International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE) Vol. 1, Issue 2, January 2015

A Review of Modern Paradigms in Data Compression Techniques

[1]Mr. Giridhar S. Sudi,[2]Dr. Meghana Kulkarni,[3]Mr. Vikrant K. Shende
[1]giridharsudi@git.edu,[2]meghanak@vtu.ac.in,[3]vkshende@git.edu

Abstract-**Data compression has been extensively applied to the motion-picture industry and consumer electronics especially in high-definition TVs, advanced multi-media systems, remote sensing systems via satellite, aircraft, radar, teleconferencing, email and social media, mobile phones, hand held PCs etc. Efficient compression of data would significantly decrease the overheads in both the communication and data storage. The efficiency of a data compression scheme is measured by the compression ratio, the resulting distortion after lossy compression, and the implementation complexity. This paper reviews the evolution of compression techniques over the years and the present day modern techniques being explored.**

Artificial Intelligence (AI) approaches appear to be very promising for intelligent information processing due to their massively paralleled computing structures. The main problems solved by Artificial Intelligence (AI) techniques such as Artificial Neural Networks, Fuzzy Logic systems and Genetic Algorithms include optimization, classification, prediction and pattern recognition. Due to the efficiency of AI paradigms in solving complex problems, researchers are applying these techniques to develop new intelligent data compression systems.

Key words: **Compression, compression ratio, Artificial Intelligence, Artificial Neural Network, Fuzzy Logic, Genetic Algorithms.**

I. INTRODUCTION

Data compression is a process that reduces the size of the data, eliminating the redundant information. The compressed data is often more desirable as it reduces the costs of data storage and data transmission. Data compression methods can be classified into lossy and lossless methods. If the compression algorithm removes some parts of data which cannot be recovered during the decompression, then the method is known as lossy. The lossy algorithms are usually used when a perfect reconstruction of the original data is not necessary after the decompression. Such a situation may occur, for example, in compression of video or image data. If the recipient of the video is a human, then small changes in colors, hue or tint introduced during the compression could be imperceptible. If the algorithm does remove any part of the data during compression then it is known as lossless compression method. Such methods are usually used for text compression where loss of data cannot be tolerated. The lossy compression methods typically yield much better compression ratios than lossless algorithms.

Significance of data compression

With the information technology entering every field of life, huge amount of information in the form of digital data is accumulated every day. Therefore, without efficient data compression techniques-

- It would not be practical to put images, let alone audio and video, on websites.
- Cellular phones would never be able to provide communication with increased clarity and reliability.
- The development of digital Satellite Television and HDTV would not have been possible.
- Listening to music on MP3 player or watching a movie on DVD would not be possible.
- Various space agencies around the world could not transmit or store huge amount of data received from satellites.

The data compression technique

The data compression technique is invariably split into two stages: modeling and coding. First, the data sequence is recognized, looking for regular patterns and similarities. This is done in the modeling stage. The modeling method is specific for the type of data we would like to compress. It is obvious that in video data we will be looking for different kind of similarities than in text data. The modeling methods are often different for lossless and lossy methods. Selecting a proper modeling method is very important because the more regularity we find the more we can reduce the data sequence length.

The second stage, coding, is based on the knowledge or information obtained in the modeling stage, and it removes the redundant data. Here, we have to encode the data sequence efficiently to remove the redundancy.

II. Evolution of data compression techniques

a. Conventional compression algorithms

Many conventional compression techniques gather statistical information to identify the frequency patterns in the data. They adopt a two-pass approach. In the first pass, the algorithm gathers statistical information regarding the data to be compressed, such as the frequencies of characters or substrings. In the second pass, the actual encoding is done. A standard encoding scheme is used to ensure that the decoder is able to retrieve the original message. Conventional compression techniques suffer from problems like limited compression ratio, low speed, less reliability and adaptability for different types of data.

b. Adaptive compression algorithms

This approach has an improvement where the algorithm only needs one pass to compress the data. The main idea of adaptive compression algorithms is that the encoding scheme changes as the data being compressed. Thus, the encoding of the n^{th} symbol is based on the characteristics of the data until position n-1. The main advantage of adaptive compression is that it does not require the entire data to be stored into the memory before the compression process can begin, in order to find the statistical pattern of redundancy.

Some examples of adaptive compression algorithms are:

- Adaptive Huffman coding
- Adaptive Arithmetic Coding (AAC)
- Prediction by Partial Matching (PPM)
- Lempel and Ziv LZ77 and LZ78

Advantages of adaptive compression algorithms:

We can see that adaptive compression algorithms represent a form of intelligent systems that adapt their behavior based on the given situation. The main advantage of adaptive compression is that it uses a single pass to compress data. This speeds up the compression process and makes it attractive for use in the cases where speed is more important.

Disadvantages of adaptive compression algorithms:

The use of a single pass has the disadvantage of its inability to see ahead in the file to determine future changes which may contain valuable information for the encoding scheme and change the encoding algorithm accordingly. Also, because they learn the alphabetic distribution during the compression time, they cannot handle drastic changes in the files streams. For example, changes from text to picture. Consequently, adaptive schemes are not recommended for heterogeneous files.

c. Composite Compression Algorithms

In many real world applications, we cannot use a single compression model effectively to adapt to the irregularities of the data. Therefore, a practical approach could be to employ a composite model, which is a combination of several models where only one model can be active at any given time. It has been reported that this approach shows superior performance as compared to conventional compression algorithms by achieving higher compression. A successful composite compression algorithm can be achieved in two ways. Firstly, by running a number of conventional compression algorithms successively. Secondly, by combining a number of simple compression algorithms and heuristically selecting them where they are expected to perform best.

Some examples of composite compression algorithms are:

- Bzip2
- Lempel-Ziv-Huffman
- Markov-chain Algorithm (LZMA)

Advantages of composite compression algorithms:

These algorithms have successfully produced higher compression ratios.

Disadvantages of composite compression algorithms:

The main disadvantage of composite compression algorithms is that they are unreliable when dealing with files that are composed of multiple data types. This is because many composite compression schemes follow fixed procedures that involve applying several compression models in a particular order.

d. Artificial Intelligence techniques for data compression

Artificial Intelligence (AI) techniques attempt to understand and simulate intelligent behavior through the modeling of natural intelligence, such as evolution, insect swarms, neural systems and immune systems. This has resulted in a variety of approaches such as Artificial Neural Networks, Particle Swarm Optimization, Ant Colony, Fuzzy systems and Genetic Algorithms just to name a few. The main problems solved by AI techniques include optimization, classification, prediction and pattern recognition. Researchers have achieved significant success in solving real world problems using these techniques. Due to this reason, there is a growing tendency to apply some of these techniques to develop new intelligent data compression systems.

III. Modern data compression techniques - A review

This section reviews some of the attempts in using Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms techniques to achieve better data compression.

a. Application of Artificial Neural Networks in Data Compression

Artificial Neural Networks (ANN) have the potential to extend data compression algorithms beyond the standard methods of detecting regularities within the data. Although, they have often been avoided as they are considered to be bit slow. Nevertheless, neural networks have been applied to data compression problems.

- Artificial Neural Networks have demonstrated their superiority over classical methods when dealing with noisy or incomplete data. One such application is for data compression. Neural networks seem to be well suited to this particular function, as they have an ability to preprocess input patterns to produce simpler patterns with fewer components. This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. The compressed features may then exit the network into the external environment in their original uncompressed form. The algorithms like the Back Propagation algorithm and the Kohonen self-organizing maps have been used in their work by B. Verma, M. Blumenstein and S. Kulkarni [1].

- A self-organization neural network architecture was attempted to implement vector quantization for image compression [2]. A modified self-organization algorithm, which is based on the frequency-sensitive cost function and centroid learning rule, is utilized to construct the codebooks. Performances of this frequency-sensitive self-organization network and a conventional algorithm for vector quantization are compared. The proposed method is quite efficient and can achieve near-optimal results. Good adaptivity for different statistics of source data can also be achieved.
- An adaptive back-propagation neural network has been proposed to overcome the limitations in the basic back propagation [3]. The basic idea is that different neural networks have different performance and only certain types of networks work best with a particular set of images. As a result, a group of neural networks with an increasing number of hidden layers in the range (hmin, hmax) are used in a single system. The aim is to match the input image blocks with the most appropriate network based on their complexity. Several training schemes have been proposed to train the new adaptive architecture, including parallel training, serial training, activity-based training and direction-based training.

b. Application of Fuzzy Logic in Data Compression

- S. Sathappan from India proposes that an image can be decomposed into image blocks and then a low bit rate still image compression can be performed by compressing the indices of vector quantization and residual codebook is generated by using Modified Fuzzy Possibilistic C-Means with Repulsion and Weighted Mahalanobis Distance [4]. The residual code book is used in his proposed approach which eliminates the distortion in the reconstructed image and enhancing the quality of the image. He reports that experimental results on standard images have showed that the proposed scheme can give a reconstructed image with high PSNR value than the existing image compression techniques.
- Rohit Kumar Gangwar and others have proposed an efficient data compression technique which combines fuzzy logic with that of Huffman coding [5]. While normalizing image pixel, each value of pixel image belonging to that image foreground are characterized and interpreted. The image is sub divided into pixel which is then characterized by a pair of set of approximation. The encoding makes use of Huffman code which is statistically independent to produce more efficient code for compression and decoding makes use of rough fuzzy logic which is used to rebuild the pixel of the image. The method used here are rough fuzzy logic with Huffman coding algorithm (RFHA). They have done comparison of different compression techniques with Huffman coding and fuzzy logic is applied on the Huffman reconstructed image. They report that high compression rates and visually negligible difference between compressed images and original images can be achieved.

c. Application of Genetic Algorithms in Data Compression

Data compression algorithms often require a large search space to be explored in order to find some forms of redundancy within the data or to select an optimal encoding scheme based on the given files. Evolutionary Algorithms (EAs) can be used as a search engine for this purpose or even as a method to evolve new compression algorithms. Nevertheless, data compression is a highly sophisticated procedure and evolving a data compression algorithm is not an easy task. Yet, few attempts have been made to use EAs to evolve data compression models. Existing research mainly focuses on investigating the applications of Genetic Algorithm (GA) and Genetic Programming (GP) in order to explore their potentials in solving data compression problems.

- Koza was the first to use GP to perform compression [6]. He considered, in particular, the lossy compression of images. The idea was to treat an image as a function of two variables (the row and column of each pixel) and to use GP to evolve a function that matches the original as closely as possible. Small images of 30x30 pixels size were treated as symbolic regression problems with just the basic arithmetic operators in the functions set. The evolved function can be then considered as a lossy compressed version of the image. The technique, which was termed programmatic compression, was tested just on one small synthetic image with good success.
- GAs have been used for image compression. Mitra et al. has proposed a new GA-based method for fractal image compression [7]. The proposed method utilizes the GA for finding self similarities in the given image. Then, the system divides the given images into blocks and tries to find functions that approximate the target block. Here, the squared mean error was used as fitness function to evaluate the quality of the individuals. Results on five gray images from the NASA Galileo Mission database were very promising, with GA based compression outperforming some of the best human designed lossless compression algorithms.
- Feiel and Ramakrishnan have proposed a new vector quantization scheme using a GA (GAVQ) to optimize the compression of colored images [8]. In this work, the GA was used to find optimal, or more precisely near optimal codebooks that describe the mapping from the input data to their compressed version. Experiments showed that the results obtained are better than the LBG algorithm by of 5% to 25%. Keong et al. used GA to optimize the Generalized Lloyd Algorithm (GLA).

GLA receives the input vectors and initializes the codebook randomly, such that it maps the input sequence to a compressed digital sequence. GLA refines the codebook though an iterative process in order to reduce the average distortion. The authors introduced a Genetic GLA (GGLA), which uses a GA to find optimal codebooks. Each chromosome in the GA population represents a codebook and the set of genes corresponds to the code words. Simulations with the three versions of the proposed system have been conducted with first-order Gaussian-Markov processes. This showed that GGLAs outperformed GLA in most cases.

CONCLUSION

The quest for further improvements in data compression techniques is ever continuing. The main objectives behind the research works being carried out have been to achieve higher compression ratio, higher compression speed, lesser distortion, better quality of the reconstructed data. The main challenge is to develop a compression system for heterogeneous data type. The use of non-conventional methods likes artificial neural networks, fuzzy logic and genetic algorithms have shown promising results and can certainly help us to develop a composite compression system with enhanced performance characteristics.

REFERENCES

[1] B. Verma, M. Blumenstein and S. Kulkarni, "*A Neural Network based Technique for Data Compression*", proceedings of the IASTED International Conference on Modeling and Simulation, MSO. IASTED, Vol. 97, pp 12–16.

[2] Oscal T. C. Chen, Bing J. Sheu and Wai-Chi Fang, "*Image Compression Using Self-organization Networks*", IEEE Transactions on Circuits and Systems for Video Technology, Volume 4, Issue 5, Oct 1994, pp 480-489 .

[3] Man, Hong Ren Wu, Sophie Liu, and Xinghuo Yu, *"A New Adaptive Back propagation Algorithm Based on Lyapunov Stability Theory for Neural Networks"*, IEEE transactions on neural networks, Vol. 17, No. 6, November 2006, Zhihong.

[4] Dr.S.Sathappan, "*A Vector Quantization Technique for Image Compression using Modified Fuzzy Possibilistic C-Means with Weighted Mahalanobis Distance*", International Journal of Innovative Research in Computer and Communication Engineering, Vol.1, Issue 1, March 2013.

[5] Rohit Kumar Gangwar, Mukesh Kumar, A.K.Jaiswal, Rohini Saxena, "*Performance analysis of image compression using fuzzy logic algorithm*", Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.2, April 2014.

[6] John R. Koza, *"Genetic Programming: On the Programming of Computers by Means of selection"*, MIT Press, Cambridge, MA, USA, 1992.

[7] S. K. Mitra, C. A. Murthy and M. K. Kundu, *"Technique for fractal image compression using genetic algorithms"*, IEEE Trans. On Image Processing, vol. 7;no. 4, pp 586-593,998

[8] Harald Feiel and Sub Ramakrishnan, *"A genetic approach to color image compression"*, in SAC, 1997, pp. 252–256.