - Cs)  # Update the state
    return output, state
```

Implementation in Dash app with interactive visualization

Streaming-rANS

Streaming-rANS

To avoid using large number arithmetic, in practice stream variants are used: which enforce by renormalization: sending the least significant bits of x to or from the bitstream (usually L and b are powers of 2). [4]

Example of the problem //TODO: add example

- ANS (Asymmetric Numeral Systems) uses a single integer x, known as the "state," to represent the encoded data.
- Over time, as more symbols are encoded, the state x can grow arbitrarily large. This can lead to:
  - Overflow: If the value exceeds the precision limits of your system (e.g., a 32-bit or 64-bit register).
  - Inefficient Encoding: A larger x means you need more memory or bits to represent it.
- Using the Cfunction from the previous section to encode succeeding symbols, we can encode a sequence into a very large natural number x, which can be stored as a length $\lfloor lg(x) \rfloor + 1$ bit sequence. [2]

$[L, H]$

- How to determine L and H?

$[L, H]$

- How to determine L and H?
-

$[L, H]$

- How to determine L and H?
-
-

Method 1: ANS encoding step of symbol sfrom state x [2]

while $x > maxX[s]$ **do** {fff}

  writeDigit(mod(x,b)); $x = \lfloor x/b \rfloor$ {fff}

end while

$x = C(s, x)$ {fff}

Method 2: ANS decoding step from state x [2]

$(s, x) = D(x)$ {fff}

useSymbol(s) {fff}

while $x < L$ **do** {fff}

  $x = b \cdot x +$ readDigit() {fff}

end while

- Base decoding:
- $x_i = 2 * x_{i-1} + bit$
- Either x_i in $[L, H]$ or x_{i-1} in $[L, H]$. Assume x_{i-1} in $[L, H]$.
  - $x_{i-1} >= L$
  - $x_i >= 2 * L + bit >= 2 * L$
  - x_i can't be in $[L, H]$
  - derive: $H <= 2 * L - 1$
-

Streaming-rANS | choose L and H | Constraint 2: Encoding

```
    for i in range(10):
        print("Hello, World!")
```

Summary

Summary

- To summarize, this may help you to make a Leipzig university beamer presentation.

Reference

Reference

1. C. Shannon, “A mathematical theory of communication,” The Bell Sys- tem Technical Journal, vol. 27, pp. 379–423, 623–656, July, October
2. J. Duda, K. Tahboub, N. J. Gadgil and E. J. Delp, ”The use of asymmetric numeral systems as an accurate replacement for Huffman coding,” 2015 Picture Coding Symposium (PCS), Cairns, QLD, Australia, 2015, pp. 65-69, doi: 10.1109/PCS.2015.7170048.
3. J. Duda, “Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding,” arXiv:1311.2540.
4. Asymmetric numeral systems - Wikipedia
5. What is Asymmetric Numeral Systems? - Kedar Tatwawadi 1948.