

On Computation of Lower Bound of E Capacity for the Channel with Two-Sided State Information

Arthur R. Muradyan

Institute for Informatics and Automation Problems of NAS RA

Abstract

In this paper a discrete memoryless channel (DMC) with 2 sided state information is examined in the sense of computations of capacity formulas and random-coding exponents. It is shown that computation of channel capacity and lower bound of E capacity requires a huge amount of computational resources and is rather slow when implemented with traditional methods. Techniques are described how to massively parallelize the computations by means of GPGPU and obtain substantial acceleration.

1. Introduction

Recent years channels with state information are actively evolving. Coming from applications various cases are being developed depending on the state information availability at the encoder, decoder, or on both ends. These data transmission models have found applications in digital watermarking, information hiding, wireless communications.

The generalization of the channel with state information, where the sender and the receiver have correlated but different state information (Figure 1), are studied in [1, 2]. The channel capacity for the generalized model is obtained in [1]. The E -capacity is studied and lower bound (random coding bound) of the E -capacity is derived in [2].

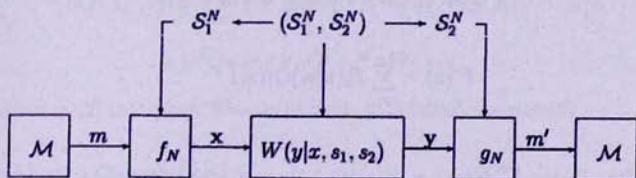


Figure 1. Channel with two-sided state information

The computation of capacity and lower bound of E -capacity requires a huge amount of computational resources due to immense number of possible probabilities. Traditional sequential implementation running on a modern processor is rather slow. Parallelization on multiple processors may cut the execution time ideally by factor of N , where N is the number of used processors.

Graphics processing unit (GPU) [3, 4] technology is chosen as a computational fabric because it contains a huge amount of microprocessors. The latest generation of graphics hardware contains couple of GPUs. As a result modern graphics hardware contains more than 450 microprocessors which helps to achieve significant acceleration.

The paper is organized as follows. Definitions of basic information theoretic quantities needed for computations are mentioned in the section 2. The description of the computational complexity and memory size needed for computation of lower bound of E -capacity is outlined in section 3. More description about chosen solution and implementation details can be found in section 4.

2. Notations and Definitions

In the paper capital letters are used for random variables (RV) S_1, S_2, U, X, Y taking values in the finite sets $S_1, S_2, \mathcal{U}, \mathcal{X}, \mathcal{Y}$, correspondingly, and lower case letters s_1, s_2, u, x, y for their realizations. Small bold letters are used for N -length vectors $\mathbf{x} = (x_1, \dots, x_N) \in \mathcal{X}^N$. The cardinality of the set \mathcal{X} we denote by $|\mathcal{X}|$. The notation $|a|^+$ will be used for $\max(a, 0)$.

The channel we consider is defined by a transition probability matrix $W(y|x, s_1, s_2)$, where x accepts values from input alphabet \mathcal{X} and y accepts values from output alphabet \mathcal{Y} . The state information is described by a pair of RVs (S_1, S_2) with given joint probability distribution (PD) $Q^* = Q_1^* \circ Q_2^* = \{Q^*(s_1, s_2) = Q_1^*(s_1)Q_2^*(s_2|s_1), s_1 \in S_1, s_2 \in S_2\}$. N -length sequences \mathbf{s}_1 are available to the encoder and \mathbf{s}_2 - to the decoder.

We use the following PDs:

$$Q = Q_1 \circ Q_2 = \{Q(s_1, s_2|u, x) = Q_1(s_1)Q_2(s_2|u, x, s_1), s_1 \in S_1, s_2 \in S_2, u \in \mathcal{U}, x \in \mathcal{X}\}.$$

$$P = P_0 \circ P_1 = \{P(u, x|s_1) = P_0(u|s_1)P_1(x|u, s_1), s_1 \in S_1, u \in \mathcal{U}, x \in \mathcal{X}\}.$$

$$V = \{V(y|u, x, s_1, s_2), s_1 \in S_1, s_2 \in S_2, u \in \mathcal{U}, x \in \mathcal{X}, y \in \mathcal{Y}\}.$$

$$\begin{aligned} Q \circ P \circ V &= \{Q(s_1, s_2|u, x)P(u, x|s_1)V(y|u, x, s_1, s_2) = \\ &= Q_1(s_1)P_0(u|s_1)P_1(x|u, s_1).Q_2(s_2|u, x, s_1)V(y|u, x, s_1, s_2), \\ &s_1 \in S_1, s_2 \in S_2, u \in \mathcal{U}, x \in \mathcal{X}, y \in \mathcal{Y}\}. \end{aligned}$$

Let

$$P'(u) = \sum_{s_1} P_0(u|s_1)Q_1(s_1) \quad (1)$$

and

$$P''(s_2, y|u) = \sum_{x, s_1} V(y|u, x, s_1, s_2)Q_2(s_2|u, x, s_1)Q_1(s_1)P_1(x|u, s_1). \quad (2)$$

The mutual information $I_{Q_1, P_0}(U \wedge S_1)$ is defined in the following way:

$$I_{Q_1, P_0}(U \wedge S_1) = \sum_{u, s_1} Q_1(s_1)P_0(u|s_1) \log \frac{P_0(u|s_1)}{P'(u)} \quad (3)$$

and $I_{Q, P, V}(U \wedge S_2, Y)$ is defined like:

$$I_{Q, P, V}(U \wedge S_2, Y) = \sum_{u, s_2, y} P'(u)P''(s_2, y|u) \log \frac{P''(s_2, y|u)}{\sum_{u'} P'(u')P''(s_2, y|u')}. \quad (4)$$

All logarithms and exponents in the paper are of the base 2.

The divergence $D(Q \circ P \circ V || Q^* \circ P \circ W)$ is defined as:

$$D(Q \circ P \circ V || Q^* \circ P \circ W) = \sum_{s_1, s_2, u, x, y} Q(s_1, s_2 | u, \tau) P(u, \tau | s_1) V(y | u, \tau, s_1, s_2) \times \\ \times \log_2 \frac{Q(s_1, s_2 | u, x) V(y | u, x, s_1, s_2)}{Q^*(s_1, s_2) W(y | x, s_1, s_2)}. \quad (5)$$

The channel capacity for the considered communication scheme is obtained in [1] and is equal to:

$$C(Q^*, W) = \max_P [C'(Q^*, W, P)] \quad (6)$$

with

$$C'(Q^*, W, P) = I_{Q^*, P, W}(U \wedge S_2, Y) - I_{Q^*, P_0}(U \wedge S_1). \quad (7)$$

Note that with PD Q^* and W (1) and (2) will become

$$P'(u) = \sum_{s_1} P_0(u | s_1) Q_1^*(s_1)$$

and

$$P''(s_2, y | u) = \sum_{x, s_1} W(y | x, s_1, s_2) Q_2^*(s_2 | s_1) Q_1^*(s_1) P_1(x | u, s_1).$$

Lower bound of E -capacity obtained in [2] is

$$R_+(E, Q^*, W) = \min_{Q_1} \max_P \min_{Q_2, V: D(Q \circ P \circ V || Q^* \circ P \circ W) \leq E} |R(E, Q^*, W, Q, P, V)|^+ \quad (8)$$

with

$$R(E, Q^*, W, Q, P, V) = I_{Q, P, V}(U \wedge S_2, Y) - I_{Q_1, P_0}(U \wedge S_1) + \\ + D(Q \circ P \circ V || Q^* \circ P \circ W) - E. \quad (9)$$

It is shown in [2] that when $E \rightarrow 0$ (8) tends to channel capacity (6).

3. Complexity Calculation

Now we evaluate how much computational resource is needed to find out the values of capacity (6) and lower bound of E -capacity (8) for given inputs. From the definitions we can see that the complexity of (7) and (9) is of the order

$$O(|S_1||\mathcal{U}|) + O(|\mathcal{X}||S_1|) + O(2|\mathcal{U}||S_2||\mathcal{Y}|)$$

and

$$O(|S_1||\mathcal{U}|) + O(|\mathcal{X}||S_1|) + O(2|\mathcal{U}||S_2||\mathcal{Y}|) + O(|S_1||S_2||\mathcal{U}||\mathcal{X}||\mathcal{Y}|)$$

correspondingly.

Expression (7) should be computed for each possible P and (9) should be computed for all possible P, Q_1, Q_2, V PDs. For given S_1 and Q_1 we have $Q_1(s_1) \in [0, 1]$ for all $s_1 \in S_1$ and

$$\sum_{s_1} Q_1(s_1) = 1.$$

The interval $[0, 1]$ should be discretized in order to get finite number of possible PDs. We can divide the interval $[0, 1]$ into segments of equal size and suppose that $Q_1(s_1) \in \{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\}$.

If we regard $Q_1(s_1) = \frac{a_i}{n}$, where a_i are natural numbers and $i = 1, \dots, |S_1|$ then we have

$$\sum_{i=1}^{|S_1|} \frac{a_i}{n} = 1.$$

which is equivalent to

$$\sum_{i=1}^{|S_1|} a_i = n.$$

The number of possible solutions for the above mentioned equation represents the number of possible Q_1 PDs which is equal to:

$$\binom{n + |S_1| - 1}{|S_1| - 1}.$$

For fixed s_1 we have

$$\sum_{u,x} P(u, x|s_1) = 1.$$

So the number of possible $P(u, x|s_1)$ for fixed s_1 is equal to $\binom{n + |\mathcal{U}||\mathcal{X}| - 1}{|\mathcal{U}||\mathcal{X}| - 1}$. The number of possible $P(u, x|s_1)$ will be $\binom{n + |\mathcal{U}||\mathcal{X}| - 1}{|\mathcal{U}||\mathcal{X}| - 1}^{|S_1|}$. Hence we get the following complexity for the capacity

$$\left(\frac{n + |\mathcal{U}||\mathcal{X}| - 1}{|\mathcal{U}||\mathcal{X}| - 1} \right)^{|S_1|} [O(|S_1||\mathcal{U}|) + O(|\mathcal{X}||S_1|) + O(2|\mathcal{U}||S_2||\mathcal{Y}|)]. \quad (10)$$

Using the same reasonings we get that the number of possible $Q_2(s_2|u, x, s_1)$ and $V(y|u, x, s_1, s_2)$ and the complexity of lower bound of E -capacity becomes:

$$\left(\frac{n + |S_1| - 1}{|S_1| - 1} \right) \left(\frac{n + |\mathcal{U}||\mathcal{X}| - 1}{|\mathcal{U}||\mathcal{X}| - 1} \right)^{|S_1|} \left(\frac{n + |S_2| - 1}{|S_2| - 1} \right)^{|\mathcal{U}||\mathcal{X}||S_1|} \left(\frac{n + |\mathcal{Y}| - 1}{|\mathcal{Y}| - 1} \right)^{|\mathcal{U}||\mathcal{X}||S_1||S_2|} \\ [O(|S_1||\mathcal{U}|) + O(|\mathcal{X}||S_1|) + O(2|\mathcal{U}||S_2||\mathcal{Y}|) + O(|S_1||S_2||\mathcal{U}||\mathcal{X}||\mathcal{Y}|)]. \quad (11)$$

To get more insight about the magnitude of complexities we can consider a simple example when smallest sets are taken $|S_1| = |S_2| = |\mathcal{U}| = |\mathcal{X}| = |\mathcal{Y}| = 2$. The computation of the capacity for this example requires approximately $24(n+1)^6$ operations and computation of the lower bound of E -capacity requires approximately $56(n+1)^{31}$ operations.

Table 1: Performance Benchmarks

Architecture	Frequency	CPUs	Technology	Time
Quad-Core AMD Phenom II	3200MHz	1	Regular C++	28.8h
Quad-Core AMD Phenom II (1 node)	3200MHz	4	POSIX Threads	6.13h
Intel Xeon (Arm Cluster) (4 nodes)	3100MHz	8	MPI	11.2h
Quad-Core AMD Opteron (4 nodes)	2200MHz	16	POSIX Threads	2.67h
GeForce GTX 295 (1 GPU)	1240MHz	240	CUDA	32min
GeForce GTX 295 (2 GPUs)	1240MHz	480	CUDA	16min
2 x GeForce GTX 295 (4 GPUs)	1240MHz	960	CUDA	8.5min

4. Description of Results

As outlined in previous section the number of operations needed for computations of capacity formulas is dependant on partitioning level and RVs set sizes and even for simple cases vast amount of computational operations are required. The sequential implementation is extremely slow even for modern CPUs. The computations should be massively parallelized to gain results in a reasonable time frame.

From the formulas (6), (8) we see that there is no data dependency [5] between calculations of (7), (9) therefore the computations can be done concurrently.

To compute the capacity and lower bound of E -capacity software applications with various architectures and technologies are implemented. The execution time of $R_r(E, Q^*, W)$ for fixed E , Q^* and W for the examined architectures/technologies is reflected in the table 1.

The GPUs are found to be the fastest due to their numerous quantities of microprocessors. In fact, number of microprocessors in GPUs is growing at an extraordinary rate and over the last decade the processing power of GPUs has been growing at a rate faster than Moore's Law [6], which governs the performance growth rate of CPUs.

Previously the number of operations that could be performed on the GPU was limited to certain fixed functions. Then GPUs have evolved to perform more and more operations. The increase in data accuracy, sufficient number operations combined with the increased programmability have lead GPUs moving towards a more general purpose processor design [7, 8].

GPUs allow efficient utilization on single instruction multiple data (SIMD) [9] architectures. The capacity computations can be viewed as SIMD type of computation, where (7), (9) is the instruction which is calculated for multiple possible PDs. The results of (7), (9) can be compared also in parallel by applying reduction principles [10]. The comparison can also be treated as SIMD type application, where comparison operation is the instruction and multiple data are the values of (7), (9).

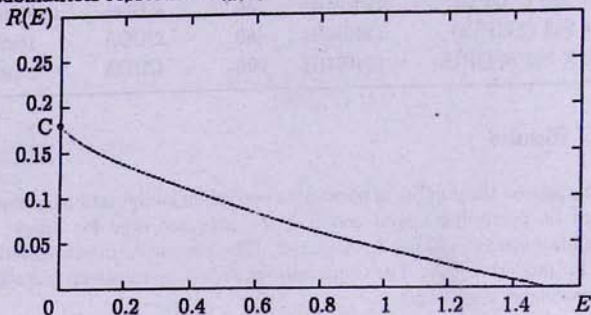
As an example, we have computed values of $C(Q^*, W)$ and $R_r(E, Q^*, W)$, when Q^* is

$$s_1/s_2 \quad \begin{matrix} 0 & 1 \\ 0 & \begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{pmatrix} \\ 1 & \end{matrix}$$

and $W(y|x, s_1, s_2)$ is

$$\begin{matrix} s_1/s_2 & 0 & 1 \\ 0 & \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} & \begin{pmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{pmatrix} \\ 1 & \begin{pmatrix} 0.2 & 0.8 \\ 0.7 & 0.3 \end{pmatrix} & \begin{pmatrix} 0.7 & 0.3 \\ 0.9 & 0.1 \end{pmatrix} \end{matrix},$$

where submatrices represent $\tilde{W}(y|x)$.



The drawing above illustrates the graph of $R_r(E, Q^*, W)$ for mentioned Q^*, W . When $E \rightarrow 0$ the transmission rate converges to the capacity and when $E = 1.5$ the transmission rate becomes equal to zero.

References

- [1] T. M. Cover and M. Chiang, "Duality between channel capacity and rate distortion with two-sided state information", *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1629-1638, 2002.
- [2] M. Haroutunian, A. Muradyan, "Lower bound for E capacity of discrete memoryless channel with two-sided state information", *Transactions of the Institute for Informatics and Automation Problems of the NAS of RA, Mathematical Problems of Computer Sciences*, vol. 31, pp. 28-39, 2008.
- [3] D. Luebke, G. Humphreys, "How GPUs work", *IEEE Computer*, 2007.
- [4] R. Wang, N. Goodnight, G. Humphreys, "Computation on programmable graphics hardware", *IEEE Computer Graphics and Applications*, 2005.
- [5] J. Hennessy, D. Patterson, "Computer architecture: a quantitative approach", ISBN 1-55860-724-2, 3rd edition, 2003.
- [6] G. Moore, "Cramming more components onto integrated circuits", *Electronics*, vol. 38, no 8, 1965.
- [7] J. Owens, D. Luebke, "A survey of general-purpose computation on graphics hardware", *Computer Graphics Forum*, vol. 26, no 1, pp. 80-113, 2007.
- [8] M. Harris, "Mapping computational concepts to GPUs", *International Conference on Computer Graphics and Interactive Techniques*, no 50, 2005.

- [9] D. Ibaroudene, "Parallel processing", Chapter 1, Motivation and History. St Mary's University, San Antonio, 2008.
- [10] T. Cormen, C. Leiserson, R. Rivest, C. Stein, "Introduction to algorithms", ISBN 978-0-262-03293-3. MIT Press. 2001

***E-ունակության ստորին գնահատականի հաշվարկումը
երկկողմանի վիճակներով կապուղիների համար***

Արթուր Մուրադյան

Ամփոփում

Աշխատանքը նվիրված է ընդհատ առանց հիշողության երկկողմանի վիճակներով կապուղիի համար նախկինում ստացված արդյունքների հաշվարկմանը: Հայտնի ունակության բանաձևի և *E*-ունակության պատահական կողավորման գնահատականի հաշվարկման համար պահանջվում է կատարել հսկայական քանակով գործողություններ: Հաշվարկների ժամանակը էապես կրճատելու նպատակով հետազոտվել են տարբեր միջավայրեր և առաջարկվել է մեթոդ գործողությունները հնարավորինս զուգահեռացնելու համար: Յույց է տրվել, որ նմանատիպ հաշվարկների համար գրաֆիկական սարքերի օգտագործումը տալիս է առավելություն այլ միջոցների նկատմամբ: Համեմատությունները կատարվել են օրինակի վրա և հաշվարկները գրաֆիկորեն պատկերվել են: