CrowdWT: Crowdsourcing via Joint Modeling of Workers and Tasks

JINZHENG TU, GUOXIAN YU*, and JUN WANG, Shandong University, China

CARLOTTA DOMENICONI, George Mason University, USA

MAOZU GUO, Beijing University of Civil Engineering and Architecture, China

XIANGLIANG ZHANG, King Abdullah University of Science and Technology, SA

Crowdsourcing is a relatively inexpensive and efficient mechanism to collect annotations of data from the open Internet. Crowdsourcing workers are paid for the provided annotations, but the task requester usually has a limited budget. It is desirable to wisely assign the appropriate task to the right workers, so the overall annotation quality is maximized whilst the cost is reduced. In this paper, we propose a novel task assignment strategy (CrowdWT) to capture the complex interactions between tasks and workers, and properly assign tasks to workers. CrowdWT first develops a Worker Bias Model (WBM) to jointly model the worker's bias, the ground truths of tasks, and the task features. WBM constructs a mapping between task features and worker annotations to dynamically assign the task to a group of workers, who are more likely to give correct annotations for the task. CrowdWT further introduces a Task Difficulty Model (TDM), which builds a Kernel ridge regressor based on task features to quantify the intrinsic difficulty of tasks and thus to assign the difficult tasks to more reliable workers. Finally, CrowdWT combines WBM and TDM into a unified model to dynamically assign tasks to a group of workers, recall more reliable even expert workers to annotate the difficult tasks. Our experimental results on two real-world datasets and two semi-synthetic datasets show that CrowdWT achieves high-quality answers within a limited budget, and has the best performance against competitive methods.

Additional Key Words and Phrases: Crowdsourcing, Task Assignment, Worker Bias Model, Task Difficulty Model, Annotation Quality

ACM Reference Format:

Jinzheng Tu, Guoxian Yu, Jun Wang, Carlotta Domeniconi, Maozu Guo, and Xiangliang Zhang. 2020. CrowdWT: Crowdsourcing via Joint Modeling of Workers and Tasks. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (September 2020), 24 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Crowdsourcing is widely used in different ways such as machine learning, graph mining, medical healthcare, and so on [7, 19, 22, 52]. Crowdsourcing comes in two forms: i) Spatial crowdsourcing [10] outsources different types of spatio-temporal tasks to the workers or participants in the real world, e.g., Uber; ii) Classical one ignores the spatial information and are mostly seen as participatory

*corresponding author

Authors' addresses: Jinzheng Tu, jinzhengtu@gmail.com; Guoxian Yu, guoxian85@gmail.com; Jun Wang, kingjun@sdu.edu.cn, Shandong University, School of Software, Jinan, China; Carlotta Domeniconi, carlotta@cs.gmu.edu, George Mason University, Department of Computer Science, USA; Maozu Guo, guomaozu@bucea.edu.cn, Beijing University of Civil Engineering and Architecture, Beijing, China, School of Electrical and Information Engineering; Xiangliang Zhang, xiangliang.zhang@kaust.edu.sa, King Abdullah University of Science and Technology, SA, Department of Computer Science.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

1556-4681/2020/9-ART1 \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Fig. 1. An example of a picture judgement task. In this task, workers need to scrutinize the picture and choose one answer from the given three options

activities based on online platforms, e.g., Amazon Mechanical Turk¹ and CrowdFlower². In this paper, we consider the background of the second form.

A drawback of crowdsourcing is that the collected annotations may have a high error rate, due to the fact that online workers lack expertise and incentives [38]. In addition, some tasks are domain specific and online workers often lack the necessary skills. Workers are paid for answering different tasks [45]. As a consequence, to achieve high-quality annotations at a limited cost, it is desirable to adopt a wise task allocation strategy that accounts for both the *tasks* and the *workers*.

Various task assignment strategies have been proposed. They focus on different aspects, such as individual worker's reliability and intention, worker's contribution to ground truth, task difficulty, and so on [3, 6, 26, 27, 34, 44, 47, 54]. These strategies generally fall into three categories: task-centered, worker-centered, and both task- and worker-centered. The task-centered strategy generally focuses on how to select the most informative tasks (that provide the most informative annotations for learning the truths) and assign them to workers. It aims to achieve an overall high quality for the informative tasks [13, 31]. The worker-centered strategy typically assumes a set of workers with known qualities, it evaluates a worker's preference on different tasks, and assigns the tasks to workers with high qualities. However, it often ignores the specific requirements of a task [33, 49]. Some recent studies account for both tasks and workers; they model the difference between worker's skills and task difficulty [17, 48]. If a worker's skill exceeds the difficulty of a task, the worker is assigned to the task.

Some researchers [2, 21, 46] investigate the impact of task features, which characterize the intrinsic complexity of a task and is not subjected to individual workers. They mainly consider the influence of task features when integrating the data. But in actual fact, workers may be influenced by task features when annotating them. If we can account for this effect in advance, we will obtain better annotations. Figure 1 gives an example of a picture judgement task. Influenced by the background of the pictures (*sea* in the left picture and *snow* in the right picture), a worker is more likely to annotate the left picture as an 'Albatross', while the true label is 'a and b'. As such, a wise task assignment strategy should jointly consider 'what is the task?' (the task intrinsic feature/complexity), and 'who can solve the problem?' (the preference and reliability of workers). In this way, more high-quality answers can be obtained within a limited budget.

In this paper, we propose a novel crowdsourcing task allocation strategy, namely *Crowdsourcing* via joint modeling of Workers and Tasks (CrowdWT). CrowdWT assumes that workers' annotations are not independent, the probability of providing the wrong annotations depends not only on

¹www.mturk.com

²<https://www.figure-eight.com>

their specialties, but also on the task itself. CrowdWT can assign the appropriate tasks to workers based on their preferences to tasks and on the task features, and thus reduce the cost. CrowdWT consists of a Worker Bias Model strategy (WBM) and a Task Difficulty Model strategy (TDM). WBM is to capture the mapping between the task features and worker specialties, and to derive a crowdsourcing annotation aggregation process. Then, TDM is introduced to distinguish easy tasks from difficult ones, and to assign the difficult ones to more reliable workers.

The main contributions of our work are summarized as follows:

- CrowdWT models the complex interactions between task features, worker annotations and the ground truths of tasks via an interpretable probability graphical model. These complex interactions are only partially accounted by existing solutions in crowdsourcing [16, 37].
- CrowdWT leverages WBM and TDM to wisely assign the appropriate tasks to the right workers. WBM generates an individual confusion matrix for each worker and a shared confusion matrix for the crowd to address the problem of sparse annotations and to allocate tasks to workers. TDM further assigns the difficult task to more reliable workers.
- Extensive experiments show that CrowdWT can obtain a better (or comparable) quality of annotations with a much reduced budget than other comparing methods [4, 15, 30, 51], and both WBM and TDM contribute to the task assignment and to save budget. In addition, WBM can more accurately aggregate annotations than other related and competitive solutions [8, 16, 21, 30, 37].

The remainder of this paper is organized as follows. Section 2 briefly reviews different assignment strategies. We elaborate on the proposed CrowdWT, including the WBM and the TDM in Section 3. The experimental setup is described in Section 4. Section 5 provides the experimental results and analysis, and Section 6 gives the conclusions and future work.

2 RELATED WORK

Crowd workers have diverse reliabilities and specialties, so they have different preferences and capabilities to the same task. As such, a wise task assignment strategy can lead to a better crowdsourcing gain. Existing crowdsourcing solutions for task assignment can be categorized into three types: task-centered, worker-centered, and the combination of tasks and workers.

Task-centered strategy aims to assign a set of appropriate tasks to a given worker. For example, Liu *et al.* [20] used a quality-sensitive answering model to define the uncertainty of each task, and then assigned the k most uncertain tasks to the worker. Simpson *et al.* [31] studied the expected error reduction strategy, which evaluates the benefits of all possible next-step assignments and selects the most beneficial assignment at each step. To overcome the uncertainty of annotations, Hu *et al.* [13] proposed an active task assignment strategy and modulated the greedy task assignment with the posterior uncertainty.

Worker-centered strategy accounts for the worker specialty and preferences when assigning tasks. For example, Ipeirotis *et al.* [14] used scalar scores to evaluate the quality of workers and to reject (or block) low-quality workers during the assignment of tasks. Fan *et al.* [9] proposed an adaptive crowdsourcing framework (iCrowd). iCrowd dynamically estimates the accuracy of a worker based on her/his performance on the completed tasks, and predicts which tasks the worker are good at. Amer-Yahia *et al.* [1] took into account workers' preferences, and helped workers finding relevant tasks and selecting tasks according to their arrival rates. Khan *et al.* used the marginal likelihood estimation to iteratively compute the expected accuracy of a worker, and for a given task it selects workers who will provide the most accurate annotations [15]. Some other solutions assume workers have different levels of skills depending on the domain. As such, they partition tasks into domains, and assign tasks to workers whose skills are a good fit [33, 49].

Both task and worker centered strategy jointly models tasks and workers when assigning tasks to workers. For example, Kurve *et al.* [17] explicitly modeled the skill and intention of individual workers, and assigned the most confident task to a particular worker. Zhao *et al.* [48] developed a Bayesian generative model to capture the latent skills of workers based on previously resolved tasks with feedback scores, and on the latent difficulty of tasks. Qi *et al.* [28] assumed that the workers may obtain their data and knowledge from the same external information source, and their answers are dependent. They proposed a probabilistic model to infer the source dependency by revealing different group structures among involved sources. Although, they considered the task difficulty, they cannot estimate the difficulty of each task. Mavridis *et al.* [23] modeled tasks and workers using a skill tree, which is equipped with a similarity distance between tasks' skills and workers' skills. Ho *et al.* [12] developed a two-phase exploration-exploitation algorithm for assigning heterogeneous items to workers with different qualities based on the online primal-dual technique. Later, Ho *et al.* [11] proposed a provably near-optimal assignment to investigate the problem of the heterogeneous classification tasks. Mehta *et al.* [24] studied the online matching problem with stochastic reward, which maximizes the number of 'successful' allocations. Tong *et al.* [35, 36] studied the more challenging online micro-task allocation with account of the spatial information of workers and tasks to allocate tasks. They highlighted the benefit of considering the average case performance instead of the rare worse-cases, and justified the greedy algorithm can be more efficient and effective than other allocation algorithms. However, their solutions cannot be directly extended to solve the problem studied in this paper, which mainly focuses on the influence of task feature and task difficulty on workers, without considering the spatial information.

The aforementioned methods do not account for both task feature and task difficulty in assigning tasks. Although some methods [21, 46] consider the task features, they usually utilize the feature information to achieve better aggregations, instead of task assignment. Our CrowdWT models the influence of task feature on workers in the process of answering tasks. In addition, CrowdWT considers the difficulty of task, and allocates difficult tasks to more reliable workers. CrowdWT leverages WBM and TDM to capture complex interactions between tasks and workers, and thus to properly assign tasks to workers. WBM constructs a mapping between task features and worker annotations to dynamically assign the task to a group of workers, who are more likely to provide correct annotations. TDM further estimates the difficulty of tasks and allocates difficult ones to the appropriate expert workers. Our experimental results show that CrowdWT not only can capture the influence of task features on worker decisions, but also can more wisely assign tasks to a group of workers than representative and competitive strategies [4, 17, 30, 51].

3 THE PROPOSED METHODOLOGY

We elaborate on the Worker Bias Model (WBM) (as shown in Figure 2) to mimic the process of worker carrying out annotations, which are generated based on worker confusion matrices and task features. Then, we introduce the Task Difficulty Model (TDM) to assign the difficult tasks to more reliable workers. After that, we unify the above two models to further wisely assign the appropriate tasks to the crowd.

Suppose there are W workers annotating N tasks into C possible classes, as shown in Figure 1. Each worker w either does not annotate an example or annotates it only once, which follows the authentic rule of current crowdsourcing systems, where each task can be accepted and completed at most once by a specific worker. $a_n^w = c$ (a worker's annotation) denotes that worker $w \in [W]$ (we use $[W]$ to denote the set $\{1, \dots, W\}$) annotates $c \in [C]$ to the n -th task. $a_n^w = 0$ means that the worker does not annotate the n -th task, $t_n \in [C]$ is the true class of the task n , $\mathcal{T} = \{t_n\}_{n=1}^N$ represents the true label set of all tasks. Each task has a d -dimensional feature vector $\mathbf{x} = [x_1, x_2, \dots, x_d]$ to represent its characteristics. $\boldsymbol{\pi}^w \in \mathbb{R}^{C \times C}$ represents a confusion matrix with respect to worker

Fig. 2. The factor graph of CrowdWT. Circular nodes are random variables and square nodes are factor nodes. The shaded nodes represent observed values (worker annotations (a_n^w) and task features (x_n)). WBM models the worker specialty (π^w) and the crowd internal commonality ($\hat{\pi}$), and then further learns a weight (m^w) per worker to bridge the gap between them. TDM builds a kernel ridge regressor ($f(x_n)$) on task features to quantify the task's difficulty.

w . $\pi_{c,c'}^w$ is the probability that worker w annotates the class c as c' . The goal of CrowdWT is to capture the complex interactions between tasks and workers based on worker annotation data $\mathcal{A} = \{a_n^w\}_{n=1, w=1}^{N, W}$ and task features $\mathbf{x}$, and thus to properly assign tasks to workers.

3.1 Worker Bias Model

The Worker Bias Model (WBM) is an extension of the Bayesian classifier combination (BCC) model [16]. BCC models the relationship between the individual workers' biases, the worker annotations, and the true labels of tasks [32]. However, BCC assumes that workers' biases are *independent* given common priors. Such an independent assumption may often be violated: worker decisions depend on their specialties and the task itself when crowdsourcing. In addition, BCC does not consider the relationship between the confusion matrices of workers and thus suffers from sparse annotations.

The factor graph of WBM is illustrated in Figure 2. Based on BCC, WBM also assumes that the true class t_n of a task n is generated from a categorical distribution with parameter $\mathbf{p}$ as follows:

$$P(t_n = c | \mathbf{p}) \sim \text{Cat}(t_n | \mathbf{p}) \quad (1)$$

where $\mathbf{p}$ models the class proportions for all tasks, and has a conjugate prior Dirichlet distribution³ parameterized by $\boldsymbol{\tau}$. $\boldsymbol{\tau}$ is initialized with uniform values to give equal prior likelihood to all classes.

$$\mathbf{p} \sim \text{Dir}(\mathbf{p} | \boldsymbol{\tau}) \quad (2)$$

WBM assumes that each worker w is associated with a confusion matrix $\boldsymbol{\pi}^w \in \mathbb{R}^{C \times C}$ (representing the worker *specialty*), with its (c, c') -entry $\pi_{c,c'}^w$ being the probability that the w -th worker annotates a task with c' when the true label is c . In addition, each worker has a probability score σ_c^w representing the log probability vector of the c -th row of the confusion matrix $\boldsymbol{\pi}^w$. WBM assumes

³The posterior probability distribution and the prior probability distribution have the same form

that worker's biases are inter-dependent. That is, each worker w has an individual score matrix σ_c^w generated from a multivariate Gaussian distribution⁴ with mean μ_c and isotropic precision Σ_c as follows:

$$\sigma_c^w \sim \mathcal{N}(\sigma_c^w | \mu_c, \Sigma_c) \quad (3)$$

where μ_c is initialized as the c -th row of the identity matrix. Since $\sigma_{c,c'}^w$ is the probability that worker w annotates the label c as c' , this initialization increases the probability that a worker will correctly annotate a task. Σ is initialized as an identity matrix. WBM applies a softmax operator to σ_c^w for the label c to derive a normalized exponentiated version of σ_c^w [37] as follows:

$$P(\pi_c^w | \sigma_c^w) = \delta(\pi_c^w - \text{softmax}(\sigma_c^w)) \quad (4)$$

where δ is the Dirac delta function to impose the equality constraint between π_c^w and the softmax of σ_c^w . Following the same assumption as BCC that a_n^w is generated by a categorical distribution conditioned on π^w and t_n

$$a_n^w | \pi^w, t_n \sim \text{Cat}(a_n^w | \pi_{t_n}^w) \quad (5)$$

Eq. (5) shows that t_n is a random variable that acts as a selector of the categorical distribution, from which the annotation by worker w is generated.

Learning a reliable individual confusion matrix per worker requires sufficient annotations per worker, which are usually unavailable in many crowdsourcing scenarios. Since workers show similar behavior traits while crowdsourcing [32], WBM additionally learns a shared confusion matrix $\hat{\pi} \in \mathbb{R}^{C \times C}$ (representing the worker *commonality*) and the probability score $\hat{\sigma}$ for the crowd to address the annotation sparsity as follows:

$$\begin{aligned} \hat{\sigma} &\sim \mathcal{N}(\hat{\sigma} | \mu_c, \Sigma_c) \\ P(\hat{\pi} | \hat{\sigma}) &= \delta(\hat{\pi} - \text{softmax}(\hat{\sigma})) \end{aligned} \quad (6)$$

Therefore, the annotation of worker w for task n can be similarly modelled as Eq. (5) as follows:

$$a_n^w | \hat{\pi}, t_n \sim \text{Cat}(a_n^w | \hat{\pi}_{t_n}) \quad (7)$$

To jointly use the specialty and commonality of workers, m^w can be seen as a binomial distribution. We impose a natural Beta distribution to approximate it as follows:

$$\psi_a(\hat{\pi}, \pi_c^w, a_n^w, m^w) = \text{Beta}(m^w | \alpha^w, \beta^w) \times (m^w \text{Cat}(a_n^w | \pi_c^w) + (1 - m^w) \text{Cat}(a_n^w | \hat{\pi}_c)) \quad (8)$$

where the weight m^w is generated from a Beta distribution with parameter α^w and β^w , α^w is a constant associated with worker w . β^w is proportional to the number of annotations provided by worker w . As such, m^w is adaptively adjusted with respect to the amount of data for worker w . For workers with few annotations, a larger weight is assigned to $\hat{\pi}_c$ (commonality). On the other hand, for workers with adequate annotations, a larger weight is given to π_c^w (specialty). In this way, WBM can not only utilize a small number of annotations to learn the shared confusion matrix for workers, but also learn an accurate confusion matrix for each worker when abundant annotations are available.

Workers who can give reliable annotations for many tasks may still be unreliable for other tasks and give wrong annotations for latter ones. In other words, task features affect workers perception and decision making, and it is not wise to assign the latter tasks to these workers. To remedy this case and obtain high-quality annotations, we model the influence of task features on worker decisions in the form of multinational logistic regression, for which the input is a vector of continuous task features $\mathbf{x}_n$ for task n . The logistic regression model links the task features

⁴In the BCC model, π_c^w is generated with a Dirichlet distribution. In our WBM model, π_c^w follows a Gaussian process, and we assume the task features and a Gaussian distribution can capture the relevance between variables.

with the regression coefficients matrix $\Omega \in \mathbb{R}^{C \times d}$, each of which follows a multivariate Gaussian distribution with parameter μ and Σ as follows:

$$\psi_a(\hat{\pi}, \pi_c^w, a_n^w, m^w, \mathbf{x}_n, \Omega) = \psi_a(\hat{\pi}, \pi_c^w, a_n^w, m^w) \prod_{j=1}^d \text{Log}(\Omega_c^j, \mathbf{x}_{nj}) \mathcal{N}(\Omega_c^j | \mu_c, \Sigma_c) \quad (9)$$

Ω_c represent the c -th row of Ω . Specifically, we assume that we have observed the class assignments of all objects from all workers (denoted by $\mathcal{A}$). Given $\mathcal{A}$, we want to infer the posterior distribution of the set of parameters: $\Theta = \{\mathbf{p}, \mathbf{t}, \pi, \hat{\pi}, \sigma, \hat{\sigma}\}$. For this inference, we utilize the Bayes theorem to compute the joint posterior distribution of Θ and the variational message passing algorithm (VMP)[43] to approximate the marginal distribution of each parameter, under the assumption that labels are independently generated as follows:

$$\begin{aligned} P(\Theta | \mathcal{A}) \propto & \text{Dir}(\mathbf{p} | \boldsymbol{\tau}) \prod_{n=1}^N \{ \text{Cat}(t_n | \mathbf{p}) \{ \prod_{w=1}^W \{ \text{Cat}(a_n^w | \hat{\pi}_{t_n}) \delta(\hat{\pi}_{t_n} - \text{softmax}(\hat{\sigma}_{t_n})) \mathcal{N}(\hat{\sigma}_{t_n} | \mu, \Sigma) \} \\ & \{ \text{Cat}(a_n^w | \pi_{t_n}^w) \delta(\pi_{t_n}^w - \text{softmax}(\sigma_{t_n}^w)) \mathcal{N}(\sigma_{t_n}^w | \mu, \Sigma) \} \text{Beta}(m^w | \alpha^w, \beta^w) \\ & (m^w \text{Cat}(a_n^w | \pi_{t_n}^w) + (1 - m^w) \text{Cat}(a_n^w | \hat{\pi}_{t_n})) \} \prod_{j=1}^d \text{Log}(\Omega_{t_n}^j, \mathbf{x}_{nj}) \mathcal{N}(\Omega_{t_n}^j | \mu, \Sigma) \} \end{aligned} \quad (10)$$

where δ operator is used to ensure that the posterior distribution is positive only when $\hat{\pi}_{t_n}(\pi_{t_n}^w)$ is equal to $\text{softmax}(\hat{\sigma}_{t_n})(\text{softmax}(\sigma_{t_n}^w))$.

As described in the generative process of WBM in Figure 2, Eq. (10) models the crowdsourcing annotation decision process by capturing the interactive relationship between the task features and worker's skills. WBM not only considers the worker-specific annotation ability by learning a confusion matrix for each worker, but also takes the crowd commonality into account to address the sparse annotations. WBM also uses the Beta distribution to bridge the gap between the worker specialty and commonality. Last but not least, WBM can capture the impact of task features on workers and obtains more reliable annotations. The experimental results confirm the advantages of WBM.

3.2 Task Difficulty Model

In real world cases, a good task assignment strategy should not only consider the task-solving ability of different workers, but also the difficulty of tasks. Obviously, a difficult task can be more reliably solved by assigning it to workers who are more knowledgeable than others.

The difficulty of a task can be approximated from the annotations of workers [21, 39, 42]. For example, if worker w provides a correct annotation for the task n with high probability, it is believed that task n is easy for worker w ; on the other hand, if w provides a wrong annotation for this task with high probability, this task is difficult for worker w . Therefore, we can compute the posterior probability of the difficulty for any task as $D_n^w = P(a_n^w \neq t_n | n)$, which quantifies the probability that worker w gives wrong annotations for task n . Then, the task difficulty with respect to the crowd W is measured as follows:

$$D_n = \sum_{w \in \mathcal{W}_n} P(a_n^w \neq t_n | n) P(w) \quad (11)$$

where $\mathcal{W}_n$ is the set of workers who annotate task n , $P(w)$ is the prior knowledge of the worker, and represents the judgement towards the given task. Eq. (11) can approximate the difficulty of task n in theory. When D_n is small, n is an easy task; otherwise, it is a difficult one. A worker tends to provide nearly random answers to a too difficult task [29, 39], it is reasonable to assume that

when $D_n \leq 1/2$, reliable answers can still be achieved from the crowd. However, it is in-feasible to compute D_n from Eq. (11) in practice, since the prior knowledge $P(w)$ and ground truth t_n are unknown in advance. Therefore, we have to find a surrogate to estimate the task difficulty.

From the WBM algorithm, we can obtain the inferred label t_n^* for each task n according to the crowdsourced annotations given by a group of workers $\mathcal{W}_n$. The tasks that the workers cannot answer correctly seem to be difficult. Thus, a good approximation of D_n can then be made by :

$$\tilde{D}_n = \begin{cases} \frac{\sum_{w=1}^{|\mathcal{W}_n|} \mathbb{I}(t_n^w \neq t_n^*)}{|\mathcal{W}_n|}, & |\mathcal{W}_n| \geq \text{Avg}(\mathcal{W}_n) \\ \frac{\sum_{w=1}^{|\mathcal{W}_n|} \mathbb{I}(t_n^w \neq t_n^*)}{\text{Avg}(\mathcal{W}_n)}, & |\mathcal{W}_n| < \text{Avg}(\mathcal{W}_n) \end{cases} \quad (12)$$

where $\text{Avg}(\mathcal{W}_n) = \sum_{n=1}^N |\mathcal{W}_n|/N$ is the average number of workers per task. If $t_n^* = t_n$, it is easy to prove that $P(|\tilde{D}_n - D_n| > \epsilon) \leq \exp(-2\epsilon^2 \max |\mathcal{W}_n|)$ with Hoeffding bound [40], where $\epsilon \in (0, 1)$ is a small constant. Similarly, if $t_n^* \neq t_n$, we have $P(|(1 - D_n) - \tilde{D}_n| > \epsilon) \leq \exp(-2\epsilon^2 \max |\mathcal{W}_n|)$, and $P(|1/2 - D_n| - |1/2 - \tilde{D}_n| > \epsilon) \leq \exp(-2\epsilon^2 \max |\mathcal{W}_n|)$. As such, we have $\tilde{D}_n \leq 1/2$ with Eq. (12) and $P(|\tilde{D}_n - D_n| > \epsilon) \leq \exp(-2\epsilon^2 \max |\mathcal{W}_n|)$ when $D_n \leq 1/2$, which indicates that $\tilde{D}_n$ is a good approximation of D_n when the true labels of the tasks are not available. Therefore, we introduce a threshold θ to determine whether a task is difficult or not. When $\tilde{D}_n \leq \theta$, the task n is easy for the crowd and the annotations from them are reliable; whilst $\tilde{D}_n \geq \theta$ means this task is rather difficult for the current workers, who may provide low-quality annotations for this task.

However, the premise for estimating the difficulty of task n by $\tilde{D}$ is that all tasks are annotated by the crowd. When $\tilde{D}_n \geq \theta$, we should not recruit erroneous annotations for the difficult task n from the crowd without caution, which wastes a great annotation budget. Given that, we develop an effective approach to estimate the task difficulty before annotating all tasks.

The approximation of task difficulty in Eq. (12) is independent from the task features. However, in the crowdsourcing process, the difficulty of a task depends on its features (as we had illustrated in Fig. 1). We want to find a model $f(\mathbf{x})$ to quantify the difficulty of a task based on its features and the approximated task difficulty. For this purpose, CrowdWT introduces a Task Difficulty Model (TDM) as follows:

$$f(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{h}\Phi(\mathbf{x})+b)}} \quad (13)$$

where $\Phi(\mathbf{x})$ denotes the kernel mapping function, which is a linear kernel for this study, $\mathbf{h}$ is the weight vector, and b is the offset. By utilizing the above formula, we can map the value of $f(\mathbf{x})$ between 0 and 1. The larger the value of $f(\mathbf{x})$ is, the more difficult the task is. Other formulas that can estimate the task difficulty with task features can also be adopted.

To find f , we randomly select a small portion of tasks $[L] \subset [N]$ from the whole $[N]$ and let the crowd annotate them. With these crowdsourced annotations, we estimate the empirical difficulty $\tilde{D}_n$. For a good f , it should accurately differentiate the easy tasks from the difficult ones. Namely if $\tilde{D}_n \leq \theta$, $f(\mathbf{x}_n) \leq \theta$; otherwise, $f(\mathbf{x}_n) > \theta$. To make this, we seek the optimal $(\mathbf{h}; b)$ using the Kernel ridge regression with constraints as follows:

$$\begin{aligned} \min \text{Loss}(\mathbf{h}, b) &= \sum_{n=1}^L (f(\mathbf{x}_n) - \tilde{D}_n)^2 + \eta \|\mathbf{h}\|^2 \\ \text{s.t. } &(\tilde{D}_n - \theta)(f(\mathbf{x}_n) - \theta) \geq 0 \end{aligned} \quad (14)$$

where the scale parameter η controls the impact of the weight vector $\mathbf{h}$. By minimizing the above equation, a good f can be found to indicate whether task n should be annotated by the crowd or

not. We integrate the constraints as follows:

$$\min \text{Loss}(\mathbf{h}, b) = \sum_{n=1}^L (f(\mathbf{x}_n) - \tilde{D}_n)^2 + \eta \|\mathbf{h}\|^2 + \gamma \sum_{n=1}^L \mathbb{I}((\tilde{D}_n - \theta)(f(\mathbf{x}_n) - \theta) \leq 0) \quad (15)$$

where $\mathbb{I}$ is an indicator function, the scale parameter γ weights the constraint. However, Eq. (15) can not be solved efficiently, since $\mathbb{I}(k \leq 0)$ is not differentiable. We replace it with a surrogate loss function $\log_2(1 + e^{-k})$ and reformulate the above function as follows:

$$\min \text{Loss}(\mathbf{h}, b) = \sum_{n=1}^L (f(\mathbf{x}_n) - \tilde{D}_n)^2 + \eta \|\mathbf{h}\|^2 + \gamma \sum_{n=1}^L \log_2(1 + e^{-(\tilde{D}_n - \theta)(f(\mathbf{x}_n) - \theta)}) \quad (16)$$

By solving the optimization problem in Eq. (16) with training set $[S] = \{(\mathbf{x}_1, \tilde{D}_{\mathbf{x}_1}), \dots, (\mathbf{x}_{|[L]|}, \tilde{D}_{\mathbf{x}_{|[L]|}})\}$, we can estimate the task difficulty based on the approximated $\tilde{D}$ and the task features $\mathbf{x}$, and predict the difficulty of tasks in $[N] - [L]$. With the predicted difficulty $f(\mathbf{x}_n)$ and the threshold θ , we can determine whether a task should be annotated by the crowd or not. When $f(\mathbf{x}_n) \leq \theta$, $n \in ([N] - [L])$, the task n should be annotated by the crowd; otherwise, the task n should be rejected by the plain crowd and more reliable workers should be recalled.

θ is a crucial parameter that determines the ratio of easy (hard) tasks. A too small θ value will lead to fewer easy tasks (more difficult tasks), and vice versa. For a crowdsourcing job with a large number of tasks, it may be impractical to specify a good value for θ by considering all tasks. To bypass this issue, we randomly sample a subset of the tasks, and compute the ratio of easy tasks on the sampled ones to estimate θ . Suppose r is the ratio of easy tasks, whose difficulties are no larger than θ . For a subset $[S]$, we set θ as follows:

$$\theta = \min\{\theta \in [0, 1] : \frac{|\{n \in [S] : \tilde{D}_n \leq \theta\}|}{|[S]|} \geq r\} \quad (17)$$

The ratio of easy tasks can be determined from the known θ as $r = \frac{|\{n \in [N] - [L] : \tilde{D}_n \leq \theta\}|}{|n \in [N] - [L]|}$. θ and r can be attained by clamping on both sides.

3.3 The Unified Model

A potential issue with TDM is that when a worker lacks background knowledge for an easy task, this task may be still difficult for him/her. Consequently, he/she cannot correctly annotate it. To remedy this, we integrate the modeled worker skills in WBM with the estimated task difficulty in TDM to achieve better assignments. If a worker's skill level is below the task difficulty, he/she is more likely to give incorrect annotations to the task. Therefore, it is better to assign the task to more reliable workers, whose skill levels are above the task difficulty. To this end, we model the capability of a worker correctly annotating the task based on the difference between the n -th task difficulty $f(\mathbf{x}_n)$ and the w -th worker's skill π^w via the *probability mass function* (*pmf*) as follows:

$$\text{pmf}(a_n^w = c) = \begin{cases} \frac{1}{1 + e^{-k'(\pi_{t_n, c} - f(\mathbf{x}_n))}} + \frac{1}{C} \frac{e^{-k'(\pi_{t_n, c} - f(\mathbf{x}_n))}}{1 + e^{-k'(\pi_{t_n, c} - f(\mathbf{x}_n))}}, & c = t_n \\ \frac{1}{C} \frac{e^{-k'(\pi_{t_n, c} - f(\mathbf{x}_n))}}{1 + e^{-k'(\pi_{t_n, c} - f(\mathbf{x}_n))}}, & \text{otherwise} \end{cases} \quad (18)$$

where C is the number of classes; $k' = \max\{m^w, (1 - m^w)\}$ controls an individual worker's propensity to answer correctly, for a given difference $\pi_{t_n, c} - f(\mathbf{x}_n)$. If $k' = m^w$, $\pi_{t_n, c} = \pi_{t_n, c}^w$; $\pi_{t_n, c} = \hat{\pi}_{t_n, c}$, otherwise. On the one hand, the larger value $\pi_{t_n, c} - f(\mathbf{x}_n)$ means the worker's skill exceeds the task difficulty. On the other hand, when the value of $\pi_{t_n, c} - f(\mathbf{x}_n)$ is small (even negative), it indicates that the worker cannot give a high-quality annotation, thus resulting a small value of *pmf*. From Eq. (18), we can see that:

- (1) when $c = t_n$, $pmf(\cdot)$ computes the probability that the worker gives a true annotation. If $\pi_{t_n, c} \geq f(\mathbf{x}_n)$, the worker answers correctly with high probability; if $\pi_{t_n, c} < f(\mathbf{x}_n)$, the quality of the answers provided by the worker is low;
- (2) when $c \neq t_n$, $pmf(\cdot)$ computes the probability that the worker gives a wrong annotation. If the worker has already provided wrong labels, the pmf value is small, regardless $\pi_{t_n, c} < f(\mathbf{x}_n)$ or $\pi_{t_n, c} \geq f(\mathbf{x}_n)$;

Therefore, Eq. (18) can well estimate the quality of a worker annotating the task with its difficulty $f(\mathbf{x})$. Then, for task n , CrowdWT selects a group of workers $\mathcal{W}_n^*$ who are more appropriate for this task as follows:

$$\mathcal{W}_n^* = \arg \max_{w \in [W]} \sum_{c=1}^C pmf(a_n^w = c) \quad (19)$$

To this end, a group of workers who can reliably provide the correct annotations for a task are called together. As such, high-quality annotations can be obtained by a much smaller number of workers, and the cost can be accordingly reduced. In practice, any given worker is not always readily-available, and assigning too many tasks to the same worker may result in latency or even abandonment. To avoid over-assignments, we use a threshold ε . Specifically, if a worker has been assigned ε tasks, he/she will not be assigned with new tasks, and the next reliable worker will be chosen instead. The resulting advantages of CrowdWT will be reported in the next section.

4 EXPERIMENT SETUP

4.1 Datasets

To quantitatively evaluate the effectiveness of CrowdWT in assigning appropriate tasks to right workers, we carry out experiments on two real-world datasets and two semi-synthetic datasets. The statistics of these datasets are listed in Table 1.

Table 1. Statistics of datasets used in the experiments.

DATASET	N	C	d	T	W	T/N
Birds	6033	4	25	30165	460	5
News	2007	4	100	14032	186	7
Landsat	2000	6	36	21684	100	11
Segment	2310	7	18	19424	40	9

T/N : the average number of annotations per task;

T : the collected annotations;

Birds 200 [41] is a challenging image dataset annotated with 200 bird species. This dataset with 6033 images (tasks) is annotated for the color pattern of ‘birds’ tails by 460 workers from Amazon Mechanical Turk. Each task is described by 25-dimensional task features. **News** [18] contains documents of 4 topics from the 20NewsGroup dataset. We used the TF-IDF (Term Frequency and Inverse Document Frequency) on the descriptive texts to extract the task features. The last two datasets are multi-class classification datasets collected from *UCI*⁵, and we adapt them to the crowd semi-synthetic task datasets using the crowd annotating model in [53]. Following the setup in [53], we assume that there are three types of annotators to label the tasks: **Experts** are those who provide correct labels with high probability, no matter how difficult the task is; **Normal** are those who lack the prior knowledge about the tasks and can give reliable answers when the task difficulty

⁵<http://archive.ics.uci.edu/ml/datasets.html>

is relatively low; **Spammers** are the type of workers who randomly annotate regardless of the task difficulty. For the three semi-synthetic datasets, the proportion of *Experts*, *Normal*, *Spammers* is 10%, 80% and 10%; 5%, 75% and 20%; or 5%, 90% and 5%, respectively. In the following experiments, the workers are not newcomers for simplicity. We take the original labels in the multi-class classification datasets as the ground truths to measure the performance. We report the results on semi-synthetic datasets with the proportion of Experts, Normal and Spammers as 10%, 80% and 10%.

4.2 Comparing Methods

To comparatively study the effectiveness of CrowdWT, we conduct two types of experiments. For the *first* type of experiments, we aim to quantitatively study which model of CrowdWT contributes to the prominent performance of CrowdWT. In addition, we also compare our task assignment strategy against the following baselines:

- **Repeated Labeling Strategy** (RLS) randomly assigns the same number (5) of workers for all tasks;
- **Quality-Aware task assignment System for Crowdsourcing Applications** (QASCA) (*worker-centered*) estimates the quality improvement if a worker is assigned with a set of k tasks, and then selects the optimal set which results in the highest expected improvement [51].
- **Dynamic Question Selection in Crowdsourcing systems** (Crowddqs) (*worker-centered*) selects the tasks with the maximum potential gain of the quality if the worker gives her/his estimated true answers [15].
- **Optimistic Knowledge Gradient** (OKG) (*task-centered*) calculates the optimistic knowledge gradient as the maximum expected accuracy increment of all possible next-step labels and selects the one with the highest optimistic KG [5].
- **Generalized EM** (GEM) [17] (*both task and worker centered*) models the skill and intention of the workers, and assign the tasks to a particular worker according to task difficulty.
- **WBM** is our proposed worker bias model, it does not use $f(\mathbf{x})$ to quantify the task difficulty.

Note that if a new worker comes without any information, we can let him/her answer a very small portion of tasks with known truths. Next, we can use the obtained answers for initialization.

For the *second* type of experiments, we investigate the effectiveness of WBM in CrowdWT. WBM can learn its parameters by the variational message passing algorithm (VMP) [43] during training, and thus can aggregate the truth labels from the training annotations. We compare our proposed WBM against four representative and closely related aggregation baselines:

- **Majority Voting** (MV) [30] is often used for the heuristic computation of crowd consensus. MV selects the label annotated by the majority as the ground truth.
- **Dawid&Skene** (DS) [8] applies expectation maximization (EM) to model the confusion matrix of each worker and to aggregate labels.
- **Bayesian Classifier Combination** (BCC) [25] learns the true labels of tasks and each worker's confusion matrix using the expectation propagation message passing algorithm.
- **Community-based Bayesian Classifier Combination** (cBCC) [37] is a proposed extension of BCC, it additionally models worker communities to deal with the annotation sparsity. We ran cBCC with the number of worker communities as 3, which is the same as the number of worker types.
- **WBM-noT** is a variant of WBM, and it does not utilize the task features in WBM. WBM-noT models the process of worker annotations being created only based on latent variables (worker confusion matrices), namely $\text{Log}(\mathbf{x}_{nj}, \Omega_{t_n})\mathcal{N}(\Omega_{t_n}|\mu, \Sigma)$ in Eq. (10) is excluded.

We want to remark that the last three comparing methods also use VMP to aggregate labels. The input parameters of these comparing methods are specified or optimized as the authors suggested

in their codes or papers. The input parameters of CrowdWT and its variant are set as $\eta = 10^{-3}$, $\gamma = 3$, $\theta = 0.2$. ε is set as T/N of each dataset. In practice, we tested $\varepsilon \in [T/N - 2, T/N + 2]$ and CrowdWT showed robust results on these four datasets. We select the same training data ($[L]$, 20% of all the data) for all methods. The influence of $[L]$ is also studied.

4.3 Evaluation Metrics

The performance of these comparing methods is evaluated by two metrics: (1) Accuracy and (2) negative log probability density (NLPD) [37]. Accuracy is defined as the ratio between the number of correct annotations and the total number of annotations. NLPD provides a more comprehensive error measure, it takes into account the uncertainty in the predictive label distribution. Let $q_{n,1}, q_{n,2}, \dots, q_{n,c}$ be the parameters of the predictive categorical distribution for task n . Then, the NLPD score is:

$$NLPD = \frac{1}{N} \sum_{n=1}^N -\log(q_{n,t_n}) \quad (20)$$

In summary, a lower NLPD but higher Accuracy indicates a better performance. NLPD is used to evaluate the uncertainty of label distribution, which is assumed to be a categorical distribution in this paper. Since these task assignment strategies do not consider the uncertainty of label distribution, NLPD is only used to quantify the performance in the *second* type of experiments.

5 EXPERIMENT RESULTS AND ANALYSIS

5.1 Results on assigning tasks

In this subsection, we conduct the *first* type of experiments. We compare CrowdWT against four representative task assignment strategies to comparatively study the strength of our strategy on accuracy. Here, WBM assign tasks to the suitable workers to maximize the expected information gain based on the confusion matrix. To quantitatively compare the effectiveness of these task assignment strategies, we use majority vote to aggregate the annotations provided by different strategies and to avoid the bias caused by different aggregation solutions. The average accuracy of these strategies is shown in Figure 3.

Fig. 3. Accuracy of different task assignment strategies on four datasets.

For each dataset, we randomly select 500 tasks, and independently repeat each experiment 10 times. Considering the fact that we should collect at least 500 annotations for 500 tasks, we start the accuracy comparison from #Annotations= 500. The accuracy of all strategies equals to $1/C$ when #Annotations= 0. In each round (task batch=50), the worker assigned with a task provides one annotation. From Figure 3, we can observe that CrowdWT can reach high accuracy (≥ 0.95) with a much smaller number of annotations than all the benchmark strategies. This is because CrowdWT can model the workers' biases and select appropriate workers to annotate the task. CrowdWT can further assign difficult tasks to the more reliable (or expert) workers and assign easy tasks to others by modeling the task difficulty. As a result, CrowdWT obtains a high accuracy with fewer annotations. WBM considers the influence of task features on workers and assigns the task without modelling task difficulty. It has a lower accuracy than CrowdWT. This comparison confirms that extra modelling the task difficulty contributes to a wiser task assignment. The accuracy of WBM is competitive with Crowddqs, OKG and QASCA, which assign confident tasks to workers. Particularly, OKG selects the worker with the highest optimistic knowledge gradient. QASCA predicts the one-step expected accuracy increment, and selects the assignment with the maximal increment at each step. GEM accounts for the interplay between worker skills, intentions and task, but its accuracy is fluctuant and lower than CrowdWT. One cause is that it neglects the impact of task features on the worker decisions. Another possible factor is that our implementation of GEM (codes not available even after several requests) is not fully consistent with the original authors. RLS always has the lowest accuracy. That is because it assigns the same number of workers to each task and generally needs more annotations to obtain a comparable accuracy.

To further investigate the influence of task features on workers in the process of crowdsourcing annotation, we report the results of several representative tasks and visualize the results in Figure 4. We observe that different strategies ask for different numbers of workers to annotate the tasks. For the first task, since the two images are very different in visual, all strategies achieve the correct results. However, for the last two tasks, the two images are similar and confusing, these compared strategies often achieve wrong results. In contrast, WBM and CrowdWT consider the influence of task features on workers, they obtain more correct annotations. In summary, considering the impact of task features on the worker annotation process contributes to better annotations.

Figure 5 shows the distribution of workers assigned to the first forty tasks on the real-world Birds dataset. We can clearly see that CrowdWT indeed assigns fewer workers than other methods to the same tasks (2 – 3 workers vs. 3 – 4 or more). We count the number of workers assigned to some tasks by these compared methods, and report the counts in Table 2. As expected, CrowdWT consistently uses a much smaller number of workers than the comparing strategies, including WBM. This is because CrowdWT additionally uses TDM to model task difficulty using task features and to further assign the tasks to the appropriate workers, who can reliably annotate the task with high probability. This fact reflects the contribution of TDM for a good task assignment. In summary, the superior accuracy of CrowdWT can be attributed to the fact that it can model the complex interactions between workers and tasks.

To further explore the advantage of our strategy, we increase the number of tasks to 1000 on the real-world dataset, which requires two times of annotations to be collected. From Figure 3 and Figure 6, CrowdWT reduces the number of annotations by half while maintaining an accuracy of 0.90. This investigation again reveals the robustness of our strategy to the scale of collected annotations. In fact, as the number of required annotations increases, the two types of confusion matrices in WBM can be better approximated. Consequently, the preferences of workers and the task difficulty can be estimated more reliably, and they both help reducing the annotation cost. The results under other compositions of experts (regular workers and spammers) are provided in Figures 6 and 8. They lead to similar observations and analysis. Under these settings, CrowdWT

Fig. 4. Annotations obtained by compared methods and CrowdWT with respect to three representative tasks in the Birds dataset. The true labels are highlighted with red rectangles in the left column, and the obtained annotations of each method are listed in right columns. These methods require different numbers of workers to do the same task.

Fig. 5. Number of assigned workers for the first 40 tasks of the Birds dataset with different assignment strategies.

again holds the best performance, and assigns fewer workers than other methods to the same tasks. CrowdWT considers the task difficulty (by Eq. (16)) when assigning tasks, and estimates the ratio of easy/hard tasks by Eq. (17). We report the task difficulty estimated by CrowdWT on the Birds dataset, and reveal them in Figure 9. From Figure 9 (left), we can find that CrowdWT can estimate the difficulty of a task. The higher the value is, the more difficult the task is. As shown in Figure 9 (right), the number of tasks with high values ($f(\mathbf{x}) > 0.8$) is very small. This observation is consistent with the reality that difficult tasks are relatively smaller than easy ones [50].

Table 2. Number of assigned workers for 8 exemplar tasks by different strategies.

#Task	#4	#8	#12	#16	#20	#24	#28	#32
RLS	5	5	5	5	5	5	5	5
GEM	5	3	3	2	2	5	5	4
OKG	2	3	4	2	4	2	4	2
QASCA	2	3	3	2	4	4	3	4
Crowddqs	2	3	3	3	4	3	4	3
WBM	3	3	4	4	3	4	3	2
CrowdWT	3	3	2	2	3	3	3	2

Fig. 6. Accuracy of different task assignment strategies on the Birds dataset with #tasks= 1000.

Fig. 7. Accuracy of CrowdWT and comparing methods under different proportions of three types of workers.

Fig. 8. Number of assigned workers by different strategies for the first 40 tasks under different proportions of three types of workers.

Fig. 9. Task difficulty estimated by CrowdWT on the first 1000 tasks of Birds and the histogram of the difficulties.

In summary, these results and analysis confirm that it is important to jointly model the workers and tasks in crowdsourcing. CrowdWT models the complex interaction between tasks and workers, and thus it more appropriately assigns the tasks to workers than these competitive and representative solutions.

5.2 Results on aggregating annotations

In this section, we conduct the *second* type of experiments to compare the annotation quality obtained by different approaches. Our strategy (WBM) uses the confusion matrices to model workers. Therefore, we compare WBM in aggregating the annotations against DS [8], BCC [16],

cBCC [37], and WBM-noT. The simplest MV [30] is also compared. Table 3 reports the results of the comparing methods on different datasets under various numbers of tasks. WBM almost always has the best performance among compared methods across all datasets, and it always gives better results than WBM-noT. This comparison suggests that task features have an impact on worker annotation decision. Learning the complex interaction between workers and task features is useful to decrease and even avoid the phenomenon that a worker is confused by some tasks, resulting in mislabeling. The accuracy of WBM on Segment is slightly lower than that of cBCC, but higher than that of others. This is possible because the used task features may not be informative enough to accurately reflect the interaction between tasks and workers. cBCC, WBM-noT and WBM consider the annotation sparsity, while MV, DS, and BCC do not. We can clearly see that the former three generally outperform the latter three across different datasets, and they hold a similar accuracy. MV generally has a lower accuracy than the other four confusion matrix-based methods. This is because MV assumes the error of each worker is uniformly distributed over all class labels, while the confusion matrix-based methods do not. This observation corroborates the fact that the probability of each worker making errors on each class is different in crowdsourcing. This observation also supports our idea of modeling the specialty and commonality of workers. In addition, we find that the Accuracy of all the comparing methods increases as the number of tasks increase, since more information is collected as more tasks accomplished.

Table 3. The Accuracy (%) and its standard deviation of WBM and compared methods. $\circ/\bullet$ indicates CrowdWT is statistically better than the comparing method, and the significance is assessed by pairwise t -test at 95% level.

Tasks	MV	DS	BCC	cBCC	WBM-noT	WBM
Landsat						
100	72.15 $\pm$ 0.00 $\bullet$	73.62 $\pm$ 0.11 $\bullet$	74.23 $\pm$ 1.40 $\bullet$	76.12 $\pm$ 1.00 $\bullet$	74.25 $\pm$ 1.51 $\bullet$	77.18 $\pm$ 1.41
1000	75.12 $\pm$ 0.00 $\bullet$	76.31 $\pm$ 1.42 $\bullet$	78.21 $\pm$ 2.41 $\bullet$	80.15 $\pm$ 1.71 $\bullet$	78.44 $\pm$ 1.41 $\bullet$	82.38 $\pm$ 1.01
21684	92.12 $\pm$ 0.00 $\bullet$	92.92 $\pm$ 2.30 $\bullet$	93.62 $\pm$ 1.42 $\bullet$	94.20 $\pm$ 1.12 $\bullet$	92.29 $\pm$ 2.10 $\bullet$	95.08 $\pm$ 2.20
Segment						
100	74.40 $\pm$ 0.00 $\bullet$	75.14 $\pm$ 1.42 $\bullet$	75.09 $\pm$ 2.20 $\bullet$	75.44 $\pm$ 1.10 $\bullet$	75.04 $\pm$ 1.10 $\bullet$	76.27 $\pm$ 0.72
1000	76.14 $\pm$ 0.05 $\bullet$	77.25 $\pm$ 2.14 $\bullet$	82.42 $\pm$ 1.61 $\bullet$	85.90 $\pm$ 1.30 $\circ$	82.32 $\pm$ 1.72 $\bullet$	85.10 $\pm$ 1.20
19424	93.20 $\pm$ 0.00 $\bullet$	93.65 $\pm$ 1.72 $\bullet$	94.12 $\pm$ 1.10 $\bullet$	95.64 $\pm$ 1.20 $\circ$	93.42 $\pm$ 0.14 $\bullet$	95.44 $\pm$ 1.10
News						
100	51.12 $\pm$ 0.00 $\bullet$	52.15 $\pm$ 0.15 $\bullet$	54.15 $\pm$ 1.22 $\bullet$	55.12 $\pm$ 0.21 $\bullet$	54.32 $\pm$ 1.10 $\bullet$	57.30 $\pm$ 0.10
1000	61.15 $\pm$ 0.00 $\bullet$	62.14 $\pm$ 1.15 $\bullet$	62.52 $\pm$ 1.60 $\bullet$	64.25 $\pm$ 1.20 $\bullet$	63.24 $\pm$ 0.12 $\bullet$	66.24 $\pm$ 0.02
14032	81.25 $\pm$ 0.00 $\bullet$	82.11 $\pm$ 1.12 $\bullet$	83.44 $\pm$ 1.20 $\bullet$	86.91 $\pm$ 0.13 $\bullet$	86.32 $\pm$ 0.06 $\bullet$	88.25 $\pm$ 0.01
Birds						
100	69.41 $\pm$ 0.01 $\bullet$	70.11 $\pm$ 0.12 $\bullet$	73.63 $\pm$ 2.11 $\bullet$	74.12 $\pm$ 2.60 $\bullet$	74.52 $\pm$ 0.43 $\bullet$	74.82 $\pm$ 1.10
1000	72.24 $\pm$ 0.00 $\bullet$	73.21 $\pm$ 1.45 $\bullet$	78.72 $\pm$ 2.13 $\bullet$	80.45 $\pm$ 0.02 $\bullet$	80.22 $\pm$ 0.71 $\bullet$	82.23 $\pm$ 1.00
30165	91.41 $\pm$ 0.00 $\bullet$	91.23 $\pm$ 2.11 $\bullet$	91.50 $\pm$ 2.21 $\bullet$	92.54 $\pm$ 1.10 $\bullet$	91.10 $\pm$ 1.42 $\bullet$	93.55 $\pm$ 1.12

From the NLPD scores on the whole datasets reported in Table 4, we can also see that WBM obtains the best NLPD score on Segment, News, and Birds, and achieves the second best NLPD score on Landsat. This observation shows that the aggregations made by WBM have lower uncertainty than the others, and suggests that WBM is able to collect more accurate annotations by jointly modeling workers and tasks.

Table 4. The NLPD scores (the lower the better) of compared methods and WBM. The best result on each dataset is highlighted in **bold** font.

Dataset	DS	BCC	cBCC	WBM-noT	WBM
Segment	1.265	1.054	0.984	0.647	0.560
Landsat	1.611	0.895	0.865	0.874	0.869
News	0.416	0.432	0.311	0.228	0.215
Birds	0.841	0.348	0.624	0.322	0.231

5.3 Robustness to annotation sparsity

In real scenarios of crowdsourcing, it's common that the majority of workers only annotate few tasks, while some workers annotate many. To evaluate the behavior of different algorithms with respect to annotation sparsity, for each dataset, we randomly removed $\{10\%, 20\%, 30\%, 50\%\}$ annotations of each worker to generate sparser annotations, and then adopted the remaining annotations for further experiments. For example, if a worker annotates 100 instances, we randomly remove $\{10\%, 20\%, 30\%, 50\%\}$ annotations of the 100 instances; as a result, 90, 80, 60, 50 annotations of this worker are retained for the experiments. In the random removal process, we ensure that each worker annotates at least one instance. We report the performance of the comparing methods under different ratios of annotations in Figure 10.

Fig. 10. Accuracy of compared methods under different ratios of removed annotations.

As the ratio of removed annotations increases, all the methods have a reduced accuracy. This pattern is expected, since the information collected from workers is gradually decreasing. WBM always outperforms the other four methods across all the four datasets. MV, DS, and BCC are more sensitive to annotation sparsity. When removing 30% annotations of each worker, the accuracy of MV, DS, and BCC has a sharp decrease. This is because MV selects the labels that are annotated by the majority as the ground truth. The accuracy of one instance is greatly reduced when fewer workers annotate it. Although DS and BCC construct a confusion matrix for each worker, they all ignore the issue of annotation sparsity, leading to an imprecise confusion matrix. When removing 50% of the annotations of each worker, WBM still holds an accuracy $>80\%$ and is more robust to annotation sparsity than the other comparing methods. This advantage can be attributed to two factors: (i) WBM not only considers the worker specialty by modeling a confusion matrix for each worker, but also takes into account the worker commonality. (ii) WBM learns the complex

interaction between workers and task features, and thus can capture the influence of task features on workers in crowdsourcing processes to obtain more reliable annotations. For these reasons, cBCC has a lower accuracy than WBM, although it also takes into account annotation sparsity.

5.4 Parameter sensitivity analysis

The three parameters η , γ , and θ may affect the performance of CrowdWT. η controls the weight vector $\mathbf{h}$ in Eq. (14), γ weights the constraint in Eq. (15), θ is a threshold used to determine whether a task is difficult or not. We conduct additional experiments to study the sensitivity of CrowdWT with respect to η , γ , and θ , by repeating the experiment 10 times, and report the average results in Figure 11, 12 and 13. All the experimental protocols are the same as those used when comparing task assignment strategies.

Fig. 11. Accuracy of CrowdWT under different combinations of η and γ

Figure 11 shows the results of CrowdWT under different combinations of η and γ . With a fixed γ , CrowdWT achieves the best performance with $\eta = 10^{-3}$. If η is too small, $\eta\mathbf{I}$ ($\mathbf{I}$ is an identity matrix) has a limited correction of $\Phi(\mathbf{x})^T\Phi(\mathbf{x})$. On the other hand, if η is large, it over-corrects the regression problem, which cannot guarantee stability. As such, a moderate η is expected. Under a fixed η , CrowdWT achieves a relatively stable performance when $\gamma > 2$. This is because a too small γ value does not make sufficient usage of the restrictive condition $(\tilde{D}_n - \theta)(f(\mathbf{x}_n) - \theta) \geq 0$. Consequently, a unreliable task difficulty regressor $f(\mathbf{x}_n)$ in Eq. (14) is learned. Based on the above analysis, we adopt $\eta = 10^{-3}$ and $\gamma = 3$ for the experiments.

Figure 12 shows the results of CrowdWT under different values of θ . The line CrowdWT-Easy denotes the accuracy of annotations on easy tasks. From Figure 12, we can observe that as θ increases, the ratio of easy tasks also increases. Meanwhile, Accuracy of CrowdWT-Easy is stable and high, which means that the crowd assigned with easy tasks can give reliable annotations.

Figure 13 shows the performance of CrowdWT (blue line), the ratio of hard tasks (blue dotted line), and Accuracy on these hard tasks (red line) with respect to different input values of θ , respectively. We can find that the performance of CrowdWT stays stable when $\theta \leq 0.2$ and decreases as $\theta > 0.2$. For example, on the Birds dataset, when $\theta = 0.2$, the ratio of hard tasks is about 30%, and the Accuracy of CrowdWT-hard is about 81%. When $\theta < 0.2$, the ratio of hard tasks will increase, more reliable workers even experts with high cost are needed, which is not realistic. Thus, we set $\theta = 0.2$ for the experiments.

Fig. 12. The accuracy of CrowdWT and the ratio of easy tasks under different task-difficulty thresholds (θ).

Fig. 13. The accuracy of CrowdWT and the ratio of hard tasks under different task-difficulty thresholds(θ).

To investigate whether the size of training set affects the estimation of the task difficulty $f(\mathbf{x}_n)$, we run experiments with different sizes (L/N) of training sets (from 5% to 40% with an interval 5%) and reveal the results on four datasets in Figure 14. We can find that the Accuracy of CrowdWT on different datasets remains relatively stable as the size increasing. It indicates that the Accuracy of CrowdWT is not sensitive to the size of training data $[L]$. This pattern shows that a good and reliable $f(\mathbf{x}_n)$ can be easily found by the Kernel ridge regression in Eq. (16).

5.5 Runtime analysis

CrowdWT and the other comparing methods (whose code were provided by the authors) are implemented using different languages; as such, it's not meaningful to compare their empirical runtime costs. Therefore, we give the time complexity of five task assignment approaches (RLS,

Fig. 14. Accuracy of CrowdWT under different sizes (L/N) of training set.

Fig. 15. Runtime of CrowdWT on the Birds dataset.

QASCA, GEM, Crowddqs and OKG) and of our CrowdWT, and report the runtime of CrowdWT in assigning tasks on the Birds dataset.

Suppose that all strategies assign N tasks to W workers into C classes. RLS randomly assigns all tasks to W workers. Therefore, the complexity of RLS is $O(WNC)$. OKG takes $O(WNC)$ to infer the posterior distribution and $O(MNW)$ to assign the tasks (M is the total budget), so its complexity is $O(WNC + MNW)$. QASCA takes $O(N + kN + WC)$ to train the model and $O(WNC)$ to assign the tasks, where $k \ll N$ is the selected size of the candidate tasks for a worker. Therefore, the complexity of QASCA is $O(N + kN + WC + WNC)$. GEM takes $O(WNC + WC)$ to infer the worker skills and intentions, and $O(NWC^2)$ to assign the tasks, so its complexity is $O(WNC + WC + NWC^2)$. Crowddqs takes $O(N^2)$ to assign the tasks, $O(NW^2)$ to estimate the worker's accuracy, and $O(NW)$ to aggregate answers. Therefore, its complexity is $O(NW^2 + NW + N^2)$. CrowdWT takes $O(dCWL)$ to train the WBM, where $L \ll N$ is the size of training set, $O(WL)$ to estimate $f(\mathbf{x}_n)$ and $O(NC)$ to assign the tasks. Thus, the computational complexity of CrowdWT is $O(dCWL + WL + NC)$. Since $\{d, L\} \ll N$, CrowdWT has a lower complexity than OKG, GEM, Crowddqs and QASCA. Although, the complexity of OKG, GEM, QASCA, Crowddqs and CrowdWT are all higher than RLS, the formers can obtain a higher accuracy than RLS.

Figure 15 reveals the runtime of crowdWT on a server with Intel(R) Xeon(R) CPU E7-4820 and 128GB RAM. We can see the runtime increases linearly as the number of tasks increase, which is in accordance with the theoretical time complexity. Since the ground truth label aggregation can be done offline, we can state that CrowdWT is efficient and it takes fewer than 45s on a large dataset (#tasks=30K, #workers=460).

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel task assignment strategy (CrowdWT) via jointly modeling tasks and workers with a family of graphical models. CrowdWT can capture the influence of task features on worker decisions and obtain more reliable annotations. By modeling task difficulty, CrowdWT further assigns the task to the more appropriate workers, whose skills are above the difficulty of the task. Extensive experimental results show that CrowdWT can more wisely assign the task than other representative and related task assignment strategies, and it can obtain the same quality of annotations with much reduced cost. In addition, the introduced worker bias model of CrowdWT also outperforms other related approaches that model workers using the similar techniques, and it is more robust to annotations sparsity than these approaches. CrowdWT implicitly assumes that all the workers have the same expense during crowdsourcing. We will extend CrowdWT with each worker having her/his own cost. Our work is offline done currently. From the results and analysis

on different datasets, we can find that CrowdWT works well in task assignment. We will extend it for a real crowdsourcing platform in the future.

ACKNOWLEDGMENTS

We appreciate the anonymous reviewers for their comments on improving our work, and the authors who kindly shared their source code and datasets with us for the experiments. This work is supported by Natural Science Foundation of China (61872300 and 61871020) and Qilu Scholarship of Shandong University.

REFERENCES

- [1] Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Mae Borromeo, and Motomichi Toyama. 2016. Task composition in crowdsourcing. In *DASFAA*. 194–203.
- [2] Kyohei Atarashi, Satoshi Oyama, and Masahito Kurihara. 2018. Semi-supervised learning from crowds using deep generative models. In *AAAI*. 1555–1562.
- [3] Ioannis Boutsis and Vana Kalogeraki. 2014. On task assignment for real-time reliable crowdsourcing. In *ICDCS*. 1–10.
- [4] Xi Chen, Qihang Lin, and Dengyong Zhou. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *ICML*. 64–72.
- [5] Xi Chen, Qihang Lin, and Dengyong Zhou. 2015. Statistical decision making for optimal budget allocation in crowd labeling. *JMLR* 16, 1 (2015), 1–46.
- [6] Peng Cheng, Lei Chen, and Jieping Ye. 2019. Cooperation-Aware Task Assignment in Spatial Crowdsourcing. In *ICDE*. 1442–1453.
- [7] Anand Inasu Chittilappilly, Lei Chen, and Sihem Amer-Yahia. 2016. A survey of general-purpose crowdsourcing techniques. *TKDE* 28, 9 (2016), 2246–2266.
- [8] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* (1979), 20–28.
- [9] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*. 1015–1030.
- [10] Bin Guo, Yan Liu, Leye Wang, Victor O. K. Li, Jacqueline C. K. Lam, and Zhiwen Yu. 2018. Task Allocation in Spatial Crowdsourcing: Current State and Future Directions. *IEEE Internet of Things Journal* 5, 3 (2018), 1749–1764.
- [11] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. 2013. Adaptive task assignment for crowdsourced classification. In *ICML*. 534–542.
- [12] Chien-Ju Ho and Jennifer Wortman Vaughan. 2012. Online task assignment in crowdsourcing markets. In *AAAI*. 45–51.
- [13] Zehong Hu and Jie Zhang. 2018. A Novel Strategy for Active Task Assignment in Crowd Labeling. In *IJCAL*. 1538–1545.
- [14] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *SIGKDD*. 64–67.
- [15] Asif R Khan and Hector Garcia-Molina. 2017. Crowddqs: Dynamic question selection in crowdsourcing systems. In *SIGMOD*. 1447–1462.
- [16] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *AISTAT*. 619–627.
- [17] Aditya Kurve, David J. Miller, and George Kesidis. 2015. Multicategory Crowdsourcing Accounting for Variable Task Difficulty, Worker Skill, and Worker Intention. *TKDE* 27, 3 (2015), 794–809.
- [18] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *ICML*. 331–339.
- [19] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. 2016. Crowdsourced data management: A survey. *TKDE* 28, 9 (2016), 2296–2319.
- [20] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. Cdas: a crowdsourcing data analytics system. *VLDB* 5, 10 (2012), 1040–1051.
- [21] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *SIGKDD*. 745–754.
- [22] Fenglong Ma, Chuishi Meng, Houping Xiao, Qi Li, Jing Gao, Lu Su, and Aidong Zhang. 2017. Unsupervised discovery of drug side-effects from heterogeneous data sources. In *SIGKDD*. 967–976.
- [23] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. 2016. Using Hierarchical Skills for Optimized Task Assignment in Knowledge-Intensive Crowdsourcing. In *WWW*. 843–853.
- [24] Aranyak Mehta and Debmalaya Panigrahi. 2012. Online matching with stochastic rewards. In *FOCS*. 728–737.
- [25] Thomas P. Minka. 2001. *A Family of Algorithms for Approximate Bayesian Inference*. Ph.D. Dissertation.
- [26] Kaixiang Mo, Erheng Zhong, and Qiang Yang. 2013. Cross-task crowdsourcing. In *SIGKDD*. 677–685.
- [27] Julien Pilourdault, Sihem Amer-Yahia, Senjuti Basu Roy, and Dongwon Lee. 2018. Task Relevance and Diversity as Worker Motivation in Crowdsourcing. In *ICDE*. 365–376.
- [28] Guo-Jun Qi, Charu C Aggarwal, Jiawei Han, and Thomas Huang. 2013. Mining collective intelligence in diverse groups. In *WWW*. 1041–1052.
- [29] Nihar Shah, Dengyong Zhou, and Yuval Peres. 2015. Approval voting and incentives in crowdsourcing. In *ICML*. 10–19.
- [30] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *SIGKDD*. 614–622.
- [31] Edwin Simpson and Stephen Roberts. 2015. Bayesian methods for intelligent task assignment in crowdsourcing systems. In *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*. 1–32.

- [32] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*. 1–35.
- [33] Dapeng Tao, Jun Cheng, Zhengtao Yu, Kun Yue, and Lizhen Wang. 2019. Domain-weighted majority voting for crowdsourcing. *TNNLS* 30, 1 (2019), 163–174.
- [34] Yongxin Tong, Lei Chen, Zimu Zhou, Hosagrahar Visvesvaraya Jagadish, Lidan Shou, and Weifeng Lv. 2019. SLADE: A Smart Large-Scale Task Decomposer in Crowdsourcing. In *ICDE*. 2133–2134.
- [35] Yongxin Tong, Jieying She, Bolin Ding, Lei Chen, Tianyu Wo, and Ke Xu. 2016. Online minimum matching in real-time spatial data: experiments and analysis. *VLDB* 9, 12 (2016), 1053–1064.
- [36] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. 2016. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*. 49–60.
- [37] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *WWW*. 155–164.
- [38] Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. 2011. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *SIGIR*. 21–26.
- [39] Wei Wang, Xiang-Yu Guo, Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. 2017. Obtaining high-quality label by distinguishing between easy and hard items in crowdsourcing. In *IJCAI*. 2964–2970.
- [40] Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *JASA* 58, 301 (1963), 13–30.
- [41] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. 2010. *Caltech-UCSD Birds 200*. Technical Report CNS-TR-2010-001. California Institute of Technology.
- [42] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2009. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In *NIPS*. 2035–2043.
- [43] John Winn and Christopher M Bishop. 2005. Variational message passing. *JMLR* 6 (2005), 661–694.
- [44] Jinfu Xia, Yan Zhao, Guanfeng Liu, Jiajie Xu, Min Zhang, and Kai Zheng. 2019. Profit-driven task assignment in spatial crowdsourcing. In *IJCAI*. 1914–1920.
- [45] Jingru Yang, Ju Fan, Zhewei Wei, and Guoliang Li. 2019. Cost-Effective Data Annotation using Game-Based Crowdsourcing. In *VLDB*. 57–70.
- [46] Jing Zhang, Victor S Sheng, and Tao Li. 2017. Label aggregation for crowdsourcing with bi-layer clustering. In *SIGIR*. 921–924.
- [47] Yan Zhao, Jinfu Xia, Guanfeng Liu, Han Su, Defu Lian, Shuo Shang, and Kai Zheng. 2019. Preference-aware task assignment in spatial crowdsourcing. In *AAAI*, Vol. 33. 2629–2636.
- [48] Zhou Zhao, Furu Wei, Ming Zhou, Weikeng Chen, and Wilfred Ng. 2015. Crowd-Selection Query Processing in Crowdsourcing Databases: A Task-Driven Approach. In *EDBT*. 397–408.
- [49] Yudian Zheng, Guoliang Li, and Reynold Cheng. 2016. DOCS: a domain-aware crowdsourcing system using knowledge bases. *VLDB* 10, 4 (2016), 361–372.
- [50] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *VLDB* 10, 5 (2017), 541–552.
- [51] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. 2015. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*. 1031–1046.
- [52] Yao Zhou, Arun Reddy Nelakurthi, and Jingrui He. 2018. Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners. In *SIGKDD*. 2817–2826.
- [53] Yao Zhou, Lei Ying, and Jingrui He. 2017. Multic2: an optimization framework for learning from task and worker dual heterogeneity. In *SDM*. 579–587.
- [54] Yao Zhou, Lei Ying, and Jingrui He. 2019. Multi-task Crowdsourcing via an Optimization Framework. *TKDD* 13, 3 (2019), 27:1–27:26.