OnTac: Online Task Assignment for Crowdsourcing

Zhe Yang*, Zhehui Zhang*, Yuting Bao*, Xiaoying Gan*, Xiaohua Tian*†, Xinbing Wang*
*Dept. of Electronic Engineering, Shanghai Jiao Tong University, China
†National Mobile Communication Research Laboratory, Southeast University, China
{youryz, qiaomai, smiling-forever, ganxiaoying, xtian, xwang8}@sjtu.edu.cn

***Abstract*—How to integrate labels from multiple labelers in order to obtain an accurate estimate of the ground truth is a major topic of crowdsourcing. One challenging issue is that, the labelers' abilities may vary significantly and the tasks distinguish each other in difficulties. Moreover, for a crowdsourcing system, task distributors have no idea in advance how many labels will be enough for each task. Consequently, an online task assignment mechanism based on the labeler expertise and question heterogeneousness becomes necessary. In this paper, we present such an online task assignment algorithm based on a probabilistic model consisting of both labeler abilities and question difficulties. We apply the online EM (Expectation Maximization) algorithm to make online estimations of system parameters, based on which we assign tasks adaptively. A series of simulation results have been demonstrated to show that our proposed scheme outperforms the conventional EM algorithm in efficiency and accuracy.**

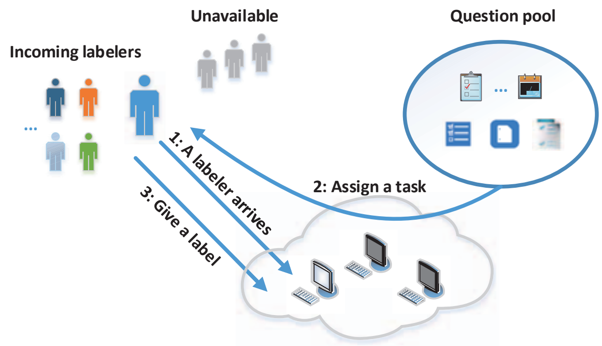

Fig. 1. A common scenario of crowdsourcing platforms.

I. Introduction

Crowdsourcing based labeling has been widely applied to many domains such as computer graphics, medical diagnoses and astronomy[1], where the basic idea is to leverage intelligence of the crowdsourcing labelers to label certain objects. A common crowdsourcing scenario is shown in Fig. 1. The current labeler is assigned with a series of tasks. He provides a number of corresponding labels and gets paid, after which this labeler becomes unavailable and does not come back for future assignments. In this scenario, task distributors are not able to predict how many labelers come in at one moment. Most of the existing works are unable to make estimates of the ground truth until the moment they receive a preset number of labels. This preset number may be either insufficient or redundant, causing a lack of accuracy or a waste of budget.

Meanwhile, it is common that the labelers vary widely in expertise, and that the difficulty of questions can also be diverse. On the one hand, these diversities introduce uncertainty to the estimation of ground truth and thus make the estimates less reliable. On the other hand, if we make adaptive assignment — assign each labeler with questions he is competent in, we can make full use of these diversities to obtain a better estimate.

However, previous works focus mainly on the integration of labels, with task assignment often overlooked[2]. As a matter of fact, an approach that considers both the labeler expertise and question difficulty remains to be studied.

Hence, how to exploit the diversity of labeler abilities and question difficulties to reach a more accurate estimation with a minimum budget is a crucial issue in the domain of crowdsourcing. In this paper we propose an approach OnTac and try to resolve this issue — Online Task Assignment for crowdsourcing.

OnTac estimates the ground truth as well as labeler abilities and question difficulties based on an online EM algorithm[3], and assigns the labelers with appropriate tasks. This algorithm consists of two phases — the assignment phase and the inference phase. *In the assignment phase*, OnTac selects a suitable question from the question pool based on the estimated parameters and asks the current labeler to provide a label. This adaptive assignment targets the proper tasks and thus enhances the estimation accuracy. *In the inference phase*, OnTac makes an online estimation of labeler competence, question difficulty and ground truth. An online inference guarantees that we can determine when we can be convinced of a ground truth in an online manner. The online algorithm also ensures it to fit the existing crowdsourcing platforms, in that the number of incoming labelers at one point is unknown. Based on the inference, OnTac eliminates incapable labelers and trusts the competent ones, which encourages the competent labelers.

The rest of the paper is organized as follows. Section II presents some related works, the probabilistic model we apply, and the original inference algorithm. We describe our algorithm OnTac in section III. The experiment evaluation part is in section IV. And finally section V is the conclusion part.

II. Preliminary

A. Literature Review

The problem of integrating labels is first considered by Dawid and Skene[4]. The basic idea of their algorithm is to infer the ground truth by EM algorithm. In [5], [6], [7], [8],

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

[9], [10], [11], the authors propose approaches focusing on obtaining reliable labels. In [12], Welinder *et al.* propose a model of the labeling process which includes label uncertainty, as well as a measure of the annotator ability, where an online algorithm is also designed without extending the model to facilitate task assignment and label accuracy improvement. Furthermore, the mechanism of calling back former labelers to do assignments is not applicable in many crowdsourcing scenarios. One cannot identify and reuse the labelers. Despite the extension of EM algorithm, several works put emphasis on task assignment. In [2], the authors propose an adaptive task assignment and label inference algorithm, considering both labeler ability and question types. The question types, however, are preassigned instead of updated from labels received. And the process of assigning golden standard questions is a waste of budget. In [13], the authors develop an active learning approach to assign the proper tasks. However, they also utilize a mechanism of calling back former labelers. In the same time, the selection of tasks and the selection of labelers are separate.

Recent works consider the difficulty of questions and put it in the model. In [14], Whitehill *et al.* introduce a model in which the ground truth of each task, the difficulty of the questions, and the expertise of each labeler are modeled. However, related works in this domain rely on offline processing [15], [5]. In [5], the authors design a crowdsourcing system minimizing the budget without loss of the accuracy, where the reliability of labelers are inferred with Belief Propagation. However, the proposed algorithm relies on batch processing, which means online task assignment is not available. Meanwhile, they do not consider the diversity of questions.

In this paper,we propose OnTac, an online task assignment approach, which differs from others since it updates inference timely with an online EM algorithm. In addition, the adaptive task assignment procedure guarantees that each labeler obtains appropriate tasks.

B. Probabilistic Model

In this part we present the probabilistic model [14] on which our algorithm is based. The model is shown in Fig. 2.

A requester has a question pool $\mathcal{T}$. The questions are of binary annotations. Namely, the true label of each question is either 0 or 1. The questions are heterogeneous, each question $j \in \mathcal{T}$ has a certain difficulty. At some point a labeler $i \in \mathcal{W}$ comes, giving a label to each question assigned to him, and getting a reward for each label. We assume that each label contributes the same amount of reward for labelers.

Every coming labeler i has a value α_i, ranging from $-\infty$ to $+\infty$, denoting the labeler's expertise. And every question j in the question pool has a value β_j, ranging from 0 to $+\infty$, denoting the difficulty of this question. The higher α_i is, the more competent labeler i is. A labeler with negative ability is defined as adversarial. We inverse the labels given by adversarial labelers to make full use of all the labels collected. The higher β_j is, the simpler the question is. The label given by labeler i on task j is denoted by l_{ij}.

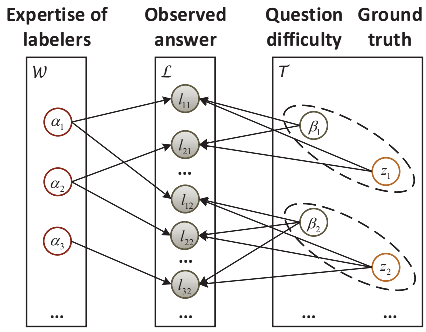

Fig. 2. Plate representation of the probabilistic model

Known the ground truth z_j, the probability that labeler i gives a correct label to question j can be expressed as:

$$P(l_{ij} = z_j) = \frac{1}{1 + e^{-\alpha_i \beta_j}}. \tag{1}$$

C. Original EM Algorithm

The original EM approach can be integrated in the inference procedure. The joint likelihood function of this problem can be expressed as:

$$p(\mathcal{L}, \mathbf{z}) = \prod_j p(z_j) \prod_{i,j} p(l_{ij}|z_j, \alpha_i, \beta_j). \tag{2}$$

The algorithm consists of two steps — E-step and M-step.

E-step We assume the estimates of parameters in the previous M-step to be $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$. Let $\mathbf{L}_j$ denote the labels received for question j, z_j denote the true label of question j, and l_{ij} denote the label given by labeler i on question j. We can calculate the posterior on the given labels as follows:

$$\begin{aligned} \hat{p}(\mathbf{z}) &= p(\mathbf{z}|\mathcal{L}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}) \\ &= \prod_j p(z_j|\mathbf{L}_j, \widehat{\boldsymbol{\alpha}}, \widehat{\beta}_j) \\ &\propto \prod_j p(z_j) \Big(\prod_i p(l_{ij}|\widehat{\alpha}_i, \widehat{\beta}_j) \Big). \end{aligned} \tag{3}$$

M-step Based on the current estimate $\hat{p}(\mathbf{z})$ and labels received $\mathcal{L}$, the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are estimated by maximizing the expectation of the joint log-likelihood function. We call it the auxiliary function $Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

$$Q(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbb{E}\Big[\sum_j \ln p(z_j) + \sum_{i,j} \ln p(l_{ij}|z_j, \alpha_i, \beta_j) \Big] \tag{4}$$

The algorithm is iterated till convergence. To echo the online EM algorithm we call the expectation of the joint log-likelihood function sufficient statistics, denoted by μ.

III. OnTac Algorithm

In this section, we are going to introduce our algorithm OnTac, based on an online EM algorithm[3]. With the aid of online estimation, a task assignment strategy is provided, which assigns proper questions to different labelers. The online EM algorithm is derived from the original EM algorithm, or the conventional EM algorithm.

A. Online Estimation

For the current crowdsourcing platforms, task distributors do not know how many labelers will come and give the corresponding labels at one moment. And with the conventional EM algorithm, task distributors cannot get informed how many labels are enough for a certain question. As the conventional EM algorithm does not select experts from random labelers, the algorithm allocates a same number of questions for the two kinds of labelers, leading to a same amount of reward, which is not a reasonable incentive.

We develop an online algorithm to estimate the model variables and parameters based on the online EM algorithm, proposed by Liang *et al.* in [3]. A single set of sufficient statistics μ, representing the previous statuses, is stored and updated with each incoming observed example. For the i-th labeler's j-th label, we compute its sufficient statistics s_{ij}. We interpolate between μ and s_{ij} with a stepsize η_k, where k is the number of labels till now.

$$\mu = (1 - \eta_k)\mu + \eta_k s_{ij} \tag{5}$$

Stepsize η_k decreases with k increasing. To guarantee the convergence of algorithm, $\sum_{k=0}^{\infty} \eta_k = \infty$ and $\sum_{k=0}^{\infty} {\eta_k}^2 < \infty$ are sufficient. In particular, we choose a decreasing function $\eta_k = (k+2)^{-a}$, where a can take any decimal from 0.5 to 1, namely, $a \in (0.5, 1]$.

In the M-step, we use the set of sufficient statistics μ to re-estimate the parameters.

We can also update the sufficient statistics with the incoming of a small number of labels instead of one single label. We call dealing with several labels each time mini-batch update.

In our particular case, at first, we initiate all the parameters and variables — $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\mathbf{z}$ as well as μ and k. When labeler i comes, we assign him with one or several questions (see subsection III-B). Based on the estimated parameters $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{\beta}}$ and the labels received so far $\mathcal{L}$, we compute the expectation of the joint log-likelihood function $Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ according to Equations 3 and 4. By maximizing this function we obtain a new estimate of the parameters $\boldsymbol{\alpha'}$ and $\boldsymbol{\beta'}$. After that we interpolate the parameters with previous estimations, utilizing Equation 6. Then we update k and assign the labeler with some new tasks or we wait for another labeler.

$$\begin{aligned} \widehat{\boldsymbol{\alpha}} &= (1 - \eta_k)\widehat{\boldsymbol{\alpha}} + \eta_k \boldsymbol{\alpha'}, \\ \widehat{\boldsymbol{\beta}} &= (1 - \eta_k)\widehat{\boldsymbol{\beta}} + \eta_k \boldsymbol{\beta'} \end{aligned} \tag{6}$$

The adoption of online algorithm enables us to largely reduce the time of computation while maintain the level of accuracy (for some datasets augment the accuracy). The conventional EM approach calculates and maximizes the auxiliary function Q on the whole dataset while the online approach takes one (or several) observed example a time, motivated from the stochastic approximation literature. Moreover, the online approach distributes the calculation into each round, so within every round the amount of calculation is small. Note that the conventional approach demands an iteration after the estimation of parameters while the online approach does not.

Meanwhile an online estimation algorithm is more suitable for task assignment due to timely update of parameters.

B. Task Assignment Strategies

We re-construct the assignment algorithm and design three criteria to allocate tasks.

Criterion 1: If we are almost certain that the competence of a labeler is very low, we do not assign any questions to this labeler any more.

Criterion 2: Once we have estimates α_i and $\boldsymbol{\beta}$, we choose the task that makes the probability from Equation 1 closest to a certain value $\mathrm{P_c}$.

Criterion 3: If the estimated posterior of a question is close to 0.5, we make it a priority to be labeled.

Criterion 4: If we are almost certain of a posterior, we do not assign this question to future incoming labelers.

If the competence of a labeler is close to 0, we think that he contributes little information gain on the questions he labels and we eliminate this kind of labelers. We should take notice here that, the adversarial labelers, namely, the labelers with negative competence, are also considered informative.

The motivation of Criterion 2 is a trade-off. The task assigned should neither be too difficult nor too simple for the current labeler. If $\mathrm{P_c}$ is set too high, every labeler answers the simplest questions, leaving the difficult ones seldom labeled. If it is set too low, the average information gain from each label is not competitive. As the difficulties of questions and the expertise of labelers are diverse, we want to guarantee that each incoming labeler gives labels on questions he is competent in. Practically we set $\mathrm{P_c} = 0.9$.

If the posterior of a question is close to 0.5, we cannot make the decision which category we should put the question in. Otherwise, if the posterior of a question is converging to 0 or 1, we are almost certain of the answer, and we no longer ask future labelers to label this task.

C. Algorithm OnTac

Based on the analyses and explanations of III-A and III-B, we propose our algorithm of online task assignment — OnTac. The algorithm is shown in **Algorithm 1**. OnTac consists of two phases — the assignment phase and the inference phase.

A labeler comes, with a maximum number of questions he labels, denoted by MaxNb.

In the assignment phase:

Algorithm 1: Online Task Assignment Algorithm (OnTac)

$k \leftarrow 0$, initialize $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{\beta}}$
for $i \in \mathcal{W}$ **do**
  $count \leftarrow 0$
  while $count \leq \mathrm{MaxNb}$ **and** *labeler i informative* **do**
    $count \leftarrow count + 1$

    /*** The assignment phase: *****/**
    Calculate the vector $\mathbf{P}$ from α'_i and $\boldsymbol{\beta}'$
    Rank $\mathbf{P}$ as to distance to $\mathrm{P_c}$, get ranking $\mathcal{R}$
    for $j \in \mathcal{W}$ **do**
      if $\hat{p}(z_j) \in (0.5 - \epsilon, 0.5 + \epsilon)$ **then**
        Prioritizes j in $\mathcal{R}$
      if $\hat{p}(z_j) \in (1.0 - \epsilon, 1.0]$ **then**
        $\mathcal{R} \leftarrow \mathcal{R} \backslash \{j\}$, $\mathcal{T} \leftarrow \mathcal{T} \backslash \{j\}$
    Assign a task j to i according to order $\mathcal{R}$

    /*** The inference phase: *****/**
    Calculate $\hat{p}(\mathbf{z})$ and $Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$
    $\boldsymbol{\alpha}', \boldsymbol{\beta}' \leftarrow \arg\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$
    $\eta_k \leftarrow (k+2)^{-a}$
    Get current estimates $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$ using Equation 5
    $k \leftarrow k + 1$
    Determine if labeler i is informative based on α'_i

a) We calculate the estimated probabilities that he labels each question correctly according to his competence and question difficulties with Equation 1, and we get the probability vector $\mathbf{P}$. We sort this vector on the basis of the distance from each value to $\mathrm{P_c}$. The order vector we obtain is denoted by $\mathcal{R}$. **b)** If the posterior calculated from last example is close to 0.5, meaning it is still ambiguous, we put it forward in $\mathcal{R}$. If the posterior is approaching 1 (or 0), we are almost certain of the answer and delete the question from the task pool. For one single example, the data may be "deceiving". We have to check whether $\hat{p}(\mathbf{z})$ has the tendency of convergence. **c)** Now we assign a task $j \in \mathcal{T}$ referring to the order vector $\mathcal{R}$ and labeler i gives his label l_{ij}. We update the labels set $\mathcal{L}$.

In the inference phase:

d) Compute the posterior $p(\mathbf{z}|\mathcal{L}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ and the auxiliary function $Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ according to Equations 2 and 3. With maximizing the auxiliary function utilising gradient ascent algorithm given in **Algorithm 2**, we obtain the current estimates $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$. In this algorithm, ∇_θ is the notation of derivation as to θ, d_{Euc} stands for Euclidean distance, t is the stepsize of one iteration and λ represents the terminal condition of iterations, $\boldsymbol{\alpha}^{(n)}$ and $\boldsymbol{\beta}^{(n)}$ stands for the parameters of the previous iteration of gradient ascent while $\boldsymbol{\alpha}^{(n+1)}$ and $\boldsymbol{\beta}^{(n+1)}$ stands for those of the current iteration. The algorithm is iterated until convergence. **e)** We combine the current estimates with previous estimates using Equation 6. **f)** Supposing that the

Algorithm 2: Parameters Update Using Gradient Ascent

Input: Auxiliary function $Q(\boldsymbol{\alpha}, \boldsymbol{\beta})$
Output: Current estimated parameters $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$
Initialize $\boldsymbol{\alpha}^{1}$ and $\boldsymbol{\beta}^{1}$, $dist \leftarrow 1$, $n \leftarrow 1$
while $dist > \lambda$ **do**
  $\boldsymbol{\alpha}^{(n+1)} \leftarrow \boldsymbol{\alpha}^{(n)} + t\nabla_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}^{(n)}, \boldsymbol{\beta}^{(n)})$
  $\boldsymbol{\beta}^{(n+1)} \leftarrow \boldsymbol{\beta}^{(n)} + t\nabla_{\boldsymbol{\beta}} Q(\boldsymbol{\alpha}^{(n)}, \boldsymbol{\beta}^{(n)})$
  $dist \leftarrow d_{Euc}(\boldsymbol{\alpha}^{(n+1)} - \boldsymbol{\alpha}^{(n)}) + d_{Euc}(\boldsymbol{\beta}^{(n+1)} - \boldsymbol{\beta}^{(n)})$
  $n \leftarrow n + 1$

absolute value of the combined estimate of α_i is smaller than a preset value, we think the labeler gives random labels so we no longer assigns him with any questions.

If the number of labeled questions for labeler i exceeds MaxNb, we do not allocate a task to labeler i any more and we wait for another labeler. Otherwise, we go back to step **a)** and re-allocate him with another task.

IV. Simulation Results

We now show that the algorithm OnTac reduces tremendously the labeling price with achieving a certain level of accuracy. We perform our simulations on a group of simulated parameters, which follow certain distributions. Based on this we generate the response matrix $\mathbf{R}$. Utilizing the matrix $\mathbf{R}$, we simulate the process of labeler incoming and giving labels. The Gaussian distribution models we choose guarantee that they can well describe a real labeling scenario with human behavior noises.

A. Experiment Settings

As mentioned above, α ranges from $(-\infty, +\infty)$, β ranges from $(0, +\infty)$. Considering the heterogeneousness of labeler competence, we set $\boldsymbol{\alpha}$ to follow a Gaussian distribution with means greater than 0. The variance of this distribution signifies the diversity of labelers. As β is greater than 0, like Whitehill *et al.*, we let $\boldsymbol{\beta} = e^{\boldsymbol{\gamma}}$ where $\boldsymbol{\gamma}$ follows a Gaussian distribution. Thus, $\boldsymbol{\beta}$ follows a logarithmic Gaussian distribution. The ground truth $\mathbf{z}$ takes the value 0 or 1 with equal probabilities.

We assume that a task distributer has a question pool $\mathcal{T}$ to be labeled, $|\mathcal{T}| = 100$. Each question has a difficulty. The logarithm of the difficulty follows $\boldsymbol{\gamma} \sim \mathcal{N}(0, 3)$. We generate a labeler set $\mathcal{W}$ where $|\mathcal{W}| = 100$. The expertise of labelers follows a Gaussian distribution, $\boldsymbol{\alpha} \sim \mathcal{N}(1, 3)$. According to Equation 1, we compute the probability that labeler i gives a correct answer of task j and we get the response matrix $\mathbf{R}$. Everyone in $\mathbf{R}$ may not be employed. Now the following simulations are all run on this dataset.

B. Accuracy Performance

First of all we evaluate the performance of our algorithm on accuracy. We fetch out 50 questions from $\mathcal{W}$ to form a new question pool. The number of labelers varies from 15 to 50. Once the algorithm gives a label to each of the questions, we measure it with the ground truth to obtain the accuracy.

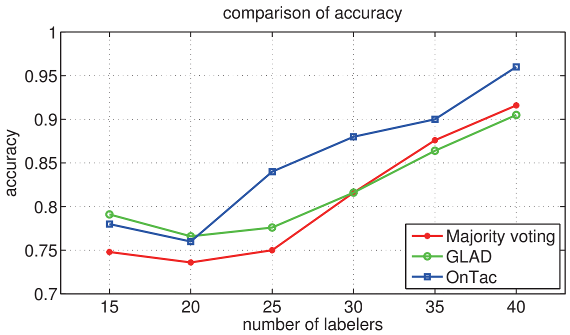

Fig. 3. The accuracies of majority voting, GLAD and OnTac. The simulation is run on 50 questions. The number of labelers vary from 15 to 40.

To compare with two other algorithms, GLAD, brought up by Whitehill *et al.*, and majority voting, we also feed them with the same set of data. The number of labelers we choose ensures that OnTac does not have confidence on the posterior. Thus the three algorithms take almost the same number of labels as input, guaranteeing the fairness of comparison.

The result is shown in Fig. 3. We can observe that the accuracy of each algorithm augments with the number of labelers increasing, meaning that the estimated hidden variables converges to the true label. The performance of GLAD is better than the other two algorithms in the beginning. As more labelers come in, GLAD and majority voting have the similar performance. Our algorithm OnTac is no better that GLAD for a small labeler set. However, OnTac outperforms the other two algorithms when the number of labelers is greater than 20 and it converges faster to a higher value.

We also provide the accuracy performance of estimating two byproducts — question difficulties and labeler abilities in Fig. 4. As the two categories of parameters are combined by multiplication, the algorithm often yields estimated parameters being proportional to, instead of directly equaling to, the true parameters (used for generating the response matrix $\mathbf{R}$). Hereby the evaluation should be with regard to the probability that labeler i gives a correct label to question j, computed with Equation 1. The color distribution on the heatmap represents the difference between true values of parameters and estimated values. We observe that on the estimation of GLAD, there exist peaks, meaning the estimation on this point can not be trusted at all. As for OnTac, the error overall can be tolerated.

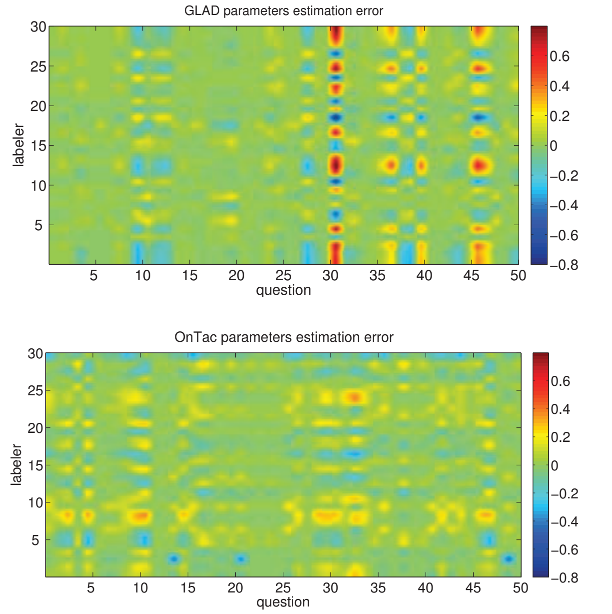

Fig. 4. The comparison of parameters estimation errors between GLAD and OnTac. As the data is two-dimensional, we use a heatmap to represent the error. We run the simulations on a sub-dataset of 50 questions and 30 labelers.

C. Cost Performance

Now we demonstrate how OnTac can tremendously save the labeling budget. We utilize the complete dataset generated previously. Each labeler gives at most 80 labels.

We observe from Fig. 5 that the accuracy obtained from algorithm OnTac converges faster to a higher value (0.9) in comparison with the original algorithm GLAD (0.84). Meanwhile, as more labelers come, the cost OnTac pays decreases tremendously compared to GLAD (shown in Fig. 6), meaning that the algorithm achieves a higher accuracy with less price paid. When the number of labelers reaches 70, the total price of OnTac is 68% of the price of GLAD. This is the effect of online task assignment. Since we only assign ambiguous tasks to competent labelers, the information gain obtained by each label is higher than that of random assignment. The act of prioritizing ambiguous questions and eliminating confident questions guarantees that each label is fully made use of.

We also observe that the gradient of the curve of OnTac decreases, due to the fact that we are confident of more questions as more labelers come in. The second derivative is the highest when there are around 25 labelers and the decrease of gradient gets slower afterwards. Our explanation is that, the remaining questions are the tough ones, demanding more labels before we are certain of the corresponding answers.

Note that in both Fig. 3 and Fig. 5, there is a fluctuation of accuracy with the number of labelers increasing. This is a reasonable phenomenon, owing to the fact that human behaviors are noisy. Still we confirm a tendency of convergency.

D. Running Time Performance

Fig. 7 illustrates the comparison of running time between the two models. The running time of GLAD is determined by the number of iterations of EM algorithm. To get a rapid result and to demonstrate the tendency compared with OnTac, we set that the algorithm stops when the Euclidean distance of parameters given by two adjacent iterations, namely, λ in **Algorithm 2**, is no greater than 0.001.

We constate that the calculations of GLAD grow linearly with labeler set growing, whereas the gradient of OnTac decreases when more labelers come, conforming to the tendency of Fig. 6. The more labelers come in, the less unconfident questions we have, the shorter the running time is. Another important point is, when a new labeler comes in, we do

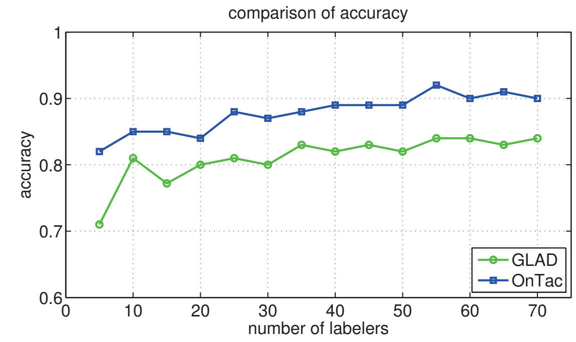

Fig. 5. This figure demonstrates that the accuracy of OnTac augments with a smaller budget (Fig. 6). We run this simulation on 100 questions. The number of labelers varies from 5 to 70. The fluctuation of accuracy is due to the noise of human behaviors whereas the convergence is clear.

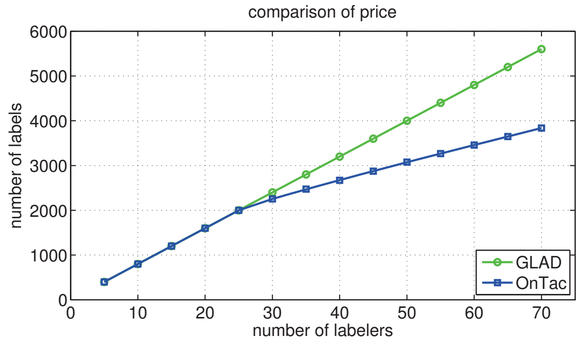

Fig. 6. The number of labels requested varies with the increasing of labelers. The number of labels is obtained in the same scenario as Fig. 5.

not need to recalculate the whole vectors of parameters and hidden variables. We only update once for every incoming label. Unlike offline EM algorithm, the running time of online algorithm is cumulative.

V. Conclusion

In this paper we studied an important issue in crowdsourcing systems — how to assign tasks to labelers in order to achieve a high accuracy with a minimum budget. We developed the algorithm OnTac, namely, Online Task assignment in a crowdsourcing system. This algorithm makes online estimations of labeler abilities, question difficulties and ground truth. Based on these estimations, the algorithm makes adaptive assignment. Once we obtain the labels, we use our inference method to recalculate the current estimations.

We have set up a series of experiments to demonstrate the performance of our algorithm. OnTac achieved a higher accuracy with a largely reduced number of labels. The experiment on running time shows that the calculations of OnTac grow more slowly that linear growth.

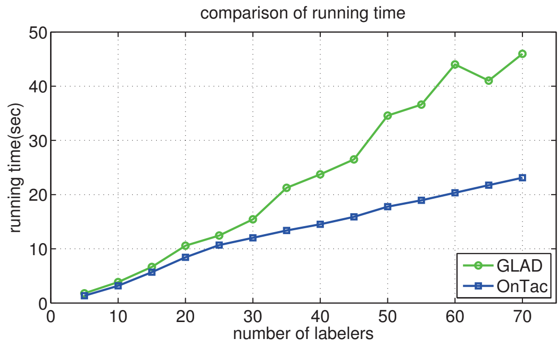

Fig. 7. The comparison of running time between GLAD and OnTac. The unity of time is in second. System setting is the same as in Fig. 5.

VI. Acknowledgement

This work is supported by National Natural Science Foundation of China (No. 61532012, 61572319, U1405251, 61325012, 61271219, 61428205, 61521062); National Mobile Communications Research Laboratory, Southeast University (No. 2014D07).

References

[1] The project Zooniverse, http://www.zooniverse.org/.
[2] C.-J. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive task assignment for crowdsourced classification," in *Proc. ICML*, 2013, pp. 534–542.
[3] P. Liang and D. Klein, "Online em for unsupervised models," in *Proc. human language technologies: NAACL HLT-2009*, 2009, pp. 611–619.
[4] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.
[5] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proc. Advances in neural information processing systems*, 2011, pp. 1953–1961.
[6] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver, "How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing," in *Proc. of the 29th international conference on machine learning (ICML-12)*, 2012, pp. 1183–1190.
[7] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing," in *Proc. ACM the 23rd international conference on World wide web*, 2014, pp. 165–176.
[8] G. Goel, A. Nikzad, and A. Singla, "Allocating tasks to workers with matching constraints: truthful mechanisms for crowdsourcing markets," in *Proc. the companion publication of the 23rd international conference on World wide web companion*. International World Wide Web Conferences Steering Committee, 2014, pp. 279–280.
[9] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *Proc. ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, 2013, pp. 81–92.
[10] V. C. Raykar and S. Yu, "An entropic score to rank annotators for crowdsourced labeling tasks," in *Proc. IEEE Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference on*, 2011, pp. 29–32.
[11] T. Pfeiffer, X. A. Gao, Y. Chen, A. Mao, and D. G. Rand, "Adaptive polling for information aggregation." in *Proc. AAAI*. Citeseer, 2012.
[12] P. Welinder and P. Perona, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in *Proc. IEEE CVPRW*, 2010, pp. 25–32.
[13] Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy, "Active learning from crowds," in *Proc. of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 1161–1168.
[14] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. NIPS*, 2009, pp. 2035–2043.
[15] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *The Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.