International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

# COMPARATIVE STUDY OF ARITHMETIC AND ADAPTIVE ARITHMETIC USING DATA COMPRESSION

**T. Annie Nisha[1], V. Pavithra[2]**

[1]*PG Scholar-Communication Systems, Department of Electronics and Communication Engineering, Government College of Engineering, Salem-11, Tamil Nadu, India*
[2] *PG Scholar-Communication Systems, Department of Electronics and Communication Engineering, Government College of Engineering, Salem-11, Tamil Nadu, India*

---***---

**Abstract -** *Data compression has to tend almost attention is text quality assessment. For this purpose different methodologies are defined. It is important to settle on the most effective machine learning algorithm. Additionally to different compression methodologies, it's also important to pick out the great data compression tool. Data compression plays a crucial role within the area of information transmission and also the data storage. Data compression is employed for compressing or reducing the info. Thanks the compression we are able to reduce the storage, it consumes less time, transmission are going to be in no time and more compression ratio is provided. Data compression is of lossless compression and also the lossy compression. The lossless compression has a number of the coding techniques. During this paper we compare and study about the various coding techniques.*

***Key Words***: **Data compression, Compression tool, Compression ratio, Lossless compression, Lossy compression.**

## 1. INTRODUCTION

Data compression is that the reduction in number of bits needed to represent the info. It reduces the cupboard space. Data compression has two components namely encoding algorithm and therefore the decoding algorithm. Encoding (compression) may be a process of remodeling the initial data into a compressed data. Decoding (decompression) may be a process of recovering the initial data from the compressed data. Data compression has two types: Lossless compression and Lossy compression.

## 2. TYPES OF COMPRESSION

### 2.1 Lossless Compression

In lossless compression there's no loss of the data. The initial data will be recovered properly within the compressed data without the loss of knowledge. Lossless compression is mostly used for applications that can't tolerate any difference between the initial and reconstructed data. It's very important that the reconstruction is a dead ringer for the text original, as very small difference may end up in statements with very different meanings. Consider the instance for the lossless compression "Do not send money" and "Do not send money". There are many situations during which it's possible to relax this requirement so as to urge more compression. In these situations we glance to lossy compression techniques.

### 2.2 Lossy Compression

In the lossy compression there's a loss of the knowledge. The first signal can't be recovered properly. In many applications, this lack of tangible reconstruction isn't a controversy. For instance, when storing or transmitting speech, the precise value of every sample of speech isn't necessary. Counting on the standard required of the reconstructed speech, varying amounts of loss of data about the worth of every sample will be tolerated. Consider instance for the lossy compression"Do not send money" and"Do now send money". Once we've developed a knowledge compression scheme, we want to be able to measure its performance.

## 3. MEASURES OF PERFORMANCE

A compression algorithm will be evaluated in an exceedingly number of various ways.

i. Relative complexity of the algorithm.
ii. The memory required to implement the algorithm.
iii. How briskly the algorithm performs on a given machine.
iv. The quantity of compression, and the way closely the reconstruction resembles the first.

## 4. COMPRESSION RATIO

Data compression ratio may be a measurement of relative reduction in size of knowledge representation that was produced by a knowledge compression algorithm. Data compression is additionally referred to as compression power. The ratio of the quantity of bits required to represent the info before compression to the quantity of bits required to represent the info after the compression. This ratio is named because the compression ratio.

## 5. COMPRESSION TECHNIQUES

### 5.1 ARITHMETIC CODING

Arithmetic coding could be an entropy encoding technique employed in the lossless data compression. Where the entropy is the measure of unpredictability. Arithmetic coding came from the Shannon observation; the symbols will be coded by their cumulative property. Employing a fixed number of bits per character a string of characters will be represented. Few bits are stored for frequency used characters and also the more bits are stored for not-so-frequently occurring characters. Within arithmetic coding the complete message is encoded into one number by a fraction n where (0.0≤n≤1.0).

### Algorithm

**Step 1:** Divide the amount range 0 to 1 into number of various symbols present within the image. Allot the intervals based the probability of the symbols.
**Step 2**: Expand the primary letter to be coded.
**Step 3**: Find the difference.

D = Upper limit – Lower limit

**Step 4**: Find the range of the each symbol.

Range = lower limit: Lower limit + difference(probability)

**Step 5**: Continue this procedure until the last character is encoded.

**Step 6**: Find the tag value.

$$T_x(X)=\frac{Upperlimit-Lowerlimit}{2}$$

### 5.2 ADAPTIVE ARITHMETIC CODING

One of the advantages of adaptive arithmetic coding is adaption. After coding is initiated the probability value will be incremented according to the previous probability value. The difference value will be calculated by using upper limit, lower limit, and increased probability value. Then the tag value will be calculated, by using these values the decoding process will be done.

### Algorithm

**Step 1:** Divide the amount range 0 to 1 into number of various symbols present within the image. Allot the intervals based the probability of the symbols.
**Step 2**: Expand the primary letter to be coded.
**Step 3**: Increase the probability value for each iteration
**Step 4**: Find the difference.

D = Upper limit – Lower limit/N

N ⟶ Increased probability value

**Step 5:** Find the range of the each symbol.

Range=D * increased probability value +Lower limit

**Step 6**: Find the tag value.

$$T_x(X)=\frac{Upperlimit-Lowerlimit}{2}$$

### 5.3 COMPARISON OF COMPRESSION TECHNIQUES

**Table -1:** Comparison of compression techniques

| PARAMETER | ARITHMETIC CODING | ADAPTIVE ARITHMETIC CODING |
|---|---|---|
| Objective | The purpose of the modeling for statistical facts compression to offer information to the code. | Updates the chances of data's from the last occurrence. |
| Compression type | Lossless | Lossless |
| Range Equation | Range = lower limit : Lower limit + difference (probability) | Range=D * increased probability value + Lower limit |
| Algorithm | Encoding and Decoding | Encoding and Decoding |
| Features | Entropy based | Entropy based |
| Advantages | Computationally efficient | Used probability distribution can be changed |
| Disadvantages | Poor error resistance | More vulnerable to corruption. |

## 6. RESULTS

**Table – 2:** Results for arithmetic coding

| Message | Size | | Tag Value | Iteration | |
|---|---|---|---|---|---|
| | Total symbols | Unique Symbols | | Encoding | Decoding |
| **GCESECE** | 7 | 4 | 0.054276673 | 7 | 7 |
| **COMMUNICATION SYSTEMS** | 21 | 12 | 0.754356288 | 21 | 21 |

## 7. CONCLUSIONS

Compression are reductions in storage hardware, data coordinated universal time, and communication bandwidth. This can lead to cost savings. Most of the info compression techniques achieves high data compression ratio. Within the data compression lossless technique numbers of the codings are used. Hence the codings are compared with the various parameters.

## REFERENCES

[1] Chen, Qingchun, Wai Ho Mow, and Pingzhi Fan. "Some new results on recursive convolutional codes and their applications." Information Theory Workshop, 2006. ITW'06 Chengdu. IEEE. IEEE, 2006.

[2] Francis, Michael. "Viterbi Decoder Block Decoding-Trellis Termination and Tail Biting." Xilinx XAPP551 v2. 0, DD (2005): 1-21.

[3] Khalid Sayood, Introduction to Data Compression. Elsevier, 2012.

[4] MacKay, David. J. C (September 2003). "Chapter 6: Stream Codes" Information Theory, Inference, and Learning Algorithms. Cambridge University Press. Archived on 22 December 2007. Retrieved 30 December 2007

[5] Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 22.6 The Arithmetic Coding "Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press.

[6] Rissanen Jorma (May 1976). "Generalized Kraft and Generate Coding". IBM Journal of Research and Development. **20** (3): 198–203. Retrieved 21 September 2007.