The Role of Information Theory and Queuing Theory in Human Computation Systems

Avhishek Chatterjee and Lav R. Varshney
Coordinated Science Laboratory, University of Illinois at Urbana-Champaign

I. Introduction

Although human computation has a long history [1], [2], crowd-based collective intelligence systems have flourished of late and now form the basis for many large-scale businesses, social enterprises, and citizen science consortia [3]–[5]. Microtask platforms with small tasks such as image labeling are perhaps most familiar, but there has been a proliferation of multifarious human computation models used in large scale by different platforms [6], [7]. Upwork operates as a freelance market and had 2.5 million workers and nearly 0.5 million clients in 2013 (then oDesk) [7]; an internal IBM crowdsourcing contest platform for software engineering drew on the company's nearly 0.5 million workers in 2013 [8]; and the microtask platform Amazon Mechanical Turk has had transaction value in the hundreds of millions of dollars per year. These varying, oft-heuristic system design models suggest myriad possible tradeoffs among basic resources like time, task accuracy, human attentional cost, communication bandwidth, and human enjoyment. To engineering systems theorists like us, this zoo of design possibilities raises the natural question: *are there fundamental limits and optimal designs for human computation?* After all, the attention and cognitive energy of people are valuable resources that should be used as efficiently and effectively as possible.

Theoretical foundations to understand what is possible and what is impossible, together with constructive approaches that achieve or approach these fundamental limits, are needed to push systems beyond heuristic and empirically-motivated designs. Previous studies on the mathematical foundations of human computation have drawn on game theory and mechanism design to understand strategic human behavior in human computation systems, e.g. [9]; machine learning theory to understand how to control quality when the outcomes of crowdsourcing are used for training algorithms, e.g. [10]; and theory of algorithms to improve efficiency of complicated crowd work, e.g. [11]. In this short position paper, we argue that information theory and queuing theory are also natural tools for studying certain fundamental limits of human computation systems. Such approaches can establish the *capacity of human-based systems to perform intelligent work*, in a form which may be computed from the properties of the people participating in them.

To demonstrate our approach, we discuss establishing limits of two types of crowdsourcing frameworks:

- skill-agnostic microtasking, where simple jobs are performed by crowd workers for a small payment, and
- skill-based crowdsourcing, where jobs can only be performed by workers with requisite skills.

A main challenge for skill-agnostic microtasking is the unreliability of random crowd workers, whereas for skill-based platforms it is the dynamic, time-varying, and random (un)availability of skilled workers. Information theory is well-suited to understand the limits of reliable computation in a system constructed from unreliable elements. On the other hand, queuing theory is naturally suited for studying limits of systems with dynamic and random resource availability. We also briefly discuss how certain important crowdsourcing scenarios necessitate a joint information- and queuing-theoretic treatment, leading to a union of these two mathematical fields in a completely novel way.

II. Limits of Reliable Microtasking

In microtasking, crowd workers may produce erroneous work, with each worker-task pair having a different probability of error. Consider binary microtasking with the simple error model [12], [13]: for each worker-task pair, the error probability p is sampled independently from some distribution on $[0, 1]$, unknown to the microtasking platform. If each task is allocated to a sufficiently large crowd and their outputs are combined appropriately (e.g. majority voting), arbitrarily high reliability can be achieved. But this reliability comes at a cost since the platform would have to pay a large number of agents. Is it possible to drive the probability of error to zero while keeping the cost from repeated or multiple processing finite, as depicted by the vertical line in Fig. 1?

This work was supported in part by the National Science Foundation under grants CCF-1623821 and IIS-1550145.

Definition 1. *A* microtasking policy *is a sequence of schemes $\{\mathcal{S}_n : n \geq 1\}$ so that for any $n \geq 1$, given n binary jobs with values $x^n \in \{0,1\}^n$, $\mathcal{S}_n$ designs a set of N_n binary jobs (without knowledge of x^n), allocates them to agents, and processes the outputs to obtain an estimate $\hat{x}^n \in \{0,1\}^n$ of job values. A microtasking policy is* reliable *if $\mathbf{P}(\hat{x}^n \neq x^n) \to 0$ as $n \to \infty$. The* redundancy factor *of a microtasking policy is given by $\limsup_{n\to\infty} N_n/n$.*

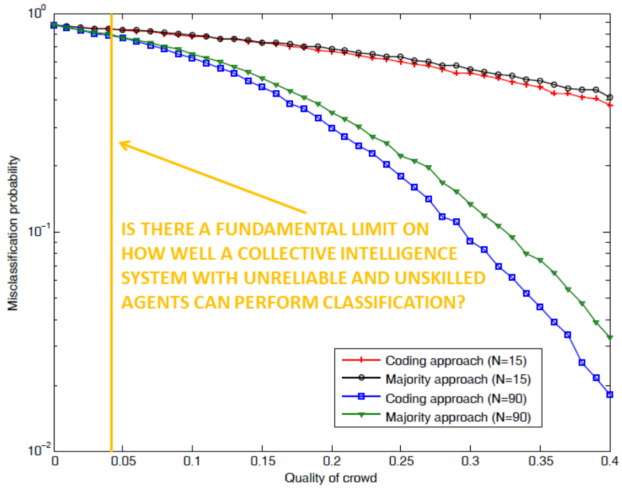

Fig. 1. Error probability for an 8-ary classification as a function of the unreliability of workers in a model crowd-based collective intelligence system. Simple majority voting schemes are shown together with schemes that use error-control codes, for different numbers of workers N [12]. In analogy to information theory, we aim to determine a vertical line that serves as a fundamental limit to the performance of any system.

The goal is to understand and achieve the minimal achievable redundancy factor among the class of reliable microtasking policies. There is a natural parallel to this problem in information theory.

To understand the system limit, first we obtain a lower bound on the best redundancy factor. Towards this a natural approach is to analyze redundancy factors of an expanded set of policies where the policy can use the knowledge of x^n to design N_n jobs to be allocated to the crowd, but cannot use the knowledge of x^n to obtain $\hat{x}^n$ from the crowd outputs. Using converse results in information theory [14], specially the strong converse results for discrete memoryless channels [15], a lower bound on the minimum achievable redundancy factor can be obtained. Using the result on capacity achieving linear codes for binary symmetric channels [16], we can show that there exists a policy that designs N_n jobs by taking an appropriate logical combinations (XOR) of the original jobs and achieves the best possible redundancy factor.

Result 1. *For any microtasking policy with $\liminf_{n\to\infty} N_n/n < 1/1 - h_2(\mathbf{E}[p])$, $\mathbf{P}(\hat{x}^n \neq x^n) \to 1$, where $\mathbf{E}[p]$ is the expectation of error probability p under the given distribution, and $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ for $x \in [0,1]$ is the binary entropy function. Moreover, for any $r > 1/1 - h_2(\mathbf{E}[p])$, there exists a microtasking policy with $\limsup_{n\to\infty} N_n/n < r$ such that $\mathbf{P}(\hat{x}^n \neq x^n) \to 0$.*

Similar results extend to settings where there may be all kinds of correlations between tasks, workers, and their responses. Though asymptotically optimal, the scheme in the proof of this result may be computationally expensive. There are simpler linear block codes [17] that achieve near-optimal performance even for a finite number of jobs. Even very short codes can improve performance compared to majority vote, Fig. 1. Note that one can specify a small fraction of erroneous jobs, i.e., to impose a milder condition $\mathbf{P}(\frac{1}{n}\sum_i \mathbf{1}(x_i \neq \hat{x}_i) > \delta) \to 0$, and obtain the minimum redundancy factor for a given blocklength under that constraint [18].

This information-theoretic approach to binary microtasking can be extended to more difficult jobs by posing them as sequences of binary queries (by using the domain taxonomy to construct an explicit [19] or implicit [20] dichotomous key). These sequences can then be encoded appropriately to produce new binary query sequences so even in the presence of erroneous answers, a correct answer to the original non-binary job can be obtained [12].

III. Limits of Allocations in Skill-based Crowdsourcing

On a skill-based crowdsourcing platform, jobs may have one or multiple parts known as *steps*, each of which requires one or more skills. In addition, steps of a job may have *precedence constraints* among them. For example in software development for mobile apps, at least three main steps must be completed in a specific order: architecture, coding, and testing. The first and last parts require knowledge of architecture and that of the particular application, whereas the middle part requires programming skills. On such a platform each worker has a specific set of skills and works only on steps requiring those skills. Consider the following model, to understand the fundamental limits of job allocation and how to achieve them.

There are N types of jobs. Each job of type j has a set of *steps* $1, 2, \ldots, K_j$, a *precedence constraint* among them $G_j = ([K_j], E_j)$, and a skill requirement $\mathcal{E}_{j,k}$ for each step k of job type j. Each step requires one unit of service (though this can be generalized [21]). Similarly there are M types of agents based on skill sets, where type i workers have skill set $\tilde{\mathcal{E}}_i$ and can offer one unit of service for each skill. Steps can be completed through collaboration among agents with complementary skills. Unlike microtasking where jobs are modeled as having values in a finite field, here steps do not share a general structure and so an error model as in microtasking does not make sense. Here we assume that if steps are allocated to appropriate agents, they are satisfactorily completed.

We consider a discrete-time dynamic system where jobs arrive over time and are allocated periodically, e.g. hourly. For each type j, the number of jobs arriving between two allocation epochs has a distribution F_j^{Job} with mean λ_j. The number of agents of type i available at an allocation epoch is time-varying and random, given by i.i.d. stochastic processes with distribution F_i^{Age} and mean μ_i. At any time t, the jobs (and steps) that have not been allocated in the previous epochs or have arrived after the last epoch can be allocated to the available agents at that time. The unallocated steps or jobs have to wait for the next allocation epoch $t + 1$.

A suitable performance metric for any allocation scheme is the accumulation of unallocated jobs after running the system for a sufficiently long time, i.e. the steady-state queue length. The smaller the steady-state queue length under an allocation scheme, the better. In a dynamic system with a fixed agent availability distribution, a scheme could have much lower queue length than another scheme for a given job arrival rate, but its accumulation becomes unbounded at a higher job arrival rate while the other scheme still maintains a finite queue length [22], [23]. To obtain a uniform benchmark of performance that avoids this, the notions of capacity and capacity-achieving scheme defined for queuing systems are useful [22], [23].

Definition 2. *Given agent availability distributions $\{F_i^{\text{Age}}\}$, the* capacity *of a crowdsourcing system is the set $\mathcal{C} \subset \mathbb{R}_+^N$, such that for any job arrival rates $\{\lambda_j\} \in \mathcal{C}$, there is a scheme such that the accumulations are finite under that scheme and for any arrival $\{\lambda_j\} \notin \mathcal{C}$ accumulations are unbounded under any scheme. A scheme is called* capacity-achieving *if the allocation rule is agnostic of the knowledge of $\{\lambda_j\}$ and accumulations are finite under the scheme for any arrival rate in $\mathcal{C}$.*

In [21], we characterized the capacity of the skill-based crowdsourcing system and provided a queue-based allocation scheme which is capacity-achieving. For a given availability of agents at any time, the algorithm maximizes a weighted sum of number of steps of each kind $((j, k))$ that are allocated. The weights are functions of the current queue lengths corresponding to the steps and the structural properties of the precedence graphs. We also showed that under certain mild conditions on $\{F_j^{\text{Job}}\}$ and $\{F_i^{\text{Age}}\}$, a simple decentralized greedy scheme that prioritizes steps higher up in the respective precedence graphs achieve capacity for sufficiently large N. Under this scheme, the total steady-state accumulation scales as $O(\log N)$.

We implicitly assume agents are truthful in reporting their skills and diligent in completing allocated jobs. Though invalid for some unstructured crowds [5], platforms like Upwork have registered agents that are closely monitored via ratings, etc. This is especially true for impact sourcing platforms like Samasource and Cloudfactory that aim to empower underprivileged workers by training them in skills and giving them work related to those skills [24].

IV. Optimal Job Dispatching to Agents in Impact Sourcing

Once an impact sourcing agent is trained in a particular skill, a stream of small jobs (often with responses in a finite set) are dispatched to the agent. Though trained, an agent may make random errors, and as noted in psychology [25], [26], performance deteriorates with increasing number of jobs waiting to be done. So there is a basic tradeoff between reliability and job dispatch rate, as fast dispatching results in larger queues and more errors. We study the maximum rate at which jobs can be dispatched to an agent while ensuring information-theoretic reliability ($\mathbf{P}(x^n \neq \hat{x}^n) \to 0$, as discussed before), and characterize the best job dispatch process [27]. This work is the first to analyze reliability of a system with queue-length dependent quality of service. Thus, studying fundamental limits of human computation yields a novel union of information and queuing theory [28]. Pushing further leads to questions on the role of feedback: already deployed in practice [19], but unstudied theoretically [29].

V. Conclusion

We have argued that two important examples of human computation—microtasking and skill-based crowdsourcing—have certain fundamental limits and optimal designs naturally described by information- and queuing-theoretic treatments. This is especially true for impact sourcing platforms. These mathematical approaches provide an alternate view on the foundations of human computation, as compared to game theory, machine learning theory, and the theory of computing. Moreover, trying to understand the limits of human computation lead to novel mathematical questions within the hard core of both information theory and queuing theory.

References

[1] D. A. Grier, *When Computers Were Human*. Princeton: Princeton University Press, 2005.

[2] ——, "Error identification and correction in human computation: Lessons from the WPA," in *Proc. AAAI Workshop Human Comput. (HCOMP'11)*, Aug. 2011, pp. 32–36.

[3] D. Tapscott and A. D. Williams, *Macrowikinomics: Rebooting Business and the World*. New York: Portfolio Penguin, 2010.

[4] J. Howe, *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. New York: Three Rivers Press, 2008.

[5] A. Marcus and A. Parameswaran, *Crowdsourced Data Management: Industry and Academic Perspectives*, 2015, in preparation.

[6] T. W. Malone, R. Laubacher, and C. Dellarocas, "The collective intelligence genome," *MIT Sloan Manage. Rev.*, vol. 51, no. 3, pp. 21–31, Spring 2010.

[7] K. J. Boudreau and K. R. Lakhani, "Using the crowd as an innovation partner," *Harvard Bus. Rev.*, vol. 91, no. 4, pp. 60–69, Apr. 2013.

[8] L. R. Varshney, "Participation in crowd systems," in *Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2012, pp. 996–1001.

[9] S. Jain and D. C. Parkes, "The role of game theory in human computation systems," in *Proc. ACM SIGKDD Workshop Human Comput. (HCOMP'09)*, June-July 2009, pp. 58–61.

[10] M. Lease, "On quality control and machine learning in crowdsourcing," in *Proc. AAAI Workshop Human Comput. (HCOMP'11)*, Aug. 2011, pp. 97–102.

[11] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: Crowdsourcing complex work," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol. (UIST '11)*, Oct. 2011, pp. 43–52.

[12] A. Vempaty, L. R. Varshney, and P. K. Varshney, "Reliable crowdsourcing for multi-class labeling using coding theory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 667–679, Aug. 2014.

[13] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Cambridge, MA: MIT Press, 2011, pp. 1953–1961.

[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

[15] J. Wolfowitz, "The coding of messages subject to chance errors," *Ill. J. Math.*, vol. 1, no. 4, pp. 591–606, Dec. 1957.

[16] P. Elias, "Coding for noisy channels," in *IRE Nat. Conv. Rec., Part 4*, 1955, pp. 37–46.

[17] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge: Cambridge University Press, 2008.

[18] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[19] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie, "The ignorant led by the blind: A hybrid human-machine vision system for fine-grained categorization," *Int. J. Comput. Vis.*, vol. 108, no. 1-2, pp. 3–29, May 2014.

[20] L. R. Varshney, P. Jyothi, and M. Hasegawa-Johnson, "Language coverage for mismatched crowdsourcing," in *Proc. 2016 Inf. Theory Appl. Workshop*, Feb. 2016.

[21] A. Chatterjee, M. Borokhovich, L. R. Varshney, and S. Vishwanath, "Efficient and flexible crowdsourcing of specialized tasks with precedence constraints," in *Proc. 2016 IEEE INFOCOM*, Apr. 2016.

[22] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. Cambridge University Press, 2014.

[23] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[24] M. Borokhovich, A. Chatterjee, J. Rogers, L. R. Varshney, and S. Vishwanath, "Improving impact sourcing via efficient global service delivery," in *Proc. Data for Good Exchange (D4GX)*, Sep. 2015.

[25] D. Kahneman, *Attention and Effort*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[26] B. Schwartz, "Queues, priorities, and social process," *Social Psychology*, vol. 41, no. 1, pp. 3–12, Mar. 1978.

[27] A. Chatterjee, D. Seo, and L. R. Varshney, "Capacity of systems with queue-length dependent service quality," in *Proc. 2016 Int. Symp. Inf. Theory Appl. (ISITA 2016)*, Nov. 2016, to appear.

[28] A. Ephremides and B. Hajek, "Information theory and communication networks: An unconsummated union," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2416–2434, Oct. 1998.

[29] O. Shayevitz and M. Feder, "Optimal feedback communication via posterior matching," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1186–1222, Mar. 2011.