Crowdsourcing Control: Moving Beyond Multiple Choice

Christopher H. Lin Mausam Daniel S. Weld
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
{chrislin,mausam,weld}@cs.washington.edu

Introduction: Crowdsourcing marketplaces (*e.g.*, Amazon Mechanical Turk) continue to rise in popularity. Unfortunately, workers come with hugely varied skill sets and motivation levels. Ensuring high quality results is a serious challenge for all requesters.

A key drawback of prior decision-theoretic approaches (Dai, Mausam, and Weld 2010; 2011) to quality control is the restriction to multiple choice questions, *i.e.*, jobs where every alternative answer is known in advance. While many tasks can be formulated in a multiple-choice fashion (*e.g.* n-ary classification), there are a large number of tasks with an unbounded number of possible answers. A common example is completing a database with workers' help, *e.g.*, asking questions such as "Find the mobile phone number of Acme Corporation's CEO."

Unfortunately, adapting multiple-choice models for these scenarios is not straightforward, because of the difficulty with reasoning about unknown answers. Requesters, therefore, must resort to using a majority-vote, a significant hindrance to achieving quality results (Dai, Mausam, and Weld 2010; 2011; Whitehill et al. 2009). Our paper tackles this challenging problem. We first create a probabilistic, generative model for tasks where workers are free to give any answer. We then present a decision-theoretic controller, LAZY-SUSAN, that uses our model to dynamically infer answers to these tasks, and finally show that it obtains a better cost-quality tradeoff compared to an agent that uses majority-voting.

Background: First, we review the Chinese Restaurant Process (Aldous 1985), a discrete-time stochastic process that generates an infinite number of labels ("tables"). Intuitively, the process may be thought of modeling sociable customers who, upon entering the restaurant, decide between joining other diners at a table or starting a new table. The greater the number of customers sitting at a table, the more likely new customers will join that table.

Formally, a Chinese Restaurant $R = (T, f, \theta)$ is a set of occupied tables $T = \{t_1, \ldots, t_n | t_i \in \mathbb{N}\}$, a function $f : T \to \mathbb{N}$ that denotes the number of customers at each table t_i, and a parameter $\theta \in \mathbb{R}^+$. A new customer can either choose to sit at one of the occupied tables, or at a new empty table. The probability that he chooses to sit at table $t \in T$ is $C_R(t) = \frac{f(t)}{N+\theta}$ where $N = \sum_{t \in T} f(t)$ is the total number of customers in the restaurant. The probability that he chooses to sit at any new unoccupied table, or, equivalently, the probability that he chooses not to sit at an occupied table is $NT_R = \frac{\theta}{N+\theta}$.

Probabilistic Model: We seek to develop a probabilistic model of workers on tasks that have a countably infinite solution space. Our model extends Dai *et al.*'s model (2010). Let $d \in [0, 1]$ be the difficulty of a given task and $\gamma_i = [0, \infty)$ be worker i's error parameter. We define the accuracy of that worker for that given task to be: $a(d, \gamma_i) = (1 - d)^{\gamma_i}$.

Let b_i be the answer that is provided by the i^{th} worker. It is determined by the correct answer v, d, and γ_i. A good model must consider correlated errors (Grier 2011). For instance, if a task encourages workers to use Google, the responses would likely by correlated with the search results. Thus, b_i is also determined by all previous responses $b_1, \ldots, b_{i-1}$. Only the responses are observable variables.

Let $\theta \in \mathbb{R}^+$ denote the task's *bandwagon coefficient*. The parameter θ encodes the concept of the "tendency towards a common wrong answer." If θ is high, then workers who answer incorrectly will tend to provide new, unseen, incorrect answers, suggesting that the task does not have "common" wrong answers. Contrastingly, if θ is low, workers who answer incorrectly will tend toward the same incorrect answer, suggesting that the task lends itself to the same mistakes.

For ease of expression, let $\mathbf{B_i}$ be the multiset of answers that workers $1, \ldots, i$ provide. Let $\mathbf{A_i}$ be the set of unique answers in $\mathbf{B_i}$. The probability that the $i + 1^{th}$ worker's ballot is correct is $P(b_{i+1} = v | d, v, \mathbf{B_i}) = a(d, \gamma_{i+1})$.

To define the probability space of wrong answers we use the Chinese Restaurant Process. Let $f(a) = |\{b \in \mathbf{B_i} | b = a\}|$, and let $R_{i,v} = (\mathbf{A_i} \setminus \{v\}, f, \theta)$ be a Chinese Restaurant Process. Then, the probability that the worker returns a previously seen incorrect answer, $y \in \mathbf{A_i} \setminus \{v\}$ is $P(b_{i+1} = y | d, v, \mathbf{B_i}) = (1 - a(d, \gamma_{i+1}))C_{R_{i,v}}(y)$. Finally, the probability that the worker returns an unseen answer is $P(b_{i+1} = u | d, v, \mathbf{B_i}) = (1 - a(d, \gamma_{i+1}))NT_{R_{i,v}}$. Here, u represents whatever the worker returns as long as $u \notin \mathbf{A_i}$. The model cares only about whether it has seen a worker's answer before, not what it actually turns out to be.

A Decision-Theoretic Agent: We now detail LAZYSUSAN, which uses our model to infer correct answers by dynamically requesting more observations as necessary. At each time-step, it can either stop and submit the most likely answer, or it can create another job and receive another response to the task from another crowdsourced worker.

To determine the agent's policy, we define its *agent state* $\mathbf{S}$, which at time i, is the set of tuples, $\mathbf{S} = \{(v, d) | v \in \mathbf{A_i} \cup \{\perp\} \wedge d \in [0, 1]\}$. $\perp$ represents the case when the true answer has not been seen by the agent so far.

Let $k = |\mathbf{A_i}|$. For LAZYSUSAN to update its posterior belief about its agent state $P(v, d|\mathbf{B_i}; i, k)$ after it receives its i^{th} ballot b_i, it requires $P(v|d; i, k)$ and $P(d; i, k)$.

Notice that for all $a \in \mathbf{A_i}$, we do not know $P(v = a|d; i, k)$. However, they must be all the same, because knowing the difficulty of the task gives us no information about the correct answer. Otherwise, we define: $P(v = \perp |d; i, k) := d^i$. This definition is reasonable since intuitively, as the difficulty of the task increases the more likely workers have not yet provided a correct answer and vice versa. Next, we choose to model $P(d; i, k) \sim Beta(\alpha, \beta)$ and define $\alpha \geq 1$ and $\beta \geq 1$ based on i, k, and θ.

To determine what actions to take, LAZYSUSAN needs to calculate the utility of its beliefs, which it does using what it currently believes the correct answer to be. LAZYSUSAN selects its actions at each time step by computing an l-step lookahead by estimating the utility of each possible sequence of l actions. If the l^{th} action is to request another response, then it will assume that it submits an answer on the $l + 1^{th}$ action. In our experiments, we use a lookahead depth of 3.

After submitting an answer, LAZYSUSAN updates its records about all the workers who participated in the task.

Experiments: We compare LAZYSUSAN to an agent that uses majority-voting, MV, using real responses generated by Mechanical Turk workers. We test these agents with 134 SAT Math questions.

We find that the workers on Mechanical Turk are surprisingly capable at solving math problems. At an average cost of 5.46 ballots per task, MV achieves a 95.52% accuracy. LAZYSUSAN almost completely eliminates the error made by MV, achieving a 99.25% accuracy at an average cost of 5.17 ballots per task.

Qualitative Discussion We examine an example sequence of actions LAZYSUSAN made for one task. In total, it requested 14 ballots, and received the following responses: 215, 43, 43, 43, 5, 215, 43, 3, 55, 43, 215, 215, 215, 215. Since MV takes the majority of 7 votes, it infers the answer incorrectly to be 43. LAZYSUSAN on the other hand, uses its knowledge of correlated answers as well as its knowledge from previous tasks that the first three workers who responded with 43 were all relatively poor workers compared to the first two workers who claimed the answer is 215. So even though a clear majority of workers preferred 43, LAZYSUSAN was not confident about the answer. While it cost twice as much as MV, the cost was a worthy sacrifice.

Related Work: Modeling repeated labeling in the face of noisy workers when the label is assumed to be drawn from a known *finite* set has received significant attention (Romney, Weller, and Batchelder 1986; Sheng, Provost, and Ipeirotis 2008; Raykar et al. 2010; Whitehill et al. 2009; Dai, Mausam, and Weld 2010; 2011; Lin, Mausam, and Weld 2012; Welinder et al. 2010; Kamar, Hacker, and Horvitz 2012; Parameswaran et al. 2010; Karger et al. 2011; Snow et al. 2008).

Acknowledgements: This work was supported by the WRF / TJ Cable Professorship, Office of Naval Research grant N00014-12-1-0211, and National Science Foundation grants IIS 1016713 and IIS 1016465.

References

Aldous, D. J. 1985. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII 1983*, volume 1117 of *Lecture Notes in Mathematics*. Springer Berlin / Heidelberg. 1–198. 10.1007/BFb0099421.

Dai, P.; Mausam; and Weld, D. S. 2010. Decision-theoretic control of crowd-sourced workflows. In *AAAI*.

Dai, P.; Mausam; and Weld, D. S. 2011. Artificial intelligence for artificial intelligence. In *AAAI*.

Grier, D. A. 2011. Error identification and correction in human computation: Lessons from the WPA. In *HCOMP*.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS*.

Karger, D. R.; Oh, S.; ; and Shah, D. 2011. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Allerton*.

Lin, C. H.; Mausam; and Weld, D. S. 2012. Dynamically switching between synergistic workflows for crowdsourcing. In *AAAI*.

Parameswaran, A.; Garcia-Molina, H.; Park, H.; Polyzotis, N.; Ramesh, A.; and Widom, J. 2010. Crowdscreen: Algorithms for filtering data with humans. In *VLDB*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; and Valadez, G. 2010. Learning from crowds. *Journal of Machine Learning Research* 11:1297–1322.

Romney, A. K.; Weller, S. C.; and Batchelder, W. H. 1986. Culture as consensus: A theory of culture and informant accuracy. *American Anthropologist* 88(2):313 – 338.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 254–263.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Whitehill, J.; Ruvolo, P.; Bergsma, J.; Wu, T.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*.