Dynamic Worker-Task Assignment for High-Quality Task Results with ML Workers

Yu Yamashita
University of Tsukuba
s1811547@s.tsukuba.ac.jp

Masaki Kobayashi
University of Tsukuba
makky@klis.tsukuba.ac.jp

Kei Wakabayashi
University of Tsukuba
kwakaba@slis.tsukuba.ac.jp

Atsuyuki Morishima
University of Tsukuba
mori@slis.tsukuba.ac.jp

Abstract

How to obtain higher quality task results with a fixed budget (labor cost) is one of the most important problems in practical crowdsourcing. We approach this problem by (1) dynamically choosing appropriate sets of tasks and workers during the worker-task assignment process and (2) introducing an ML worker that performs tasks as if it was a human worker. Our approach, named Dynamic Worker and Task Assignment with ML workers (DWTAML), chooses tasks and workers during the worker-task assignment process, based on the estimation of the reliability of workers and the confidence of the task results at each moment. It extends the Dawid&Skene model and IEThresh. The results of an extensive set of experiments showed that DWTAML is effective in most cases.

Introduction

Quality management of task results has remained one of the critical concerns in crowdsourcing and many methods have been proposed in the literature. Some of them are compatible with practical scenarios and easy to implement on the current crowdsourcing platforms but not necessarily cost-effective. Others are more sophisticated, but not necessarily easy to implement on the crowdsourcing platforms today.

This paper proposes a method, named Dynamic Worker and Task Assignment with ML workers (DWTAML), that can be used in the practical setting and also more cost-effective than existing ones. The method assumes a practical scenario today, where the followings hold (1) requesters want to obtain better results of a set of tasks for a fixed budget, (2) they do not want to do a batch assignment to a fixed set of workers (Chen, Lin, and Zhou 2013), but want to do an online task assignment where we assign the best task to a worker who requests a task in the situation where workers join and leave while other tasks are processed.

DWTAML has three features: (1) **Dynamic blacklisting** that dynamically estimates the reliability of workers and updates the blacklist to exclude workers with low reliability from the worker pool by extending IEThresh (Donmez, Carbonell, and Schneider 2009). (2) **Task prioritization** that dynamically estimates the confidence of task results by applying the Dawid&Skene model (Dawid and Skene 1979), and assigns a task with an uncertain result to the worker. (3) **ML worker participation** that assigns an ML algorithm as a 'worker' tasks so that we can apply the Dawid&Skene model in the situation we have the answer from one human worker only for each task.

We conducted some simulation experiments. The results showed that DWTAML is effective in most cases and especially effective with a small budget.

Proposed Method: DWTAML

We have as input (1) a fixed budget (labor cost) C. For simplicity, we assume that the monetary cost for a worker to perform a task is one, i.e., we will have C task assignments. We also have (2) the set I of tasks and (3) the set J of labels that will be assigned to data objects shown in the tasks. The output is the estimated true label, denoted by $\hat{A} = \{\hat{a}_i\}_{i \in I}$.

Avoid assigning the tasks to the workers with low reliability. Interval Estimation (IE) (Kaelbling 1990) (Moore and Schneider 1996) estimates the upper bound of confidence interval $UI_{k,t}$ of a worker k at the time when t tasks were performed, based on the agreement ratio between k's answers and correct ones so far. In DWTAML, we used Dawid&Skene model to estimate correct answers from the workers' answers $r_{i \in I}^{k \in K}$ where K is the set of workers who performed the tasks. Each worker k starts with $UI_{k,t} = 1.0$ (as long as we do not have enough number of (i.e., two) answers from him or her for obtaining t-distribution) and the values become more precise as the number of answers increases. We make a blacklist of workers whose upper bounds are lower than a threshold θ. For example, we can use the average of the upper bound of all workers:

$$BW_t = \{k | UI_{k,t} < \sum_{k' \in K} UI_{k',t} / |K|\} \quad (1)$$

where BW_t denotes a set of workers to be in the blacklist at t. It is so difficult to set the good threshold, but as we showed later, the average threshold worked well in our preliminary experiments.

Assign tasks with uncertain results to workers. Dawid&Skene model estimates the posterior distribution of

Table 1: The accuracy of EM, DWA, DWTA, DWTAML(LR), and DWTAML(RF)

worker	$\beta(8,2)$			$\beta(3,2)$			$\beta(2,3)$			Hammer-Spammer			Real		
budget	6000	9000	12000	6000	9000	12000	6000	9000	12000	6000	9000	12000	3000	4500	6000
EM	0.832	0.914	0.944	0.576	0.679	0.771	0.363	0.414	0.433	0.890	0.961	0.980	0.502	0.548	0.579
DWA	0.832	0.930	0.962	0.576	0.728	0.834	0.363	0.432	0.476	0.890	0.998	1.0	0.502	0.560	0.610
DWTA	0.832	0.955	0.972	0.576	0.817	0.912	0.363	0.468	0.561	0.890	1.0	1.0	0.502	0.577	0.607
DWTAML(LR)	0.909	0.955	0.974	0.708	0.845	0.917	0.409	0.442	0.479	0.999	0.999	0.999	0.572	0.596	0.612
DWTAML(RF)	0.895	0.940	0.963	0.712	0.804	0.901	0.434	0.459	0.482	0.997	0.999	0.999	0.562	0.596	0.616

the true labels $e_{ij} = p(a_i = j|r_i^{k_1}, r_i^{k_2}, \ldots, r_i^{k_{|K|}})$. We take $\max_{j \in J} e_{ij}$ as the confidence of task i. We assign the task i' with the least confidence to the worker, where i' is:

$$i' = \arg\min_i (\max_{j \in J} e_{ij}) \quad (2)$$

Assign an ML worker as the second worker in each task. Dawid&Skene model requires more than one worker for each task to estimate the confidence of task results. DWTAML employs an ML algorithm as if it was another human worker so that it is always able to apply the David&Skene model. Therefore, we call such an algorithm to be used as a worker an ML worker. An ML worker learns the answers estimated by Dawid&Skene model at $t-1$ and outputs the labels of all tasks in I. Note that our purpose of using ML algorithms is to obtain another worker, not to obtain a good classifier. We also note that there are ML algorithms whose outputs are prediction probabilities. We interpret the Dawid&Skene model in a probabilistic setting in order to deal with probabilistic answers from the ML worker.

Algorithm 1 summarizes DWTAML. It first assigns every task in I to workers, assuming that $BW_t = \phi$ for $0 \leq t < |I|$ (Line 1). Then the ML worker learns the answers and outputs labels (Line 2). Next, DWTAML performs David&Skene model to compute $\hat{A}$ at that moment. At last, update the blacklist and the assigned task (Line 4). After receiving one answer to every task in I, the following iteration proceeds each time it receives the answer of a task from a worker; Each iteration updates the blacklist of workers and the assigned task. If a worker (not in the blacklist) requests a task, the task which has the results with the least confidence is assigned to the worker as long as $t < C$.

Algorithm 1 DWTAML

Input: A fixed budget C, the set I of tasks, and the set J of labels
Output: The estimated true labels $\hat{A}$
1: Have everytask $i \in I$ performed by workers; $t \leftarrow |I|$
2: Let ML worker learn the answers and outputs the labels (probabilities) for all tasks $i \in I$
3: Perform David&Skene; Set $\hat{a}_i \leftarrow \arg\max_j(e_{ij})$
4: Update the blacklist and the assigned task with Eq 1 and 2
5: **for** each time a task result is submitted; $t++$ **do**
6: Let the ML worker learn $\hat{A}$ and output the labels (probabilities) for all tasks $i \in I$
7: Perform Dawid&Skene; Set $\hat{a}_i \leftarrow \arg\max_j(e_{ij})$
8: Update the blacklist and the assigned task with Eq 1 and 2
9: **end for**

Experiments

We conducted some simulation experiments.

Tasks and Workers. We used two distributions of worker accuracies: (1) Synthesized distribution (Beta distribution model and Hammer-Spammer model) and (2) Real distribution taken from Amazon Mechanical Turk (AMT). In Hammer-Spammer model, 60 percent of all workers are hammers (always respond to the correct labels) and 40 percent of workers are spammers (respond to the labels uniformly at random). In Beta distribution model, the worker's reliability distribution follows the defined beta distribution. We use $\beta(8,2)$ (average accuracy 0.8), $\beta(3,2)$ (average accuracy 0.6) and $\beta(2,3)$ (average accuracy 0.4). For the task, we use CIFAR-10 (three classifications of Dog, Horse, and Deer) and Fashion-MNIST (three classifications of Coat, Pullover, and Shirt). CIFAR-10 was used for the synthesized distribution and Fashion-MNIST was used for the real distribution. It has 3000 tasks (images) and 200 workers in the synthesized distribution. It has 1500 tasks (images) and 280 workers in the real distribution.

Budget. In CIFAR-10, we set the budget to 6000 (redundancy 2), 9000 (redundancy 3), and 12000 (redundancy 4). In Fashion-MNIST, we set the budget to 3000 (redundancy 2), 4500 (redundancy 3), and 6000 (redundancy 4).

Comparative Experiments. (1) EM: Dawid&Skene model. (2) DWA: It omits steps 2, 6, 7 in DWTAML algorithm. The task i' to be assigned is decided in a round-robin manner in (1) EM and (2) DWA. (3) DWTA: It omits steps 2, 7 in DWTAML algorithm. (4) DWTAML (LR): ML algorithm is Logistic Regression. (5) DWTAML (RF): ML algorithm is Random Forest. Logistic Regression and Random Forest implemented by scikit-learn with default hyperparameters.

Result. Table 1 shows the accuracy of EM, DWA, DWATA, DWTAML(LR), and DWTAML(RF) in some simulation experiments. We found that (1) dynamic blacklisting and (2) task prioritization can improve task results from almost DWTA $>$ DWA $>$ EM. In addition, we found that (3) ML worker participation can improve task results with a small budget. At last, we found that DWTAML can improve task results in most cases and especially effective with a small budget.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR16E3 including AIP challenge, Japan.

References

Chen, X.; Lin, Q.; and Zhou, D. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *International conference on machine learning*, 64–72.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28(1):20–28.

Donmez, P.; Carbonell, J. G.; and Schneider, J. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 259–268.

Kaelbling, L. P. 1990. Learning in embedded systems.

Moore, A. W., and Schneider, J. G. 1996. Memory-based stochastic optimization. In Touretzky, D. S.; Mozer, M. C.; and Hasselmo, M. E., eds., *Advances in Neural Information Processing Systems 8*. MIT Press. 1066–1072.