A Simple Quantitative Model of Cognitive Tradeoff Phenomenon

Griselda Acosta[1], Eric Smith[2], and Vladik Kreinovich[3]
[1]Department of Electrical and Computer Engineering
[2]Department of Industrial, Manufacturing, and Systems Engineering
[3]Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
gvacosta@miners.utep.edu, esmith2@utep.edu, vladik@utep.edu

Abstract

A recent study of chimpanzees has shown that on the individual basis, they are, surprisingly, much better than humans in simple tasks requiring intelligence and memory. A usual explanation – called cognitive tradeoff – is that a human brain has sacrificed some of its data processing (computation) abilities in favor of enhancing the ability to communicate; as a result, while individual humans may not be as smart as possible, jointly, we can solve complex problems. A similar cognitive tradeoff phenomenon can be observed in computer clusters: the most efficient computer clusters are not formed from the fastest, most efficient computers, they are formed from not-so-fast computers which are, however, better in their communication abilities than the fastest ones. In this paper, we propose a simple model that explains the cognitive tradeoff phenomenon.

1 Formulation of the Problem

Interesting empirical phenomenon. A recent study of chimpanzees [1, 2, 4] showed, somewhat surprisingly, that on the individual basis, they are much better than human in many tasks requiring intelligence. For example, they can remember more objects in images, and in conflict situations their behavior is much closer to the optimal behavior (as recommended by game theory) than the behavior of humans.

Cognitive tradeoff: an explanation for this phenomenon. A current explanation for this phenomenon is based on what is called *cognitive tradeoff*: humans have better communication abilities, and so, human brain has to sacrifice some individual intellectual abilities to leave space for parts needed for efficient communication.

The need for such a tradeoff is not limited to humans. A similar tradeoff phenomenon can be observed not only in humans, but in computers as well. The world's fastest computations are performed on so-called high performance computers. Each of them is, in effect, a large number of processors constantly communicating with each other.

In principle, there exist processors which are very fast and efficient, but modern super-computers are not formed from these processors: they are formed from simpler processors – similar to the ones we use in not-very-expensive home computers. One of the reasons for this choice is that these simple processors communicate well, as opposed to more efficient processors; these more efficient processors individually perform better but which take much longer time to communicate (another reason is that simple processors are usually much cheaper, which allows the designers to combine many more such processors within the same budget).

The ubiquity of cognitive tradeoff motivates the desired to have a universal quantitative model. The fact that cognitive tradeoff occurs in many situations, from human to computer communications, shows that there must be a simple quantitative explanation for this phenomenon.

In this paper, we provide a simple quantitative model that explains the main ideas behind this phenomenon. We hope that this simple model can be used as a basis for more complex – and more realistic – models that would not only qualitatively explain this phenomenon, but that would also lead to quantitative predictions.

2 Description of a Model

Main idea behind the model. We have a computing device – be it a computer or a brain – that is involved in communication with other computing devices so that together, they can solve a certain important problem.

The main difficulty with communication is that we cannot just send the internal signals out. It does not work for humans: we sometimes do not even understand each other's gestures or words, we need to translate our knowledge from our internal knowledge-representation language to a more universal one. Similarly, computers cannot just send out signals describing 0s and 1s that serve as internal representations of the corresponding knowledge: even if the two computers use the same way of representing, e.g., arrays of real numbers, the actual representation includes the information on where exactly this arrays is stored in the computer memory – the information that is useless for the computer that receives this information.

So, in general, to communicate, computing devices need to translate their internal signals into a different, more universal communication language. For this translation, we need a dictionary stored in the computing device.

In computing devices, usually, there are several levels of information storage. There is an operating memory where access to information is fast but the size of this memory is limited. There is usually a much larger second-tier memory that can store a much larger amount of information but where access takes much longer. There are usually several more layers, but in this paper, for simplicity, we will simply assume that we have two memory layers.

Details. Let a denote the overall computational ability related to the top (fastest-to-access) memory level. Some part of this level memory is taken by the most frequent "words" in the dictionary – so that translation of these words and thus, sending a message would go faster. Let a_0 denote the part of this level memory that is focused on this translation; then, we have $a - a_0$ ability remaining for general computations.

Let us denote by t_0 the part of the memory that is needed, on average, to store a translation of one word. Then, in the part a_0, we can store the translations of $w \stackrel{\text{def}}{=} \dfrac{a_0}{t_0}$ words.

Let us assume that we need:

- to perform some fast computations – whose overall running time will be denoted by C – and
- to send several (M) messages (of average length of ℓ words per message); this means that overall, in addition to computations, we need to communicate $W = M \cdot \ell$ words.

Let d be the size of the dictionary, i.e., the overall number of words that can be used for communication.

In this arrangement, what is the best division of top layer memory a into parts a_0 and $a - a_0$ under which both computation and communication tasks will be performed as fast as possible?

Zipf's law. In our analysis, we will rely on the known law that describes how frequently different words appear in a message. According to this law – known as Zipf's law – if we sort all the words from a dictionary in the decreasing order of their frequency, then the frequency f_i with which the i-word appears is equal to $f_i \approx \dfrac{c}{i}$, for some constant c; see, e.g., [3].

The constant c can be determined from the condition that the sum of all the frequencies $f_1, \ldots, f_d$ should be equal to 1. Thus, we get

$$\frac{c}{1} + \frac{c}{2} + \ldots + \frac{c}{d} = 1,$$

i.e., equivalently,

$$c \cdot \left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{d}\right) = 1.$$

The sum in parentheses is an integral sum for the integral

$$\int_1^d \frac{1}{x}\, dx = \ln(x)|_1^d = \ln(d) - \ln(1) = \ln(d),$$

thus

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{d} \approx \ln(d),$$

hence $c \cdot \ln(d) = 1$, so $c = \dfrac{1}{c} = \dfrac{1}{\ln(d)}$ and

$$f_i = \frac{1}{\ln(d)} \cdot \frac{1}{i}.$$

Towards formulas for computation and communication times. We have $a - a_0$ elementary computational devices to perform the overall amount C of needed computations. So, if we distribute these computation tasks between these $a - a_0$ devices, then we need the time

$$\frac{C}{a - a_0}$$

to perform all these computations.

Let us now estimate the amount of computations needed to send all M needed messages. In the fast memory layer, we can store w words. To speed up computations, it is reasonable to store, in the fast memory, translations to w most frequent words. If a message contains other words, we need to spend some time either computing its translation, or, alternatively, bringing this translation from the slower memory layer. Let us denote the average time needed to translate a not-stored-in-fast-memory word by t.

Among all $W = M \cdot \ell$ words that we need to communicate, we need the translate for all the words except for the w most frequent ones, i.e., for all the words whose frequencies are $f_{w+1}, \ldots, f_d$. The overall frequency f of all such words can be obtained by adding up all these frequencies; so, we get

$$f = f_{w+1} + \ldots + f_d = \frac{c}{w+1} + \ldots + \frac{c}{d} = c \cdot \left(\frac{1}{w+1} + \ldots + \frac{1}{d} \right).$$

The sum in the last expression is also an integral sum, this time for the integral

$$\int_{w+1}^{d} \frac{1}{x}\, dx = \ln(x)|_{w+1}^{d} \approx \ln(d) - \ln(w).$$

Thus, the frequency f is approximately equal to

$$f = c \cdot (ln(d) - \ln(w)) = \frac{\ln(d) - \ln(w)}{\ln(d)}.$$

Among all W words, we thus need to spend time on $f \cdot W$ words. Translating each word requires time t, so overall, we need to spend time $f \cdot W \cdot t$ on this translation.

Substituting the above expression for f and the formula $W = M \cdot w_0$ into this formula, we conclude that the overall time for sending M messages is equal to

$$\frac{\ln(d) - \ln(w)}{\ln(d)} \cdot M \cdot w_0 \cdot t,$$

i.e., taking into account that $w = a_0/t_0$ and thus, $\ln(w) = \ln(a_0) - \ln(t_0)$, we get

$$\frac{\ln(d) + \ln(t_0) - \ln(a_0)}{\ln(d)} \cdot M \cdot w_0 \cdot t.$$

By adding the computation and communication time, we get the following formula for the overall time.

Resulting formula for overall computation and communication time. The overall time T needed for computation and communication is equal to

$$\frac{C}{a - a_0} + \frac{\ln(d) + \ln(t_0) - \ln(a_0)}{\ln(d)} \cdot M \cdot w_0 \cdot t. \tag{1}$$

3 Analysis of the Model: What Is the Optimal Tradeoff Between Computation and Communication

Main idea. The desired tradeoff is described by the parameter a_0. We want to find the value of this parameter for which the overall time T needed to perform all the tasks (including both computation and communication) is the smallest possible. In other words, the expression (1) for this time T is our objective function.

Towards an explicit expression for the optimal value a_0. To find the optimal value a_0, let us differentiate the objective function (1) with respect to a_0 and equate the derivative to 0. As a result, we get the following formula:

$$\frac{C}{(a - a_0)^2} - \frac{M \cdot w_0 \cdot t}{\ln(d)} \cdot \frac{1}{a_0} = 0.$$

Multiplying both sides of this equality by $(a - a_0)^2 \cdot a_0$, we get a quadratic equation:

$$C \cdot a_0 - \frac{M \cdot w_0 \cdot t}{\ln(d)} \cdot (a - a_0)^2 = 0.$$

Dividing both sides by the coefficient at $(a - a_0)^2$ and changing the sign of both sides, we get

$$(a - a_0)^2 - k \cdot a_0 = a_0^2 - (k - 2) \cdot a \cdot a_0 + a^2 = 0,$$

where we denoted

$$k \overset{\text{def}}{=} \frac{C \cdot \ln(d)}{M \cdot w_0 \cdot t}.$$

Dividing both sides by a^2, we get the following quadratic equation to the fraction $r_0 \overset{\text{def}}{=} \dfrac{a_0}{a}$ of the top-level memory allocated for communications:

$$r_0^2 - (k - 2) \cdot r_0 + 1 = 0.$$

The solution of this quadratic equation is

$$r_0 = \frac{k-2}{2} \pm \sqrt{\left(\frac{k-2}{2}\right)^2 - 1},$$

and $a_0 = a \cdot r_0$.

Analysis of the problem. When there are practically no communications, i.e., when the number of messages M is very small, the second term in the expression (1) for the objective function is negligible, so the objective function is approximately equal to its first term:

$$T \approx \frac{C}{a - a_0}.$$

This expression is the smallest when the difference $a - a_0$ is the largest, i.e., when the value a_0 is the smallest possible – and the smallest possible value of a_0 is 0.

Thus, in situations when we do not need to perform many communications, it makes sense not to allocate any top-level memory for communications, and use it all (or almost all) for computations.

On the other hand, if the number of messages is large, then, vice versa, we can ignore the first term in the expression (1) for the objective function and conclude that the objective function is approximately equal to its second term:

$$T \approx \frac{\ln(d) + \ln(t_0) - \ln(a_0)}{\ln(d)} \cdot M \cdot w_0 \cdot t.$$

In this case, the larger a_0, the larger is $\ln(a_0)$ and thus, the smaller is the above expression. So, for this expression to be as small as possible, we need to select the value a_0 which is as large as possible. The largest possible value of the communication-related portion a_0 of the top-level memory is the whole amount a of this memory: $a_0 = a$.

Thus, in situations when we need to perform a large number of communications, it makes sense to allocate practically all top-level memory for communications, and leave only the bare minimum for computations.

These are the two extreme cases, but they show that the more communications we need, the larger portion of the top-level memory should be allocated for communication purposes (and the above explicit formula for the optimal value of a_0 confirms this conclusion).

This is exactly what we observe, both in chimps and in computer networks, in terms of a tradeoff between communication and computation. Thus, our simple model indeed captures – at least on the qualitative level – the cognitive tradeoff phenomenon.

Acknowledgments

This work was partially supported by the US National Science Foundation via grant HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] J. Cohen, *Almost Chimpanzee: Searching for What Makes Us Human, in Rainforests, Labs, Sanctuaries, and Zoos*, Times Books, Henry Holt and Co., New York, 2010.

[2] C. F. Martin, R. Bhui, P. Bassaerts, T. Matsuzawa, and C. Camerer, "Chimpanzee choice rates in competitive games match equilibrium game theory predictions", *Scientific Reports*, 2014, Vol. 4, Paper 5182.

[3] B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, San Francisco, California, 1983.

[4] T. Matsuzawa, "Evolution of the brain and social behavior in chimpanzees", *Current Opinion in Neurobiology*, 2013, Vol. 23, pp. 443–449.