Advances in Mathematics: Scientific Journal **9** (2020), no.11, 9055–9062
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.11.10

A STUDY ON DECIPHERING A TAG AND GENERATING BINARY CODE IN ARITHMETIC CODING

S. S. ARCHA[1], P. A. AZEEM HAFIZ, AND K. S. ZEENATH

ABSTRACT. In this paper we attempt to introduce the basic ideas behind arithmetic coding.For distinguishing a sequence of symbols from another sequence of symbols it is enough to tag it with a unique identifier. In arihmetic coding, a tag is generated for the sequence which is to be encoded and a binary fraction corresponds to the tag becomes the binary code.

Here we generated tag for sequence of symbols and find the binary code for the corresponding tags by considering some examples and we see that it is more efficient for generating code words for sequence of symbols and it is uniquely decodable code.

1. INTRODUCTION

One of the most powerful compression techniques is arithmetic coding. It is used in a variety of lossless and lossy compression applications. It is a form of entropy encoding used in lossless data compression. It replaces a stream of input symbols with single floating point number. The main aim of arithmetic coding is to assign an interval to each symbol. Then each interval is assigned a decimal number.

Several papers appeared in arithmetic coding providing the algorithms of practical arithmetic coding.The most well known among them is the paper by Rissanen and Langdone [4].

[1] *corresponding author*

2020 *Mathematics Subject Classification.* 11T71.

Key words and phrases. arithmetic coding, tag, deciphering, binary code .

Definition 1.1. *The cumulative distribution function (cdf) [3] of the random variable associated with the source is a function that maps random variables and sequence of random variables into the unit interval.*

Definition 1.2. *If $A = \{a_1, a_2, \dots a_n\}$ is the alphabet for a discrete source and x is a random variable. We can define the mapping as $x(a_i) = i$, where $a_i \in A$.*

Definition 1.3. *If $A = \{a_1, a_2, \dots a_m\}$ is the alphabet for a discrete source and x is a random variable. The cumulative density function can be defined as[3]*

$$F_x(i) = \sum_{k=1}^{i} p(x = k), \tag{1.1}$$

where $p(x_i) = a_i$ and $a_i \in A$.

Definition 1.4. *Let$A = \{a_1, a_2, \dots a_m\}$ and if the symbol a_i maps to the real number i. We can define*

$$\widetilde{T}_x = \sum_{k=1}^{i-1} p(x = k) + \frac{1}{2}p(x = i) = F_x(i-1) + \frac{1}{2}p(x = i). \tag{1.2}$$

The unique value $\widetilde{T}_x$ for each symbol a_i is called the tag [1] for a_i.

Note: For the case of longer sequence we can extend as

$$\widetilde{T}_x^m = \sum_{y < x_i} p(y) + \frac{1}{2}p(x_i), \tag{1.3}$$

where $y < x_i$ means y precedes x_i.

2. Generating Tag for a Sequence

The tag generation process works by reducing the size of the interval in which the tag lies. The procedure starts by dividing the unit interval into subintervals.

Definition 2.1. *For any sequence $x = (x_1, x_2, \dots x_n)$,*

$$l^n = l^{n-1} + u^{n-1} - l^{n-1}F_x(x_{n-1}) \tag{2.1}$$

$$u^n = l^{n-1} + (u^{n-1} - l^{n-1}F_x(x_n). \tag{2.2}$$

The tag [3],

$$\widetilde{T}_X(x) = \frac{l^n + u^n}{2}. \tag{2.3}$$

Example 1. *Consider sequence 1231 with probability as given as follows*

x_i	$p(x_i)$
1	*0.8*
2	*0.02*
3	*0.18*

The cdf can be written as

$$F_x(1) = 0.8$$
$$F_x(2) = 0.82$$
$$F_x(3) = 1$$

and

$$F_x(k) = 1, k \geq 3$$

We can initialize as $l^0 = 0, u^0 = 1.$

For the first element 1,

$$\begin{aligned} l^1 &= l^0 + (u^0 - l^0)F_x(1-1) \\ &= 0 + (1-0)0 \\ &= 0 \; u^1 = l^0 + (u^0 - l^0)F_x(1) \\ &= 0 + (1-0)0.8 = 0.8. \end{aligned}$$

The tag contained in the interval (0,0.8). For the second element 2:

$$\begin{aligned} l^2 &= l^1 + (u^1 - l^1)F_x(2-1) \\ &= 0 + (0.8-0)0.8 = 0.64 \\ u^2 &= l^1 + (u^1 - l^1)F_x(2) \\ &= 0 + (0.8-0)0.82 = 0.656. \end{aligned}$$

Tag lies in the interval (0.64,0.656). For the third element 3:

$$\begin{aligned} l^3 &= l^2 + (u^2 - l^2)F_x(3-1) \\ &= 0.64 + (0.656 - 0.64)0.82 = 0.65312 \\ u^3 &= l^2 + (u^2 - l^2)F_x(3) \\ &= 0.64 + (0.656 - 0)0.82 = 0.656. \end{aligned}$$

Tag lies in the interval (0.65312,0.656). For the element 1:

$$\begin{aligned} l^4 &= l^3 + (u^3 - l^3)F_x(1-1) \\ &= 0.65312 + (0.656 - 0.65312)0 = 0.65312 \\ u^4 &= l^3 + (u^3 - l^3)F_x(1) \\ &= 0.65312 + (0.656 - 0.65312)1 = 0.656. \end{aligned}$$

Hence the required $tag = \dfrac{0.65312 + 0.656}{2} = 0.65456.$

3. Deciphering the Tag

By deciphering the tag we can decode a sequence

Definition 3.1. *The process of converting a text written in code or coded signal into normal language is called deciphering.*

3.1. Algorithm for decipher a tag.

(1) Initialize $l^0 = 0$ and $u^0 = 1$.

(2) For each k, find $t^* = \dfrac{tag - l^(k-1)}{u^(k-1) - l^(k-1)}$.

(3) Find the value of x_k for $F_x(x_{k-1}) \leq t^* \leq F_x(x_k)$.

(4) Update u^k and l^k.

(5) Go to step 2 and continue until the entire sequence has been decoded.

Example 2. *Given a sequence {1, 2, 3} with* $p(1) = 0.8, p(2) = 0.02, p(3) = 0.18$ *If the tag of the sequence ,* $\widetilde{T}_x(x) = 0.772352$ *. If the length of the sequence is given as 4, identify the sequence* x*:*

$$F_x(0) = 0, F_x(1) = 0.8, F_x(2) = 0.82, F_x(3) = 1, F_x(k) = 1 \, \textit{for} \ k \geq 3.$$

Set $l^0 = 0$ *and* $u^0 = 1$*. Then:*

$$t^* = \frac{tag - l^(k-1)}{u^(k-1) - l^(k-1)} = \frac{0.772352 - 0}{1 - 0} = 0.772352$$
$$F_x(0) = 0 \leq t^* \leq 0.8 = F_x(1) \longrightarrow 1$$
$$l^1 = l^0 + (u^0 - l^0)F_x(1-1) = 0 + (1-0)0 = 0$$
$$u^1 = l^0 + (u^0 - l^0)F_x(1) = 0 + (1-0)0.8 = 0.8.$$

Now

$$t^* = \frac{0.772352 - 0}{0.8 - 0} = 0.96544$$
$$F_x(2) = 0.82 \leq t^* \leq 1 = F_x(3) \longrightarrow 13$$
$$l^2 = l^1 + (u^1 - l^1)F_x(2) = 0 + (0.8 - 0)0.82 = 0.656$$
$$u^2 = l^1 + (u^1 - l^1)F_x(3) = 0 + (0.8 - 0)1 = 0.8$$
$$t^* = \frac{0.772352 - 0.656}{0.8 - 0.656} = 0.808$$
$$F_x(1) = 0.8 \leq t^* \leq 0.82 = F_x(2) \longrightarrow 132$$
$$l^3 = l^2 + (u^2 - l^2)F_x(1) = 0.656 + (0.8 - 0.656)0.8 = 0.7712$$
$$u^3 = l^2 + (u^2 - l^2)F_x(2) = 0.656 + (0.8 - 0.656)0.82 = 0.77408$$
$$t^* = \frac{0.772352 - 0.7712}{0.77408 - 0.7712} = 0.4$$
$$F_x(0) = 0 \leq t^* \leq 0.8 = F_x(1) \longrightarrow 1321.$$

Since it is given that the length of the sequence is 4, we can stop the process and the required sequence is 1321.

Example 3. *A table for the probability model is given as follows*

letter	*probability*
a_1	*0.2*
a_2	*0.3*
a_3	*0.5*

Decode a sequence of length 3 with the tag 0.63215699, given that $p(a_1) = 0.2, p(a_2) = 0.3$ *and* $p(a_3) = 0.5$. *Hence we have,*

$F_x(a_i) = 0$ *for* $i = 0$

$F_x(a_1) = 0.2, F_x(a_2) = 0.5$ *and* $F_x(a_3) = 1$.

We can set $l^0 = 0$ *and* $u^0 = 1$. *Now*

$t^* = \dfrac{0.63215699 - 0}{1 - 0} = 0.63215699$

$F_x(a_2) = 0.5 \leq 0.63215699 \leq 1 = F_x(a_3) \longrightarrow a_3$

$l^1 = l^0 + (u^0 - l^0)F_x(a_2)$

$= 0 + (1 - 0)0.5 = 0.5$

$u^1 = l^0 + (u^0 - l^0)F_x(a_3)$

$= 0 + (1 - 0)1 = 1$

Now $t^* = \dfrac{0.63215699 - 0.5}{1 - 0.5} = 0.2643$

$F_x(a_1) = 0.2 \leq 0.2643 \leq 0.5 = F_x(a_2) \longrightarrow a_3a_2$

$l^2 = l^1 + (u^1 - l^1)F_x(a_1)$

$= 0.5 + (1 - 0.5)0.2 = 0.6$

$u^2 = l^1 + (u^1 - l^1)F_x(a_2)$

$= 0.5 + (1 - 0.5)0.5 = 0.75$

Now $t^* = \dfrac{0.63215699 - 0.6}{0.75 - 0.6} = 0.21438$

$F_x(a_1) = 0.2 \leq 0.21438 \leq 0.5 = F_x(a_2) \longrightarrow a_3a_2a_2$

Hence a_3, a_2, a_2 *is the required sequence.*

4. Generating a Binary code

A binary code for the tag $\widetilde{T}_X(x)$[6] can be obtained by taking the binary representation of this number and truncating it to the length $l(x) = \lceil log(\frac{1}{p(x)})\rceil + 1$ bits.

Example 4. *Consider a source A that generates letters from an alphabet of size four,* $A = a_1, a_2, a_3, a_4$ *with probabilities* $p(a_1) = 1/2$, $p(a_2) = 1/4$, $p(a_3) = 1/8$, $p(a_4) = 1/8$.

For the symbol a_1, $F_x(a_1) = \frac{1}{2}$ *and* $\widetilde{T}_x(a_1) = 0 + \frac{1}{2}(0.5) = 0.25$. *In binary form,*

$0.25 \times 2 = 0.5 \longrightarrow 0$

$0.5 \times 2 = 1.0 \longrightarrow 1.$

Hence the binary form of a_1 *is 0.01. Length of the code is* $\lceil log(\frac{1}{p(x)})\rceil + 1 = \lceil log(2^1)\rceil + 1$*=1+1=2.*

For the symbol a_2, $F_x(a_2) = 0.5 + 0.25 = 0.75$ *and* $\widetilde{T}_x(a_2) = 0 + \frac{1}{2} = 0.625$. *In binary form,*

$0.625 \times 2 = 1.25 \longrightarrow 1$

$1.25 \times 2 = 2.50 \longrightarrow 0$

$2.5 \times 2 = 5 \longrightarrow 1.$

Therefore the binary form of a_2 *is 0.101. Length of the code is* $\lceil log\frac{1}{p(x)}\rceil + 1 = \lceil log(2^2)\rceil + 1 = 2 + 1 = 3.$

For the symbol a_3, $F_x(a_3) = 0.75 + 0.125 = 0.8125$ *and* $\widetilde{T}_x(a_3) = 0.75 + \frac{1}{2}(0.125) = 0.8125$. *In binary form,*

$0.8125 \times 2 = 1.625 \longrightarrow 1$

$1.625 \times 2 = 3.25 \longrightarrow 1$

$3.25 \times 2 = 6.5 \longrightarrow 0$

$6.5 \times 2 = 13 \longrightarrow 1.$

So the binary form of a_3 *is 0.1101. Length of the code is* $\lceil log\frac{1}{p(x)}\rceil + 1 = \lceil log(2^3)\rceil + 1 = 3 + 1 = 4.$

For the symbol a_4, $F_x(a_4) = 0.875 + 0.125 = 1$ *and* $\widetilde{T}_x(a_4) = 0.875 + \frac{1}{2}(0.125) = 0.9375$. *In binary form,*

$0.9375 \times 2 = 1.875 \longrightarrow 1$

$1.875 \times 2 = 3.75 \longrightarrow 1$

$3.75 \times 2 = 7.5 \longrightarrow 1$

$7.5 \times 2 = 15 \longrightarrow 1.$

Hence the binary form of a_4 *is 0.1111. Length of the code is* $\lceil log\frac{1}{p(x)}\rceil + 1 = \lceil log(2^3)\rceil + 1 = 3 + 1 = 4.$

We can tabulate the data as:

TABLE 1. Binary code

Symbol	F_x	$\widetilde{T}_x$	*Binary form*	$\lceil log\frac{1}{p(x)}\rceil + 1$	*code*
a_1	*0.5*	*0.25*	*0.01*	*2*	*01*
a_2	*0.75*	*0.625*	*0.101*	*3*	*101*
a_3	*0.875*	*0.8125*	*0.1101*	*4*	*1101*
a_4	*1.0*	*0.9375*	*0.1111*	*4*	*1111*

5. CONCLUSION

We have shown that in arithmetic coding it is possible to encode sequences directly. It is a part of many international standards. Through the example we see that deciphering the tag is as simple as generating it. Also we can decipher a tag with minimum computational cost. By generating the binary code it is clear that the code is unique for a sequence.

REFERENCES

[1] I. MENGYI PU: *Fundamental Data Compression* , Elsevier, USA, 2006.

[2] G. G. LANGDON: *An Introduction to Arithmetic coding*, IBM Journal of Research and Development, **28**(2) (1984), 135-149,

[3] K. SAYOO: *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Fransisco, 2006.

[4] J. RISSANEN, G. G. LANGDON: *Arithmetic Coding* , IBM Journal of Research and Development, **23**(2) (1979), 149-162.

[5] S. L. CHAOPINGXING: *Coding Theory A First Course*, Cambridge University Press, 2004.

[6] I. H. WITTEN, R. M. NEAL, J. G. CLEARY : *Arithmetic Coding for Data Compression*, Communications of the ACM, **30**(6) (1987), 520-540.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF KERALA, INDIA
Email address: archakmr@gmail.com

PRINCE SATTAM BIN ABDELAZIZ UNIVERSITY
AL KHARJ, RIYADH, SAUDI ARABIA
Email address: azeem parayil@gmail.com

SDE, UNIVERSITY OF KERALA, INDIA
Email address: zeenath.ajmal@gmail.com