

# Distributed Arithmetic Coding for Sources with Hidden Markov Correlation

Yan Zhang, Nan Yang and Yong Fang  
 Chang'an University, Xi'an, China  
 E-mail: {2019124014, nyang, fy}@chd.edu.cn

**Abstract**—Distributed arithmetic coding (DAC) has been shown to be effective for Slepian-Wolf coding, especially for short data blocks. In this paper, we propose to use the DAC to compress memory-correlated sources. More specifically, the correlation between sources is modeled as a hidden Markov process. Because image pixels are correlated, the image is modeled as a hidden Markov source then DAC compression is implemented, and forward algorithms are embedded in the decoding process. Experimental results show that the performance is close to the theoretical Slepian-Wolf limit. When the image is used as a hidden Markov source for DAC compression, it shows lower error rate.

## I. INTRODUCTION

We consider the problem of Slepian-Wolf Coding (SWC) with decoder Side Information (SI). The encoder compresses discrete source  $X = \{x_t\}_{t=1}^N$  in the absence of  $Y = \{y_t\}_{t=1}^N$ , discretely correlated SI. Slepian-Wolf theorem points out that lossless compression is achievable at rates  $R \geq H(X|Y)$ , the conditional entropy of  $X$  given  $Y$ , where both  $X$  and  $Y$  are discrete random processes [1]. Conventionally, channel codes, such as turbo codes [2] or Low-Density Parity-Check (LDPC) codes [3], are used to deal with the SWC problem.

Some SWC techniques based on entropy coding are proposed, such as Distributed Arithmetic Coding (DAC) [4, 5] and Overlapped Quasi-Arithmetic Coding (OQAC) [6]. These schemes can be seen as an extension of classic Arithmetic Coding (AC) whose principle is to encode source  $X$  at rates  $H(X|Y) \leq R < H(X)$  by allowing overlapped intervals, where  $H(X)$  is the entropy of the source. The overlapping leads to a larger final interval and hence a shorter codeword. However, ambiguous codewords are unavoidable at the same time. A soft joint decoder exploits SI  $Y$  to decode  $X$ . Afterwards, the time-shared DAC (TS-DAC) [7] is proposed to deal with the symmetric SWC problem. To realize rate-incremental SWC, the rate-compatible DAC is proposed in [8]. [10] develops the DAC spectrum to assist in decoding and analyzed its complexity for equivalent sources, detailed analysis of the coding efficiency of binary DAC coding in [11].

In this paper, we research how to use the DAC to compress sources with hidden Markov correlation.

## II. BACKGROUND REVIEW

### A. Binary Distributed Arithmetic Coding

Distributed arithmetic code is a coding scheme based on distributed source coding and using arithmetic code as the core

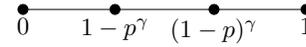


Fig. 1. distributed arithmetic coding.

of the codec.

Let  $p$  be the bias probability of binary source  $X$ , i.e.,  $p = P(x_t = 1)$ . In the classic AC, source symbol  $x_t$  is iteratively mapped onto sub-intervals of  $[0, 1]$ , whose lengths are proportional to  $(1 - p)$  and  $p$ . The resulting rate is  $R \geq H(X)$ . In the DAC [4], interval lengths are proportional to the modified probabilities  $(1 - p)^\gamma$  and  $p^\gamma$ , where  $0 \leq \gamma \leq 1$  is the overlap factor. To fit the  $[0, 1]$  interval, the sub-intervals have to be partially overlapped. More specifically, symbols  $x_t = 0$  and  $x_t = 1$  correspond to intervals  $[0, (1 - p)^\gamma]$  and  $[1 - p^\gamma, 1]$ , respectively,  $[1 - p^\gamma, (1 - p)^\gamma]$  is called overlap interval as shown in Fig. 1.

It is just the overlapping that leads to a larger final interval, and hence a shorter codeword. However, as a cost, the decoder can not decode  $X$  unambiguously without  $Y$ .

For binary sources, the rate after message encoding is

$$R = -p \log_2 p^\gamma - (1 - p) \log_2 ((1 - p)^\gamma) = \gamma H(X). \quad (1)$$

If the source  $X$  is a binary equivalent source, that is  $p = 0.5$ , then  $H(x) = 1$ ,  $R = \gamma$ . According to Slepian-Wolf theorem,

$$R = \gamma H(X) \geq H(X|Y). \quad (2)$$

Therefore, the range of the overlap factor is

$$H(X|Y)/H(X) \leq \gamma < 1. \quad (3)$$

To describe the decoding process, we define a ternary symbol set  $\{0, \chi, 1\}$ , where  $\chi$  represents a decoding ambiguity. Let  $C_X$  be the codeword and  $\tilde{x}_t$  be the  $t$ -th decoded symbol, then

$$\tilde{x}_t = \begin{cases} 0, & 0 \leq C_X < 1 - p^\gamma \\ \chi, & 1 - p^\gamma \leq C_X < (1 - p)^\gamma. \\ 1, & (1 - p)^\gamma \leq C_X < 1 \end{cases} \quad (4)$$

When the  $t$ -th symbol is decoded, if  $\tilde{x}_t = \chi$ , the decoder performs a branching: two candidate branches are generated, corresponding to two alternative symbols  $x_t = 0$  and  $x_t = 1$ . For each new branch, its metric is updated and the corresponding interval is selected for next iteration. To reduce complexity,

every time after decoding a symbol, the decoder uses the  $M$ -algorithm to keep at most  $M$  branches with the best partial metric, and prunes others [4].

Note that the metric is not reliable for the last symbols of a finite length sequence  $X$  [5]. The problem is solved by encoding the last  $T$  symbols without interval overlapping [5]. It means that for  $1 \leq t \leq (N - T)$ ,  $x_t$  is mapped onto  $[0, (1 - p)^\gamma]$  and  $[1 - p^\gamma, 1]$ ; while for  $(N - T + 1) \leq t \leq N$ ,  $x_t$  is mapped onto  $[0, 1 - p]$  and  $[1 - p, 1]$ .

Therefore, a binary DAC system can be described by four parameters:  $\{p, \gamma, M, T\}$ .

### B. Hidden Markov Model

Hidden Markov Model (HMM) is a statistical model, which is used to describe a Markov process with hidden unknown parameters.

The sequence that can be directly observed in the hidden Markov model is called the sequence of observations, and its value depends on the sequence of hidden states. Each hidden state of the HMM generates an observation value with a certain probability distribution, so the information of the sequence of hidden states can be obtained through the information of the sequence of observations.

Let  $S = \{s_t\}_{t=1}^N$  be a sequence of hidden states and  $Z = \{z_t\}_{t=1}^N$  be a sequence of observations. A hidden Markov process is defined by  $\lambda = (A, B, \pi)$ :

$A = \{a_{ji}\}$ : state transition probability matrix, where  $a_{ji} = P(s_t = i | s_{t-1} = j)$ ;

$B = \{b_i(k)\}$ : observation probability distribution, where  $b_i(k) = P(z_t = k | s_t = i)$ ;

$\pi = \{\pi_i\}$ : initial state distribution, where  $\pi_i = P(s_1 = i)$ .

## III. METHODOLOGY

As mentioned above, the HMM includes observation sequence and hidden sequence. We can obtain the hidden sequence with the help of observation sequence. The distributed arithmetic code obtains the source with the assistance of side information, so we can use the hidden Markov process to establish the correlated source, and decode by solving the hidden sequence. Because of the correlation between image pixels, we apply this scheme to the image.

### A. DAC for Hidden Markov Correlation

Assume that binary source  $X$  and SI  $Y$  are correlated by  $Y = X \oplus Z$ , where  $Z$  is generated by a hidden Markov model with parameter  $\lambda$ .  $X$  is encoded using a  $\{p, \gamma, M, T\}$  DAC encoder.

There are three different ways to solve different hidden Markov problems, the aim of forward algorithm is to compute  $P(z_1, \dots, z_t | \lambda)$ , given observation  $\{z_1, \dots, z_t\}$  and model  $\lambda$ , so we can embed the forward algorithm into the DAC decoder.

The decoding process is very similar to what described in [4]. The only difference is that the metric of each branch is replaced by  $P(z_1, \dots, z_t | \lambda)$ , where  $z_t = x_t \oplus y_t$ . The forward algorithm process is as follows:

Let  $\alpha_t(i)$  be the probability of observing the partial sequence  $\{z_1, \dots, z_t\}$  such that state  $s_t$  is  $i$ , i.e.,

$$\alpha_t(i) = P(z_1, \dots, z_t, s_t = i | \lambda). \quad (5)$$

Initially, we have

$$\alpha_1(i) = \pi_i b_i(z_1). \quad (6)$$

For  $t > 1$ ,  $\alpha_t(i)$  can be induced through iteration

$$\alpha_t(i) = \left\{ \sum_j [\alpha_{t-1}(j) a_{ji}] \right\} b_i(z_t). \quad (7)$$

Therefore,

$$P(z_1, \dots, z_t | \lambda) = \sum_i \alpha_t(i). \quad (8)$$

In practice,  $\alpha_t(i)$  is usually normalized by

$$\alpha_t(i) = \frac{\alpha_t(i)}{\delta_t}, \quad (9)$$

where

$$\delta_t = \sum_i \alpha_t(i). \quad (10)$$

In this case, we have

$$P(z_1, \dots, z_t | \lambda) = \prod_{t'=1}^t \delta_{t'}. \quad (11)$$

### B. Image Compression

The pixels of an image are related, so we can use DAC-based HMM to compress the image. We use one line of the image as the source  $X$ , so for an  $n$ -line image,  $n$  encoding and decoding processes are required respectively. Before encoding, we need to obtain the hidden Markov model parameter  $\lambda$ .

The Baum-Welch is an algorithm for solving HMM parameter. It is an unsupervised learning algorithm, because the Baum-Welch algorithm only uses the observation sequence when solving the model parameter, instead the hidden sequence.

When we use  $\lambda$  to describe the hidden Markov process, the Baum-Welch algorithm looks for the local maximum of  $\lambda^* = \arg \max_{\lambda} P(Z | \lambda)$ , which is the HMM  $\lambda$  that can maximize the probability of the observed sequence.

The Baum-Welch algorithm requires forward variables, which can be obtained by (6)-(7), and the backward variables are as follows, let  $\beta_t(i) = P(z_{t+1}, \dots, z_N | s_t = i, \lambda)$  be the probability that the state at time  $t$  is  $i$ , the remaining part of the observation sequence is  $\{z_{t+1}, \dots, z_N\}$ , we calculate  $\beta_t(i)$  as,

$$\beta_N(i) = 1, \quad (12)$$

$$\beta_t(i) = \sum_j [\beta_j(t+1) a_{ij}] b_j(z_{t+1}). \quad (13)$$

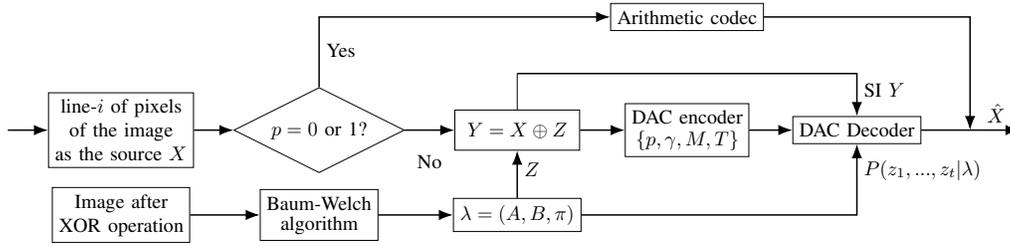


Fig. 2. DAC-based HMM applied to image.

Let  $\gamma_t(i)$  denote the probability of being in the state  $i$  at time  $t$  given the observation sequence and parameter  $\lambda$  :

$$\begin{aligned} \gamma_t(i) &= P(s_t = i | Z, \lambda) \\ &= \frac{P(s_t = i, Z | \lambda)}{P(Z | \lambda)} \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_j \alpha_t(j) \beta_t(j)}. \end{aligned} \quad (14)$$

Let  $\xi_t(i, j)$  denote the probability that the time  $t$  is state  $i$  and the time  $t + 1$  is in the  $j$  given the observation sequence and parameter  $\lambda$  :

$$\begin{aligned} \xi_t(i, j) &= P(s_t = i, s_{t+1} = j | Z, \lambda) \\ &= \frac{P(s_t = i, s_{t+1} = j, Z | \lambda)}{P(Z | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(z_{t+1})}{\sum_k \sum_w \alpha_t(k) a_{kw} \beta_{t+1}(w) b_w(z_{t+1})}. \end{aligned} \quad (15)$$

Then update the parameters,

$$\pi_i^* = \gamma_i(1), \quad (16)$$

which is the probability that the state is  $i$  at time 1.

$$a_{ij}^* = \frac{\sum_{t=1}^{N-1} \xi_t(i, j)}{\sum_{t=1}^{N-1} \gamma_i(t)}, \quad (17)$$

$$b_i^*(k) = \frac{\sum_{t=1}^N 1_{z_t=k} \gamma_i(t)}{\sum_{t=1}^N \gamma_i(t)}, \quad (18)$$

where

$$1_{z_t=k} = \begin{cases} 1 & \text{if } z_t = k \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

Repeat the above steps until convergence, the parameter  $\lambda$  of HMM can be obtained .

In an image with a size of  $n * N$ , XOR two adjacent rows of pixels to obtain an image of  $(n - 1) * N$  size. Let each line of pixels in image after XOR operation as the observation sequence, and obtain HMM  $\lambda_i$  through Baum-Welch algorithm.

The model parameter of image compression can be described as

$$\lambda = \frac{\sum_{i=0}^{n-1} \lambda_i}{n-1}. \quad (20)$$

Each line of the image is compressed as the source  $X$ , when  $p = 0$  or  $p = 1$ , i.e., this line of the image is all white or all black, use arithmetic coding for the current line,  $\gamma = 0$  at this time. Otherwise, the DAC encoder compresses this line. The specific process is shown in Fig. 2.

#### IV. EXPERIMENTAL RESULTS

In the experimental part, we finished two parts of work. First, we compressed the virtual source and compared it with LDPC-based HMM [9]. In the second part, we compressed two different types of images.

##### A. Compared with LDPC

We have implemented a 16-bit DAC codec system. The bias probability of  $X$  is  $p = 0.5$ . According to the recommendation of [5], we set  $M = 2048$  and  $T = 15$ . The same 2-state (0 and 1) and 2-output (0 and 1) sources as in [9] are used in simulations (see Table I). The length of data block used in each test is  $N = 1024$ .

To achieve lossless compression, each test starts from  $\gamma = H(X|Y)$  (see Table II). If the decoding fails, we increase  $\gamma$  with 0.01. Such process is iterated until the decoding succeeds. For each model, results are averaged over 1000 trials. Experimental results are listed in Table II.

For comparison, experimental results for the same settings from [9] are also included in Table II. In each test of [9],  $N = 16384$  source symbols are encoded using an LDPC code. In addition, to synchronize the hidden Markov model,  $N\alpha$  original source symbols are sent to the decoder directly without compression.

The results show that the DAC performs similarly to or slightly better than (for models 1 and 2) the LDPC-based approach [9]. Moreover, for hidden Markov correlation, the DAC outperforms the LDPC-based approach in two aspects:

TABLE I  
MODELS FOR SIMULATION

model	$\{a_{00}, a_{11}, b_0(0), b_1(1)\}$
1	$\{0.01, 0.03, 0.99, 0.98\}$
2	$\{0.01, 0.065, 0.95, 0.925\}$
3	$\{0.97, 0.967, 0.93, 0.973\}$
4	$\{0.99, 0.989, 0.945, 0.9895\}$

TABLE II  
EXPERIMENTAL RESULTS

model	$H(X Y)$	[9]	DAC
1	0.24	0.36	0.345449
2	0.52	0.67	0.633043
3	0.45	0.58	0.583429
4	0.28	0.42	0.424827

1). The LDPC-based approach requires longer codes to achieve better performance, while the DAC is insensitive to code length [4].

2). For the LDPC-based approach, to synchronize the HMM, a certain proportion of original source symbols must be sent to the decoder as “seeds.” However, it is hard to determine  $\alpha$ , the optimal proportion of “seeds.” The results reported in [9] were obtained through an exhaustive search, which limits its application in practice.

*B. Application in image*

For this part of the experiment, we chose two binary images with the pixel value of 0 or 1, which can be used as binary source.

First, we encode a black-and-white text image in the Fig. 3(a), its size is 512\*512, therefore  $N = 512$ . Fig. 3(b) is the image after XOR operation. Like above, we set  $M = 2048$  and  $T = 15$ . In order to achieve lossless compression, therefore test starts from  $\gamma = H(X|Y)/H(X)$ , if the decoding fails, we increase  $\gamma$  with 0.01 until the decoding succeeds. Fig. 4 is the distribution of the overlap factor of each line when the image is completely correctly decoded. There can be seen a portion of the overlap factor is 0, which means that the pixels in these rows are all black or white, arithmetic coding.

When the overlap factor of each line is set to  $\gamma = H(X|Y)/H(X)$ , we compare with the Maximum A Posteriori(MAP) decoder [4], and the results are shown in Table III. We found that the frame-error-rate(FER) and symbol-error-rate(SER) of the DAC-based HMM method are lower.

Source coding is a conversion of purpose of improving the communication, or the conversion for purpose of reducing or redundancy. Specifically, it is to fit according to the statistical cha source output symbol sequence, to output symbol sequence into the sequence, so that the average an carried by each symbol of the latte at the same time, the original an

Source coding is a conversion of purpose of improving the communication, or the conversion for purpose of reducing or redundancy. Specifically, it is to fit according to the statistical cha source output symbol sequence, to output symbol sequence into the sequence, so that the average an carried by each symbol of the latte at the same time, the original an

(a)

(b)

Fig. 3. (a) is the original image before compression, (b) is the image subjected to XOR operation, which obtains the HMM parameter  $\lambda$  through the Baum-Welch algorithm.

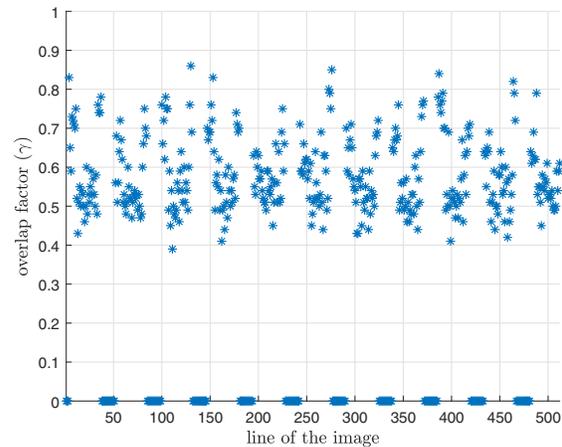


Fig. 4. distribution of the overlap factor.

TABLE III  
ERROR RATE OF IMAGE

method	value
HMM-SER	0.068264
MAP-SER	0.370422
HMM-FER	0.625
MAP-FER	0.712891

For the second test image, we chose an international standard test image. This paper is aimed at binary sources, so we binarized the image as shown in Fig. 5(a).

Different overlap factors are used in the test. In Fig. 5(b), when  $\gamma$  takes the minimum value  $H(X|Y)/H(X)$  of the range, the error rate is relatively high, but the person in the picture can basically be recognized. It can be seen from Fig. 5(b)-(d) that the larger the overlap factor, the higher the restoration degree of the picture, but at the same time the rate  $R = \gamma H(X)$  will also increase. Therefore, in actual applications, the value of the overlap factor can be set according to the need for data accuracy.

The above simulation results show that the image compression method proposed in this paper is feasible. Compared with the classic method MAP, the error rate is reduced a lot. In addition, the experimental results show that the overlap factor has a significant impact on the accuracy of image decoding, so we need to choose an appropriate overlap factor.

V. CONCLUSION AND FUTURE WORK

This paper researches the compression of sources with hidden Markov correlation using the DAC. The forward algorithm is incorporated into the DAC decoder. The results are similar to that of the LPDC-based approach. Compared to the LDPC-based approach, the DAC is more suitable for practical applications. Therefore, this paper applies it to image

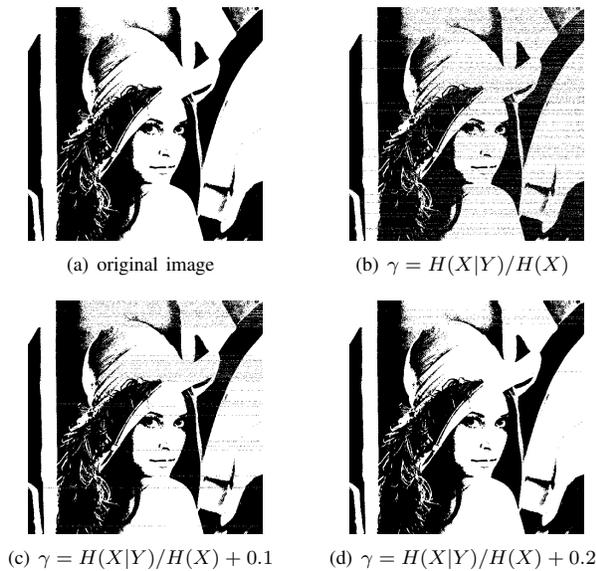


Fig. 5. (a) is the original image before compression, (b)-(d) is the image decoded by setting the value of different overlap factor  $\gamma$ .

compression. For decoder embedded with forward algorithm, it shows a lower error rate than MAP, and we analyze the effect of overlap factor on error rate.

This paper is aimed at binary sources, we intend to expand to multi-valued sources in the future. Moreover, HMM is

a widely used statistical model, which can be applied to modeling in many fields. The distributed arithmetic coding method based on HMM proposed in this paper can be applied to more fields.

REFERENCES

- [1] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Info. Theory*, vol. 19, no. 4, pp. 471-480, Jul. 1973.
- [2] J. Garcia-Frias, "Compression of Correlated Binary Sources Using Turbo Codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417-419, Oct. 2001.
- [3] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440-442, Oct. 2002.
- [4] M. Granelto, E. Magli, and G. Olmo, "Distributed Arithmetic Coding," *IEEE Commun. Lett.*, vol. 11, no. 11, pp. 883-885, Nov. 2007.
- [5] M. Granelto, E. Magli, and G. Olmo, "Distributed Arithmetic Coding for the Asymmetric Slepian-Wolf problem," *arXiv preprint arXiv:0712.0271*, 2007.
- [6] X. Artigas, S. Malinowski, C. Guillemot, and L. Torres, "Overlapped Quasi-Arithmetic Codes for Distributed Video Coding," *IEEE International Conference on Image Processing (ICIP)*, San Antonio, Texas, USA, Sep. 2007.
- [7] M. Granelto, E. Magli, and G. Olmo, "Symmetric Distributed Arithmetic Coding of Correlated Sources," *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Crete, Greece, Oct. 2007.
- [8] M. Granelto, E. Magli, R. Tron, and G. Olmo, "Rate-Compatible Distributed Arithmetic Coding," *IEEE Commun. Lett.*, vol. 12, no. 8, pp. 575-577, Aug. 2008.
- [9] J. Garcia-Frias and W. Zhong, "LDPC Codes for Compression of Multi-Terminal Sources With Hidden Markov Correlation," *IEEE Commun. Lett.*, vol. 7, no. 3, pp. 115-117, Mar. 2003.
- [10] Y. Fang, "DAC spectrum of binary sources with equally-likely symbols," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1584-1594, Apr. 2013.
- [11] Y. Fang and L. Chen, "Improved Binary DAC Codec with Spectrum for Equiprobable Sources," *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 256-268, January 2014, doi: 10.1109/TCOMM.2013.112613.120796.