Balancing Human and Machine Contributions in Human Computation Systems

R. Jordan Crouser, Alvitta Ottley, and Remco Chang

Motivation

As we enter an age of increasingly larger and noisier data, dynamic interplay between human and machine analysis grows ever more important. Researchers and toolbuilders work to better understand and support the analytical process through systems that employ novel visual interactive interfaces along with computational support. These systems leverage the acuity of the human visual system as well as our capacity to understand and reason about complex data, nuanced relationships, and changing situations. In designing and building these systems, we rely on the intuition that the lived experience, perceptual advantage, and adaptability of the human analyst may prove crucial in areas where purely computational analyses fail. Similarly, by pairing the human analyst with a machine collaborator we hope to overcome some of the limitations imposed by the human brain such as limited working memory, bias, and fatigue.

With many promising examples of human-machine collaboration in the literature and everyday life, how do we tell if a new problem would benefit from human-computer collaboration and how should we allocate computational tasks? At present, balancing the cost of building and deploying a collaborative system with the benefits afforded by its use is precarious at best. We rely heavily on researcher intuition and current field-wide trends to decide which problems to approach using collaborative techniques. While this has led to many successes, it has also led to the investment of significant time and energy into inefficient collaborative solutions for problems that might better have been (or have already been) solved by human or machine alone.

While we have come a long way from listing tasks best assigned to human or machine (Fitts 1951), appropriate function allocation in collaborative systems is still far from a perfect science (Sheridan 2000). However, the effectiveness of any

R.J. Crouser (✉) • A. Ottley • R. Chang
Department of Computer Science, Tufts University, Medford, MA 02155, USA
e-mail: rcrous01@cs.tufts.edu; alvittao@cs.tufts.edu; remco@cs.tufts.edu

P. Michelucci (ed.), *Handbook of Human Computation*,
DOI 10.1007/978-1-4614-8806-4_48,

collaborative system is heavily dependent on how well it leverages the skills that humans and machines have to offer while minimizing waste. In the absence of a secret formula to prescribe this interplay, how do we balance the expected contributions of human and machine during the design process, and how do we evaluate the effectiveness of systems once we've built them? Herein, we seek to address these questions.

Toward that end, this chapter is organized into three main sections. Section 1 provides a broad definition of human computation and human-computer collaboration, as well as comparisons between a few well-known problems and their human-computer collaborative solutions. Section 2 considers the relative strengths of human and machine collaborators. Section 3 discusses the open problem of function allocation in human-computer collaborative systems, and will provide some insight on applying this knowledge during the design process. We hope that this chapter will leave the reader with an improved understanding of the complementary strengths of human and machine, as well as actionable information about best practices for real world design.

1 - Beyond Crowdsourcing: Human Computation as Human-Computer Collaboration

In this section, we develop a working definition for the term *human computation*. It is important to note that human computation is not synonymous with terms such as *collective intelligence*, *crowdsourcing*, and *social computing,* although they are related. Before we continue, we will first define a few of these terms in the interest of developing a context for defining human computation. For a more detailed discussion on definitions of Human Computation, please see Chapter 9 - "Synthesis and Taxonomy of Human Computation".

Crowdsourcing is the practice of obtaining services, ideas, or content by soliciting contributions from a large group of people.

Collective intelligence is the notion that groups of individuals working together can display intelligent behavior that transcends individual contributions.

Social computing is the intersection between people's social behaviors and their interactions with technology.

In many cases, a single system could be classified under more than one of these headings. At the same time, none of them fully captures the notion of human computation. As such, there are many working definitions of human computation in the literature:

> …using human effort to perform tasks that computers cannot yet perform, … (Law and von Ahn 2009)
>
> …a technique that makes use of human abilities for computation to solve problems. (Chan et al. 2009)

> A computational process that involves humans in certain steps… (Yang et al. 2008)
>
> …systems of computers and large numbers of humans that work together in order to solve problems that could not be solved by either computers or humans alone. (Quinn and Bederson 2009)

Working from these definitions, we can begin to come to consensus regarding what constitutes human computation. First, the problem must involve some form of *information processing*. This may occur as part of an algorithmic process, or may emerge through the observation and analysis of technology-mediated human behavior. Second, human participation must be *integral to the computational system or process*. In this work, we do not consider systems with only superficial human involvement to fall under the umbrella of Human Computation.

We can think of human-computation as a kind of human-computer collaboration. In this context, collaboration as defined as *a process in which two or more agents work together to achieve shared goals,* and human-computer collaboration as *collaboration involving at least one human and at least one computational agent* (Terveen 1995). This has also been called *mixed-initiative systems* (Horvitz 1999). In a mixed-initiative system, both the human and the machine can initiate action, access information and suggest or enact responses (Thomas and Cook 2005). The field of Visual Analytics is a perfect example of human-computer collaboration, as Visual Analytics systems leverage both analyst intelligence and machine computation in a collaborative effort to solve complex problems.

Under this definition, we see that crowdsourced computation is just the tip of the HC iceberg. Along a continuum between human-heavy and machine-heavy collaboration such as the one posed by Bertini and Lalanne in 2009, crowdsourced computation falls at one extreme:

With few exceptions, the computational burden falls almost entirely on the human collaborators in typical crowdsourcing applications such as image labeling and text translation. Human-based genetic algorithms also fall on the human-heavy end of the continuum, as the human agents determine both population fitness and genetic variation. In these systems, the primary role of the machine collaborator is to distribute tasks and collect results, a role with relatively trivial computational requirements. On the other extreme, algorithms for unsupervised learning functions with near autonomy from the human collaborator. Here, the human's role is to set the parameters of the algorithms and to verify the results. While less common, there are an increasing number of algorithmic approaches that attempt to maximize the contributions from both collaborators.

Without question, the term *human computation* spans a wide range of possible applications and computational distributions. Among all these, many of the most interesting and successful human computation systems not only balance the contribution of human and machine, but also leverage the complementary computational strengths of both parties. In the following sections, we will explore some of these strengths and how they can impact the distribution of labor in a human computation system.

2 - Complementary Computation

Both human and machine bring to the partnership varying strengths and opportunities for action, and during collaboration, each must be able to perceive and access these opportunities in order for them to be effectively leveraged. These affordances define the interaction possibilities of the team, and determine the degree to which each party's skills can be utilized during collaborative problem solving. The set of problems warranting a collaborative technique is equivalent to the set problems where there is an opportunity to effectively leverage affordances on both sides of the partnership in pursuit of the solution.

Instead of deciding who gets (stuck with) which task, we can begin to reason about which party can best contribute to the collective goal at each stage. The answer may not be only the human, or only the machine, but could in fact be both. By designing such that both human and machine are aware of the affordances made available to them by their collaborators, we encourage the development of more flexible procedures for collective problem solving.

Relative Strengths of Human and Computers

Fitts made the first published attempt in 1951 categorize tasks when he created a list of tasks the *humans are better at* and *machines are better at* (see Table 1). This is often abbreviated in the literature as *HABA-MABA*. While for many years this list was viewed as mantra for the division of labor, frequent and consistent technological advances in computation, automation and robotics make function allocation and the HABA-MABA list a moving target. The distinction between human and machine

Table 1 An outdated comparison of the relative strengths of humans and machines

Humans are better at:	Machines are better at:
Detecting small amounts of visual and auditory energy	Responding quickly and applying great force smoothly and precisely
Perceiving patterns of light or sound	Performing repetitive, routine tasks
Improvising/using flexible procedures	Storing information briefly, then erasing it completely
Storing large amounts of information and performing selective recall	Reasoning deductively
Reasoning inductively	Multitasking
Exercising judgment	

Table 2 A modern comparison of human vs machine affordances

Human Affordances	Machine Affordances
Visual perception	Computational Power
Spatial ability	Scalable, persistent storage
Linguistic ability	Efficient/reliable data transfer
Creativity	Freedom from bias
Adaptability	Precision
Sociocultural awareness	Parahuman sensing
Expertise/lived experience	Environmental Tolerance

is now less clear. For example, while in the 1950s humans were indeed better at storing large amounts of information, today's machines far exceed the capacity previously imagined, and the advent of distributed storage is rapidly enabling the outpacing of human memory by machines.

While Fitts' list was aimed at simply comparing the two for basic labor division, for many years it was incorrectly interpreted as gospel for function allocation for human-machine collaborative systems. Jordan (1963) criticized this approach stating that the underlying foundation should be that humans and machines are complementary rather than antithetical. Price (1985) also supported this view, arguing that function allocation could be better viewed as an interactive process rather than a divisive listing and that there may exist several optimal solutions for a given problem. Although the comparative approach to division of labor in human-computer collaborative systems was unwarranted, Fitts' list laid the foundation for thinking about the respective strengths of humans and machines.

Affordances: A Changing Perspective

In recent years, researchers have argued that the original perception of function allocation and Fitts' list no longer makes sense (Sheridan 2000; Dekker and Woods 2002). Dekker and Woods (2002) also provided counterargument to the validity of Fitts' list. They discussed how human-machine collaboration transforms human practice and forces analysts to adapt theirs skills and analytics processes. They argue for a shift in attention, moving away from allocation of tasks to a focus centered on how to design for harmonious human-machine cooperation. That is, how do we get humans and machines to play nicely, and work effectively? In response, a more recent framework (Crouser and Chang 2012) categorizes tasks based on relative strengths or affordances—opportunities for action. For a listing of some of the affordances they examined, see Table 2.

Human computation is an ideal approach to problems where there is an opportunity to leverage both human and machine affordances in pursuit of the solution. By framing potential collaboration in terms of the affordances at our disposal, we can then consider which of these affordances could be used to approach a problem and construct a solution.

3 - Effectively Leveraging Human and Machine Affordances

The success of human-computer collaborative systems hinges on leveraging the skills of both the human and the computer. That said, in order to address the problem of balancing and allocating workload in a human-computer collaborative system, it is first necessary to explore the space of problem difficulty relative to human and machine.

While problem difficulty for a machine can be defined as space and time complexity, for the human we propose that problem difficulty is attributed to two main sources: knowledge necessary to solve the problem and time investment required to solve a problem. The level of difficulty for one party may not necessarily transfer to the other. For instance, some problems such as character recognition are inherently easy for a human and can sometimes be performed in constant time but can be computationally expensive or unsolvable for the machine. The inverse can also be true. We can think about the problem space as having two orthogonal dimensions: human difficulty and machine difficulty. Figure 1 depicts some well-known sample problems within in this space.

In this diagram, problems appearing in the lower left region are trivial; that is, they are comparatively easy for both humans and machines. These problems, such

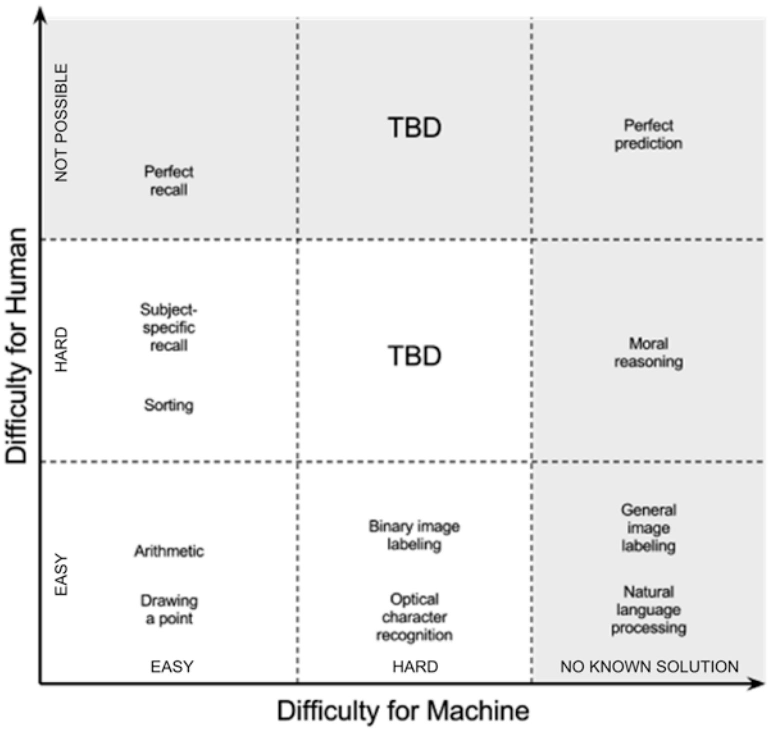

Fig. 1 A matrix of example problems based on respective difficulty levels for humans and machines

as arithmetic or simple shape rendering, generally do not warrant a human-computer collaborative solution. As we move to the right along the x axis, we encounter many of the problems addressed in early human computation systems: image labeling, character recognition, language processing, etc. These problems are difficult for machines, but relatively straightforward for humans. Here, the overhead cost incurred by involving human processing power is minimal compared with the resources required to achieve comparable performance using a machine.

As the field of human computation progresses, we are becoming more invested in applying collaborative techniques to solve problems that are difficult or impossible for *either* humans or machines alone, but which may be solvable through collaboration. In these problems, we are especially interested in how to best allocate the computational resources of the human and machine collaborators, allowing each party to play to its strengths.

Technically speaking, a human can simulate any process the machine can execute. After all, we designed the algorithms in the first place. Given an understanding of the process, enough paper and a sufficient supply of pencils, a human operator could write out the contents of each register, perform each bitwise operation, and record each result by hand. However, the time and resources required to compute exactly the same result are exorbitant. In addition, humans are susceptible to fatigue, and we are arguably limited by the capacity of our working memory and unreliable recall. In this sense, human operations are *expensive*, and there are cases where it is possible to reduce the number of human operations while maintaining optimal performance.

Consider the following example (Shahaf and Amir 2009): Imagine that we are given n randomly selected samples that we wish to classify. We know that the classifiers are simple threshold functions:

$$h_w(x) = \begin{cases} 1, & | \; x > w \\ 0, & | \; x \leq w \end{cases}$$

with the value of w depending on the input. Assume that we do not know w, but a human can classify the data correctly without too much trouble. We can think of the human as an *oracle*, a black box which is able to correctly answer specific questions in a single operation. Using the human as an oracle, there are several ways to approach this problem, each with benefits and drawbacks:

1. We could ignore the human and use a pure machine computational approach, first sorting the set of samples according to their x values and then choosing a random threshold value that falls between the lowest and highest values. This requires $O(n \log n)$ time to sort the list, and guarantees that at least two of the samples will be classified correctly. This is not a very promising bound on accuracy.
2. We could use a pure human computational approach, asking the human to classify each sample in the dataset. Because as we assumed that the human can always classify samples correctly, this method guarantees 100 % accuracy. In addition, this method requires $c \times n = O(n)$ operations, where c corresponds to the amount of time it takes the human to classify one sample. Under the usual metrics for evaluating algorithmic complexity, the method is technically "faster".

However, the value of the constant c may be enormous, which means that for all reasonably-sized input sets, this approach really isn't much better.

3. Finally, we could try a collaborative solution. First, the machine sorts the set of samples according to their x values, requiring $n \log n$ operations. Next, the human is asked to classify the sample that falls in the middle of the sorted list. If she answers "1", we know that all the samples above should also be labeled "1". Similarly, if she answers "0", we know that all the samples below should also be labeled "0". From here, the human is recursively asked about the middle point in the remaining half of the list that remains unlabeled. This is simply binary search, which implies that the human will be asked to classify at most $\log n$ samples for a cost of $c \times \log n$. Using this algorithm, we are able to dramatically reduce the workload for the human operator while maintaining 100 % accuracy simply by being clever regarding which samples to ask her about.

In this example, the third approach is clearly superior to the other two in terms of maximizing accuracy and efficiency. However, there are several key assumptions that need to be dressed. For example, what is the scale of the constant c? In human computation, we argue that this scale depends on the affordance being leveraged. This is perhaps most readily apparent in the field of information visualization. Through visualization, we transform the task of assessing abstract numerical information to evaluating visual information, leveraging the human visual processing system and thereby decreasing the per-operation cost c. As designers, it is important to consider the implications of leveraging various combinations of affordances between human and machine.

A few caveats: under this model, there is an explicit assumption the human oracle will always be able to provide the correct answer at a fixed (albeit large) cost. In reality, humans don't work this way. Intuition and experience indicate that humans eventually get tired or bored, and as a consequence their speed and accuracy suffer. Even under optimal working conditions, humans are fallible; instead of modeling the human as omniscient, we may wish to model human oracles as accurate with some probability $p < 1$. This fallibility may require measures to ensure that the overall probability of correctness is higher than that of any single oracle. In addition, the human may wish to request more information from the system before she can make a determination. This mutual querying behavior cannot be captured by an oracle model; we assume that the oracle takes input, and gives a correct answer using only that input. Because of this, it is important to continue to develop more nuanced models human of behavior in human computation systems, and to design metrics by which we can more robustly evaluate algorithmic complexity and performance in human-machine collaborative systems.

Conclusion

While history has often depicted human and machine as antithetical, we argue that in many cases their relative strengths prove complementary. Though originally designed simply to relieve human operators of tedious tasks, today it is perhaps more fitting to view machines as collaborators in the pursuit of solutions to challenging problems. We hope that this chapter has left the reader with a better understanding of some of the intricacies of balancing human and machine contributions in human computation systems as well as candidate methods for evaluating the optimality of that balance. We believe that work in the emerging field of human computation will help us to expand our understanding of what is computable, and that human-computer collaboration could lead to significant advances in tackling currently intractable problems.

References

Bertini E, Lalanne D (2010) Investigating and reflecting on the integration of. ACM SIGKDD Explor Newslett 11(2):9–18

Chan KT, King I, Yuen M (2009) Mathematical modeling of social games. Proc CSE 1205–1210

Crouser RJ, Chang R (2012) An affordance-based framework for human computation and human-computer collaboration. IEEE Trans Vis Comput Graph 18(12):2859–2868

Dekker SW, Woods DD (2002) Maba-maba or abracadabra? Progress on human–automation coordination. Cogn Technol Work 4(4):240–244

Ferguson G, Allen J (1998) Trips: an integrated intelligent problem- solving assistant. In: Proceedings of the national conference on artificial intelligence, Wiley, pp 567–573

Fitts PM (1951) Human engineering for an effective air-navigation and traffic-control system. National Research Council, Division of Anthropology and Psychology, Committee on Aviation Psychology, Washington

Gibson J (1977) The theory of affordances. In: Perceiving, acting, and knowing, pp 67–82

Horvitz E (1999) Principles of mixed-initiative user interfaces. In: Proceedings of the SIGCHI conference on human factors in computing systems: the CHI is the limit, ACM, pp 159–166

Jordan N (1963) Allocation of functions between man and machines in automated systems. J Appl Psychol 47(3):161

Law E, von Ahn L (2009) Input-agreement: a new mechanism for collecting data using human computation games. Proc CHI 3:1197–1206

Price H (1985) The allocation of functions in systems. Human factors. J Hum Factors Ergon Soc 27(1):33–45

Quinn A, Bederson BB (2009) A taxonomy of distributed human computation. Tech report HCIL-2009-2023, University of Maryland

Rickel J, Johnson W (1999) Animated agents for procedural training in virtual reality: perception, cognition, and motor control. Appl Artif Intell 13(4–5):343–382

Shahaf D, Amir E (2007) Towards a theory of AI completeness. In: 8th international symposium on logic formalizations of commonsense reasoning

Sheridan TB (2000) Function allocation: algorithm, alchemy or apostasy? Int J Hum-Comput Stud 52(2):203–216

Terveen L (1995) Overview of human-computer collaboration. Knowl Based Syst 8(2–3):67–81

Thomas J, Cook K (2005) Illuminating the path: the research and development agenda for visual analytics, vol 54. IEEE

Valdés-Pérez R (1999) Principles of human-computer collaboration for knowledge discovery in science. Artif Intell 107(2):335–346

Yang Y, Zhu BB, Guo R, Yang L, Li S, Yu N (2008) A comprehensive human computation framework: with application to image labeling. Proc MM