OPEN ACCESS

Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Review Article

Mathematical Preliminaries in the Case of Lossless Compression Markov Models

Tran Dang Hung[1, 2*], Jan Platoš[2]

[1]Faculty of Applied Science, Ho Chi Minh City University of Food Industry,140 Le Trong Tan Street, Tay Thanh Ward, Tan Phu District, Ho Chi Minh City 70000, Vietnam
[2]Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 17.Listopadu 15/2172, 70833 Ostrava, Czech Republic

DOI: 10.36348/sjet.2023.v08i05.003 | **Received:** 16.04.2023 | **Accepted:** 20.05.2023 | **Published:** 26.05.2023

***Corresponding author:** Tran Dang Hung
Faculty of Applied Science, Ho Chi Minh City University of Food Industry,140 Le Trong Tan Street, Tay Thanh Ward, Tan Phu District, Ho Chi Minh City 70000, Vietnam

Abstract

Compression schemes can be divided into two categories, lossy and lossless, but this paper presents lossless data compression models and the original data can be correctly recovered from the data compressed material. Some mathematical results are assumed; the results of probability tests are assumed and used to evaluate the compression techniques we will discuss. To learn more about math concepts for some of the topics in this article, see [2, 3]. First, we look at several ideas in information theory that provide a standard for the development of lossless data compression schemes are briefly reviewed. We next look at several ways to model data that lead to efficient data compression encryption schemes.

Keywords: Compression, Lossless Compression, entropy, Markov Models.

I. INTRODUCTION

Compression technique or compression algorithm is included two algorithms. The compression algorithm takes an input X *(original data)* and generates a representation Y *(compressed data)* that requires fewer bits. The reconstruction algorithm operates on the compressed representation Y to generate the reconstruction Z. These operations are shown schematically in Figure 1. We by convention refer to both compression and reconstruction algorithms together to mean compression algorithm.

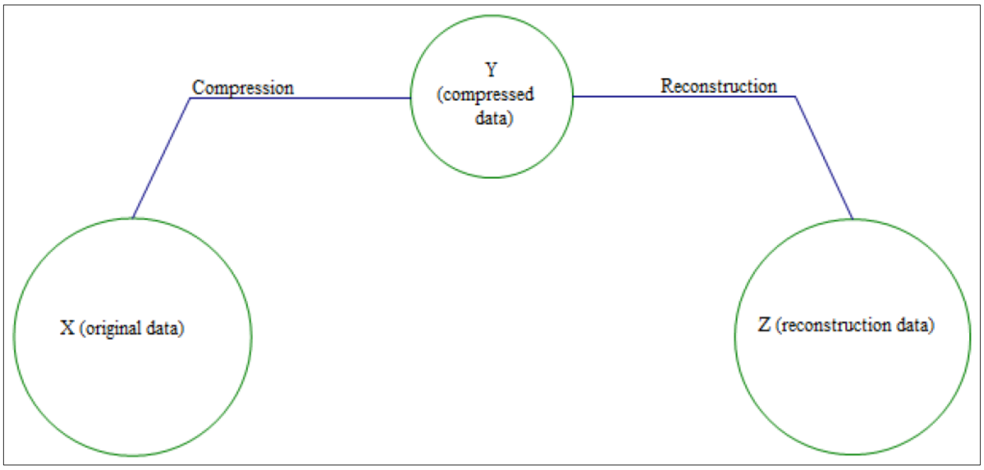

Figure 1: Compression and reconstruction

Data compression schemes can be divided into two schemes: lossless compression schemes, in which Z is the same to X, and lossy compression schemes, which generally provide much higher compression than lossless compression but allow Z to be different from X.

Lossless Compression

Lossless compression techniques involve no loss of information. Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data (Text compression is an example for lossless compression). Data have been losslessly compressed; the original data can be recovered exactly from the compressed data.

Lossy Compression

Lossy compression techniques involve some loss of information. Data have been compressed using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction, we can generally obtain much higher compression ratios than is possible with lossless compression. For example, when storing or transmitting speech, the exact value of each sample of speech is not necessary.

II. INFORMATION THEORY

Claude Elwood Shannon, an American mathematician, developed information theory [12]. Shannon defined a quantity called self-information. Suppose we have an event *X*, which is a set of outcomes of some random experiment. If *P(X)* is the probability that event *X* will occur, then the self-information associated with *X* is given by

$$i(X)=\log_b \frac{1}{P(X)}=-\log_b P(X). \qquad (1)$$

The unit of information depends on the base of the log. If we use log base 10, the unit is hartleys; if we use log base e, the unit is nats; and we use log base 2, the unit is bits. Using the logarithm to obtain a measure of information was not an arbitrary choice. First, let's see if the use of a logarithm in this context makes sense from an intuitive point of view. The fact that −log(x) increases as x decreases from one to zero. In another word, *i(X)* increases as *P(X)* decreases from one to zero. It means that, if the probability of an event is high, the information associated with it is low; if the probability of an event is low, the amount of self-information associated with it is high.

Another property of this mathematical definition of information is that the information obtained from the occurrence of two independent events is the sum of the information obtained from the occurrence of the individual events. Suppose *X* and *Y* are two independent events. The self-information associated with the occurrence of both event *X* and event *Y* is, by Equation (1),

$$\begin{aligned} i(XY) &= -\log_b P(XY) \\ &= -\log_b \left[P(X)P(Y)\right] \\ &= -\log_b P(X)-\log_b P(Y) \\ &= \ i(X)+i(Y). \end{aligned}$$

Definition 2.1. If *S* is the sample space of experiment $\mathcal{S}$ that we have a set of independent events X_i, which are sets of outcomes of some experiment $\mathcal{S}$, such that $\bigcup X_i = S$

then the average self-information associated with the random experiment is given by

$$H=\sum P(X_i)i(X_i)=-\sum P(X_i)\log_b P(X_i)$$

The quantity *H* is called the entropy associated with the experiment. If the experiment is a source that puts out symbols X_i from a set $\mathcal{X}$, then the entropy is a measure of the average number of binary symbols needed to code the output of the source. Then, the best that a lossless compression scheme can do is to encode the output of a source with an average number of bits equal to the entropy of the source.

The set of symbols $\mathcal{X}$ is often called the *alphabet* for the source, and the symbols are referred to as *letters*. For a general source $\mathcal{S}$ with alphabet $\mathcal{X}=\{1,2,...,m\}$ that generates a sequence $\{X_1,X_2,...\}$, the entropy is given by

$$H(\mathcal{S})=\lim_{x\to\infty}\frac{1}{n}G_n \qquad (2)$$

Where $G_n=-\sum_{i_1=1}^{i_1=m}\sum_{i_2=1}^{i_2=m}...\sum_{i_n=1}^{i_n=m}P(X_1=i_1,X_2=i_2,...,X_n=i_n)\log P(X_1=i_1,X_2=i_2,...,X_n=i_n)$

and $\{X_1,X_2,...,X_n\}$ is a sequence of length *n* from the source. If each element in the sequence is independent and identically distributed (*iid*), then we can show that

$$G_n=-n\sum_{i_1=1}^{i_1=m}P(X_1=i_1)\log P(X_1=i_1) \qquad (3)$$

and the equation for the entropy becomes

$$H=-\sum P(X_i)\log P(X_i) \qquad (4)$$

For most sources Equations (2) and (4) are not identical. If we need to distinguish between the two, we will call the quantity computed in (4) the *first-order entropy* of the source, while the quantity in (2) will be referred to as the *entropy* of the source.

In general, it is not possible to know the entropy for a physical source, so we have to estimate the entropy. The estimate of the entropy depends on our assumptions about the structure of the source sequence.

Consider the following sequence:
1 2 3 4 5 4 5 6 7 8 7 8 9 10 11 10 11 12 13 14 13 14 15 16 17 16 17 18 19 20 19 20

Assuming the frequency of occurrence of each number is reflected accurately in the number of times it appears in the sequence, we can estimate the probability of occurrence of each symbol as follows:

$$P(1)=P(2)=P(3)=P(6)=P(9)=P(12)=P(15)=P(18)=\frac{1}{32}$$

$$P(4)=P(5)=P(7)=P(8)=P(10)=P(11)=P(13)=P(14)=P(16)=P(17)=P(19)=P(20)=\frac{1}{16}.$$

Assuming the sequence is *iid*, the entropy for this sequence is the same as the first-order entropy as defined in (4). The entropy can then be calculated as

$$H=-\sum_{i=1}^{20}P(i)\log_2 P(i).$$

With our stated assumptions, the entropy for this source is 4.25 bits. This means that the best scheme we could find for coding this sequence could only code it at 4.25 bits/sample.

However, if we assume that there was sample-to-sample correlation between the samples and we remove the correlation by taking differences of neighboring sample values, we arrive at the *residual* sequence
1 1 1 1 -1 1 1 1 1 -1 1 1 1 1 -1 1 1 1 1 -1 1 1 1 1 -1 1 1 1 1 -1 1

This sequence is constructed using only two values with probabilities $P(1)=\frac{13}{16}$ and $P(-1)=\frac{3}{16}$. The entropy in this case is $H=-\sum_{i=1}^{2}P(i)\log_2 P(i)=0.69$ bits per symbol. Of course, knowing only this sequence would not be enough for the receiver to reconstruct the original sequence. The receiver must also know the process by which this sequence was generated from the original sequence. The process depends on our assumptions about the structure of the sequence. These assumptions are called the model for the sequence. In this case, the model for the sequence is

$$x_n=x_{n-1}+r_n$$

Where x_n is the *n*th element of the original sequence and r_n is the *n*th element of the residual sequence. This model is called a static model because its parameters do not change with *n*. A model whose parameters change or adapt with *n* to the changing characteristics of the data is called an adaptive model.

Basically, we see that knowing something about the structure of the data can help to "reduce the entropy." We have put "reduce the entropy" in quotes because the entropy of the source is a measure of the amount of information generated by the source. As long as the information generated by the source is preserved (in whatever representation), the entropy remains the same. What we are reducing is our estimate of the entropy. The "actual" structure of the data in practice is generally unknowable, but anything we can learn about the data can help us to estimate the actual source entropy. Theoretically, as seen in Equation (2), we accomplish this in our definition of the entropy by picking larger and larger blocks of data to calculate the probability over, letting the size of the block go to infinity.

III. MARKOV MODEL

One of the most popular ways of representing dependence in the data is through the use of Markov models, named after the Russian mathematician Andrey Andreyevich Markov (1856 - 1922). For models used in lossless compression, we use a specific type of Markov process called a discrete time Markov chain.

Definition 3.1. Let $\{x_n\}$ be a sequence of observations. This sequence is said to follow a *k*th-order Markov model if

$$P(x_n|x_{n-1},\dots,x_{n-k})=P(x_n|x_{n-1},\dots,x_{n-k},\dots)\qquad(5)$$

In other words, knowledge of the past *k* symbols is equivalent to the knowledge of the entire past history of the process. The values taken on by the set $\{x_{n-1},\dots,x_{n-k}\}$ are called the states of the process. If the size of the source alphabet is *l*, then the number of states is l^k. The most commonly used Markov model is the first-order Markov model, for which

$$P(x_n | x_{n-1}) = P(x_n | x_{n-1}, x_{n-2}, x_{n-3}, \ldots). \qquad (6)$$

Equations (5) and (6) indicate the existence of dependence between samples. However, they do not describe the form of the dependence. We can develop different first-order Markov models depending on our assumption about the form of the dependence between samples. If we assumed that the dependence was introduced in a linear manner, we could view the data sequence as the output of a linear filter driven by white noise. The output of such a filter can be given by the difference equation

$$x_n = \rho x_{n-1} + \varepsilon_n$$

Where ε_n is a white noise process. This model is often used when developing coding algorithms for speech and images.

The use of the Markov model does not require the assumption of linearity. For example, consider a binary image. The image has only two types of pixels, white pixels and black pixels. We know that the appearance of a white pixel as the next observation depends, to some extent, on whether the current pixel is white or black. Therefore, we can model the pixel process as a discrete time Markov chain. Define two states A_w and A_b (A_w would correspond to the case where the current pixel is a white pixel, and A_b corresponds to the case where the current pixel is a black pixel). We define the transition probabilities $P(w/b)$ and $P(b/w)$, and the probability of being in each state $P(A_w)$ and $P(A_b)$. The Markov model can then be represented by the state diagram shown in Figure 2.

The entropy of a finite state process with states A_i is simply the average value of the entropy at each state:

$$H = \sum_{i=1}^{M} P(A_i) H(A_i) \qquad (7)$$

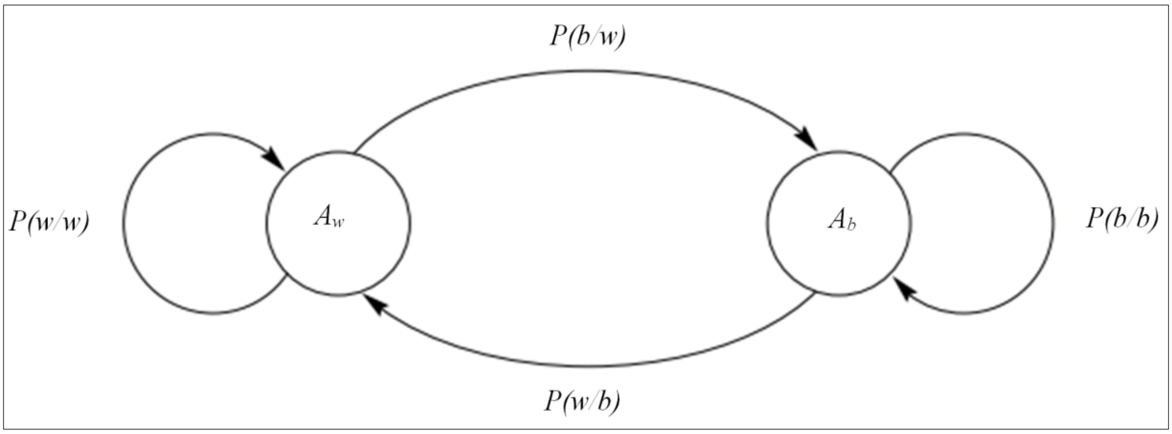

Figure 2: A two-state Markov model for binary images

For our particular example of a binary image

$$H(A_w) = -P(b/w)\log P(b/w) - P(w/w)\log P(w/w)$$

Where $P(w/w) = 1 - P(b/w)$. $H(S_b)$ can be calculated in a similar manner.

$$H(A_b) = -P(w/b)\log P(w/b) - P(b/b)\log P(b/b)$$

To see the effect of modeling on the estimate of entropy, let us calculate the entropy for a binary image. First using a simple probability model and then using the finite state model described above. Let us assume the following values for the various probabilities:

$$P(A_w) = \frac{30}{31}; P(A_b) = \frac{1}{31}$$

$$P(w/w) = 0.99; P(b/w) = 0.01; P(b/b) = 0.7; P(w/b) = 0.3$$

Then the entropy using a probability model and the *iid* assumption is (using Equation (4))

$$\begin{aligned} H &= -\sum P(A_i)\log P(A_i) \\ &= -P(A_w)\log P(A_w) - P(A_b)\log P(A_b) \\ &= -\frac{30}{31}\log\frac{30}{31} - \frac{1}{31}\log\frac{1}{31} = 0.206 \ \ bits \end{aligned}$$

Now using the Markov model

$$H(A_b) = -P(w/b)\log P(w/b) - P(b/b)\log P(b/b)$$

$$= -0.3\log 0.3 - 0.7\log 0.7 = 0.881 \ \ bits$$

and

$$H(A_w) = -P(b/w)\log P(b/w) - P(w/w)\log P(w/w)$$

$$= -0.01\log 0.01 - 0.99l\,og\, 0.99 = 0.081 \ \ bits$$

which, using Equation (7), results in an entropy for the Markov model of:

$$H = \sum P(A_i)H(A_i)$$

$$= P(A_w)H(A_w) + P(A_b)H(A_b)$$

$$= \frac{30}{31}(0.081) + \frac{1}{31}(0.881)$$

$$= 0.107 \ bits$$

about a half of the entropy obtained using the *iid* assumption.

IV. CONCLUSION

In this article, we took a rather brief visit to the basic definitions of information theory based on some mathematical results and some of the ways we can reduce entropy. However, the coverage in this paper will sufficient to take us through some further specializations, such as Coding, Arithmetic Coding, Dictionary Techniques, Context-Based Compression, Lossless Image Compression. The concepts introduced in this paper will allow us to estimate the number of bits we need to represent the output of a source given the probability model for the source, and this will use in the case when we describe different encryption algorithms, modeling.

REFERENCES

1. Waerden, van der B. L. (1985). *A History of Algebra*, Springer-Verlag.
2. Berger, T. (1971). *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Prentice-Hall, Englewood Cliffs, NJ.
3. Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*, Wiley Series in Telecommunications. John Wiley & Sons Inc.
4. Hamming, R. W. (1986). *Coding and Information Theory (2nd edition)*, Prentice-Hall.
5. Ash, R. B. (1990). *Information Theory*. Dover.
6. Gray, R. M. (1990). *Entropy and Information Theory*, Springer-Verlag.
7. Storer, J. A. (1988). *Data Compression - Methods and Theory*, Computer Science Press.
8. Abramson, N. (1963). *Information Theory and Coding*, McGraw-Hill.
9. Jelinek, F. (1968). *Probabilistic Information Theory*, McGraw-Hill.
10. Nelson, M., & Gailly, J.-L. (1996). *The Data Compression Book*. M&T Books, CA.
11. Tate, S. (2003). Complexity Measures. In K. Sayood, editor, 2003, *Lossless Compression Handbook*, pages 35–54. Academic Press.
12. Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal, 27*, 379–423, 623–656.
13. Robinson, T. (1994). *SHORTEN: Simple Lossless and Near-Lossless Waveform Compression*, Cambridge Univ. Eng. Dept., Cambridge, UK. Technical Report 156.
14. Langdon, G. G., & Rissanen, J. J. (1981). Compression of black-white images with arithmetic coding. *IEEE Transactions on Communications, 29*(6), 858–867.
15. Ziv, J., & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory, IT-24*(5), 530–536.