real responses generated by Mechanical Turk workers. We develop a “triangle” task (Figure 5) that presents workers with a triangle drawn on a grid, and asks them to find the area of the triangle, rounded down. We posted 200 of these tasks and solicited 5 responses for each. These tasks are difficult since many of the responses are off by 1. Our EM algorithm achieves an accuracy of 65.5% while MV achieves an accuracy of 54.1%.

6 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 et al. [Romney et al., 1986] are one of the first to incorporate a worker accuracy model to improve label quality. Sheng et al. [Sheng et al., 2008] explore when it is necessary to get another label for the purpose of machine learning. Raykar et al. [Raykar et al., 2010] propose a model in which the parameters for worker accuracy depend on the true answer. Whitehill et al. [Whitehill et al., 2009] and Dai et al. [Dai et al., 2010] address the concern that worker labels should not be modeled as independent of each other unless given problem difficulty. Welinder et al. [Welinder et al., 2010] design a multidimensional model for workers that takes into account competence, expertise, and annotator bias. Kamar et al. [Kamar et al., 2012] extracts features from the task at hand and use Bayesian Structure Learning to learn the worker response model. Parameswaran et al. [Parameswaran et al., 2010] conduct a policy search to find an optimal dynamic control policy with respect to constraints like cost or accuracy. Karger et al. [Karger et al., 2011] develops an algorithm based on low-rank matrix approximation to assign tasks to workers and infer correct answers, and analytically prove the optimality of their algorithm at minimizing a budget given a reliability constraint. Snow et al. [Snow et al., 2008] show that for labeling tasks, a small number of Mechanical Turk workers can achieve an accuracy comparable to that of an expert labeler. None of these works consider tasks that have an infinite number of possible solutions.

For more complex tasks that have an infinite number of possible answers, innovative workflows have been designed, for example, an iterative improvement workflow for creating complex artifacts [Little et al., 2009], find-fix-verify for an intelligent editor [Bernstein et al., 2010], and others for counting calories on a food plate [Noronha et al., 2011].

An AI agent makes an efficient controller for these crowdsourced workflows. Dai et al. [Dai et al., 2010, Dai et al., 2011] create a POMDP-based agent to control an iterative improvement workflow. Shahaf and Horvitz [Shahaf and Horvitz, 2010] develop a planning-based task allocator to assign subtasks to specific humans or computers with known abilities. We [Lin et al., 2012] create a POMDP-based agent to dynamically switch between workflows.

Weld et al. [Weld et al., 2011] discuss a broad vision for the use of AI techniques in crowdsourcing that includes workflow optimization, interface optimization, workflow selection and intelligent control for general crowdsourced workflows. Our work provides a more general method for intelligent control.

7 Conclusion & Future Work

This paper introduces LazySUSAN, an agent that takes a decision-theoretic approach to inferring the correct answer of a task that can have a countably infinite number of possible answers. We extend the probabilistic model of [Dai et al., 2010] using the Chinese Restaurant Process and use l-step lookahead to approximate the optimal number of crowdsourcing jobs to submit. We also design an EM algorithm to jointly learn the parameters of our model while inferring the correct answers to multiple tasks at a time. Live experiments on Mechanical Turk demonstrate the effectiveness of LazySUSAN. At comparable costs, it yields an 83.2% error reduction compared to majority vote, which is the current state-of-the-art technique for aggregating responses for tasks of this nature. Live experiments also show that that our EM algorithm outperforms majority-voting on “triangle” tasks.

In the future, we would like to address the ambiguity between high difficulty, low $\theta$ problems and low difficulty problems. We also hope to develop a generative model that does not change as responses are gathered from workers. We also hope to extend the ability of LazySUSAN to solving tasks that have multiple correct answers. Indeed, workers oftentimes provide the same answer in different forms (e.g., in different units). Other questions may have several answers (e.g., top executives often carry two mobiles). While multiple correct answers may be reduced with crisply-written instructions, an improved model may also prove useful.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. 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.