Dynamic Markov Compression Using a Crossbar-Like Tree Initial Structure for Chinese Texts

Ghim-Hwee Ong and Jun-Ping Ng
Department of Computer Science, School of Computing
National University of Singapore, Republic of Singapore
{onggh, ngjunpin}@comp.nus.edu.sg

Abstract

This paper proposes the use of a crossbar-like tree structure to use with Dynamic Markov Compression (DMC) for the compression of Chinese text files. DMC had previously been found to be more effective than common compression techniques like **compress** *and* **pack** *and gives a compression gain of between 13.1% and 32.0%. This initial structure is able to improve on DMC's compression results, and outperforms the various initial structures commonly adopted, such as the single-state, linear, tree or braid structures by a gain ranging from 1.5% to 9.6%.*

1. Introduction

This paper aims at studying a method of Dynamic Markov Compression (DMC) [1] to compress large Chinese text files which are encoded with the GB2312 encoding scheme [2, 3]. In particular, this paper proposes the use of a crossbar-like tree structure to improve the compression performance achieved by DMC on Chinese text files.

In the following sections, the various steps involved in DMC are explained, and important aspects of the method are highlighted. Then, various initial structures that can be used in DMC are revised. The proposed initial structure that gives the best performance to date is presented next. A short introduction to Chinese text files and the GB2312 encoding follows, before the key findings of this research are summarized.

2. Dynamic Markov Compression

DMC [1] is a one-pass adaptive compression scheme, based on finite-state models. The compression process in DMC consists of two steps – Modeling and Encoding [4, 5]. Input data is processed symbol by symbol. As each symbol is processed, a Markov model is updated and this generates a new probability distribution for the coder which produces a corresponding output data stream.

Central to the performance of DMC is *cloning*. When specified cloning criteria are met, a new state is created out of the original one. In essence, the new node is to capture the context where the current outgoing transition is seen. Such information helps the model make a more accurate prediction of the next occurring bit. A fixed amount of memory is allocated to the building of new states. When the memory available for new states runs out, Cormack et al. suggested that the model built so far be discarded and reset to the original one [1].

DMC uses an arithmetic coder to encode the input data given probabilities supplied by the Markov model [5, 6]. An in-depth treatment of arithmetic coding is given in [6]. At any point in time, the probability distribution supplied by the Markov model will be the probability distribution of the current state of the model.

3. Revision of Initial Structures

It is known that arithmetic coding provides an optimal solution to the coding phase of compression. Modeling thus holds the key for improving compression results when applying DMC. The study into modeling revolves mainly around the design of an initial structure to be used by DMC. The cloning mechanism will then be responsible to grow this initial structure such that it represents the characteristics of the text to be compressed. There are several initial structures that can be used with DMC.

The ***single state*** structure is the simplest structure. It comprises of a single state with transitions leading from and back to itself as shown in Figure 1.

The ***linear*** structure consists of several states chained together in a straight line (Figure 2). The transitions leading out of a state lead to the state directly below it. The last state in the chain will have transitions leading back to the first state to form a cycle.

The ***tree*** structure resembles a binary tree where the

Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)
0-7695-2316-1/05 $20.00 © 2005 **IEEE**

transitions form the branches and the states form the nodes (Figure 3). Transitions out of the leaf nodes will lead back to the root.

The ***braid*** structure is a generalization of the tree structure where a bit sequence follows transitions from any top level node back to an unique top level node determined by the sequence (Figure 4).

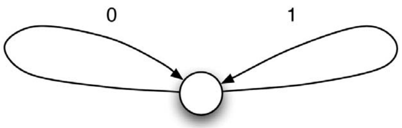

Figure 1 : A Single-state Markov chain structure

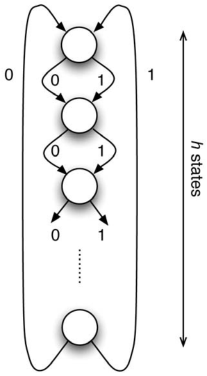

Figure 2 : Example of an initial Markov linear structure

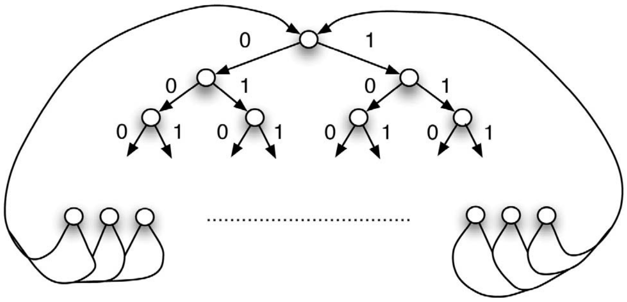

Figure 3 : Example of an initial Markov tree structure

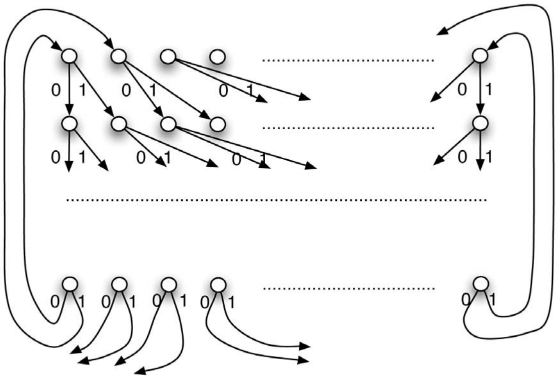

Figure 4 : Example of an initial Markov braid structure

4. Proposal - Crossbar-Like Tree Structure

This paper proposes an initial structure – *Crossbar-like Tree Structure* - which has empirically been shown to perform better than the initial structures reviewed so far, when used to compress GB-encoded Chinese text files.

The structure consists of 2^h trees, each of height h, where $h=1,2,\ldots$. The various trees are all interconnected systematically, such that there is exactly one transition from one of the 2^{h-1} leaf states of a tree going to one of the 2^h trees. Figure 5 shows the structure with a height h of 8.

The structure is termed such because it behaves conceptually like a crossbar network. Each tree is traversed according to the input encountered, and will eventually lead to the root node of another tree. The destination tree can be any of the other trees, including itself. This is logically similar to a crossbar network, where each tree is a node in the network. Starting from a node, a given input determines which path is activated. This path will lead to the destination node.

5. Chinese Text Files and GB2312 Encoding

Chinese texts are very different from English texts. While the smallest units of English words are the 26 letters in the English alphabet, there are more than 10,000 different commonly used Chinese characters. Also, unlike an ordinary English text file which consists of purely ASCII characters, a Chinese text file contains both codes for Chinese characters, like 2-byte GB codes [2] or 3-byte Big-5 codes [7], and 1-byte codes for ASCII characters. This is because codes for Chinese characters are not designed to replace ASCII codes but to supplement ASCII codes for representation of Chinese text files.

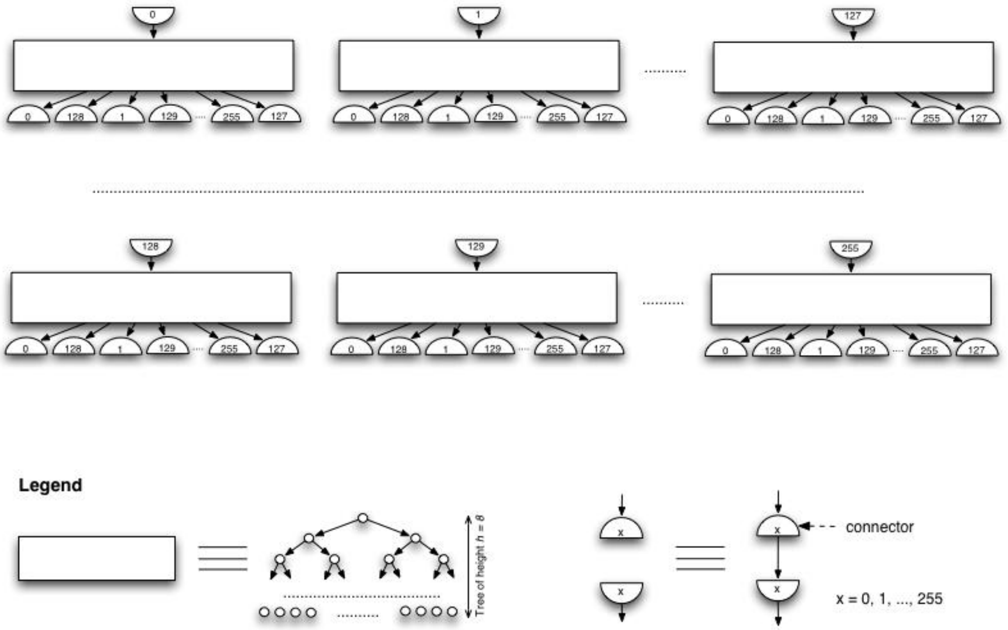

Figure 5 : Crossbar-like tree structure with h = 8

One of the widely used character sets for internal representation in computers is the GB2312-80 Coded Chinese Graphic Character Set for Information Interchange (Primary Set) [2]. It contains a total of 6,763 prearranged Chinese characters and 682 non-Chinese characters (including Chinese punctuation marks).

6. Results and Discussion

The following results are of the compression achievable by DMC for each of the six test files. These files are used as the sample files on which the compression algorithms are run:

FILE1 (44KB)	→	Collection of science articles and news
FILE2 (200KB)	→	Collection of short articles and commentary
FILE3 (491KB)	→	Short story 《灵山》
FILE4 (559KB)	→	Collection of children tales
FILE5 (948KB)	→	Short story 《大宅门 》
FILE6 (1110KB)	→	Text of Chinese classic 《三国演义》

The percentages shown represent the ratio of the compressed file size over the original file size. The compression results are obtained with threshold values of 2 for *MIN_CNT1 and MIN_CNT2* [1, 6]. The memory available for cloning is set so that there is always enough memory to create new states if the cloning process so requires. These runs, and other following runs, are done on an Apple iBook with an IBM G4 PowerPC 933MHz processor and 640MB of RAM. It should be noted however that the platform does not affect the compression results. Execution times may differ but the results are similar because the compression will produce the same output given the same input data.

Table 1 gives the results for the various initial structures discussed earlier in Section 3. It has been determined empirically that using a height of 8 gives the best results for the other three initial structures, and thus only the results for the initial structures with a height of 8 are shown.

Table 2 gives the results for the crossbar-like tree structure. The whole array of results for varying heights is given in the table. It is observed that the crossbar-like tree structure with heights of 1, 2, 4, 6, 8 give better compression than those of 3, 5, and 7. This can be similarly explained as have been done for the other initial structures in [8].

Table 1 : Compression results (%) for various initial structures

Text Files	Initial Structure Used			
	Single	Linear	Tree	Braid
FILE1	71.21	61.31	61.18	59.63
FILE2	70.40	60.56	59.55	58.85
FILE3	59.69	53.84	53.57	52.22
FILE4	52.09	47.62	47.34	46.40
FILE5	43.05	39.77	39.05	38.38
FILE6	58.63	51.76	51.15	50.50
Average	59.18	52.48	51.97	51.00

Table 2 : Compression results (%) for Crossbar-like tree structure

Text Files	Height of Crossbar-Like Tree Structure							
	1	2	3	4	5	6	7	8
FILE1	71.01	67.11	107.28	62.91	113.23	75.33	105.13	57.19
FILE2	67.65	65.52	96.80	60.73	113.16	70.75	104.94	57.24
FILE3	58.70	56.95	75.32	53.85	78.47	60.97	81.84	51.06
FILE4	52.06	50.59	62.62	48.16	113.68	55.15	104.58	45.08
FILE5	42.33	41.41	52.61	39.22	58.26	45.02	51.41	37.42
FILE6	57.26	54.08	62.22	51.67	113.41	60.49	105.65	49.24
Average	58.17	55.94	76.14	52.76	98.37	61.29	92.26	49.54

As can be seen from the tables, the crossbar-like tree structure of height 8 outperforms the other initial structures. Comparing the results obtained, the crossbar-like tree structure gives an additional compression gain of about 1.5% to as much as 9.6%.

This observation also reinforces that the initial structure has an important bearing on the compression effectiveness of DMC. A structure like the crossbar-like tree initial structure, when used for compressing Chinese text files, have some likeness to the characteristics of the input data. This reduces the time needed by the cloning mechanism to model the structure to suit the characteristics of the input data, leading to better compression.

Consistent with the results reported in [8], DMC when coupled with the crossbar-like tree structure, will be able to compress Chinese text files better than the other existing approaches. Table 3 shows the compression achieved by some of the common compression approaches.

Table 3 : Compression results (%) of various approaches.

Text Files	Compression Approaches		
	compress	pack	SAC
FILE1	68.69	78.67	80.07
FILE2	68.16	77.86	79.12
FILE3	64.07	76.19	82.46
FILE4	59.11	76.34	81.51
FILE5	51.27	74.73	86.44
FILE6	64.52	76.94	79.57
Average	62.64	76.79	81.53

Compared with the effectiveness of other compression algorithms, the crossbar-like tree structure allows DMC (49.54%) to do better than Compress (62.64%), Pack (76.79%) and Static Arithmetic Coding (SAC) (81.53%).

7. Conclusion

The proposed crossbar-like tree structure is suitable for use in DMC to compress Chinese text files. The structure is able to give the best compression ratio with a height of 8 and is able to give an additional compression gain by 1.5% to as much as 9.6% over other initial structures. Further, gains of as much as 13.1% to 32.0% are obtained with employing the crossbar-like tree structure with DMC, when compared with common compression approaches like ***compress*** and ***pack***.

8. Acknowledgements

This authors' work was supported by the National University of Singapore Academic Research Fund under grant R-252-000-118-112.

10. References

[1] G. V. Cormack and R. N. S. Horspool, "Data Compression Using Dynamic Markov Modeling*", The Computer Journal*, Vol.30, No.6, 1987, pp.541 – 550.

[2] The People's Republic of China National Standards Institute, "GB2312-80 – The Code of Chinese Graphic Character Set for Information Interchange (Primary Set)", Beijing, China, 1980.

[3] http://i18nwithvb.com/surrogateime/codepages/gbk.htm

[4] J. Rissanen and G. Langdon, "Universal Modeling and Coding", *IEEE Trans. Inf. Theory*, IT-27, 1981, pp.12-23.

[5] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression", Comm. ACM, Vol.30, No.6, 1987, pp.520 – 540.

[6] T. C. Bell, J. G. Cleary, and I. H. Witten*, Text Compression*, New Jersey: Prentice Hall, 1990.

[7] Chinese Character Analysis Group, *Symbol and Character Tables of Chinese Character Code for Information Interchange*, Vol. II, 2nd Ed., National Central Library, Taiwan, November 1982.

[8] G. H. Ong and J. P. Ng, "Exploring the Initial Structures of Dynamic Markov Modeling for Chinese Text Compression", *International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2004, pp.460 – 463.