Assuring Privacy and Reliability in Crowdsourcing with Coding

Lav R. Varshney[†], Aditya Vempaty[‡], and Pramod K. Varshney[‡]
[†] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
[‡] Department of Electrical Engineering and Computer Science, Syracuse University

Abstract—**Crowd workers are often unreliable and anonymous. Hence there is a need to ensure reliable work delivery while preserving some level of privacy to the requester's data. For this purpose, we use a combination of random perturbation to mask the sensitive data and error-correcting codes for quality assurance. We also consider the possibility of collusion attacks by malicious crowd workers. We develop mathematical models to study the precise tradeoffs between task performance quality, level of privacy against collusion attacks, and cost of invoking a large crowd. Such a study provides design strategies and principles for crowd work. The use of classification codes may improve efficiency considerably.**

We also comment on the applicability of these techniques for scalable assessment in education via peer grading, e.g. for massive open online courses (MOOCs).

I swear, by Galois and all the gods, that I will be true to the noble traditions of coding theory; that I will speak of it solely in the secret language of higher algebra known only to my fellow initiates; and above all, that I will vigilantly guard our sacred theory from those who would profane it by practical applications.

— James L. Massey (1971)

Today coding theory is badly profaned and most coding theorists are contaminated by practical applications. This is because practical application of error-correcting codes have been rapidly spreading all over the area of digital communication and storage systems. Mainly this is due to the advance of semiconductor technology.

— Hideki Imai (1991) [1]

I. Introduction

By the time 2011 rolled around, twenty years after Imai's pronouncement that the use of error-correcting codes were proliferating in communication and storage systems, crowdsourcing platforms such as Amazon Mechanical Turk were gaining maturity [2], as enabled by high-speed communication systems. In such a platform, a requestor breaks work into several encapsulated microtasks and posts them on a paid platform. Crowd workers complete this cognitive piecework and are rewarded with small payments for successful completion. Prominent services delivered through this model include text transcription, scientific experimentation, image classification, data entry, recommendation, and proofreading.

Error-control codes have enabled the underlying information technologies for crowdsourcing just as semiconductor technology enabled digital communications, but can they also be used to make the human effort itself within these emerging sociotechnical systems more efficient? After all both humans and machines are central [3].

Using error-control methods for human knowledge work actually has a long history. For example, one major activity of the Works Progress Administration (WPA) in the 1930s was the Mathematical Tables Project.[1] There, human computers created large, highly accurate tables, but no complete or coherent theory of error correction emerged. Their computing techniques were a form of craft mathematics [4], [5]. Leveraging the development of formal coding theory (by Galois and all the gods) after the 1930s, our aim in this paper is to develop a principled and systematic approach to error control for crowdsourcing. As we will see, this not only helps with obtaining reliable performance from unreliable and unknowledgeable crowd workers, but also allows a stochastic perturbation approach for preserving some privacy of work tasks from workers.

Those requesting work are typically not allowed to ask for identifying information about crowd workers and further crowdsourcing platforms typically do not verify user profiles. Hence crowd workers should be thought of as anonymous, but see [6]. Given the anonymity of workers, requestors may want to obscure work to protect intellectual capital or for other organizational, regulatory, or ethical reasons. There may be a desire to preserve privacy on the work being done.

Since microtasks are often tedious and payments are small, crowdsourced work is often unreliable. Indeed, workers may supply information totally unrelated to the posted task. To produce reliable work, we should introduce redundancy by giving each microtask to several workers and aggregating the individual results by some fusion rule or decoding rule.

Workers may further not have sufficient domain expertise to perform the full task and may only be able to perform simpler subtasks; for example in an M-ary classification task, workers may only be able to make simpler binary distinctions among objects [7]. Although it is possible to teach crowdworkers as they perform tasks [8], coding can be used to generate simpler tasks whose results can be combined using a decoding procedure.

This short paper describes a technique for jointly providing

[1] The WPA was required by law to organize tasks that used the largest possible amount of labor. Here we want to use human labor as efficiently as possible.

privacy and reliability in crowdsourcing using error-control codes. A key element is perturbing microtask data before dispatch to crowd workers so as to preserve privacy. The other element is introducing redundancy and aggregation through coding/decoding, not only to ensure quality in the face of worker unreliability but also the privacy-preserving perturbations.

We study tradeoffs in the space of privacy, reliability, and cost (as measured through the number of microtasks required) for crowdsourcing using coding. In particular we consider the privacy-reliability problem formulation of [9], but using coding methods developed in [10], [11]. Much more mathematical detail, as well as background material, is provided in [9], [11]. This idea was first presented at a non-archival industry conference [12].

Note that in Section III, we also discuss the possibility of using these coded crowdsourcing techniques for scalable, reliable, and private assessment in education via peer grading, e.g. for massive open online course (MOOC) platforms.

II. Background on Microtask Crowdsourcing

In this section we briefly discuss properties of expertise, reliability, and privacy that arise largely due to low pay and anonymity of crowd workers. These will drive the remainder of the paper.

A. Expertise

Consider fine-grained image labeling work, a kind of M-ary classification task, such as classifying images of mammals of the order *Carnivora* by species. This order includes common animals like cats and dogs, but also species such as jackals, coatis, kinkajou, olingos, weasels, mink, polecats, ferrets, martens, wolverines, badgers, mongooses, hyenas, bear-dogs, civets, and otters, which are not as well-known. Since crowd workers may not be carnivore experts, they may not be able to directly classify and so it makes sense to ask simpler questions. For example, although the kinkajou and the olingo look very similar, the olingo has gray to yellow coloration whereas the kinkajou is golden-brown. The same basic coloration question also distinguishes between the Corsac fox and the red fox. Thus asking a simple binary question on coloration would provide significant knowledge for the larger task. Questions can be designed using taxonomy and dichotomous keys [13].

Although typical crowd workers may lack expertise to perform the full task, malicious crowd workers may actually have extra ability. For example they may be able to identify whether a given image is of Silverton Blanco (the Wonder Dog), Blanco (a pet of Lyndon Johnson), or Pal (who played Lassie), and thereby break task privacy when given unperturbed task data.

B. Reliability

Given the anonymous worker pool, it is difficult to enforce the quality of their low-pay work. Verifying the quality of every submitted piece of work would be very expensive. Indeed the time and cost required to verify correctness is typically comparable to the time and cost for performing the work itself [14]. Hence the estimation of worker quality and ensuring reliability of crowdsourced work has emerged as an active research area in recent years.

A common approach to quality control is based on redundancy, e.g. having several workers perform the same task and then using a simple voting rule to determine the final output [15], [16]. Fusion rules better matched to the problem domain than simple voting, e.g. for speech transcription have also been used [17].

Beyond fusion rules, coding-theoretic ideas have been used to inject redundancy in a more efficient way than simple repetition [18]–[20]. Beyond machine-implemented decoding, people can do error detection and correction through a tournament selection-based quality control process for tasks with more than one possible correct answer [21].

C. Privacy

We are concerned with the requestor preserving some level of privacy regarding the work itself from the crowd workers. In the domain of text transcription, this problem has previously been noted, where it is said that “privacy, or securing the anonymity of the individuals who filled the forms, is a key issue, which often limits the use of outsourcing, let alone crowdsourcing” [22]. Similar privacy concerns also arise in data release [23], supply chain operations [24], and a variety of other settings [25].

One particular approach espoused herein is using randomized data-distortion to mask data for preserving privacy; this methodology attempts to hide sensitive data by randomly modifying data values with random noise.

III. Scaling Assessment in Massive-Scale Education

Assessment is a critical part of education since feedback allows students to learn from their mistakes, and with the increasing prominence of massive open online courses (MOOCs) there is a need for scalable assessment. Indeed many of these courses have tens of thousands of students enrolled. Although automatic grading has been a primary method used for MOOCs, *peer grading* has a long history in traditional classrooms [26] and is also emerging in MOOCs [27]. In fact this may be the only viable approach for assessing complicated, open-ended questions.

One can think of the students in a MOOC as crowd workers, and form a correspondence to the crowdsourcing setting described above.

- **[Expertise]** Students, almost by definition, do not have strong expertise in the content they are to assess. When given standard grading rubrics that decomposes assessment into smaller queries, students do seem to be able to provide scores effectively [26]. An alternative approach is to elicit ordinal ratings rather than cardinal ones by having students perform pairwise comparisons; such an approach has been shown to be robust to the lack of expertise of the students [28].

- **[Reliability]** Just as with any other crowdworkers, due to small incentives for high-quality work, there may be unreliability and bias. Effective peer grading must deal with the effects of inconsistent subjective evaluation; prior work has used matrix factorization approaches to normalize subjective differences among different peers [29].
- **[Privacy]** There may be compelling ethical, regulatory, or legal reasons to preserve some level of privacy of student work, even from peers. As an example, in a 2002 United States Supreme Court case, *Owasso Independent School District v. Falvo*, it was held under the federal Family Educational Rights and Privacy Act (Buckley Amendment) that student work is personal and potentially shielded by privacy laws [30] (but also held that peer grading is allowed for other reasons).

With these equivalences, it is clear how results for the microtask crowdsourcing setting would apply directly to the setting of peer assessment, and enable scaling despite lack of expertise, limits on reliability, and the need to preserve privacy of submitted work.

IV. A Coding Scheme for Reliable and Privacy-Preserving Crowdsourcing

In this section, we discuss the basic concept of using error-correcting codes to achieve reliable and privacy-preserving classification in a crowdsourcing system. We first briefly describe the distributed classification fusion using error-correcting codes (DCFECC) approach proposed by Wang et al. [31] that serves as a basic building block for us.

A. Distributed Classification Fusion using Error-Correcting Codes

In DCFECC, a distributed classification problem is represented using a binary code matrix $\mathbf{A}$. If there are M hypotheses $H_0, H_1, \ldots, H_{M-1}$ that need to be distinguished by n agents, the code matrix $\mathbf{A}$ is of size $M \times n$. Each row, a codeword of $\mathbf{A}$, corresponds to one of the possible hypotheses and the columns represent the decision rules of the agent.

Given a code matrix, agent j sends its binary decision ($u_j \in \{0, 1\}$). The fusion center receives the n-bit vector, $\mathbf{u} = [u_1, \cdots, u_n]$ and makes the final classification decision using a minimum Hamming distance decoding rule. The error-correction property of the code matrix $\mathbf{A}$ provides fault tolerance. Two heuristic methods have been proposed for DCFECC code design: cyclic column replacement and simulated annealing [31].

B. Ensuring Reliability using Crowds

The DCFECC approach can be used to design questions to be posed to the crowd workers. Recall the setting of an image to be classified into one of M fine-grained categories. Binary questions are designed for crowd workers using code matrix $\mathbf{A}$ so the task manager may reliably infer correct classification even with unreliable workers.

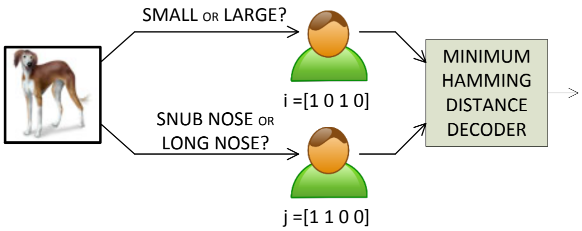

Fig. 1. A schematic diagram showing binary questions posed to workers and the decoding rule used by the task manager.

As part of modeling, assume worker j decides the true class with probability p_j and makes an error equiprobably:

$$p(y_j|H_m) = \begin{cases} p_j & \text{if } y_j = m \\ \frac{1-p_j}{M-1} & \text{otherwise.} \end{cases} \quad (1)$$

For every worker j, let a_j be the corresponding column of $\mathbf{A}$ and recall hypothesis $H_l \in \{H_0, H_1, \cdots, H_{M-1}\}$ is associated with row l in $\mathbf{A}$. Workers send a binary answer u_j based on decision y_j and column a_j. The task manager makes the final classification as the hypothesis corresponding to the codeword (row) that is closest in Hamming distance to the received vector of decisions. An illustrative example is shown in Fig. 1 for a dog breed classification task.

We use this classification code and decoding algorithm as a building block.

C. Preserving Privacy through Perturbation

Think of a microtask as a transformation of input data into output data. For example, a labeling task might be to transform the raw text of an email into a label drawn from a finite set like amusing, serious, irrelevant. Also suppose that this transformation is Lipschitz continuous so that small changes in the input lead to small changes in the output.

As has been noted, we need to provide some level of privacy. That is, there may be a requirement for the crowd workers not to have direct and precise knowledge of the input data: some form of perturbation should be used to change the data in some way so that it is not revealed precisely.

As depicted in Figure 2 for image processing tasks, one approach is to add random noise to the original image to create a perturbed image. Due to Lipschitz continuity, the performance of the crowd worker will degrade smoothly due to the noise. Since the crowd worker only has access to the perturbed version of the task, privacy is preserved: obfuscation through perturbation. It is clear that in the image labeling task, noise makes it difficult to determine the specific identity of the animal (thereby preserving privacy) but still allows answering simple questions.

D. Decoding with Privacy-Preserving Perturbation

Having developed building blocks, we put them together as in Figure 3 for the image classification problem. Each crowd

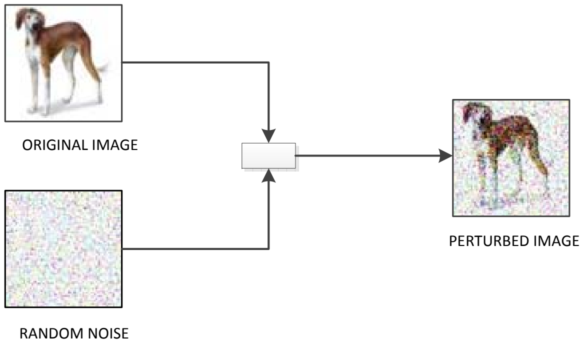

Fig. 2. Perturbation using random noise to preserve privacy.

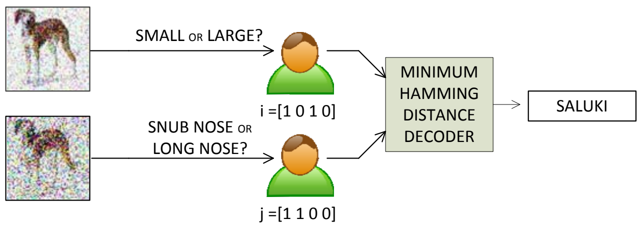

Fig. 3. Using decoding to mitigate the effects of worker unreliability, which are increased due to privacy-preserving noise perturbation.

worker gets a perturbed version of the input data, conditionally independent given the original data. This is then processed in the usual way by the crowd worker and individual results are combined by minimum distance decoding. Since the crowd workers only have access to perturbed versions of input data, privacy is preserved. Due to this perturbation however, their performance will typically degrade.

There is a tradeoff between the levels of privacy and unreliability, governed by the strength of the perturbation.

E. Collusion Attacks

By using independent perturbations for the several crowd workers, we ensure random discrepancies among answers given by different crowd workers that can be mitigated through decoding. There is, however a problem with this approach we need to consider: what if crowd workers share their perturbed data? By working together, as depicted in Figure 4 for image labeling, perhaps crowd workers can recover the original input data (such as the identity of the animal).

V. Mathematical Analysis

Now let us analyze the basic tradeoffs among privacy, reliability, and cost. Let an (n, M, Q, μ, δ) crowdsourcing system have n workers, with reliabilities which are i.i.d. random variables with mean μ, performing an M-ary classification task using data which has been perturbed by noise δ to protect the true identity among Q possibilities.

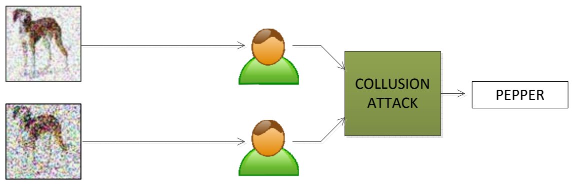

Fig. 4. A collusion attack for recovering the original input data from independent privacy-preserving perturbations.

Using an (n, M, Q, μ, δ) crowdsourcing system, the expected misclassification probability using code matrix $\mathbf{A}$ is, following arguments in [11]:

$$P_e(\mu,\delta) = \frac{1}{M}\times \sum_{\underline{i},l}\prod_{j=1}^{n}\left[\left(\tilde{\mu}a_{lj} + \frac{(1-\tilde{\mu})}{(M-1)}\sum_{k\neq l}a_{kj}\right)(2i_j-1)+(1-i_j)\right]C^l_{\underline{i}},$$

where

$$\tilde{\mu} = \frac{(1-\mu)\delta}{(M-1)} + \mu(1-\delta)$$

is the mean of effective reliabilities of the crowd workers.

Now suppose further that a subset of $\mathfrak{K}$ crowdworkers collude to try to determine the input data from their individual privacy-perturbed data. There are various ways of conducting a collusion attack, but suppose that the colluders use a plurality rule with random tie-breaking to vote on what they think the input data is. If there is just one colluder, the probability of success is:

$$P_c^{(1)} = 1-\delta.$$

If there are two colluders, the probability of success is:

$$P_c^{(2)} = (1-\delta)^2 + (1-\delta)\delta.$$

If there are three colluders, the probability of success is:

$$P_c^{(3)} = (1-\delta)^3 + 3(1-\delta)^2\delta + \left(\frac{(Q-1)^2-1}{(Q-1)^2}\right)(1-\delta)\delta^2.$$

Similar expressions can be developed for larger numbers of colluders $P_c^{(\mathfrak{K})}$ and are modifications of geometric random variable expressions.

Armed with expressions for probability of correct classification as well as probability of successful collusion attack, let us investigate the basic tradeoffs between privacy, reliability, and cost.

Figure 5 shows basic tradeoff results for the setting with $(n, M = 4, Q = 16, \mu = 0.9, \delta)$, n increasing by tens from 10 to 200, and δ taking values $0.5, 0.6, 0.7$ with three colluders. A suboptimal code matrix $\mathbf{A}$ formed by concatenating the following base matrix $\mathbf{A_0}$ together $n/10$ times was used.

$$\mathbf{A_0} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

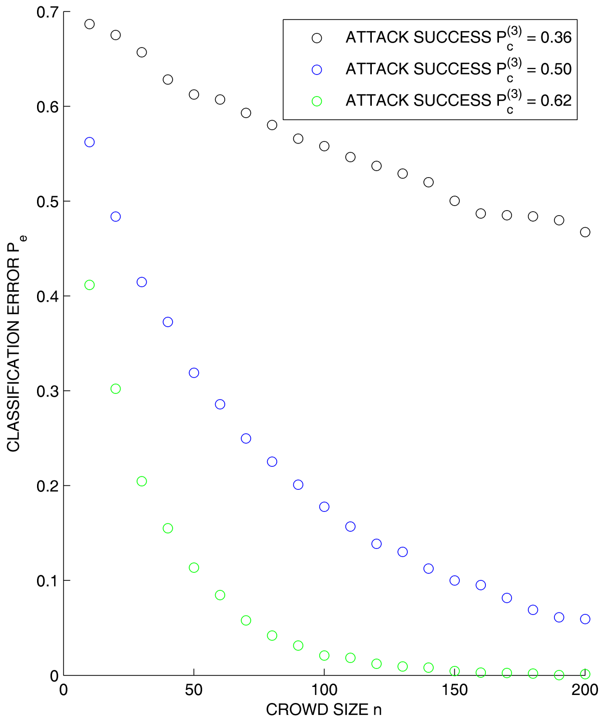

Fig. 5. Tradeoffs among cost, reliability, and privacy using coding.

We see a clear tradeoff among the three key quantities: privacy, reliability, and cost. Comparing results to those from a scheme based on majority vote [9] rather than error-control codes, indicates the benefit of using codes.

VI. Conclusion

In this short paper, we studied reliable and privacy-preserving crowdsourcing by putting together the problem formulation of [9] with the ideas of coding from [11]. Three novel features of crowdsourcing were considered: first that workers are unreliable due to low pay, tedium, and anonymity; second that workers may not have requisite expertise to perform complete tasks; and third that data may need to be protected from crowd workers due to privacy considerations.

We also argued that the problem formulation and proposed methods are apropos not just for microtask crowdsourcing platforms but also for peer grading in massive online education.

A system based on random privacy-preserving perturbations of input data together with code-based local computations and a global decoding rule to combine the computations of several crowd workers was proposed. A mathematical model was used to determine precise tradeoffs between task performance quality, level of privacy protection against collusion attacks, and cost of invoking a large crowd.

It would be useful going forward to study asymptotic regimes and develop information-theoretic results on the fundamental limits of reliable and privacy-preserving crowdsourcing. Also, to profane algebraic coding theory further by pushing it to practical applications in sociotechnical systems, it would be interesting to find algebraic characterizations of good distributed classification codes.

Acknowledgments

We thank K. Eswaran for making us aware of the Coding Theorist's Pledge due to Massey and of the response due to Imai.

References

[1] H. Imai, "Recent applications of error-correcting codes," *IEICE Trans.*, vol. E74-A, no. 9, pp. 2473–2482, Sep. 1991.

[2] P. G. Ipeirotis, "Analyzing the Amazon Mechanical Turk marketplace," vol. 17, no. 2, pp. 16–21, Winter 2010.

[3] O. L. de Weck, D. Roos, and C. L. Magee, *Engineering Systems: Meeting Human Need in a Complex Technological World*. Cambridge, MA: MIT Press, 2011.

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

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

[6] M. Lease, J. Hullman, J. P. Bigham, M. S. Bernstein, J. Kim, W. Lasecki, S. Bakhshi, T. Mitra, and R. C. Miller, "Mechanical Turk is not anonymous," Mar. 2013. [Online]. Available: http://dx.doi.org/10.2139/ssrn.2228728

[7] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *Computer Vision – ECCV 2010*, ser. Lecture Notes in Computer Science, L. Buttyán, V. Gligor, and D. Westhoff, Eds. Berlin: Springer, 2010, vol. 6314, pp. 438–451.

[8] A. Singla, I. Bogunovic, G. Bartók, A. Karbasi, and A. Krause, "On actively teaching the crowd to classify," in *NIPS Workshop on Data Driven Education*, Dec. 2013.

[9] L. R. Varshney, "Privacy and reliability in crowdsourcing service delivery," in *Proc. SRII Global Conf. 2012*, Jul. 2012, pp. 55–60.

[10] A. Vempaty, L. R. Varshney, and P. K. Varshney, "Reliable classification by unreliable crowds," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2013)*, May 2013, pp. 5558–5562.

[11] ——, "Reliable crowdsourcing for multi-class labeling using coding theory," arXiv:1309.3330v2 [cs.IT]., Jan. 2014.

[12] ——, "Error-correcting codes allow privacy and quality assurance in crowdsourcing," in *CrowdConf 2013*, Oct. 2013.

[13] J. Rocker, C. M. Yauch, S. Yenduri, L. A. Perkins, and F. Zand, "Paper-based dichotomous key to computer based application for biological indentification," *J. Comput. Sci. Coll.*, vol. 22, no. 5, pp. 30–38, May 2007.

[14] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," in *Proc. ACM SIGKDD Workshop Human Comput. (HCOMP'10)*, Jul. 2010, pp. 64–67.

[15] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empirical Meth. Natural Language Process. (EMNLP'08)*, Oct. 2008, pp. 254–263.

[16] J. Le, A. Edmonds, V. Hester, and L. Biewald, "Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution," in *Proc. SIGIR 2010 Workshop Crowdsourcing for Search Eval. (CSE 2010)*, Jul. 2010, pp. 17–20.

[17] M. Marge, S. Banerjee, and A. I. Rudnicky, "Using the Amazon Mechanical Turk for transcription of spoken language," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2010)*, Mar. 2010, pp. 5270–5273.

[18] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal crowdsourcing using low-rank matrix approximations," in *Proc. 49th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2011, pp. 284–291.

[19] ——, "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.

[20] ——, “Efficient crowdsourcing for multi-class labeling,” in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, Jun. 2013, pp. 81–92.

[21] Y.-A. Sun, S. Roy, and G. D. Little, “Beyond independent agreement: A tournament selection approach for quality assurance of human computation tasks,” in *Proc. AAAI Workshop Human Comput. (HCOMP’11)*, Aug. 2011, pp. 113–118.

[22] E. D. Karnin, E. Walach, and T. Drory, “Crowdsourcing in the document processing practice,” in *Current Trends in Web Engineering*, ser. Lecture Notes in Computer Science, F. Daniel and F. M. Facca, Eds. Berlin: Springer, 2010, vol. 6385, pp. 408–411.

[23] X. Xiao, Y. Tao, and M. Chen, “Optimal random perturbation at multiple privacy levels,” in *Proc. 35th Int. Conf. Very Large Data Bases (VLDB’09)*, Aug. 2009, pp. 814–825.

[24] D. Y. Zhang, Y. Zeng, L. Wang, H. Li, and Y. Geng, “Modeling and evaluating information leakage caused by inferences in supply chains,” *Comput. Ind.*, vol. 62, no. 3, pp. 351–363, Apr. 2011.

[25] L. Sankar, W. Trappe, K. Ramchandran, H. V. Poor, and M. Debbah, “The role of signal processing in meeting privacy challenges: An overview,” *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 95–106, Sep. 2013.

[26] P. M. Sadler and E. Good, “The impact of self- and peer-grading on student learning,” *Educational Assessment*, vol. 11, no. 1, pp. 1–31, 2006.

[27] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller, “Tuned models of peer assessment in MOOCs,” in *Proc. 6th Int. Conf. Educational Data Mining (EDM 2013)*, Jul. 2013, pp. 153–160.

[28] N. B. Shah, J. K. Bradley, A. Parekh, M. Wainwright, and K. Ramchandran, “A case for ordinal peer-evaluation in MOOCs,” in *NIPS Workshop on Data Driven Education*, Dec. 2013.

[29] J. Díez, O. Luaces, A. Alonso-Betanzos, A. Troncoso, and A. Bahamonde, “Peer assessment in MOOCs using preference learning via matrix factorization,” in *NIPS Workshop on Data Driven Education*, Dec. 2013.

[30] B. Brinkman, “An analysis of student privacy rights in the use of plagiarism detection systems,” *Sci. Eng. Ethics*, vol. 19, no. 3, pp. 1255–1266, Sep. 2013.

[31] T.-Y. Wang, Y. S. Han, P. K. Varshney, and P.-N. Chen, “Distributed fault-tolerant classification in wireless sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 724–734, Apr. 2005.