A Study of Unsupervised Adaptive Crowdsourcing

G. Kesidis[1,2] and A. Kurve[2]
[1]CS&E Dept and [2]EE Dept
The Pennsylvania State University, University Park, PA, 16802
{gik2 and ack205}@psu.edu

***Abstract*—We consider unsupervised crowdsourcing performance based on the model given in [13] wherein the responses of end-users are essentially rated according to how their responses correlate with the majority of other responses to the same subtasks/questions. In one setting, we consider an independent sequence of identically distributed crowdsourcing assignments (meta-tasks), while in the other we consider a single assignment with a large number of component subtasks. Both problems yield intuitive results in which the overall reliability of the crowd is a factor.**

***Index Terms*—Crowdsourcing, unsupervised learning, consensus, design, performance, error rate.**

I. Introduction

On-line crowdsourcing addresses the problem of solving a large meta-task by decomposing it into a large number of small tasks/questions and assigning them to an online community of peers/users. Examples of decomposable meta-tasks include [4], [11]:

- annotating (including recommending) or classifying a large number of consumer products and services, or data objects such as documents [12], web sites (*e.g.*, answering which among a large body of URLs contains pornography), images, videos;
- translating or transcribing a document [6] possibly including decoding a body of CAPTCHAs [17];
- document correction through proofreading [5], [20]; and
- creating and maintaining content, *e.g.*, Wikipedia and open-source communities.

General purpose platforms for on-line crowdsourcing include Amazon's Mechanical Turk [1], [12], [14] and Crowd Flower [7].

Users responding to questions may do so with different degrees of reliability. If p is the probability that a user correctly answers a question, let the expectation $\mathsf{E}p$ be taken over the ensemble of users. Thus, $\mathsf{E}p$ is a measure of the reliability of the *majority* and, fundamentally, whether the positive correlation with the majority ought to be sought for individual users (as is typically assumed in many online unsupervised "polling" systems).

A user population is arguably reliable ($\mathsf{E}p > 0.5$) when the population *itself* ultimately decides the issue (*e.g.*, confidence intervals for an election poll), or the questions concern a commonplace issue with commonplace expertise among the population (*e.g.*, whether a web site contains pornography), or the population is significantly financially incentivized to be accurate (*i.e.*, incentivized to acquire the required expertise to be accurate). Some market-based crowdsourcing scenarios (*e.g.*, questions of investing in stocks of complex companies), or analogies to bookmaking (setting odds so that the house always profits), may not be relevant here, *i.e.*, scenarios where questions are pushed to users who minimally profit by answering them correctly. That is, for some specialized technical issues, it may be possible that the "crowd" will be unhelpful ($\mathsf{E}p \approx 0.5$) or incorrectly prejudiced/biased ($\mathsf{E}p < 0.5$). In many cases, the users may need to be paid for questions answered [19]. Thus, the crowdsourcer is incentivized to determine the reliability of individual users in a scalable fashion.

This paper is organized as follows. The iterative, unsupervised framework and assumptions of [13] in Section II. In Section III, we find expressions for the mean and variance of the parameters (y) used to weight user answers after one iteration, under certain assumptions related to the regularity of connectivity of the bipartite graph matching users to questions/sub-tasks. We also state the existence of a fixed point for a normalized version of the user-weights iteration. To derive an asymptotic result, we consider the user weight iteration spanning a sequence of independent and identically distributed (i.i.d.) meta-tasks, with one iteration per meta-task. We give the results of a numerical study for the original system (multiple iterations for single meta-task) in Section V. In Section VI, how the crowdsourcing system of [13] is related to LDPC decoding is decribed. Finally, we conclude with a summary in Section VII.

This work is supported in part by NSF CNS grant 0916179.

II. Model Background

In [13], a single meta-task is divided into a group of $|Q|$ similar subtasks/questions $i \in Q$ for which the true Boolean answers are encoded $z_i \in \{-1, 1\}$. These questions are assigned to a group of U users $a \in U$. If a is assigned question i, then his/her answer is $A_{ia} \in \{-1, 1\}$. Again, the questions i are assumed similar so we model user a with a task-independent parameter p_a which reflects the reliability of the user's answer: for all i,

$$\mathsf{P}(A_{ia} = z_i) = p_a \quad \text{and} \quad \mathsf{P}(A_{ia} = -z_i) = 1 - p_a,$$

so that

$$\begin{aligned} \mathsf{E}A_{ia} &= z_i p_a - z_i(1-p_a) \;=\; z_i(2p_a - 1) \;\text{ and} \\ \text{var}(A_{ia}) &= 1 - (2p_a - 1)^2 \;=\; 4p_a(1-p_a). \end{aligned}$$

Suppose that the response to question i is determined by the crowdsourcer as

$$\hat{z}_i \quad = \quad \text{sgn}\left(\sum_{b\in\partial i} A_{ib} y_{b\to i}\right), \tag{1}$$

where $\partial i \subset U$ is the group of users assigned to question i and $y_{b\to i}$ is the *weight* given to user b for question i.

If $y_{b\to i}$ is the same positive constant for all b, i, then the crowdsourcer is simply taking a majority vote without any knowledge of the reliability of the peers.

One approach to determining weights y is to assess how each user a performs with respect to the majority of those assigned to the same question i. The presumption is that the majority will tend to be correct on average. Given that, how can the crowdsourcer identify the unreliable users/respondents so as to avoid them for subsequent tasks? Accordingly, a different weight $y_{i\to a}$ can be iteratively determined for each user a's response to every question i in the following way [13]:

- Initialize i.i.d. $y^{(0)}_{a\to i} \sim \mathsf{N}(1,1)$, *i.e.*, initially assume each user is roughly reliable with $\mathsf{E}y^{(0)} = 1$ and $\mathsf{P}(y^{(0)} > 0) \approx 0.84$.
- For step $k \geq 1$:
  - k.1: $x^{(k)}_{j\to a} = \sum_{b\in\partial j\backslash a} A_{jb} y^{(k-1)}_{b\to j}$, *i.e.*, consider the weighted answer to question j not including user a's response.
  - k.2: $y^{(k)}_{a\to i} = \sum_{j\in\partial^{-1}a\backslash i} A_{ja} x^{(k)}_{j\to a}$, *i.e.*, correlate the responses of the other users with those of user a over all questions assigned to a except i.

Here $\partial^{-1}a$ is the set of questions assigned to user a. The distribution of $y^{(k)}_{a\to i}$ as a function of iteration k is studied in [13] for degree-regular assignment of questions to users. Note that by simply eliminating $x^{(k)}_{j\to a}$ we can write

$$y^{(k)}_{a\to i} \quad = \quad \sum_{j\in\partial^{-1}a\backslash i} \sum_{b\in\partial j\backslash a} A_{ja} A_{jb} y^{(k-1)}_{b\to j}. \tag{2}$$

So, $y^{(1)}_{a\to i}$ depends on the responses of a's *one-hop neighbors* in U,

$$N^{(1)}_{a\to i} \quad := \quad \{b \in U \mid \exists j \in \partial^{-1}a\backslash i \text{ s.t. } b \in \partial j\backslash a\},$$

i.e., not including a itself or any one-hop neighbors of a (in U) also assigned to question i.

III. Distribution of user weights after one iteration

A. Degree-regular graph

For the degree-regular assignment, we can relate the number of users per question $r := |\partial i|\ \forall i$ to the number of questions per user $s := |\partial^{-1}a|\ \forall a$:

$$r|Q| \quad = \quad s|U|.$$

In the following, we will assume

$$r \geq 2 \quad \text{and} \quad s \geq 2.$$

Furthermore, we may assume that all sets $N^{(1)}_{a\to i}$ are the same size N, where generally

$$N \quad \leq \quad (r-1)(s-1) \text{ members},$$

but that the number of terms summed in (2) is always equal to $(r-1)(s-1)$.

To form such degree-regular assignments, one can simply iterate over the (enumerated) questions:

0. $i = 1$ (first question $\in \{1, 2, ..., |Q|\}$).
1. assign i to r different users $\in U$ chosen uniformly at random.
2. $\forall a \in U$ such that $|\partial^{-1}a| = s$, $U \to U\backslash\{a\}$.
3. if $i < |Q|$, $i \to i+1$ and go to step 1.

Note that since $r|Q| = s|U|$, the questions will be exhausted just when the users are (*i.e.*, when $i \to |Q|$, $U \to \emptyset$).

B. First-iteration variance and mean of user weights, y

Let $O^{(0)}_{ja} = \sum_{b\in\partial j\backslash a} A_{jb} y^{(0)}_{b\to j}$ and note that $O^{(0)}_{ja}$ is independent of $O^{(0)}_{j'a}$ for all a and $j \neq j'$. So, if $\mu^{(0)}_{a\to i} := \mathsf{E}y^{(0)}_{a\to i} = 1$, the mean of $y^{(1)}_{a\to i}$ is

$$\begin{aligned}
\mu^{(1)}_{a\to i} \quad &:= \quad \sum_{j\in\partial^{-1}a\backslash i} \mathsf{E}A_{ja} \sum_{b\in\partial j\backslash a} \mathsf{E}A_{jb} \cdot 1 \quad \text{(by indep.)} \\
&= \quad \sum_{j\in\partial^{-1}a\backslash i} z_j(2p_a - 1) \sum_{b\in\partial j\backslash a} z_j(2p_b - 1) \\
&= \quad (2p_a - 1) \sum_{j\in\partial^{-1}a\backslash i} \sum_{b\in\partial j\backslash a} (2p_b - 1)
\end{aligned}$$

(since $z_j^2 = 1$ a.s.). Also, assume the variance $\text{var}^{(0)}_{a\to i} := \text{var}(y^{(0)}_{a\to i}) = 1$ ($\Rightarrow \mathsf{E}(y^{(0)}_{a\to i})^2 = 2$). So,

$$\begin{aligned}
\text{var}^{(1)}_{a\to i} \quad &= \quad \sum_{j\in\partial^{-1}a\backslash i} \text{var}(A_{ja} O^{(0)}_{ja}) \quad \text{(by indep.)} \\
&= \quad \sum_{j\in\partial^{-1}a\backslash i} [\mathsf{E}(O^{(0)}_{ja})^2 - (2p_a - 1)^2 (\mathsf{E}O^{(0)}_{ja})^2] \\
&= \quad \sum_{j\in\partial^{-1}a\backslash i} [\text{var}(O^{(0)}_{ja})^2 + (1 - (2p_a - 1)^2)(\mathsf{E}O^{(0)}_{ja})^2] \\
&= \quad \sum_{j\in\partial^{-1}a\backslash i} [\{\sum_{b\in\partial j\backslash a} \text{var}(A_{jb} y^{(0)}_{b\to j})\} + \\
&\qquad (1 - (2p_a - 1)^2)(\mathsf{E}O^{(0)}_{ja})^2] \quad \text{(by indep.)} \qquad (3) \\
&= \quad \sum_{j\in\partial^{-1}a\backslash i} [\{\sum_{b\in\partial j\backslash a} (2 - (2p_b - 1)^2)\} + \\
&\qquad (1 - (2p_a - 1)^2)(\sum_{b\in\partial j\backslash a} 2p_b - 1)^2]. \qquad (4)
\end{aligned}$$

C. Assumption of large number of users per question, r

Finally, for simplicity, we may additionally assume sufficiently large r (number of users per question) and the neighbor selection is uniformly distributed so that, for all a, j,

$$\frac{1}{r-1} \sum_{b\in\partial j\backslash a} (2p_b - 1) \quad \approx \quad \mathsf{E}(2p - 1) \; = \; 2\mathsf{E}p - 1. \tag{5}$$

The following lemma is now obtained simply by substitution.

Lemma 1. *For (2) under (5), for all a, i:*

$$\mu^{(1)}_{a\to i} \quad \approx \quad (s-1)(r-1)(2p_a - 1)(2\mathsf{E}p - 1) \tag{6}$$

and

$$\begin{aligned} var^{(1)}_{a\to i} &\approx (s-1)(r-1)[\mathsf{E}(2-(2p-1)^2) \\ &+ (1-(2p_a-1)^2)(r-1)(2\mathsf{E}p-1)^2]. \quad (7)\end{aligned}$$

Though the $U\times Q$ matrix $\mathbf{Y}^{(k)}$ with elements $Y_{a,i} := y^{(k)}_{a\to i}$, is Markovian, directly proceeding along these lines for $\text{var}^{(k)}_{a\to i}$, $k\geq 2$, is complicated by the dependence of the terms involved through the structure of the bipartite graph mapping users U to questions Q, *cf.*, Section IV.

D. Discussion: Normalized weights

It's possible that the weights $y^{(k)}$ may be unbounded in k. Instead of (2), for a degree-regular assignment suppose the weights are, for all a,i,

$$\hat{y}^{(k)}_{a\to i} = \frac{1}{(s-1)(r-1)} \sum_{j\in\partial^{-1}a\backslash i}\sum_{b\in\partial j\backslash a} A_{ja}A_{jb}\hat{y}^{(k-1)}_{b\to j} \quad (8)$$

Let $\hat{\mathbf{Y}}^{(k)}$ be the $|U|\times|Q|$-matrix with elements $\hat{Y}_{a,i} := \hat{y}^{(k)}_{a\to i}$.

Proposition 1. *For (8), if $\hat{\mathbf{Y}}^{(0)} \in [-1,1]^{|U|\times|Q|}$, then the sequence $\hat{\mathbf{Y}}^{(k)}$ has a fixed point in $[-1,1]^{|U|\times|Q|}$.*

Proof: Simply by the triangle inequality and induction, if $\hat{\mathbf{Y}}^{(0)} \in [-1,1]^{|U|\times|Q|}$ then $\hat{\mathbf{Y}}^{(k)} \in [-1,1]^{|U|\times|Q|}$ for all k. As the mapping (8) is continuous, we can apply Brouwer's fixed point theorem [3] to get existence. □

Note that, generally, fixed points of a continuous linear operator on a bounded domain needn't be unique.

IV. A SERIES OF SIMILAR META-TASKS WITH ONE ITERATION PER META-TASK

Let

$$\begin{aligned} \delta &:= (s-1)(r-1) \ > 1, \text{ and} \\ \phi &:= \mathsf{E}(2p-1)^2 \ \in [0,1]. \end{aligned}$$

We now consider a *series* of similar meta-tasks indexed k and a *single* iteration as (2) for each on its component questions (all questions similar to each other too). Moreover, each meta-task will reassign the component questions using an independently sampled degree-regular assignment. Obviously, the answers A will be independently resampled too. That is, here

$$\tilde{y}^{(k)}_{a\to i} = \frac{1}{\delta}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}\sum_{b\in\partial j^{(k)}\backslash a} A^{(k)}_{ja}A^{(k)}_{jb}\tilde{y}^{(k-1)}_{b\to j} \quad (9)$$

where now the $A^{(k)}_{jb}$ and $\tilde{y}^{(k-1)}_{b\to j}$ terms are independent. The following asymptotic analysis is facilitated by this assumption on successive i.i.d. meta-tasks.

Lemma 2. *If (5) and $\tilde{\mu}^{(0)}_{a\to i} = 1$ for all a,i, then for all $k\geq 1$,*

$$\tilde{\mu}^{(k)}_{a\to i} \approx (2p_a-1)(2\mathsf{E}p-1)\phi^{k-1}. \quad (10)$$

Proof: First note that by the argument for (6) and definition (9), (10) holds for $k=1$. The lemma is simply proven by induction. □

Lemma 3. *If (5), $\tilde{\mu}^{(0)}_{a\to i} = 1$ and $v\tilde{a}r^{(0)}_{a\to i} = v_0$ for all a,i, then for $k\geq 1$*

$$var^{(k)}_{a\to i} \leq v_0\delta^{-k} + r(2\mathsf{E}p-1)^2\delta^{-1}\frac{\phi^{2k}-\delta^{-k}}{\phi^2-\delta^{-1}}. \quad (11)$$

Proof: Proceeding as for (4), and using the independence at (3) afforded by (9), gives

$$\begin{aligned} \tilde{\text{var}}^{(k)}_{a\to i} &= \frac{1}{\delta^2}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}[\{\sum_{b\in\partial j^{(k)}\backslash a}\text{var}(A^{(k)}_{jb}\tilde{y}^{(k-1)}_{b\to j})\} \\ &\quad + (1-(2p_a-1)^2)\{\sum_{b\in\partial j^{(k)}\backslash a}(2p_b-1)\tilde{\mu}^{(k-1)}_{b\to j}\}^2] \\ &= \frac{1}{\delta^2}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}[\{\sum_{b\in\partial j^{(k)}\backslash a}\mathsf{E}(\tilde{y}^{(k-1)}_{b\to j})^2 \\ &\quad - (2p_b-1)^2(\tilde{\mu}^{(k-1)}_{b\to j})^2\} \\ &\quad + (1-(2p_a-1)^2)\{\sum_{b\in\partial j^{(k)}\backslash a}(2p_b-1)\tilde{\mu}^{(k-1)}_{b\to j}\}^2] \\ &= \frac{1}{\delta^2}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}[\{\sum_{b\in\partial j^{(k)}\backslash a}\tilde{\text{var}}^{(k-1)}_{b\to j} \\ &\quad + (1-(2p_b-1)^2)(\tilde{\mu}^{(k-1)}_{b\to j})^2\} \\ &\quad + (1-(2p_a-1)^2)\{\sum_{b\in\partial j^{(k)}\backslash a}(2p_b-1)\tilde{\mu}^{(k-1)}_{b\to j}\}^2] \\ &\leq \sum_{j\in\partial^{-1}a^{(k)}\backslash i}[\{\sum_{b\in\partial j^{(k)}\backslash a}\tilde{\text{var}}^{(k-1)}_{b\to j} + (\tilde{\mu}^{(k-1)}_{b\to j})^2\} \\ &\quad + \{\sum_{b\in\partial j^{(k)}\backslash a}(2p_b-1)\tilde{\mu}^{(k-1)}_{b\to j}\}^2]. \end{aligned}$$

Thus,

$$\begin{aligned} \tilde{\text{var}}^{(k)}_{a\to i} &\leq \frac{1}{\delta^2}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}\sum_{b\in\partial j^{(k)}\backslash a}\tilde{\text{var}}^{(k-1)}_{b\to j} \\ &\quad + \frac{(2\mathsf{E}p-1)^2(\mathsf{E}(2p-1)^2)^{2k}}{\delta} \\ &\quad + \frac{(r-1)(2\mathsf{E}p-1)^2(\mathsf{E}(2p-1)^2)^{2k}}{\delta} \\ &= \frac{1}{\delta^2}\sum_{j\in\partial^{-1}a^{(k)}\backslash i}\sum_{b\in\partial j^{(k)}\backslash a}\tilde{\text{var}}^{(k-1)}_{b\to j} \\ &\quad + \frac{r(2\mathsf{E}p-1)^2\phi^{2k}}{\delta}. \end{aligned}$$

The proof then follows by induction, *i.e.*, dropping dependence on a,i, the previous display is

$$\tilde{\text{var}}^{(k)} \leq \frac{1}{\delta}\tilde{\text{var}}^{(k-1)} + \frac{1}{\delta}r(2\mathsf{E}p-1)^2\phi^{2k}.$$

□

By direct substitution, we arrive at the following.

Proposition 2. *For (9)*[1] *under (5), if*

$$\frac{1}{(s-1)(r-1)} \quad \leq \quad [\mathsf{E}(2p-1)^2]^2 \ < \ 1, \qquad (12)$$

then for all a, i with $p_a \neq 0.5$,

$$\frac{\sqrt{v\tilde{a}r^{(k)}_{a\to i}}}{\tilde{\mu}^{(k)}_{a\to i}} \quad = \quad O(1),$$

i.e., *the relative error is bounded as $k \to \infty$.*

Note that (12) is just $\delta^{-1} \leq \phi^2 < 1$.

In (10), the product $(2p_a-1)(2\mathsf{E}p-1)$ determines the *sign* of $\mu^{(k)}_{a\to i}$ (corresponding to the weight of user a for question i). So, if the crowd tends to be correct ($\mathsf{E}p > 0.5$) then (as expected): if a particular user a tends to be correct ($p_a > 0.5$) then the sign of a's weight $y_{a\to i}$ will tend to be positive, else negative.

Revisiting (1), we can instead estimate the answer to question i of task k using *normalized* weights as

$$\hat{z}^{(k)}_i \quad = \quad \mathrm{sgn}\left(\sum_{b\in\partial i} A^{(k)}_{ib}\tilde{y}^{(k)}_{b\to i}\right)$$

An immediate consequence of the previous proposition is the following (noting ϕ in (10) is positive).

Corollary 1. *If the limiting relative error is sufficiently small, then*

$$\hat{z}^{(k)}_i \quad \sim \quad \mathrm{sgn}\left((2\mathsf{E}p-1)\sum_{b\in\partial i} A^{(k)}_{ib}(2p_b-1)\right) \qquad (13)$$

as $k \to \infty$.

This expression is intuitive pleasing as the reliability of the individual user b's responses A_{ib} are weighted by their own reliabilities $(2p_b-1)$ essentially learned through the iterations. Also, the overall reliability of the crowd, $\mathsf{E}(2p-1) = 2\mathsf{E}p - 1$, is a factor so that if the crowd is on average unreliable ($\mathsf{E}(2p-1) < 0$) the opposite response (sign change) of the system which favors the majority (the summation) will result.

V. Numerical Study

Considering the term $\phi := \mathsf{E}(2p-1)^2$ in (12), note that the crowd is unreliable if $\mathsf{E}p$ is close to 0.5. If $\mathsf{E}(2p-1)^2 \neq 0$ but is small, then intuitively the unreliability of the crowd can can be compensated for by sufficient correlation with the majority, *e.g.*, if (12) holds and a sufficient number of iterations k are performed, otherwise the weights $\mathbf{Y}$ (or $\tilde{\mathbf{Y}}$) may be too close to $\mathbf{0}$ giving indeterminant decisions (1).

A model of a crowdsourcing assignment with *single* meta-task was simulated using Netlogo [16], a multi-agent simulation tool. The meta-task was an aggregate of 100 subtasks/questions to be assigned to a subset of 100 users. Using the method described in Section III-A, we performed degree regular question-to-user assignment. Each user was assigned 10 questions, *i.e.*, $s = r = 10$. We generated random reliabilities p_a for each user a using normal distribution with a known mean $\mathsf{E}p$ and variance $\mathsf{V} = \mathsf{E}(p^2) - (\mathsf{E}p)^2$. Using these reliabilities, random answers were generated by each user for the questions assigned. The weights $y_{a\to i}$ for each link between user a and question i were randomly initialized with normal distribution with mean and variance equal to 1 as in [13]. We computed the values of $y_{a\to i}$ for all a and i according to (2) for $k = 15$ iterations of message passing between the users and the questions and vice-versa. For each value of $\mathsf{E}p$ and V we repeated the previous step (consisting of 15 iterations of message passing) 50 times, each time generating new answers A_{ia} for all i and a based on the reliabilities of the users. Figure 1 is a plot for a (typical) link (a, i) of $\frac{\sqrt{\hat{\mathbf{var}}_{\mathbf{a\to i}}}}{\hat{\mu}_{\mathbf{a\to i}}}$ versus iteration index (k), for different values of $\mathsf{E}p$ and V, where $\hat{\mu}_{a\to i}$ and $\hat{\mathrm{var}}_{a\to i}$ are the sample mean and sample variances respectively of the weights of edge (a, i) for a given iteration. Note that $\frac{1}{(s-1)(r-1)} = \frac{1}{81} = 0.0123$ and when $\mathsf{E}p = 0.5$ (*i.e.*, an unreliable group of users), $[\mathsf{E}(2p-1)^2]^2 = 0.008 < 0.0123$, there is no convergence, otherwise there is rapid convergence of the relative error to zero, so that the condition of Corollary 1 is met for this single meta-task experiment.

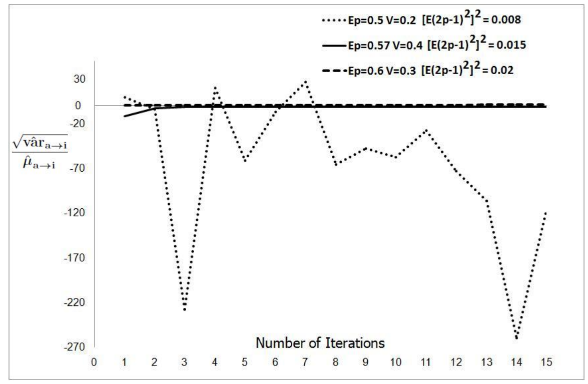

Fig. 1. Relative error $\frac{\sqrt{\hat{\mathbf{var}}_{\mathbf{a\to i}}}}{\hat{\mu}_{\mathbf{a\to i}}}$ with iterations

In our next experiment we varied r *i.e.*, the number of users assigned to a question and observed the average percentage error (the number of questions with incorrect answers derived through the weighted majority correlation method). The error values were steady state values *i.e.*, when the iterative calculation of weights $y_{a\to i}$ converged and we took the weighted correlation with the majority of users. The average was taken for 50 different random realizations of the question-to-user assignment and answers for the given value of r , E_p and V. Fig 2 shows the percentage error for different values of $\mathsf{E}p$ as it decreases with r. Note that for all of these three pairs of E_p and V, the value of $[\mathsf{E}(2p-1)^2]^2$ was well above $\frac{1}{(s-1)(r-1)}$. We observed an initial increase in the error approximately at $r = 2, 3$. A possible explanation for this could be the labeling of reliable users as unreliable (by lowering the weight $y_{a\to i}$ for the user) in the light of insufficient samples to obtain correct correlation.

VI. Similarity with LDPC Decoding via Message Passing

The consensus framework described above has a marked resemblance with message passing algorithms of belief propagation for decoding of low-density-parity-check (LDPC) codes

[1] With or without the normalizing factor $\frac{1}{\delta}$.

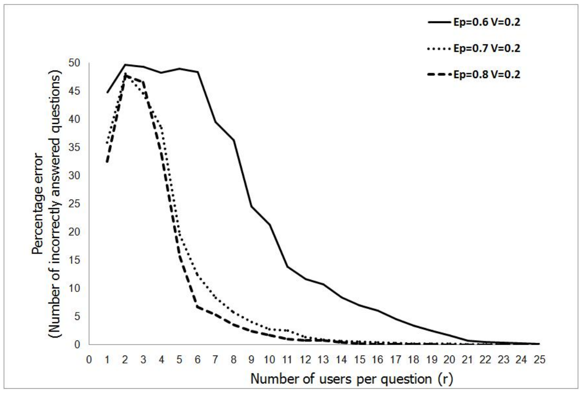

Fig. 2. Effect of r on the percentage error for different user reliabilities

[21]. LDPC codes were introduced in early sixties [9], but became popular only recently because of their ability to reach the Shannon limits on communication rates. LDPC codes typically have large lengths and rely on message passing over sparse bipartite graphs to reach a consensus on the transmitted codeword. One can think of leveraging the already mature theory of LDPC coding and decoding for the crowdsourcing model. For instance, the convergence properties of LDPC codes are normally studied under an independence assumption, *i.e.*, that the messages received by each node are independent. This assumption is true for the first few iterations defined by the "girth length" of the codes, which is the length of the smallest cycle in the graph. In our model, girth length will depend on the question-to-user assignment.

Consider a model where the weights $y_{a\to i} \in [0,1]$ because instead of (2),

$$y_{a\to i}^{(k)} = \frac{1}{s-1} \sum_{j\in\partial^{-1}a\backslash i} \text{sgn}^{+}\left(A_{ja}\ \text{sgn}\left(\sum_{b\in\partial j\backslash a} A_{jb} y_{b\to j}^{(k-1)}\right)\right),$$

where $\text{sgn}^{+}(a) = 1$ if $a > 0$ and 0 otherwise. Here, $x_{j\to b}$ gives the answer based on the expected value of answers given by all users $a \in \partial j\backslash b$, and $y_{a\to i}$ is the posterior probability that a answers i correctly given that we know the correct answers for all questions $j \in \partial^{-1}a\backslash i$. Alternatively, soft decisions can be used, *e.g.*, by defining $x_{j\to b}$ as the log-likelihood of the answer being 0 or 1. The question nodes compute the user-reliability expectations, while the user nodes maximize the log-likelihood of the observed answers over their (estimated) reliabilities. So, this is similar to the Expectation-Maximization (EM) algorithm [22].

The use of EM algorithm is natural in this scenario since we have to estimate a set of decision variables (correct answers) along with latent variables (the user reliabilities) [8]. Apart from the user reliabilites, one can also think of considering difficulty of the tasks as another set of latent variables [23]. The M-step of the EM algorithm poses some computational challenge and most of the known work in this area seeks to find a solution to the maximization step by using softwares that use numerical optimization techniques. From this perspective the message passing framework can be viewed as an alternate local or distributed optimization technique that takes node-by-node decisions iteratively. It will be interesting to find the cost of using such a framework in terms of the loss in optimality.

VII. Summary

This paper studied iterative, unsupervised crowdsourcing frameworks wherein the weights of users' answers are determined by correlating their responses to the majority. We considered the case of a single meta-task and multiple independent meta-tasks, deriving an asymptotic result for the latter. Numerical experiments for multiple iterations on a single meta-task show that the iteration does not converge when the the crowd is unreliable, but rapid convergence otherwise results. Finally, we briefly described how these crowdsourcing frameworks are related to LDPC decoding and EM.

References

[1] Amazon Mechanical Turk. http://www.mturk.com
[2] Z.D. Bai. Methodologies in spectral analysis of large dimensional random matrices, a review. *Statistica Sinica* **9**:611-677, 1999.
[3] K.C. Border. Fixed Point Theorems with Applications to Economics and Game Theory. Cambridge University Press, London, 1985.
[4] Crowdsourcing Seminar. http://husk.eecs.berkeley.edu/courses/cs298-52-sp11/index.php/Main_Page
[5] M.S. Bernstein, G. Little, R.C. Miller, B. Hartmann, M.S. Ackerman, D.R. Karger, D. Crowell, and K. Panovich. Soylent: A word processor with a crowd inside. In *Proc. ACM Symp. User Interface Software and Tech.*, New York, NY, 2010.
[6] Casting Words. http://castingwords.com
[7] Crowd Flower. http://crowdflower.com
[8] A.P. Dawid and A.M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *J. Royal Stat. Soc. C* **28**(1):20-28, 1979.
[9] G. Gallager. Low Density Parity-Check Codes. In *MIT Press*, Cambridge, MA. 1963.
[10] B. Golub and M.O. Jackson. How homophily affects learning and diffusion in networks. Feb. 2009, available at http://arxiv.org/PS_cache/arxiv/pdf/0811/0811.4013v2.pdf
[11] Leah Hoffman. Crowd Control. *Commun. of the ACM* **52**(3):16-17, 2009.
[12] P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *Proc. ACM SIGKDD Workshop on Human Computation*, New York, NY, 2010.
[13] D.R. Karger, S. Oh and D. Shah. Iterative Learning from a Crowd (abstract). In *Proc. Interdisciplinary Workshop on Information and Decision in Social Networks*, MIT, May 31 - June 1, 2011. http://wids.lids.mit.edu/wids_program_final.pdf p.32-33.
[14] A. Kulkarni, M. Can, B. Hartmann. *Turkomatic: Automatic, Recursive Task and Workflow Design for Mechanical Turk*. (Poster) In *Proc. AAAI Human Computation Workshop (HCOMP)*, 2011.
[15] Marchenko-Pastur Distribution. Available at http://en.wikipedia.org/wiki/Marchenko-Pastur_law
[16] NetLogo itself: Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
[17] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G.M. Voelker and S. Savage. Understanding CAPTCHA-Solving from an Economic Context. In *Proc. USENIX Security Symposium*, Washington, DC, August 2010.
[18] V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from Crowds. In *Journal of Machine Learning Research*, 11(7):12971322, 2010.
[19] Y. Singer and M. Mittal. Pricing Mechanisms for Online Labor Markets. In *Proc. AAAI Human Computation Workshop (HCOMP)*, 2011.
[20] Soylent. http://projects.csail.mit.edu/soylent/
[21] A. Shokrollahi. LDPC codes: An introduction. In *Coding, cryptography and combinatorics, Progress in Computer Science and Applied Logic*, 2004
[22] C. Tomasi. Estimating Gaussian Mixture Densities with EM A Tutorial. http://citeseer.nj.nec.com
[23] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 22:20352043, 2009.