Entropy Coding and Different Coding Techniques

Sandeep Kaur
Asst. Prof. in CSE Dept, CGC Landran, Mohali

Sukhjeet Singh
Asst. Prof. in CS Dept, TSGGSK College, Amritsar.

Abstract – In today's digital world information exchange is been held electronically. So there arises a need for secure transmission of the data. Besides security there are several other factors such as transfer speed, cost, errors transmission etc. that plays a vital role in the transmission process. The need for an efficient technique for compression of Images ever increasing because the raw images need large amounts of disk space seems to be a big disadvantage during transmission & storage. This paper provide a basic introduction about entropy encoding and different coding techniques. It also give comparison between various coding techniques.

Index Terms – Huffman coding, DEFLATE, VLC, UTF-8, Golomb coding.

1. INTRODUCTION

1.1 Entropy Encoding

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium.

One of the main types of entropy coding creates and assigns a unique prefix-free code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix-free output codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes.

1.2 Entropy as a measure of similarity

Besides using entropy encoding as a way to compress digital data, an entropy encoder can also be used to measure the amount of similarity between streams of data and already existing classes of data. This is done by generating an entropy coder/compressor for each class of data; unknown data is then classified by feeding the uncompressed data to each compressor and seeing which compressor yields the highest compression. The coder with the best compression is probably the coder trained on the data that was most similar to the unknown data.

2. CODING TECHNIQUES

1. Huffman Coding-

In computer science and information theory, a Huffman code is a particular type of optimal prefix code that is commonly used for lossless data compression.

1.1 Basic principles of Huffman Coding-

Huffman coding is a popular lossless Variable Length Coding (VLC) scheme, based on the following principles: (a) Shorter code words are assigned to more probable symbols and longer code words are assigned to less probable symbols.

(b) No code word of a symbol is a prefix of another code word. This makes Huffman coding uniquely decodable.

(c) Every source symbol must have a unique code word assigned to it. In image compression systems), Huffman coding is performed on the quantized symbols.

Quite often, Huffman coding is used in conjunction with other lossless coding schemes, such as run-length coding,. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code.

1.2 Applications

Huffman coding today is often used as a "back-end" to some other compression methods. DEFLATE (PKZIP's algorithm) and multimedia codecs such as JPEG and MP3 have a front-end model and quantization followed by Huffman coding

2. Unary coding

Unary coding, sometimes called thermometer code, is an entropy encoding that represents a natural number, n, with n ones followed by a zero (if natural number is understood as non-negative integer) or with n − 1 ones followed by a zero (if natural number is understood as strictly positive integer). For example 5 is represented as 111110 or 11110. Some representations use n or n − 1 zeros followed by a one. The ones and zeros are interchangeable without loss of generality. Unary coding is both a Prefix-free code and a Self-synchronizing code.

2.1 Uses -

1. Golomb Rice code, unary encoding is used to encode the quotient part of the Golomb code word.

•In UTF-8, unary encoding is used in the leading byte of a multi-byte sequence to indicate the number of bytes in the sequence, so that the length of the sequence can be determined without examining the continuation bytes.

•Instantaneously trained neural networks use unary coding for efficient data representation.

2.2 Unary coding in biological networks

New research has shown that unary coding is used in the neural circuits responsible for birdsong production. The nucleus in the brain of the songbirds that plays a part in both the learning and the production of bird song is the HVC (high vocal center). This coding works as space coding which is an efficient strategy for biological circuits due to its inherent simplicity and robustness.

3. ARITHMETIC CODING

Arithmetic coding is a form of entropy encoding used in lossless data compression. Arithmetic coding differs from other forms of entropy encoding, such as Huffman coding, in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number, a fraction n where $[0.0 \leq n < 1.0)$.

3.1 Comparison Arithmetic Coding versus Huffman

In contrast to this the Huffman coding always produces rounding errors, because its code length is restricted to multiples of a bit. This deviation from the theoretical optimum is much higher in comparison to the arithmetic coding's inaccuracies.

The efficiency of an arithmetic code is always better or at least identical to a Huffman code.

4. SHANNON–FANO CODING

In the field of data compression, Shannon–Fano coding, , is a technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). It is suboptimal in the sense that it does not achieve the lowest possible expected code word length like Huffman coding; however unlike Huffman coding, it does guarantee that all code word lengths are within one bit of their theoretical ideal.

Basic technique

In Shannon–Fano coding, the symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1". As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes. When a set has been reduced to one symbol this means the symbol's code is complete and will not form the prefix of any other symbol's code.

The algorithm produces fairly efficient variable-length encodings; when the two smaller sets produced by a partitioning are in fact of equal probability, the one bit of information used to distinguish them is used most efficiently. Unfortunately, Shannon–Fano does not always produce optimal prefix codes; the set of probabilities {0.35, 0.17, 0.17, 0.16, 0.15} is an example of one that will be assigned non-optimal codes by Shannon–Fano coding.

Shannon–Fano coding is used in the IMPLODE compression method, which is part of the ZIP file format.

5. ELIAS GAMMA CODING

Elias gamma code is a universal code encoding positive integers developed by Peter Elias. It is used most commonly when coding integers whose upper-bound cannot be determined beforehand.

5.1 Uses

Gamma coding is used in applications where the largest encoded value is not known ahead of time, or to compress data in which small values are much more frequent than large values.

Gamma coding is a building block in the Elias delta code.

5.2 Comparison

Exponential-Golomb coding generalizes the gamma code to integers with a "flatter" power-law distribution, just as Golomb coding generalizes the unary code. It involves dividing the number by a positive divisor, commonly a power of 2, writing the gamma code for one more than the quotient, and writing out the remainder in an ordinary binary code.

6. TUNSTALL CODING

In computer science and information theory, Tunstall coding is a form of entropy coding used for lossless data compression.

6.1 Properties-

1. Tunstall coding is a code which maps source symbols to a fixed number of bits.

2.Tunstall coding parses a stochastic source with codewords of variable length.

3.for a large enough dictionary, the number of bits per source letter can be infinitely close to , the entropy of the source.

7 GOLOMB Coding

Golomb coding is a lossless data compression method using a family of data compression codes. Alphabets following a geometric distribution will have a Golomb code as an optimal prefix code, making Golomb coding highly suitable for situations in which the occurrence of small values in the input stream is significantly more likely than large values.

8. RICE CODING

Rice coding (invented by Robert F. Rice) denotes using a subset of the family of Golomb codes to produce a simpler (but possibly suboptimal) prefix code. Rice used this set of codes in an adaptive coding scheme; "Rice coding" can refer either to that adaptive scheme or to using that subset of Golomb codes. Whereas a Golomb code has a tunable parameter that can be any positive integer value, Rice codes are those in which the tunable parameter is a power of two. This makes Rice codes convenient for use on a computer, since multiplication and division by 2 can be implemented more efficiently in binary arithmetic.

Rice coding is used as the entropy encoding stage in a number of lossless image compression and audio data compression methods.

The Golomb–Rice coder is used in the entropy coding stage of Rice Algorithm based lossless image codecs.

9. UNIVERSAL CODE (DATA COMPRESSION)

In data compression, a universal code for integers is a prefix code that maps the positive integers onto binary codewords, with the additional property that whatever the true probability distribution on integers, as long as the distribution is monotonic (i.e., $p(i) \geq p(i + 1)$ for all positive i), the expected lengths of the codewords are within a constant factor of the expected lengths that the optimal code for that probability distribution would have assigned. A universal code is asymptotically optimal if the ratio between actual and optimal expected lengths is bounded by a function of the information entropy of the code that, in addition to being bounded, approaches 1 as entropy approaches infinity.

In general, most prefix codes for integers assign longer codewords to larger integers. Such a code can be used to efficiently communicate a message drawn from a set of possible messages, by simply ordering the set of messages by decreasing probability and then sending the index of the intended message..

Relationship to practical compression

universal codes are useful when Huffman coding cannot be used — for example, when one does not know the exact probability of each message, but only knows the rankings of their probabilities.

Universal codes are also useful when Huffman codes are inconvenient. For example, when the transmitter but not the receiver knows the probabilities of the messages, Huffman coding requires an overhead of transmitting those probabilities to the receiver. Using a universal code does not have that overhead.

10. SHANNON CODING

In the field of data compression, Shannon coding, named after its creator, Claude Shannon, is a lossless data compression technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). It is suboptimal in the sense that it does not achieve the lowest possible expected code word length like Huffman coding, and never better but sometime equal to the Shannon-Fano coding.

In Shannon coding, the symbols are arranged in order from most probable to least probable, and assigned codewords by taking the first digits from the binary expansions of the cumulative probabilities

11. RANGE ENCODING

Range encoding is an entropy coding method which effectively rediscovered the FIFO arithmetic code. Given a stream of symbols and their probabilities, a range coder produces a space efficient stream of bits to represent these symbols and, given the stream and the probabilities, a range decoder reverses the process.

Range coding is very similar to arithmetic encoding, except that encoding is done with digits in any base, instead of with bits, and so it is faster when using larger bases (e.g. a byte) at small cost in compression efficiency After the expiration of the first (1978) arithmetic coding patent,] range encoding appeared to clearly be free of patent encumbrances. This particularly drove interest in the technique in the open source community.

11.1 Relationship between range coding and arithmetic coding

Arithmetic coding is the same as range encoding, but with the integers taken as being the numerators of fractions. These fractions have an implicit, common denominator, such that all the fractions fall in the range [0,1). Accordingly, the resulting arithmetic code is interpreted as beginning with an implicit "0.". As these are just different interpretations of the same coding methods, and as the resulting arithmetic and range codes are identical, each arithmetic coder is its corresponding range encoder, and vice versa. In other words, arithmetic coding and range encoding are just two, slightly different ways of understanding the same thing.

12. EXPONENTIAL- GOLOMB CODING

An exponential-Golomb code (or just Exp-Golomb code) is a type of universal code. To encode any nonnegative integer x using the exp-Golomb code:

1.Write down x+1 in binary

2.Count the bits written, subtract one, and write that number of starting zero bits preceding the previous bit string.

This is identical to the Elias gamma code of x+1, allowing it to encode 0.[2]

13. FIBONACCI CODING

It is a universal code] which encodes positive integers into binary code words.

13.1 Comparison with other universal codes

Fibonacci coding has a useful property that sometimes makes it attractive in comparison to other universal codes: it is an example of a self-synchronizing code, making it easier to recover data from a damaged stream. With most other universal codes, if a single bit is altered, none of the data that comes after it will be correctly read. With Fibonacci coding, on the other hand, a changed bit may cause one token to be read as two, or cause two tokens to be read incorrectly as one, but reading a "0" from the stream will stop the errors from propagating further. Since the only stream that has no "0" in it is a stream of "11" tokens, the total edit distance between a stream damaged by a single bit error and the original stream is at most three.

14. CONCLUSION

Compression is an important technique in the multimedia computing field. This is because we can reduce the size of data and transmitting and storing the reduced data on the Internet and storage devices are faster and cheaper than uncompressed data. Many image and video compression standards such as JPEG, JPEG2000, and MPEG-2, and MPEG-4 have been proposed and implemented. In all of them entropy coding, arithmetic and Huffman algorithms are almost used. In other words, these algorithms are important parts of the multimedia data compression standards. From comparison between different coding techniques we know that Huffman coding is easier than arithmetic coding. Arithmetic algorithm yields much more compression ratio than Huffman algorithm while Huffman coding needs less execution time than the arithmetic coding. Also we learn that Exponential-Golomb coding generalizes the gamma code to integers. Universal codes are useful when Huffman codes are inconvenient.

REFERENCES

[1] D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098–1102. Huffman's original article.
[2] Ken Huffman. Profile: David A. Huffman, Scientific American, September 1991, pp. 54–58
[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 16.3, pp. 385–392.
[4] MacKay, David J.C. (September 2003). "Chapter 6: Stream Codes". Information Theory, Inference, and Learning Algorithms (PDF/PostScript/DjVu/LaTeX). Cambridge University Press.ISBN 0-521-64298-1. Archived from the original on 22 December 2007. Retrieved 2007-12-30.
[5] Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, 5BP (2007). "Section 22.6. Arithmetic Coding". Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.
[6] Rissanen, Jorma (May 1976). "Generalized Kraft Inequality and Arithmetic Coding" (PDF). IBM Journal of Research and Development **20** (3): 198–203. doi:10.1147/rd.203.0198. Retrieved 2007-09-21.
[7] Rissanen, J.J.; Langdon, G.G., Jr (March 1979). "Arithmetic coding" (PDF). IBM Journal of Research and Development **23** (2): 149–162. doi:10.1147/rd.232.0149. Archived (PDF)from the original on 28 September 2007. Retrieved 2007-09-22.
[8] Witten, Ian H.; Neal, Radford M.; Cleary, John G. (June 1987). "Arithmetic Coding for Data Compression" (PDF). Communications of the ACM **30** (6): 520–540.doi:10.1145/214762.214771. Archived (PDF) from the original on 28 September 2007. Retrieved 2007-09-21.
[9] Rodionov Anatoly, Volkov Sergey (2010) “p-adic arithmetic coding” Contemporary Mathematics Volume 508, 2010 Contemporary Mathematics
[10] Rodionov Anatoly, Volkov Sergey (2007) ” p-adic arithmetic coding”, http://arxiv.org/abs//0704.0834v1