On the random property of compressed data via Huffman coding

Yi-Ping You
Department of Computer Science and Information Engineering
National Chi-Nan University
u5321009@ncnu.edu.tw

Shi-Chun Tsai
Department of Computer Science and Information Engineering
National Chi-Nan University
tsai@csie.ncnu.edu.tw

Abstract

Though Huffman codes [2,3,4,5,9] have shown their power in data compression, there are still some issues that are not noticed. In the present paper, we address the issue on th random property of compressed data via Huffman coding. Randomized computation is the only known method for many notoriously difficult #P-complete problems such as permanent, and some network reliability problems, etc [1,7,8,10]. According to Kolmogorov complexity [6,10], a truly random binary string is very difficult to compress and for any fixed length there exist incompressible strings. In other words, incompressible strings tend to carry higher degree of randomness. We study this phenomenon via Huffman coding. We take compressed data as random source that provides coin flips for randomized algorithms. A simple randomized algorithm is proposed to calculate the value of π with the compressed data as random number. Experimental results show that compressed data via Huffman coding does provide a better approximation for calculating π, especially in the first few round of th compressions. We t ry several different types of files and obtain similar results that compressed data is random for the testing example.

Keyword: data compression, Huffman coding, rando number generator, randomized algorithms, Kolmogorov complexity

1. Introduction

Recentl , researches of Huffman coding [2,4,5,9] have been devoted to the efficiency of data compression techniques, especially for time and space saving during encoding and decoding. Obviously, data after compression is smaller than the original in general. Properties of compressed data and original data would not be the same. We focus on the random property of compressed data via Huffman coding. Cryptanalysis on compressed data via Huffman coding has been studied before [3]. This paper is the first one t address the random property of Huffman codes.

Randomized method is the only known method for man difficult problems, such as #P-complete problems, which can be much more difficult than NP-complete problem [1,7,8,10]. For the prime testing problem, the most efficient method is randomized algorithm.[1,7,8,10]. In randomized algorithms, we need random numbers (or sources) to sample objects. The performance of a randomized algorithm relies heavily on the random source. However, there is no real random source available in computers or real world. So there are all kinds of simulations for pseudo-random number generator. This is one of the major motivations to study randomness for randomized computation.

In early 1960's, Kolmogrov initiated the study of randomness from the computational approach [6]. The so-called Kolmogrov complexity for a string is the shortest program that can produce it. This implies that incompressible strings are more random. It is also known that for any fixed length there are strings that cannot be compressed in the sense of Kolmogrov complexity.

In this paper, motivated by the above background knowledge, we are interested in the random property of compressed data and relationship between the random property and compression iterations. The random property of data is regarded as the degree of entropy order of data. If the distribution of data is quite tumultuous, it is said that the entropy of the data is high. To study the random property of data, we take the compressed data as random number generator. By using a simple randomized algorithm which is proposed to calculate the value of π in this paper, we can justify the quality of randomness from the compressed data by the accuracy of π. The closer it is t the real value of π, the higher the entropy of the data is. To verify the relationship between random properties and compressing iterations, we make several experiments on different types of data, such as pure text, image and zip files.

The experimental results show that there is sig nificant randomness gain on the compressed data, especially in the first few rounds of compressions, and that the entropy order of data tends toward higher after being compressed via Huffman coding. That is, the more iterations the data is compressed, the more tumultuous the distribution of data is.

The remainder of this paper is organized as follows. randomized algorithm for approximating π is proposed in Section 2. Experiments on random properties of compressed data are made in Section 3. We conclude t his paper in Section 4.

2. A testing randomized algorithm

The proposed approach uses a simple randomized algorithm to calculate the approximate value of π. This randomized algorithm is for testing the quality of randomness from compressed data. For self-contained we choose this simple algorithm for easy illustration. The basi idea of this method is to use a random number generator as a parameter. If generator generates numbers randomly on the premise that the numbers are mass enough, the estimated value will be quite accurate to the real value. Suppose that the generator generates points with two-dimensional coordinate, say (x, y), where $0 \leqq x \leqq r$, $0 \leqq y \leqq r$, and x and y are real numbers. That is, points fall inside the gray area in Figure 1. We draw a curve from point $(r, 0)$ to point $(0, r)$, which is a quarter of a circle with center $(0, 0)$ and radius r, and then divide these points into two groups **INSIDE** and **OUTSIDE**. If a point falls on the same side as $(0, 0)$ or just falls on the curve, put the point into INSIDE group; otherwise, put it into OUTSIDE group.

Figure 2 demonstrates that there are 15 points in INSIDE group and 4 points in OUTSIDE group. If the radius of the inscribed circle is r, then its area is $\pi r^2 / 4$, where as that of the square target is r^2, so the average proportion of the points that fall inside the circle is $\pi r^2 / 4r^2 = \pi / 4$. Since points are distributed over the zone randomly, the relationship of the number of points in INSIDE and OUTSIDE group is:

$$\pi \times N(\text{INSIDE} + \text{OUTSIDE}) = 4 \quad N(\text{INSIDE}),$$

where $N(\ )$ denotes the number of points in some group.

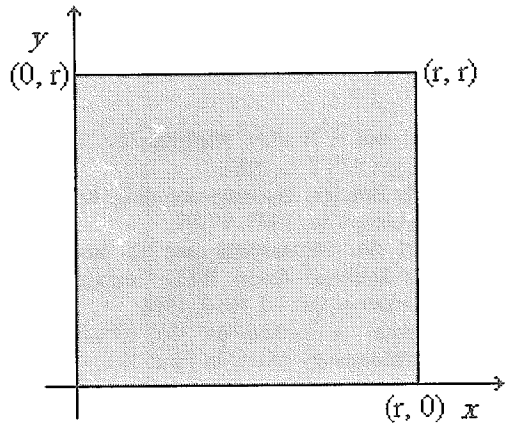

Figure 1. The area where points fall on

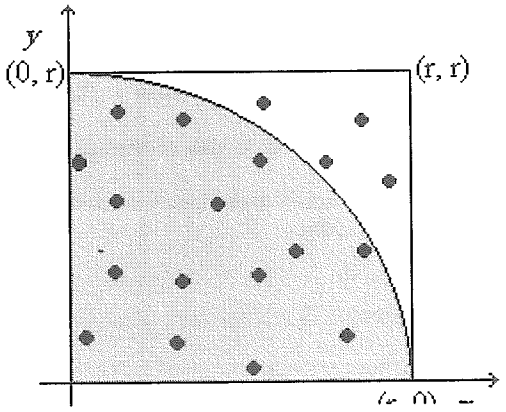

Figure 2. An example of estimate of π with random points. The estimated value of π is $4 \times (15 / 9) = 3.157894736842$

To measure the random properties of data, we take data itself as a random source. A parameter called W is added to decide how many of bits should be fetched each time when generating random numbers. Clearly, the larger the W is, the more random numbers are generated, which means w partition the square into a more refined grid. By comparing the computed value of π, we can measure the random property of compressed data. The detailed procedure for π approximation is shown below.

1. Fetch $2 \times W$ bits from compressed data t generate two numbers between zero and r.
2. Take the two numbers as a coordinate (x, y) on the plane. One is as x-component, and the other is as y-component.
3. If the point falls within the qu arter circle, mark the point with **INSIDE**; otherwise, mark the point with **OUTSIDE**.
4. Repeat step 1 to step 3 until the data is read over.
5. Make statistics of total points.
6. Calculate the value of π from

$$4 \times \frac{\text{The number of INSIDE points}}{\text{The number of total points}}$$

More complicated randomized algorithms can also be used for testing random property of compressed data. However we think this simple randomized algorithm suffices t justify the phenomenon implied by Kolmogorov complexity.

3. Experimental results

Based on the above proposed approach and algorithm, the experiments have been done on a Sun workstation. Data is compressed with Huffman coding repeatedly, which means the compressed data is regarded as source data to be encoded again, until the data is incompressible. In order to examine the quality of the result from large window size W, when fetching bits, we do some experiments with window sizes: 5, 10, 15, and 20. Figure 3 shows the results of experiments on four types of file: text, image, and zip. It shows that there is a tendency that the larger the value o f W is, the more accurate and stable the estimated value of π is.

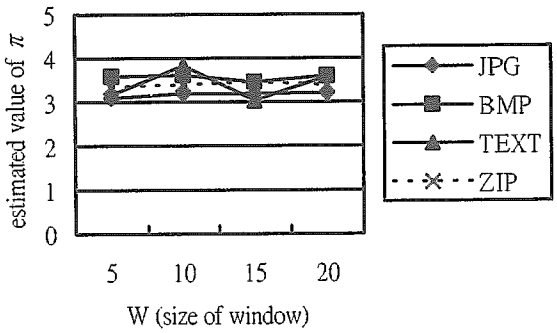

Figure 3. The effect of W on different types of files.

Next, we experiment on data that are compressed k times, where k is a positive integer and $0 \leqq k \leqq 20$, with a fixed value of W =15. As it is mentioned above, data is compressed repeatedly until it is incompressible. Hence compression may stop before k reaches 20. The results are shown in figure 4. From the results, we can see that the estimated value has a tendency toward real value of π. That is to say, data trend to be more random after being compressed several times. Note that in the first f ew iterations we see the most gains on randomness. However, after a while the test with compressed jpg file behaves not as stable as the other cases. The reason is that after a fe iterations, the file could no longer be compressible and then the irregularity happens. After-all, in this note we are interested in the first few compressions, which are suffice to justify the randomness v.s. compressibility, i.e. data after Huffman compression does carry more random property.

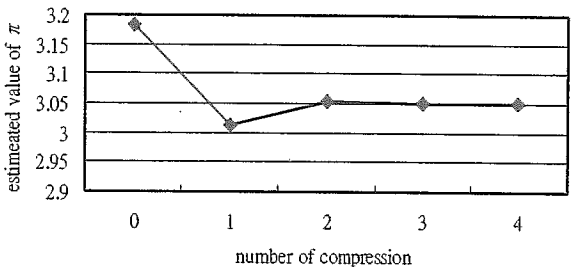

Figure 4. (a) The estimated value of π with a jpg file

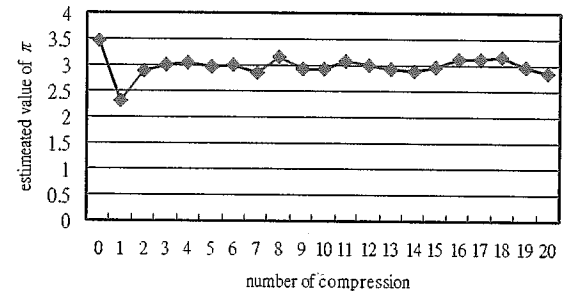

Figure 4. (b) The estimated value of π with a bmp file

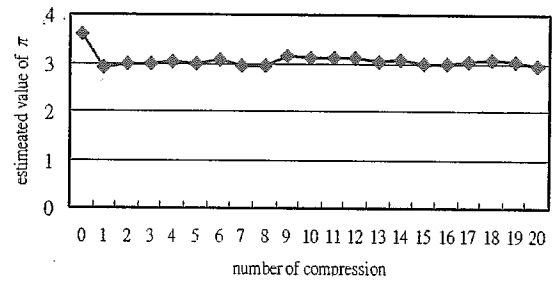

Figure 4. (c) The estimated value of π with a text file

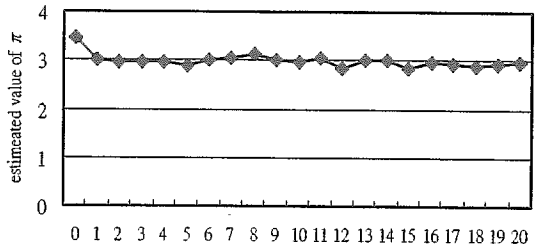

Figure 4. (d) The estimated value of π with a zip file

4. Conclusion

In this paper, we study the random properties of compressed data via Huffman coding. The simulation results show that data is more random after compression in most cases. That is, compressed data has a higher complexity than original data. According to Kolmogorov complexity, we believe other compression methods, such as arithmetic coding or DCT, will have similar property. In the future, we will experiment with other compression methods. This paper shows an alte rnative way of generating random numbers easily without calling a traditional random number generation system call. Still we need some theoretical evidence to prove the feasibility. Thus a concrete model for this type random number generation might be necessary in order to justify this approach of pseudo-random number generation.

Reference

[1] G. Brassard and P. Bratley. Algorithmics: theory and practice, pages 228-230, 1987.

[2] K.L. Chung. Efficient Huffman decoding, Information Processing Letters 61 (1997) 97-99.

[3] D. Gillman, M. Mohtashemi and R. Rivest, On breaking a Huffman Code, IEEE Transactions on Information theory,Vol. 42, No 3, 1996.

[4] D.A. Huffman. A method for the construction of minimum redundancy codes. In Proc. IRE 40 (1951), 1098-1101.

[5] D.E. Knuth. Dynamic Huffman Coding. Journal of Algorithms, 6:163-180, 1985.

[6] M. Li and P. Vitanyi. Introduction to Kolmogorov Complexity and its Applications. Springer-Verlag, 1993.

[7] R. Motwani and P.Raghavan. Randomized Algorithms, Cambridge University Press, 1995.

[8] C. Papadimitriou. Computational Complexity Addison-Wesley, 1994.

[9] K. Sayood. Introduction to data compression, pages 43-50, 1996

[10] M. Sipser. Introduction to the theory of computation, pages 213-220, 1996.