DATA COMPRESSION STATISTICS AND IMPLICATIONS

Sheila Horan
New Mexico State University

ABSTRACT

Bandwidth is a precious commodity. In order to make the best use of what is available, better modulation schemes need to be developed, or less data needs to be sent. This paper will investigate the option of sending less data via data compression. The structure and the entropy of the data determine how much lossless compression can be obtained for a given set of data. This paper shows the data structure and entropy for several actual telemetry data sets and the resulting lossless compression obtainable using data compression techniques.

KEY WORDS

Lossless data compression, Bandwidth efficiency, Data compression

INTRODUCTION

One of the main topics for discussion in the modulation area of communications lately seems to be frequency allocation and the protection of spectrum. The importance of this topic can be seen in the formation of ARTM (the Advanced Range Telemetry group) which is tasked to "evaluate technology which can potentially improve the efficiency of aeronautical telemetry utilization. The technologies which emerge from this project with the greatest potential will be integrated into new range operational capabilities through the ARTM program." [1]. One of the areas of concern for ARTM is bandwidth efficient modulation. Again, in publications like Satellite News [2], the importance of spectrum and preserving and conserving spectrum is evident. There are two ways to make the optimum use of the spectra that is available. One way is to change the modulation of the signals by using higher orders of modulation, or more efficient modulation techniques. The other way is to reduce the amount of data to be sent either by taking less data, or by compressing the data.

Entropy is a measure of the information content of data. The entropy of the data will specify the amount of lossless data compression that can be achieved. However, finding the entropy of data sets is non-trivial [3]. Approximations to the entropy can be obtained by calculating various orders of entropy. The actual entropy can be approximated as the limit as n approaches infinity of the nth order entropy [3].

Many lossless data compression algorithms exist. Some of the main techniques in use are the Huffman [4], Arithmetic [3], Lempel-Ziv [5], runlength, predictive coding or variations and combinations of these. Each of these methods can be found in most data compression texts. The Huffman code is a very efficient code, that is built using variable length codes where the least probable symbol is assigned the longest codeword and the most probable symbol is assigned the smallest codeword. The Arithmetic code is the mapping of several symbols to a specific region on the number line. To code the sequence, one simply sends a numerical value from inside the specific region of the number line. Lempel-Ziv is a dictionary based code which uses information from what has been sent to code what is being sent. Runlength codes are used to code long runs of single symbols or long runs of strings of symbols. Predictive coding can take many forms from simply taking the difference between symbols to modeling the underlying physical process generating the data. The trick to coding is finding what works best with the data that needs to be compressed.

TEST DATA

Thirteen actual telemetry data files were obtained from the Advanced Range Telemetry (ARTM) group. The data were then analyzed to find the 4 bit, 8 bit, and 12 bit entropies. The entropy for a set of data is given by:

$$H(S) = -\lim_{n\to\infty}\frac{1}{n}\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m}\{P(X_1=i_1,X_2=i_2,\ldots,X_n=i_n)\bullet \log P(X_1=i_1,X_2=i_2,\ldots,X_n=i_n)\} \quad (1)$$

Where H (S) is the entropy of the source (the data)
X_i are the elements of all possible sequences of the data. The X_i can be bits, symbols, or words.
n = the length of the sequence
P is the probability
m = the size of the alphabet used for coding; for binary data, m=2, for words of 4 bits in length, m=16, etc.

To find the 4 bit entropy, all possible combinations of four elements from the binary alphabet were found. The frequency of occurrence of each of these 4 bit words was found. This frequency was used as an estimate of the probability of occurrence for each of these words. From these probabilities, the estimate for the 4 bit entropy was found using equation 1 where n=4 and m=2. For the 8 bit entropy, 256 different words are possible, and for the 12 bit entropy, 4096 words are possible. It can be seen that letting n approach infinity will quickly become impractical. The formats for each data set varied. Some of the data were coded into words of 10 bits, some 12 bits, etc. A summary of the data statistics is given in Table 1.

Table 1. Data Set Entropies and Word Size

Data Set	Data word size In bits	4 bit Entropy	8 bit Entropy	12 bit Entropy
SDS001	12	3.9	7.5	10.3
SDS002	12	3.3	6.0	7.5
SDS003	12	3.8	7.0	9.0
SDS004	10	3.4	5.9	7.3
SDS005	10	3.6	6.6	8.6
SDS006	10	3.7	5.9	7.6
SDS007	10	2.9	5.9	7.5
SDS008	12	2.6	4.2	4.7
SDS009	12	3.2	5.8	6.9
SDS010	12	2.7	4.3	4.8
SDS011	12	2.8	4.9	5.9
SDS012	16	2.5	3.7	5.2
SDS013	24	2.8	3.7	4.6

DATA COMPRESSION RESULTS

To determine the amount of compression that should be possible, the entropy per number of bits must be used. Hence % compression for k bit entropies = (1-(k bit entropy / k))*100. The amount of compression for each data set can then be found. If the k bit entropy is equal to the entropy, then the predicted compression will be a lower bound for all data compression techniques. A plot of the predicted compression from the calculated entropies is given in Figure 1. It can be seen that the compression increases as the number of bits per symbol is increased. This indicates that the entropies do not yet equal the entropy for the data sets. Consequently, compression techniques should be able to achieve values better than these predicted values.

The Huffman, an Adaptive Huffman, Arithmetic, and Lempel-Ziv compression algorithms were applied using programs from Mark Nelson's text [6]. The Huffman codes and Arithmetic codes use an 8-bit word length in the code. Two variations of the Lempel-Ziv algorithm were used. The LZSS uses a pair of values to indicate the location of the match and the length of the match in the dictionary. The LZW algorithm involves only sending one element instead of a pair of elements, and using a start up alphabet in the dictionary consisting of all the letters of the source alphabet. The percent compression for each technique along with the predicted compression is given in Table 2.

Table 2. Data Compression Results

Data Set	Huffman	Adapt. Huffman	LZSS	LZW	Arith-metic	% 4 bit Predicted compress	% 8 bit Predicted compress	% 12 bit predicted compress
SDS001	6.3	6.9	15.5	-9.9	6.6	2.5	6.3	14.2
SDS002	24.1	25.5	61.1	33.8	24.7	17.5	25.0	37.5
SDS003	11.5	15.5	39.7	-7.3	11.9	5.0	12.5	25.0
SDS004	25.1	27.7	50.2	11.5	25.5	15.0	26.3	39.2
SDS005	18.5	19.0	34.7	18.5	18.7	10.0	17.5	28.3
SDS006	25.1	25.9	40.2	32.5	25.4	7.5	26.3	36.7
SDS007	25.9	26.5	41.9	28.2	41.9	27.5	26.3	37.5
SDS008	46	48.3	71.0	53.9	46.7	35.0	47.5	60.8
SDS009	27.5	29.4	64.7	26.3	27.7	20.0	27.5	42.5
SDS010	45.2	47.5	68.9	52.7	45.9	32.5	46.3	60.0
SDS011	37.6	39.7	64.5	46.0	38.1	30.0	38.8	50.8
SDS012	51.9	51.9	65.9	63.7	51.9	37.5	53.8	56.7
SDS013	48.6	53.7	65.5	64.8	51.4	30.0	53.8	61.7

A negative with the compression value indicates that the file was expanded instead of compressed. It can be observed that each data file compressed differently. Figure 2 contains the plot of the results of the compression techniques. It can be seen from Figure 2 that certain files will compress very well. A 60% compression would mean that the file would take up less than half its original size. In all cases, there is at least one technique that provides compression of 10% or more. With the demands on spectra, even this little gain can be worth something. Since the Huffman and Arithmetic codes that were tried work with 8 bit word sizes, if the compression obtained by these techniques is compared with the 8 bit entropy, we see a very close match. All but 2 of the results are within 10% or less of the 8 bit entropy. Since the LZ algorithms achieve larger compression than this indicates, the 8 bit word size is not the best choice and that this 8 bit entropy is not the entropy. Also, since the k bit entropies continue to increase with k, this also indicates that the actual entropy has yet to be found. Hence even larger compressions can be expected.

CONCLUSION

Data compression is a viable possibility to aid in optimal use of frequency spectra. Less data transmitted translates into less bandwidth necessary to transmit the data.

Further work is necessary. Making use of the data structure of the data sets can result in the use of prediction techniques, which can provide more compression. Work in this area is needed. Also, analysis of how data performs in the channel will also determine its ultimate usefulness. This work will continue by simulating the data compression in aeronautical channels.

ACKNOWLEDGMENTS

The author gratefully acknowledges ARTM and the Air Force Office of Scientific Research (AFOSR) for funding this project. The author also credits James Woods, an undergraduate at New Mexico State University, for running and developing the programs used in this paper.

REFERENCES

[1] Irving, Chuck, “Advanced Range Telemetry (ARTM) Concept Exploration”, Air Force Flight Test Center Edwards AFB, CA, October 6, 1997

[2] Satellite News, Vol. 22, Issue 14, April 5, 1999

[3] Sayood, K., Data Compression, Morgan Kaufman Publishers, 1997

[4] Huffman, D. A., “A Method for the construction of minimum redundancy codes”. Proceedings IRE, 40, 1951, pages 1098-1101

[5] Ziv, J. and Lempel, A., “A universal algorithm for data compression”. IEEE Transactions on Information Theory, IT-23(3), May 1977, pages 337-343

[6] Nelson, M., The Data Compression Book, M&T Books, California, 1987.

Figure 1. Predicted Compression from Data Statistics

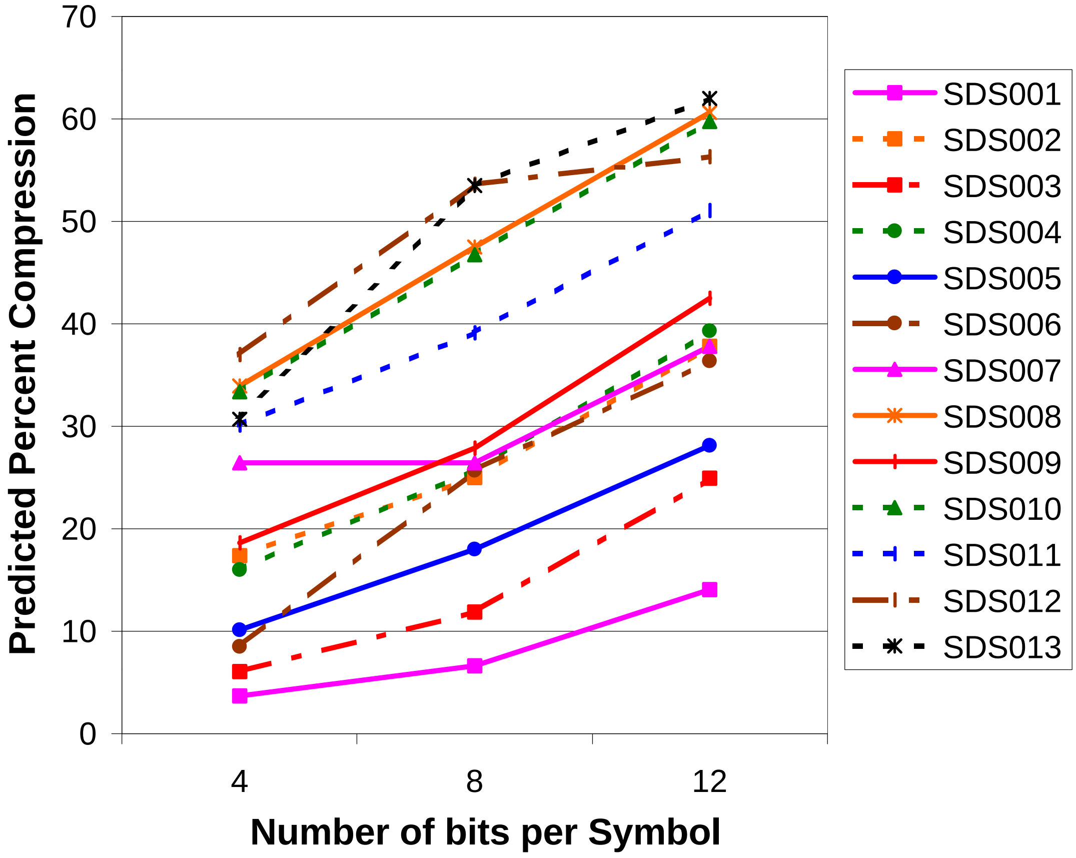
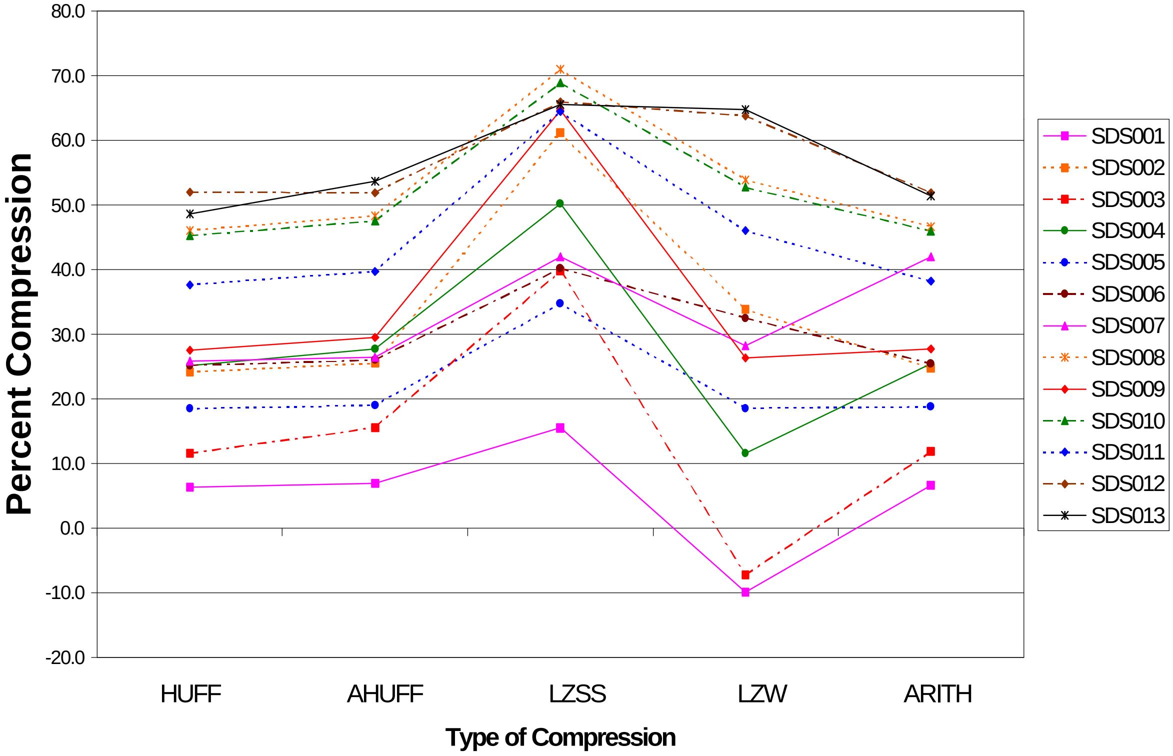


Figure 2.
Data Compression Results


Percent Compression
HUFF
AHUFF
LZSS
LZW
ARITH
Type of Compression
SDS001
SDS002
SDS003
SDS004
SDS005
SDS006
SDS007
SDS008
SDS009
SDS010
SDS011
SDS012
SDS013