GENERAL ARITHMETIC CODING FOR QUANTUM COMPRESSION

A PREPRINT

Cong Yu
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA 90007
congyu@usc.edu

Todd Brun
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA 90007
tbrun@usc.edu

December 13, 2023

ABSTRACT

Based on existed reversible quantum arithmetic compression of quantum Bernoulli source, a generalized compression procedure has been proposed to tackle any dimensional quantum sources. This technique is the hand compilation of classical arithmetic compression, which offers the possibility to compress more complex quantum source distribution as the classical counterpart. Using this procedure, the compression rate can nearly achieve the entropy of quantum source asymptotically while the expected fidelity achieves 1 asymptotically as well.

Keywords Arithmetic coding · Quantum computation · Quantum information theory · Quantum typical subspace measurement

1 Introduction

Quantum compression, which has been regarded as relatively more theoretical rather than practical, actually plays an important role just as the classical compression. The counterpart of classical Shannon compression – Schumacher's quantum compression theory [1, 2] has proved that the Von Neumann entropy of the quantum density operator is the theoretical lower bound of any viable compression scheme.

However, not much work about how to implement a practical one has been done. Some proposal like [3] studied how to compress a quantum Bernoulli source by reversible "arithmetic coding", some generic compression [4] tried to compress the quantum information source by compiling the classical correspondence block coding. As the statement from those literature, it's not hard to see the practical algorithm compressing a quantum memoryless Bernoulli source is pretty mature now, but the limitation is obvious too. The encoding is so special that can only be implemented in binary quantum source, which is really unlike the classical arithmetic coding being used in any D-ary sources. Furthermore, Schumacher compression does not constraint the dimension of quantum information source, hence theoretically, it offers the possibility to find the generalized compression scheme to tackle high dimensional quantum source. The main result of this paper is to adapt the classical arithmetic coding and decoding procedures [5] into the corresponding quantum version instead of reinventing a completely new scheme to address only specific situations.

2 Preliminaries

The basic idea of classical block compression is to construct a function which maps the original random variable with any distribution to a new random variable with uniform distribution, which means the new one is incompressible and forms the compressed version of the original information. Arithmetic coding, motivated by Shannon-Fano-Elias coding and the compressing mapping, can compress any sequence made of I.I.D. elements with a rate slightly higher than the entropy-theoretic limit. The basic scheme of classical arithmetic coding is to map a sequence to a subinterval in the $[0, 1)$, and tag the midpoint of the interval to represent the codeword.

For example, if now we have a M-ary alphabet $\{A_1, A_2 \ldots, A_m\}$, and the corresponding probability is $\{p_1, p_2, \ldots, p_m\}$. Then we can partition the unit interval $[0, 1)$ to $[0, p_1), [p_1, p_1 + p_2), \ldots, [\sum_{k=1}^{m-1} p_k, 1)$ n subintervals as the probability distribution. Each length of the subintervals equals to the corresponding probability. Input the primal sequence $\mathcal{X} = X_1 X_2 \ldots X_n$ composed of n elements from the M-ary alphabet, which can be regarded as a real random variable $0.X_1 X_2 \ldots X_n$. Initial states of the upper bound and lower bound are 1 and 0, then we check the members from X_1 to X_n one by one. Every time one symbol is determined we shall upgrade the upper bound and lower bound to those of the subinterval which represents the symbol. Then we reallocate the partitions in the new subinterval as the probability distribution again, and detect the next element from the primal sequence... Do the iteration until the last element is checked. Finally we will get a small subinterval representing the whole sequence, we choose the midpoint of the interval to be the codeword , then according to the Shannon-Fano-Elias coding theory, we can truncate the midpoint to the $\left\lceil \log \frac{1}{p(\mathcal{X})} \right\rceil + 2$ most important bits, which can also be losslessly decoded [5].

In the quantum compression case, we consider a generalized quantum information source. The quantum ensemble is $\{p_k, |\psi_k\rangle\}_k$. We are given a sequence of independent identically distributed pure-states from the ensemble, which is the primal information sequence needed to be compressed. Unlike the classical information sequence, the elements of the quantum sequence don't have to be orthogonal to each other, hence the goal of compression is different too. The quantum compression is to reduce the dimension of the Hilbert space while still containing most part of the information source to maintain the high fidelity between the final output sequence and the original one [6].

Consider the quantum information ensemble $\{p_k, |\psi_k\rangle\}_k$. Suppose those pure states in the ensemble can span to a d-dimension Hilbert space, then the density operator can be expressed as $\rho = \sum_k p_k |\psi_k\rangle \langle\psi_k|$, which can be eigendiagonalized as $\sum_{i=1}^{d} \lambda_i |x_i\rangle \langle x_i|$. We can expand the original sequence density operator as below:

$$\rho^{\otimes n} = (\sum_{i=1}^{d} \lambda_i |x_i\rangle \langle x_i|)^{\otimes n} = \sum_{m=1}^{d^n} p(\mathcal{X}_m) |\mathcal{X}_m\rangle \langle \mathcal{X}_m|$$

Where $|\mathcal{X}_m\rangle$ is one indexed permutation sequence composed of n eigenstates, and $p(\mathcal{X}_m)$ is the corresponding probability mass function, which equals to the product of the probabilities of each members. Define quantum random variable $|\mathcal{X}\rangle = |X_1\rangle \otimes |X_2\rangle \cdots \otimes |X_n\rangle$, each $|X\rangle$ is the random variable with the sample space $\{|x_i\rangle, \lambda_i\}_{i=1\ldots d}$. We can associate the sequence $|\mathcal{X}\rangle$ with the real random variable $0.X_1 X_2 \ldots X_n$, which can allow us use a fractional real number to represent the quantum state. By the property of tensor product, $\{\mathcal{X}\}_m$ are a set of basis of Hilbert space of the sequences, which means $\sum_{m=1}^{d^n} p(\mathcal{X}_m) |\mathcal{X}_m\rangle \langle \mathcal{X}_m|$ is the eigendecomposition of $\rho^{\otimes n}$. If we are able to compress the strong typical sequences, even we lose all the information of the atypical sequences during the compression, the fidelity can still remain high.

3 Quantum Arithmetic Coding and Decoding

In order to tackle general quantum source like general classical information source, we need to keep the essence of the scheme, hence we still adopt the similar encoding and decoding procedures and adapt them. The problem is the design of the adaption. The other reason offering the possibility of quantum arithmetic coding is that the relevant quantum arithmetic and scientific computing circuits and algorithms are well developed [3, 7, 8, 9, 10, 11, 12, 13, 14], also the relevant controlled quantum computation with different subsystem dimensions has been proposed[15, 16], we can adopt their models to construct more complex computing.

We will begin by introducing several modules which will be put together later. Those modules formulate the encoding and decoding circuits and algorithms.

3.1 "e" module

Consider the input sequence $|\mathcal{X}\rangle$, composed of $|X_1\rangle$, $|X_2\rangle \ldots |X_n\rangle$ I.I.D. random variables. Instead of using the upper bound and lower bound of the interval which represents the sequence to do the iteration, we use another 2 variables-one representing the midpoint and the other representing the probability of counted elements. Hence we need to change the iteration by some straightforward calculation.

Algorithm 1: Framework of "e" module

Input: The source sequence, $|\mathcal{X}\rangle$; The initial state of midpoint variable, $|C_{mid}\rangle = \left|\frac{1}{2}\right\rangle$; The initial state of probability variable, $|Prob\rangle = |1\rangle$

Output: The source sequence, $|\mathcal{X}\rangle$; The midpoint variable, $|C_{mid}\rangle$; The probability variable, $|Prob\rangle$;

for $i = 1$ *to* n **do**
  if $|X_i\rangle = |x_m\rangle$ **then**
    $|C_{mid}\rangle \otimes |Prob\rangle \leftarrow M_m(|C_{mid}\rangle \otimes |Prob\rangle)$;
final ;
return $|\mathcal{X}\rangle$, $|C_{mid}\rangle$ and $|Prob\rangle$;

Based on the essence of classical arithmetic coding, if We define the upper bound as H, lower bound as L. Then if the x_m is detected, straightforwardly according to the encoding procedure[5], the iteration should be:

$$\begin{bmatrix} H \\ L \end{bmatrix} \leftarrow \begin{bmatrix} \sum_{k=1}^{m} \lambda_k & \sum_{k=m+1}^{d} \lambda_k \\ \sum_{k=1}^{m-1} \lambda_k & \sum_{k=m}^{d} \lambda_k \end{bmatrix} \begin{bmatrix} H \\ L \end{bmatrix}$$

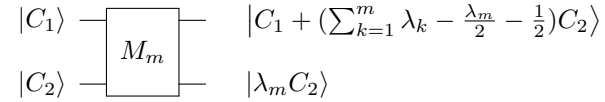
However, because we use a new pair of variables to replace the upper/lower bounds, the transformation is:

$$\begin{bmatrix} C_{mid} \\ Prob \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} H \\ L \end{bmatrix}$$

Therefore, the M_m matrix in the pseudocode is:

$$M_m = \begin{bmatrix} 0.5 & 0.5 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \sum_{k=1}^{m} \lambda_k & \sum_{k=m+1}^{d} \lambda_k \\ \sum_{k=1}^{m-1} \lambda_k & \sum_{k=m}^{d} \lambda_k \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 1 & -1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & (\sum_{k=1}^{m} \lambda_k - \frac{\lambda_m}{2} - \frac{1}{2}) \\ 0 & \lambda_m \end{bmatrix}$$

Hence the conditional quantum computing block is defined as:



By exploiting the block computing circuits above we can construct "e" module corresponding to the pseudocode to implement the encoding procedure:

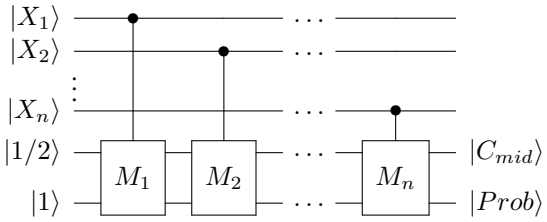


Figure 1: A quantum circuit implementing the Algorithm 1. The M is the same conditional computing block controlled by different elements of the sequence.

The final output $|C_{mid}\rangle$ is the midpoint of the interval representing the input sequence, and the output $|Prob\rangle$ is the total probability of the sequence.

3.2 Strong Typical Measurement

One of the major distinctions between classical and quantum compression is that a fixed classical compression procedure can hardly compress any sequence, especially the atypical ones, but the probability of atypical sequences only occupy a small proportion, hence on average the compression can be achieved. However, a quantum sequence which can be regarded as a superposition of "classical sequences", encompasses nearly all the possible basis of the Hilbert space, we have to compress the sequences in the typical subspace and discard the atypical part.

For convenience we will exploit the quantum method of types to choose the typical quantum sequences[6]. By the definition of strong quantum typicality, each quantum type class corresponds to one unique probability, which means the map between quantum type class and probability is one-to-one. Hence if we sum all the typical type class projectors together, we can construct a strong typical subspace projector, then we can implement strong typical measurement. We will input the probability quantum variable got from "e" module, and also define a new index variable $|I_\tau(\mathcal{X})\rangle$ to indicate whether the quantum sequence $|\mathcal{X}\rangle$ is in the strong typical subspace. The pseudocode of the operations is below.

Algorithm 2: Framework of strong typical measurement

Input: The source sequence, $|\mathcal{X}\rangle$; The index variable in initial state, $|I_\tau(\mathcal{X})\rangle = |0\rangle$; The probability variable, $|Prob\rangle$;
Output: The source sequence, $|\mathcal{X}\rangle$; The index variable, $|I_\tau(\mathcal{X})\rangle$; The probability variable in initial state, $|Prob\rangle = |1\rangle$;

1 **if** $Prob \geq \tau$ **then**
2 $\quad |I_\tau(\mathcal{X})\rangle \leftarrow |I_\tau(\mathcal{X}) \oplus 1\rangle$;
3 **for** $i = 1$ *to* n **do**
4 $\quad$ **if** $|X_i\rangle = |x_m\rangle$ **then**
5 $\quad\quad |Prob\rangle \leftarrow |Prob \div \lambda_m\rangle$
6 **final** ;
7 **return** $|\mathcal{X}\rangle$, $|I_\tau(\mathcal{X})\rangle$ and $|Prob\rangle = |1\rangle$;

The standard we choose the types is to determine a threshold τ. The probabilities greater than the threshold are what we define as typical sequences. The reason why we only set the lower bound here is that the greater the probability is, the less codeword bits we need to recover the sequence according to the Shannon-Fano-Elias coding theory[5], hence there's no need to set the upper bound exactly as the definition of strong typicality.

After resetting the probability variable, the entanglement between probability variable and others has been removed. Then we can measure the index variable, if the value is $|1\rangle$, then the sequences are wanted, otherwise we discard it. By the definition of strong typicality, the probability is near to 1 while n is really large[6]. Then we can assume the measurement is successful, and assume the sequence is typical and design the encoding/decoding schemes.

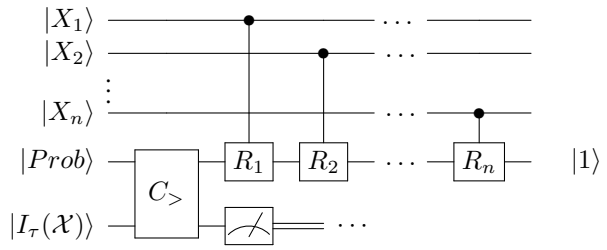


Figure 2: A quantum circuit implementing the Algorithm 2. The $C_>$ implements the lines 1-2, R implements the lines 3-5.

3.3 Framework of "d" module

We have to remove the entanglement between the midpoint variable and the input sequence after "e" module. Here we borrowed the idea from [4], by transforming the classical decoding procedure to the quantum version, we can do the entanglement removal. Here is the pseudocode:

There are two reasons we cannot access the exact value of the output of the midpoint variable. The first one is that quantum multiplication and addition can get the exact result by fixed point form, but the division will generate some errors[7], and it can varies depending on which division algorithms is applied here. The other reason is the intrinsic uncertainty about the final value of the codeword from the classical arithmetic decoding procedure. We can only estimate the value is approximately $\frac{1}{2}$ but later we will explain that the exact value doesn't matter and only is a intermediate state of the midpoint codeword.

Here is the specific circuit implementation of the "d" module:

Algorithm 3: Framework of "d" module

Input: n initial quantum registers, $\left|0_{[n]}\right\rangle$; The midpoint variable, $|C_{mid}\rangle$;

Output: Intermediate state of midpoint variable, $\left|\approx \frac{1}{2}\right\rangle$; The decoded sequence, $|\mathcal{X}\rangle$;

1 **Define** $\sum_{k=1}^{0} \lambda_k = 0$, $\sum_{k=1}^{d} \lambda_k = 1$;
2 **for** $i = 1$ *to* n **do**
3 **find** m **that** $\sum_{k=1}^{m-1} \lambda_k \leq C_{mid} \leq \sum_{k=1}^{m} \lambda_k$;
4 **do**: $|0_i\rangle \leftarrow |x_m\rangle$;
5 **if** $|X_i\rangle = |x_m\rangle$ **then**
6 $|C_{mid}\rangle \leftarrow \left|(C_{mid} - \sum_{k=1}^{m-1} \lambda_k) \div \lambda_m\right\rangle$
7 **final** ;
8 **return** $\left|\approx \frac{1}{2}\right\rangle$, $|\mathcal{X}\rangle$;

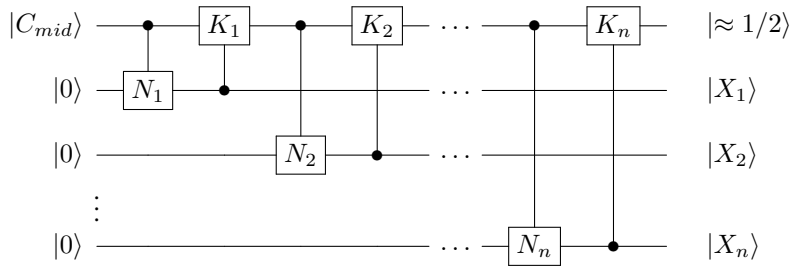


Figure 3: A quantum circuit implementing the Algorithm 3. The N implements the lines 3-4 representing the conditional value assignment on the $\left|0_{[n]}\right\rangle$, K implements the lines 5-6 representing conditional iteration applied on C_{mid} itself.

3.4 Scheme of the "E" encoder

It's time to put the modules together with all the preparation above. With the circuit implementation we can remove the

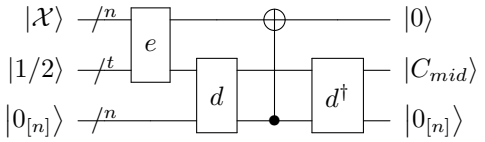


Figure 4: The quantum circuit combining the modules above to remove the entanglement between the codeword and the input sequence.

entanglement between the midpoint variable and any other variables by resetting them to the fixed values. In this way we can only keep the midpoint and discard all the others. As we can see, there is no need to know the exact value of intermediate state, which will be recovered to C_{mid} again. We are able to construct the encoder block now with t digit accuracy.

The last problem remaining to solve is to truncate the midpoint to k most important digits because at last our goal is to attain the compressed quantum sequence–the truncated midpoint(t is usually really big to maintain high computational accuracy), and there is already successful scheme for the practical quantum truncation circuits. What we have now are two encoding/decoding computing blocks:

The "D" decoder is exactly made by literally running the encoder in reverse: $D = E^\dagger$. Hence can we directly take the same scheme from [3] to truncate the midpoint variable. By this way we will finally complete the quantum arithmetic encoder, and decoder can be attained straightforwardly by the reverse.

References

[1] Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738, 1995.

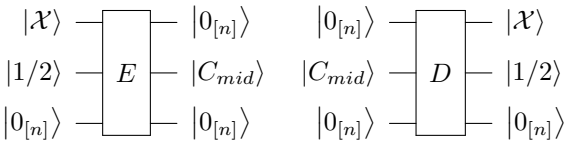


Figure 5: The schematic symbols for the encoder/decoder circuits

[2] Richard Jozsa and Benjamin Schumacher. A new proof of the quantum noiseless coding theorem. *Journal of Modern Optics*, 41(12):2343–2349, 1994.

[3] Isaac L Chuang and Dharmendra S Modha. Reversible arithmetic coding for quantum data compression. *IEEE Transactions on Information Theory*, 46(3):1104–1116, 2000.

[4] John Langford. Generic quantum block compression. *Physical Review A*, 65(5):052312, 2002.

[5] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[6] Mark M Wilde. From classical to quantum shannon theory. *arXiv preprint arXiv:1106.1445*, 2011.

[7] Mihir K Bhaskar, Stuart Hadfield, Anargyros Papageorgiou, and Iasonas Petras. Quantum algorithms and circuits for scientific computing. *arXiv preprint arXiv:1511.08253*, 2015.

[8] David Beckman, Amalavoyal N Chari, Srikrishna Devabhaktuni, and John Preskill. Efficient networks for quantum factoring. *Physical Review A*, 54(2):1034, 1996.

[9] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004.

[10] Thomas G Draper. Addition on a quantum computer. *arXiv preprint quant-ph/0008033*, 2000.

[11] Eleanor G Rieffel and Wolfgang H Polak. *Quantum computing: A gentle introduction*. MIT Press, 2011.

[12] Yasuhiro Takahashi and Noboru Kunihiro. A linear-size quantum circuit for addition with no ancillary qubits. *Quantum Information & Computation*, 5(6):440–448, 2005.

[13] Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. Quantum addition circuits and unbounded fan-out. *arXiv preprint arXiv:0910.2530*, 2009.

[14] Vlatko Vedral, Adriano Barenco, and Artur Ekert. Quantum networks for elementary arithmetic operations. *Physical Review A*, 54(1):147, 1996.

[15] Jamil Daboul, Xiaoguang Wang, and Barry C Sanders. Quantum gates on hybrid qudits. *Journal of Physics A: Mathematical and General*, 36(10):2525, 2003.

[16] Gerardo A Paz-Silva, Gavin K Brennen, and Jason Twamley. Globally controlled universal quantum computation with arbitrary subsystem dimension. *Physical Review A*, 80(5):052318, 2009.