

Optimizing microtask assignment on crowdsourcing platforms using Markov chain Monte Carlo

Alireza Moayedikia^{a,*}, Hadi Ghaderi^a, William Yeoh^b

^a Department of Business Technology and Entrepreneurship, Swinburne Business School, Swinburne University of Technology, Hawthorn 3122, VIC, Australia

^b Department of Information Systems and Business Analytics, Faculty of Business and Law, Deakin University, Burwood 3125, VIC, Australia



ARTICLE INFO

Keywords:

Crowdsourcing
Task assignment
Markov chain
Crowd labeling
Quality estimation

ABSTRACT

Microtasking is a type of crowdsourcing, denoting the act of breaking a job into several tasks and allocating them to multiple workers to complete. The assignment of tasks to workers is a complex decision-making process, particularly when considering budget and quality constraints. While there is a growing body of knowledge on the development of task assignment algorithms, the current algorithms suffer from shortcomings including: *after-worker quality estimation*, meaning that workers need to complete all tasks after which point their quality can be estimated; and *one-off quality estimation* method which estimates workers' quality only at the start of microtasking using a set of pre-defined quality-control tasks. To address these shortcomings, we propose a Markov Chain Monte Carlo-based task assignment approach known as MCMC-TA which provides iterative estimations of workers' quality and dynamic task assignment. Specifically, we apply Gaussian mixture model (GMM) to estimate workers' quality and Markov Chain Monte Carlo to shortlist workers for task assignment. We use Google Fact Evaluation dataset to measure the performance of MCMC-TA and compare it against the state-of-the-art algorithms in terms of AUC and F-Score. The results show that the proposed MCMC-TA algorithm not only outperforms the rival algorithms, but also offers a spammer-resistant result that maximizes the learning of workers' quality with minimal budget.

1. Introduction

Crowdsourcing is the act of breaking a job into smaller pieces, so-called tasks (or microtasks), and outsourcing the tasks to an undefined network of people [16,17]. Given the availability of crowdsourcing platforms and knowledgeable workers, microtasking has become an emerging business model, allowing a wide range of tasks to be allocated and completed by individuals [8,27]. Microtasking has received additional momentum in recent years as a cost-effective approach to leverage crowd-based human computation and intelligence [7,9,19]. A typical scenario for microtasking platforms is to employ a repeated labeling approach [34], where tasks are distributed online, and then, completed and submitted by workers [24,41]. Accordingly, workers are equally rewarded upon submission of their work for the tasks they submitted.

However, the repeated labeling method used in microtask crowdsourcing platforms has limitations, which may negatively affect the quality of the final job and performance of the platform. A major weakness of this approach is untargeted task assignments, where work is allocated to the crowd without examining the quality of their skills

and interest [10]. Untargeted assignments may not only increase the cost of the job, but also deteriorate the quality of results as those workers with insufficient skills may produce inferior results [26].

The existence of such challenges has motivated many researchers to consider solutions capable of optimizing the allocation of tasks, known as task assignment algorithms. Task assignment algorithms aim to personalize and optimize the process of allocating tasks to workers by using attributes determined by the crowdsourcing platform. For example, the Appen crowdsourcing platform considers a quality index that requires a job owner to design and distribute quality-control indicators to measure the workers' performance [13,25,31].

Some researchers have proposed solutions to further enhance the performance of conventional crowdsourcing platforms. For example, Yu et al. [42] proposed the surprise-minimization-value-maximization (SMVM) approach to optimize task allocation, while maximizing social welfare through establishing a worker desirability index (WDI). WDI represents three elements of a worker – reputation, workload and motivation. Dai et al. [4] introduced AI agents that use Bayesian network learning and an inference-based model with partially observable Markov Decision Processes to achieve a balance between cost and

* Corresponding author.

E-mail address: amoayedikia@swin.edu.au (A. Moayedikia).

quality. The algorithm proposed by Dai et al. [4] considers tasks' difficulty. However, task difficulty remains a questionable area in crowdsourcing research since a task might appear difficult to one worker while easy to another one. In short, the existing task assignment algorithms suffer from the following shortcomings:

- *After-work estimation*: The after-work quality estimation [1,11] means a pool of workers must first attempt and complete the assigned tasks and get paid, before an algorithm can estimate worker quality according to their provided answers. This estimation will be used for task assignment in future jobs. Although such algorithms consider the quality of workers before aggregation of the collected answers, they are not efficient in terms of the cost and accuracy of microtasking. This is because worker quality estimation occurs *after* task assignment.
- *One-off estimation*: To overcome the problems of the after-work quality estimation approach, some algorithms have proposed the idea of injecting quality-control tasks in the microtasking process [14,15,35,36]. In particular, the job owner designs some quality-control tasks, where their ground truths are known, to be completed by the workers prior to the start of microtasking. While using quality-control tasks, workers' quality can be estimated, and eligible workers are selected for microtasking. However, the issue with this approach is that workers' quality is measured only once, the workers' quality during the microtasking process might change. Therefore, the performance of workers is not monitored continually during the course of the job.
- *Task-difficulty consideration*: Some algorithms use tasks' difficulty as one of the elements to assign tasks [15,35,40]. Consideration of task difficulty is challenging since a task that is hard for some workers might be easy for others. Also, in algorithms that request workers to attempt some initial quality-control tasks, matching the difficulty of quality-control tasks and those tasks in the actual job is another challenge. If quality-control tasks are easier or harder than tasks in the actual job, the evaluated quality may not truly represent workers' quality. Hence, consideration of task difficulty is a challenge in task assignment algorithms and may result in poor understanding of workers' ability and quality in completing the tasks [3].

To overcome the above-mentioned shortcomings, we propose Markov Chain Monte Carlo Task Assignment (MCMC-TA) approach, which is the first study to jointly employ MCMC and Gaussian Mixture Model (GMM) for assigning tasks on crowdsourcing platforms. We further evaluate the proposed MCMC-TA approach in terms of cost effectiveness and resistance against spammers in budget-driven and population-based experiments. In our proposed MCMC-TA algorithm, tasks arrive sequentially [2,23,25] and will be assigned to multi-task workers (i.e., workers with different skills such as annotation, labeling, transcription and etc. [24]) to attempt. Next, using GMM component, the quality of worker will be estimated, followed by the MCMC component of the algorithm that predicts the likelihood of workers' suitability for the next task.

The scope of current research is in line with the growing body of knowledge that addresses the functionality of decision support systems applied in crowdsourcing platforms [21,26]. Given the inherent complexity associated with design and operations of large-scale crowdsourcing platforms, such system functionality is of particular interest from both research and practice perspectives [12].

The remainder of this paper is organized as follows. Section 2 presents the literature review, followed by a description of the details of the MCMC-TA algorithm in Section 3. Section 4 evaluates the performance of the MCMC-TA algorithm in comparison with the state-of-the-art algorithms, and finally Section 5 discusses conclusion and contributions.

2. Literature review

While the crowdsourcing concept presents new opportunities to access resources in a more efficient manner, many of the platforms struggle with turning their promises into reality because of complexities associated with the assignment of work to the right crowd workers [43]. Given such challenges, research in task assignment has received momentum over the past decade as a result of growth in the number and diversity of crowdsourcing platforms. Although academic research on crowdsourcing task assignment has been accumulating, there remains several gaps and limitations in the current body of knowledge, which opens opportunities for developing more robust and practical algorithms that address the challenges associated with the quality and performance of crowdsourcing platforms.

Algorithms proposed in the literature mostly focus on modeling the quality of individual workers [39], with two general methods to pair workers with tasks [24]. In the first approach, the aim is to select the right worker from a set of existing candidates for a newly registered task. Whereas, the second method aims to identify the right job from a pool of tasks for a newly registered worker. Because the ultimate goal is to optimize the allocation of tasks to workers, this section provides a review of these two approaches.

The early stages of the literature demonstrate several algorithms that require human intervention. Algorithms such as those developed by Khattak and Salleb-Aouissi [18] and Pfeiffer et al. [29] involve humans as a one-off approach in which an expert designs quality measures to identify the reliable workers on the basis of the answers received by them. However, workers' quality is subject to change over the microtasking process, which means a worker who seems to be reliable might become unreliable, and vice versa.

Therefore, the one-off human-intervention approach for quality evaluation may not truly detect the reliable workers for future task allocations. Similarly, Ho and Vaughan [14] proposed the dual task assigner (DTA) algorithm, which relies on human input to measure workers' quality. By formulating the problem in the online primal-dual framework, the DTA assumes that worker expertise is unknown and should be learned by exploration. Such assumptions allow spammers to falsely demonstrate high levels of competency to receive more tasks, which may harm the performance of crowdsourcing platforms. Despite the practicality of the human-intervention approach for small-scale microtasking, in large-scale microtasking platforms where tasks and workers are both homogeneous and heterogeneous, human intervention is not practical or feasible [26].

The limitations with the one-off human-intervention approach have been largely resolved through automated task assignment approaches in more recent studies. However, these approaches are limited by the answering type (i.e., an algorithm works on binary-answer tasks but does not work on multiple-choice tasks). For instance, the work proposed by Ertekin et al. [6], called CrowdSense, requires tasks with binary type of answers and operates in online settings where tasks arrive one at a time. The algorithm dynamically samples subsets of workers by using an exploration/exploitation criterion and then creates a combined weight of each subset's votes that approximates the opinions of the workers. Such models are beneficial to create a full picture of the entire crowd by using a representative subset. Similarly, Tarasov et al. [37] proposed a method to dynamically estimate workers' quality in regression (DER^3) by benefiting from a multi-armed bandit (MAB) technique. Such methods of estimating quality are practical for situations that involve labeling or rating corpora for use in supervised machine learning.

More recent works have benefited from EM algorithms. Such algorithms encompass an *E*-step, which aims at estimating the workers' quality index, and an *M*-step, which estimates the rough ground truth. In this context, Long et al. [20] proposed a probabilistic model that

actively learns from the crowd by using Gaussian process classifier. Applying two levels of a flip model, the model is capable of characterizing the overall label noise and skill level of individual labelers. Such probabilistic mechanisms allow the model to dynamically assign the tasks to the crowd with quality expertise. As another example of an EM-style algorithm, we can refer to the work of Rodrigues et al. [32], in which the authors proposed a probabilistic approach for sequence labeling by using conditional random fields (CRFs). The method can be used for situations where ground truths do not exist but label sequences from multiple annotators are available. Nevertheless, the EM-style computation has limitations as it benefits from heuristic techniques but lacks in solution rigor [37].

To address the shortage of EM methods in relation to task dependency, Moayedikia et al. [26] proposed ROUGH set based eXpertise estimation (ROUX), which uses a rough dataset approach in conjunction with a harmony search to detect the most suitable community of workers and estimate crowd expertise. Once experts are known, they can be ranked and assigned to a task. The limitation with this work is the assumption that the pool of workers does not change, and all the workers are available to be assigned to any tasks. Overall, the existing task assignment algorithms suffer from two main limitations. The first limitation lies with the one-off quality estimation approach, which leads to uncertainty in the overall quality assurance of the platform. The second limitation is associated with the after-work assessment of workers' performance. Therefore, the need for development of efficient task assignment algorithms capable of addressing these issues is important for microtasking platforms. In following Section 3 we introduce a novel algorithm that overcomes these limitations by iteratively estimating the workers' quality and making dynamic task assignment for each round of the microtasking process.

3. Proposed Markov chain Monte Carlo task assignment algorithm

This section introduces our proposed MCMC-TA algorithm by first giving an overview on the problem and our proposed solution, followed by detailed explanations of the two main components: Markov Chain Monte Carlo (MCMC) and Gaussian Mixture Model (GMM). Table 1 provides a description of frequently used notations.

3.1. Problem definition and solution overview

In microtask crowdsourcing platforms, the microtasking scenario involves jobs, job owners, workers, and a platform owner, where every job¹ contains several tasks (also known as microtasks). In MCMC-TA, workers $W \in \{w_1, \dots, w_k\}$, are a set of self-registered candidates with a quality index of $R = \{\mu_1, \mu_2, \mu_3, \dots, \mu_k\}$ who are willing to attempt the available tasks $X = \{x_1, x_2, x_3, \dots, x_n\}$ of a job. Every worker after finishing a task gets paid according to the allocated budget of the task $B = \{b_1, b_2, b_3, \dots, b_X\}$. Once a job is finished (i.e., all of its tasks have been attempted), the job owner collects the final aggregated answers for each task.

In this microtasking scenario, the problem is to identify high-quality workers when the tasks are assigned sequentially. The problem definition is based on the assumption that: 1) on a microtasking platform, the pool of workers is mixed, including genuine and non-genuine workers with a varying quality index; and 2) similar to Appen platform, we assumed that job owners design several quality-control tasks that are attempted by the workers prior to the start of microtasking. This enables MCMC-TA to have some prior knowledge about the quality of the workers.

To solve this problem, we propose a task assignment algorithm (so-called MCMC-TA) that works based on Gaussian Mixture Model (GMM)

¹ Jobs might be similar, but they are independent. This means completion of a job is not dependent on completion of another one.

Table 1
Frequently used symbols and notations in this paper.

Notations	Description
W	Set of available workers on the platform ready to attempt the tasks
w_k	The k th worker from set W
R	Set of quality of all workers
μ_k	Quality of the k th worker
X	Set of all tasks belong to a job
x_n	The n th task from set X
B	The entire budget allocated to all tasks
b_X	Budget allocated to the X th task
μ	Average of quality of a worker
Σ	Range of possible changes to a worker quality
D	Number of tasks a worker has completed so far
π	Size of the Gaussian function
W_{nk}	Worker k attempted task n
π_k	Size of Gaussian function of worker k
$\theta = \{\pi, \mu, \Sigma\}$	The current latent parameters of a worker
$\theta^* = \{\pi^*, \mu^*, \Sigma^*\}$	The estimated latent parameters of a worker using GMM
d_i	The i th task completed previously by a worker
Q_c	Set of quality control tasks

and Monte Carlo Markov Chain (MCMC). Recent advances in MCMC methods have encouraged its application in many fields such as hospital ward management [2], marketing and finance [28]. Although MCMC is an effective tool for capturing useful information about distributions in the Bayesian inference [38], its application in microtasking research is scarce, with few examples using MCMC-based methods for the inference of variables [22,33].

MCMC-TA first starts with an initialization stage in which jobs (that contain several tasks) are advertised by job owners on the platform. Then, interested workers are asked to solve the designated quality-control tasks. The aim of quality-control tasks is to assess the quality of workers before they attempt the actual job. For this purpose, we apply the work proposed by Khattak and Salleb-Aouissi [18] as the base algorithm to measure the workers' initial quality. Workers who refrain from attempting the quality tasks will not be considered for the job. The MCMC component takes the quality index of workers estimated in initialization stage and shortlists some workers for the next stage, known as the microtasking stage.

Within the microtasking stage, MCMC-TA starts by assigning the first iteration of tasks to the shortlisted workers (i.e., those workers who were shortlisted in the initialization stage), collecting their answers and estimating their quality index using GMM. The quality indices of shortlisted workers (along with quality indices of other workers from the previous task assignment iteration) are then used by the MCMC component to evaluate the likelihood of a worker being suitable for the next iteration of task assignment. Fig. 1 shows the cycle of quality estimation and task assignment in MCMC-TA. As shown in Fig. 1, Stage 1 is task assignment, followed by Stage 2 in which the quality of workers will be estimated based on their attempts made in Stage 1. Finally, Stage 3 uses MCMC and considers all workers (i.e., workers shortlisted and not shortlisted for the most recent task assignment) and their quality indices to shortlist workers for Stage 1 and the next round of task assignment. The algorithm cycles through Steps 1 to 3 iteratively until all tasks are accomplished.

3.2. GMM for quality estimation

The Gaussian Mixture Model is used for calculation of workers' quality according to the collected answers. Let X be a task attempted by a worker, μ is the average of quality of a worker, Σ be the range of possible changes to a worker quality, D is the number of tasks a worker is going to complete and, π is the size of the Gaussian function. A worker's quality metrics of μ and Σ after solving a task can be estimated using Gaussian density function as shown in Eq. (1):

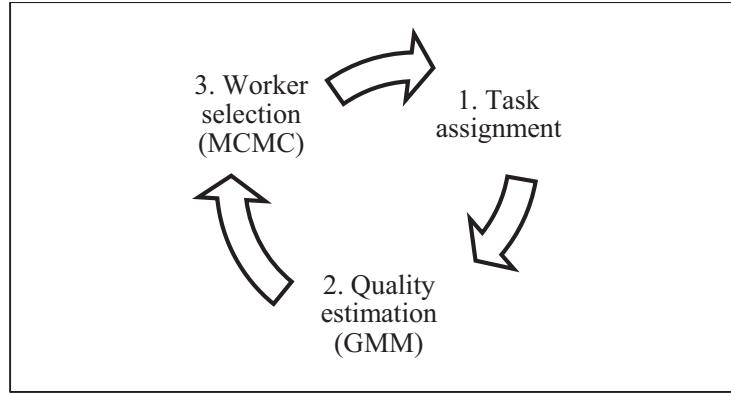


Fig. 1. Steps involved in the microtasking stage of MCMC-TA.

$$N(X | \mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu)^T \Sigma^{-1}(X - \mu)\right) \quad (1)$$

with its log format shown in Eq. (2):

$$\ln N(X | \mu, \Sigma) = -\frac{D}{2} \ln 2\pi - \frac{1}{2} \ln \Sigma - \frac{1}{2}(X - \mu)^T \Sigma^{-1}(X - \mu) \quad (2)$$

Expression $N(X | \mu, \Sigma)$ indicates given a worker with average quality μ and quality variation of Σ , how suitable the worker is for task X . In this equation there are three latent variables that should be estimated to measure a worker suitability on task X . In Eq. (2) there are three latent variables of $\theta = \{\pi, \mu, \Sigma\}$, where estimating three latent variables in one equation is challenging. Other authors [13,31] have used Expectation-Maximization (EM) as an effective approach to estimate several latent variables. Hence, we consider an EM style estimation to measure θ latent variables, as detailed in Section 3.3. To better understand how EM works, we need to know the probability of a task assigned to and solved by a worker as shown in Eq. (3),

$$P(W_{nk} = 1 | x_n) \quad (3)$$

where W_k is a latent variable that only receives 0 or 1, indicating whether the worker k attempted task n . Specifically, this equation means, what is the likelihood of a task being suitable for worker k . Knowing the probability of occurrence of $W_k = 1$ helps with estimating Gaussian mixture parameters, as will be discussed later. To generalise Eq. (3), we introduce Eq. (4)

$$p(W_k = 1) = \pi_k \quad (4)$$

This equation means, what is the probability of a worker being suitable for a job. Considering set W , as the set of all workers $W \in \{w_1, \dots, w_K\}$, where the performance of every worker is independent from other workers; the suitability of every worker for a task is estimated using Eq. (5):

$$p(w) = p(w_1 = 1)^{w_1} p(w_2 = 1)^{w_2} \dots p(w_K = 1)^{w_K} = \prod_{k=1}^K \pi_k^{w_k} \quad (5)$$

And the probability of assigning a task to a suitable worker is calculated using Eq. (6):

$$P(x_n | w) = \prod_{k=1}^K N(x_n | \mu_k, \Sigma_k)^{w_k} \quad (6)$$

However, our aim is to estimate the probability (i.e., quality) of

worker w being qualified for task x_n . This can be done through the Bayes rule; where Eqs. (5) and (6) help to measure this probability:

$$P(x_n, w) = p(x_n | w)p(w) \quad (7)$$

In Eq. (7), expression $p(x_n | w)p(w)$ is calculated using Eqs. (5) and (6) and through marginalization we can sum up the terms on W and finally estimate $p(x_n)$ as shown in Eq. (8):

$$P(x_n) = \sum_{k=1}^K p(x_n | w)p(w) = \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k) \quad (8)$$

Maximum likelihood can be used to determine the optimal values for parameters shown in Eq. (8). This is done by considering likelihood as the joint probability over all tasks as defined in Eq. (9):

$$P(X) = \prod_{n=1}^N p(x_n) = \prod_{n=1}^N \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k) \quad (9)$$

And by applying a logarithm on Eq. (9):

$$P(X) = \sum_{n=1}^N \ln \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k) \quad (10)$$

In Eq. (10) the logarithm affecting the second summation makes it hard to calculate the optimal parameters. Section 3.3.1 explains how this logarithm will be removed and make EM calculations simple. However, going back to our initial aim (i.e., determining if a worker is qualified for a task) using Bayes rule we know that:

$$P(w_k = 1 | x_n) = \frac{p(x_n | w_k = 1)p(w_k = 1)}{\sum_{j=1}^K p(x_n | w_j = 1)p(w_j = 1)} \quad (11)$$

Also, as we showed previously in this section:

$$p(W_k = 1) = \pi_k, p(x_n | w_k = 1) = N(x_n | \mu_k, \Sigma_k)$$

By replacing these expressions in Eq. (11) we can come up with Eq. (12):

$$P(w_n = 1 | x_n) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)} = \gamma(w_{nk}) \quad (12)$$

Eq. (12) will be used in expectation maximization measure workers' quality as discussed in the next section.

3.3. MCMC for task assignment

In MCMC-TA, Expectation-Maximization starts by an initialization step. In this step θ variables will be assigned to some values based on the following heuristics: We initially assign ten quality control micro-tasks and estimate workers quality based on the algorithm proposed by Khattak and Salleb-Aouissi [18]. μ is the average of every worker's quality across ten different tasks and Σ is the lowest and highest quality of a worker on ten quality control tasks. The size of Gaussian function (π) has been defined as one divided by number of workers.

3.3.1. Expectation step

This step calculates Eq. (13):

$$Q(\theta^*, \theta) = \mathbb{E} \left[\ln p(X, W | \theta^*) \right] = \sum_W p(W | X, \theta) \ln p(X, W | \theta^*) \quad (13)$$

where θ^* is the revised parameters. In Eq. (13) the value of $p(W | X, \theta)$ expression has been calculated using Eq. (12). For Gaussian Mixture Models, the expectation step calculates the value of γ in Eq. (12) using the old parameter values (i.e., θ). By replacing Eq. (12) in Eq. (13), we will have:

$$Q(\theta^*, \theta) = \sum_W \gamma(w_{nk}) \ln p(X, W | \theta^*) \quad (14)$$

To calculate $\ln p(X, W | \theta^*)$ and consequently θ^* , we take a complete likelihood of our model that gives us:

$$p(X, W | \theta^*) = \prod_{n=1}^N \prod_{k=1}^K \pi^{w_{nk}} N(x_n | \mu_k, \Sigma_k)^{w_{nk}} \quad (15)$$

Eq. (15) resulted from the joint probability of over workers' performance on allocated tasks, answers collected from workers, and an extension of the initial derivations for (X). Applying a logarithm on Eq. (15), generates:

$$\ln p(x_n, w_k | \theta^*) = \sum_{n=1}^N \sum_{k=1}^K w_{nk} [\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)] \quad (16)$$

By replacing Eq. (16) into (14) we get Eq. (17) as:

$$Q(\theta^*, \theta) = \sum_{n=1}^N \sum_{k=1}^K \gamma(w_{nk}) [\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)] \quad (17)$$

This helps use to eliminate the logarithm operation appeared in Eq. (10). Eq. (16) makes it easier to calculate latent variables by maximizing Q with respect to the parameters as discussed in the next section.

3.3.2. Maximization step

Maximization step calculates the revised parameters of the Gaussian Mixture Model using:

$$\theta^* = \operatorname{argmax}_{\theta} Q(\theta^*, \theta) \quad (18)$$

However, we need to incorporate the fact that the size of all Gaussian functions of all workers is equal to one (i.e., $\sum_{k=1}^K \pi_k = 1$). This can be done by adding a suitable Lagrange multiplier to Eq. (17) to generate Eq. (19) where the parameters can be easily determined using maximum likelihood.

$$Q(\theta^*, \theta) = \sum_{n=1}^N \sum_{k=1}^K \gamma(w_{nk}) [\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)] - \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) \quad (19)$$

By taking the derivative of Q with respect to π and set it equal to

zero:

$$\frac{\partial Q(\theta^*, \theta)}{\partial \pi_k} = \sum_{n=1}^N \frac{\gamma(w_{nk})}{\pi_k} - \lambda = 0 \quad (20)$$

Eq. (21) extracted from Eq. (20) by reorganising the terms and adding a summation over k operation to the sides of Eq. (20):

$$\sum_{n=1}^N \frac{\gamma(w_{nk})}{\pi_k} = \pi_k \lambda \rightarrow \sum_{k=1}^K \sum_{n=1}^N \gamma(w_{nk}) = \sum_{k=1}^K \pi_k \lambda \quad (21)$$

Summing up all coefficients π of all workers equals one (i.e., $\sum_{k=1}^K \pi_k = 1$). Also, summing up the probabilities γ over all workers equals one (i.e., $\sum_{k=1}^K \gamma_k = 1$), and hence $\lambda = |W|$. Using this, we can solve for π :

$$\pi_k = \frac{\sum_{n=1}^N \gamma(w_{nk})}{|W|} \quad (22)$$

where $|W|$ is the total number of workers registered on a platform.

Finally, workers' latent variables will be calculated through differentiating Q with respect to μ and Σ , assuming derivative is equal to zero and solving the parameters through log-likelihood (see Eq. (2)):

$$\mu_k^* = \frac{\sum_{n=1}^N \gamma(w_{nk}) x_n}{\sum_{n=1}^N \gamma(w_{nk})}, \Sigma_k^* = \frac{\sum_{n=1}^N \gamma(w_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(w_{nk})}$$

The revised values will be used to determine γ in the next EM iteration. The iterative EM computation continues until convergence in the likelihood value occurs. Once EM converged, the values of μ_k^* and Σ_k^* in the final EM iteration will be used to shortlist workers for task assignment.

3.4. Markov chain Monte Carlo for task decisions

This section discusses how the integration of MCMC with GMM works to shortlist workers based on their predicted quality values θ^* (i.e., μ_k^* and Σ_k^*) and their previous quality values. Workers' quality during microtasking changes as they work on several tasks. On some tasks, workers have better knowledge while in some other tasks their knowledge is less. That is, a worker for instance, might perform well on the first 5 tasks while perform moderately on the subsequent 5 tasks. Hence, we need a solution to consider the sequence of task attempts of a worker and based on that decide on worker's eligibility for the next task assignment. Hence, we integrate GMM with Markov Chain Monte Carlo (MCMC) to predict eligible workers for the next task allocation, where on the one hand GMM generates and preserves a sequence of workers' quality indices, and on the other hand MCMC uses that sequence to shortlist workers for the next task to be assigned.

The MCMC component used in our proposed algorithm relies on Metropolis–Hastings (MHs) [22] as an efficient approach to handle high-dimensional data. In the MCMC-TA setting, high dimensionality denotes a situation in which workers are assigned many tasks to complete. The MHs technique requires an initial distribution, which is called the transition model $\varphi\left(\frac{\theta^*}{\theta}\right)$, to draw suitable workers. MHs uses φ function to randomly explore the distribution space and finally decide on acceptance or rejection of workers. In MHs to predict the next set of workers for the next task, MCMC draws a symmetric distribution from $\varphi\left(\frac{\theta^*}{\theta}\right)$. To decide if the worker based on the estimated distribution can be accepted, Eq. (23) must be calculated:

$$\frac{P\left(\frac{\theta^*}{D}\right)}{P\left(\frac{\theta}{D}\right)} \quad (23)$$

Using Bayes formula, this can be re-formulated as:

$$\frac{P(\theta^*)P\left(\frac{\theta^*}{D}\right)}{P(\theta)P\left(\frac{\theta}{D}\right)} \quad (24)$$

which is also equivalent to:

$$\frac{\prod_i^{|D|} f(d_i/\theta^*)P(\theta^*)}{\prod_i^{|D|} f(d_i/\theta)P(\theta)} \quad (25)$$

Hence, the rule for acceptance is formulated as:

$$P(\text{acceptance}) = \begin{cases} \frac{\prod_i^{|D|} f(d_i/\theta^*)P(\theta^*)}{\prod_i^{|D|} f(d_i/\theta)P(\theta)}, & \prod_i^{|D|} f(d_i/\theta)P(\theta) > \prod_i^{|D|} f(d_i/\theta^*)P(\theta^*) \\ 1, & \prod_i^{|D|} f(d_i/\theta)P(\theta) \leq \prod_i^{|D|} f(d_i/\theta^*)P(\theta^*) \end{cases} \quad (26)$$

This means if θ^* is more likely than θ , then the worker will be accepted, unless otherwise. As shown in Algorithm 1, MCMC-TA first starts by assigning quality control tasks to workers to estimate an initial reliability rate for workers using the approach proposed by Khattak and Salleb-Aouissi [18].

Algorithm 1. Markov Chain Monte Carlo Task Assignment

Input:

- W : set of workers
- X : set of tasks
- Q_t : set of quality control tasks
- b_x : budget allocated to task x once completed by worker w

Output:

- A : Final aggregated answers

Algorithm:

1. **foreach** quality task qt in Q_t
 2. $A_{qt} \leftarrow$ get all workers to attempt quality control task qt
 3. $QA_{qt} \leftarrow$ apply Gaussian Mixture Model on A_{qt} to measure workers quality
 4. $R \leftarrow R \cup QA_{qt}$
5. **end foreach**
6. $R \leftarrow \{\}$
7. **foreach** task x in X
 8. **foreach** worker w in set W
 9. offer task x to w
 10. **if** w accepts x
 11. assign task x to w
 12. Pay b_x amount to worker w
 13. $A_{qt} \leftarrow$ collect and aggregate the answers of worker w
 14. $QA_{qt} \leftarrow$ use Gaussian Mixture Model to measure the quality of all workers in set W
 15. $R \leftarrow R \cup QA_{qt}$
16. $\theta^* \leftarrow$ using workers reliability set R and they collected responses, estimate their latent variables
17. $w \leftarrow$ using θ^* and Equation (26) of MCMC predict the next set of eligible workers in W
18. **end foreach**

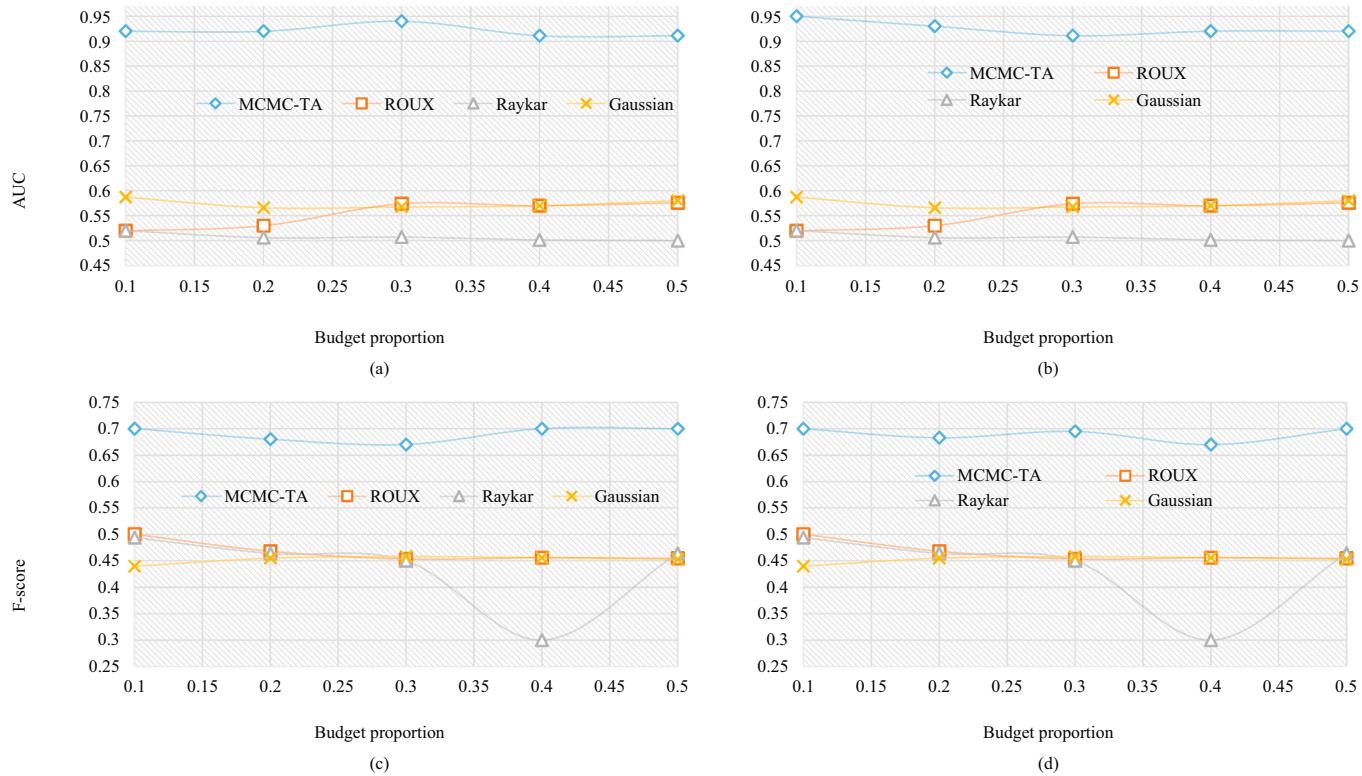


Fig. 2. Budget-driven MCMC-TA performance and comparisons with rival algorithms using AUC and F-score (a)(c) $T = 550$ and (b)(d) $T = 750$.

4. Analyses and discussions

This section evaluates the performance of the MCMC-TA algorithm using a real-world dataset called Google Fact Evaluation² (GFE). This is a text dataset with 57 workers, in which workers are asked to judge more than 40,000 instances of textual data. To ensure that the MCMC-TA's performance has been tested and compared in a fair experimental condition, the rival algorithms of Raykar et al., [31], ROUX [26], and Gaussian [13] along with MCMC-TA were implemented and tested by considering two evaluation metrics of F-Score and Area Under Curve (AUC) as discussed in Section 4.1. The experiments in Section 4.1 investigate whether the budget spent on learning about workers' responses to tasks influences MCMC-TA³ performance in comparison with other novel algorithms. Section 4.2 details the impact of population size (i.e., when a pool of workers is small versus large) to study how it can boost MCMC-TA's performance, followed by a Wilcoxon signed-rank statistical test to investigate the significance of MCMC-TA's performance in comparison to rivals from a statistical point of view.

4.1. Budget-driven experiments

In budget-driven experiments, we investigate how increasing our budget in measuring workers' quality affects the performance of MCMC-TA in comparison with other novel algorithms such as Raykar et al., [31], ROUX [26], and Gaussian [13] algorithms. In MCMC-TA budget-driven experiments, we consider two sets of tasks with sizes of 550 and 750, where for each set, we assume that the proportion of quality-control tasks to the tasks of the actual job varies between 10 and 50%. For instance, when a job owner has a total of 550 tasks, then a 10%

proportion indicates that there are 55 quality-control tasks and 550 tasks in the actual job.

Fig. 2 shows the performance comparison between MCMC-TA and the rival algorithms of ROUX, Gaussian, and Raykar. In the rival algorithms, as the number of quality-control tasks increases, the performance either enhances or deteriorates. This might be due to the presence of noisy workers who provide random answers, which leads to the deterioration of performance. The results also indicate that the rival algorithms can be affected by the number of quality-control tasks. Only as the number of quality-control tasks increases can the rival algorithms learn better about the quality of the workers.

The results shown in Fig. 2 indicate that, in MCMC-TA, allocation of 10 or 50% of the microtasking budget for learning from workers generates a similar performance (e.g., Fig. 3(a) and (d)). In other words, if we allocate either 10% or 50% to learn from workers, the impact of this learning on MCMC-TA's performance does not change. To a microtask owner this means MCMC-TA can yield remarkable performance even with minimal budget, through maximizing the learning of workers' quality with only 10% of the total budget.

Most importantly, MCMC-TA outperforms the rival algorithms in terms of F-score and AUC metrics, achieved through the combined utility of GMM and MCMC components. This performance could be justified as MCMC filters out low-quality workers in an iterative manner. After each round of task allocation, all workers' answers are collected by MCMC and fed to GMM to estimate the quality of workers using an expectation maximization manner. The outcome of GMM is workers' quality. Using Eqs. (16) and (17), MCMC then predicts if the candidate workers are eligible for the next round of task assignment.

4.2. Population-based experiments

This section investigates the robustness of MCMC-TA against the availability of spammers in the pool of workers. As introduced by Raykar and Yu [30], workers can generally be divided into two groups

²The dataset can be downloaded from: <https://sites.google.com/site/crowdscale2013/shared-task/task-fact-eval>

³The python code can be downloaded from: <https://bit.ly/2ZS96Eb>

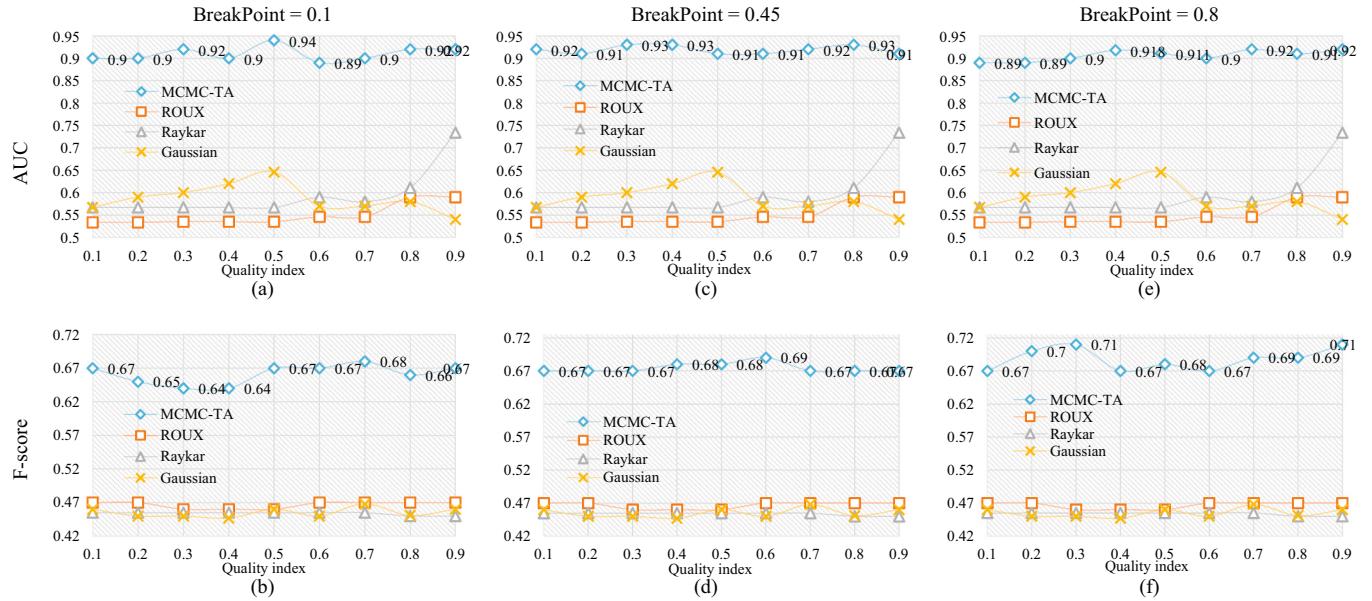


Fig. 3. Evaluation of MCMC-TA in terms of F-score and AUC with comparisons with rival algorithms.

of spammers and hammers. A ‘Spammer’ refers to a non-genuine worker who aim to collect financial rewards or harm the platform, while a ‘Hammer’ is a genuine worker who is willing to attempt tasks and participate in microtasking. We adopt this definition from Raykar and Yu [30] and apply it in MCMC-TA, in which ‘Spammer’ are considered as low-quality workers whereas ‘Hammer’ are high-quality workers.

To make the definition of ‘high’ and ‘low’ clear, we introduce a term as $BreakPoint \in \{0.1; 0.45; 0.8\}$ to distinguish high-quality workers from low-quality workers. In a pool of workers, those with a quality index lower than the specified $BreakPoint$ are considered low-quality, while the rest are high-quality. In the population-based experiments, we also discuss how the MCMC-TA’s performance is affected if the population is mostly spammer or hammer. For instance, a proportion with a value of 0.1 means that only 10% of workers have a quality index lower than the $BreakPoint$ (i.e., they are spammers) and that the rest are above the defined $BreakPoint$ (i.e., they are hammers).

As shown in Fig. 3 when the $BreakPoint$ increases, the population quality decreases, which means there are fewer hammer workers in the pool. For the aim of consistency of experimentation, we conduct the experiments of MCMC-TA and the rival algorithms in 10 separate runs, where each run consists of one round of task allocation and quality estimation through the actual algorithms (e.g., MCMC-TA or other rival algorithms). The final results (i.e., the values of F-score and AUC) are the average values across the best value of each run.

The results shown in Fig. 3 indicate that MCMC-TA’s performance is only changed slightly when the pool of workers is affected by the existence of spammers (as also shown in Fig. 4). In contrast to MCMC-TA, the Gaussian [13] and Raykar et al., [31] algorithms are affected markedly by the proportion of spammers in the pool of workers. In the Gaussian algorithm, the AUC metric on average fluctuates within 10% variation, and Raykar’s AUC metric increases from 55% to around 75%. However, the changes in the AUC metric in MCMC-TA are not more than 4%. Even though ROUX’s resistance against spammer availability is remarkable, this algorithm is a one-off quality estimation algorithm that estimates workers’ quality in an offline mode (i.e., when workers are not solving any tasks) and therefore is outperformed by MCMC-TA. Overall, MCMC-TA shows significant performance compared to rival algorithms due to two reasons.

First, it benefits from the combination of the GMM for quality estimation and MCMC for shortlisting suitable workers. The rival algorithms shown in Fig. 3 are only based on Gaussian formulation, while

MCMC-TA not only uses GMM but also MCMC. This indicates that the combination of these two components is effective in improving task assignment, specifically Gaussian-based task assignment.

Second, MCMC-TA is capable of handling high dimensionality (i.e., when workers attempt a large number of tasks) through Metropolis–Hastings (MHs) [22]. In the GFE dataset we assign 40,000 tasks to workers, noting that MCMC-TA generates better results (i.e., higher AUC and F-Score) and hence handles a large number of tasks more efficiently than its rival algorithms.

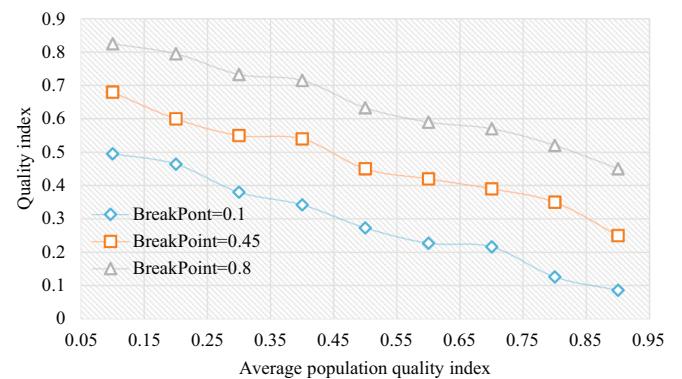


Fig. 4. The impact of various $BreakPoint$ values on the average of workers’ quality in MCMC-TA.

Table 2
Wilcoxon signed-rank test results.

	MCMC-TA	ROUX	Raykar	Gaussian
AUC	0.1 0.45 0.8	0 (0.00742) 0 (0.00757) 0 (0.00763)	0 (0.00747) 0 (0.00747) 0 (0.00757)	0 (0.00768) 0 (0.00763) 0 (0.00763)
F-score	0.1 0.45 0.8	0 (0.00747) 0 (0.0066) 0 (0.00768)	0 (0.00747) 0 (0.00705) 0 (0.00763)	1 (0.0108) 1 (0.0107) 1 (0.0108)

4.3. Wilcoxon statistical test

Clearly, the practical utility of MCMC-TA lies in its performance for shortlisting eligible workers for task assignment. Further, to evaluate if the results of MCMC-TA are significant and independent from the other rival algorithms (i.e., rejecting the null hypothesis), we conduct Wilcoxon signed-rank test [5]. The Wilcoxon signed-rank test is a more sensible measure than other statistical tests such as the *t*-test. In the Wilcoxon signed-rank test, the aim is to understand the differences between the performance of two given algorithms, and the absolute magnitude quantifying these differences is not of interest. The Wilcoxon test is also a safer test as it does not assume normal distributions and outliers have less effect on the final result. Table 2 shows the test results in the form of $h(p)$, where h refers to the test value and p indicates to what extent the null hypothesis has been rejected. The test results indicate that the null hypothesis is rejected to various degrees between MCMC-TA and other rival algorithms in AUC and F-score measures. The rejection of the null hypothesis indicates that the algorithms are independent (or dissimilar).

5. Conclusions

In this paper, we present a novel task assignment algorithm (called MCMC-TA) for microtask crowdsourcing platforms, which introduces an iterative quality measurement through GMM and MCMC for task assignment. The GMM part estimates the quality index of workers, and the MCMC component determines the list of suitable candidates to be assigned for a task by using the input from the GMM. The performance of the algorithm was tested using GFE dataset and compared with the baseline and three state-of-the-art algorithms. As demonstrated in the simulation study, MCMC-TA outperforms the rival algorithms in terms of AUC and F-score.

This paper contributes to the research of crowdsourcing and microtasking, by showing the effectiveness of MCMC when integrated with GMM. It represents the first study to jointly employ MCMC and GMM for microtask assignment on crowdsourcing platforms. As shown in the budget-driven experiments, MCMC-TA maximizes the learning of workers by using only 10% of the microtasking budget, while other algorithms need more expenditure to learn about workers. In the population-based experiments, we also show that the integrated approach is an effective approach to obtain spammer-resistant performance compared with other existing approaches. Our work also contributes to the literature of decision support system, by developing an effective approach for predicting suitable candidate workers for an outsourced job on crowdsourcing platforms. MCMC-TA algorithm uses prior distributions generated based on workers' performance in the assigned tasks to approximate workers' reliability before the actual task assignment. This filtration is performed using the MH mechanism embedded in the structure of our algorithm. Furthermore, the MCMC-TA approach proposed in this research addresses the shortcomings of the current microtasking algorithms. MCMC-TA overcomes the problem of after-work quality estimation in a more effective fashion by formulating microtasking assignment as a Markov Chain Monte Carlo problem and estimate workers' quality once they finish a microtask.

Our research also has implications for practice, particularly for crowdsourcing platform owners. Conventional crowdsourcing platforms suffer from the existence of spammers and low-performing workers. Prior knowledge of workers' performance before the actual task assignment is an important piece of information to filter out unfit candidates and spammers from the pool of available workers. MCMC-TA assists crowdsourcing platforms to identify a pool of workers that perform better, resulting in lower payments and higher job quality, and eventually, platform success. In addition, the iterative quality estimation approach proposed in this research overcomes the limitations of one-off quality estimation approach. While the one-off quality estimation ignores the change in workers' performance over the course of task

completion, iterative estimation ensures a fair and consistent payment in accordance with workers' performance.

The task assignment algorithm proposed in this research is generalizable not only to other online crowdsourcing settings such as citizen journalism, macro-tasking, idea generation, etc., but also for business applications where effective task assignment is a core operation (e.g., crowd delivery/shipping). Subject to modification, MCMC-TA can be adapted for crowdsourced parcel delivery platforms, in which the quality of delivery is measured based on reliability and speed of service. Furthermore, MCMC-TA can be applied to any environment where the quality of work performed by an individual/entity is the determinant of future task assignment.

Like all other studies, this study has some limitations that provide avenues for future research. First, our research is based on this assumption that all workers are available on the crowdsourcing platform to attempt the outsourced tasks. Future studies can consider the possibility that workers may quit during the microtasking process. Second, the MCMC-TA algorithm was tested using GFE dataset. The experiments could be further expanded by using other datasets from other domains. Finally, the MCMC-TA algorithm developed in this research was initially designed for assigning online tasks such as annotation and labeling. Future research can adapt MCMC-TA for assignment of spatial tasks such as crowd-shipping or car sharing.

References

- [1] H. Amirkhani, M. Rahmati, Agreement/disagreement based crowd labeling, *Appl. Intell.* 41 (2014) 212–222.
- [2] A.R. Andersen, B.F. Nielsen, L.B. Reinhardt, Optimization of hospital ward resources with patient relocation using Markov chain modeling, *Eur. J. Oper. Res.* 260 (2017) 1152–1163.
- [3] I. Boutsis, V. Kalogeraki, On task assignment for real-time reliable crowdsourcing, *IEEE 34th International Conference on Distributed Computing Systems*, 2014, IEEE, 2014, pp. 1–10.
- [4] P. Dai, C.H. Lin, D.S. Weld, POMDP-based control of workflows for crowdsourcing, *Artif. Intell.* 202 (2013) 52–85.
- [5] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [6] S. Ertekin, C. Rudin, H. Hirsh, Approximating the crowd, *Data Min. Knowl. Disc.* 28 (2014) 1189–1221.
- [7] M.J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, R. Xin, CrowdDB: Answering queries with crowdsourcing, *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 2011, pp. 61–72.
- [8] X. Gan, X. Wang, W. Niu, G. Hang, X. Tian, X. Wang, J. Xu, Incentivize multi-class crowd labeling under budget constraint, *IEEE J. Select. Areas Commun.* 35 (2017) 893–905.
- [9] J. Gao, X. Liu, B.C. Ooi, H. Wang, G. Chen, An online cost sensitive decision-making method in crowdsourcing systems, *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 2013, pp. 217–228.
- [10] D. Geiger, M. Schader, Personalized task recommendation in crowdsourcing information systems—current state of the art, *Decis. Support. Syst.* 65 (2014) 3–16.
- [11] R. Gilyazev, D.Y. Turdakov, Active learning and crowdsourcing: a survey of optimization methods for data labeling, *Program. Comput. Softw.* 44 (2018) 476–491.
- [12] H. Gimpel, V. Graf-Drasch, R.J. Laubacher, M. Wöhl, Facilitating like Darwin: supporting cross-fertilisation in crowdsourcing, *Decis. Support. Syst.* 113282 (2020).
- [13] P. Groot, A. Birlutiu, T. Heskes, Learning from multiple annotators with Gaussian processes, *International Conference on Artificial Neural Networks*, Springer, 2011, pp. 159–164.
- [14] C.J. Ho, J.W. Vaughan, Online task assignment in crowdsourcing markets, *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [15] C.J. Ho, S. Jabbari, J.W. Vaughan, Adaptive task assignment for crowdsourced classification, *Int. Conf. Mach. Learn.* (2013) 534–542.
- [16] J. Howe, The rise of crowdsourcing, *Wired Mag.* 14 (2006) 1–4.
- [17] K.R. Jespersen, Crowdsourcing design decisions for optimal integration into the company innovation system, *Decis. Support. Syst.* 115 (2018) 52–63.
- [18] F.K. Khattak, A. Salleb-Aouissi, Quality control of crowd labeling through expert evaluation, *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011, p. 5.
- [19] K. Lee, J. Caverlee, S. Webb, The social honeypot project: Protecting online communities from spammers, *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 1139–1140.
- [20] C. Long, G. Hua, A. Kapoor, A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing, *Int. J. Comput. Vis.* 116 (2016) 136–160.
- [21] C. Miao, H. Yu, Z. Shen, C. Leung, Balancing quality and budget considerations in mobile crowdsourcing, *Decis. Support. Syst.* 90 (2016) 56–64.
- [22] K. Mo, E. Zhong, Q. Yang, Cross-task crowdsourcing, *Proceedings of the 19th ACM*

- SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 677–685.
- [23] A. Moayedikia, K.-L. Ong, Y.L. Boo, Yeoh, W. Bee colony based worker reliability estimation algorithm in microtask crowdsourcing, 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, IEEE, 2016, pp. 713–717.
- [24] A. Moayedikia, W. Yeoh, K.-L. Ong, Y.-L. Boo, Framework and literature analysis for crowdsourcing's answer aggregation, *J. Comput. Inf. Syst.* 60 (2017) 49–60.
- [25] A. Moayedikia, K.-L. Ong, Y.L. Boo, W.G. Yeoh, Task assignment in microtask crowdsourcing platforms using learning automata, *Eng. Appl. Artif. Intell.* 74 (2018) 212–225.
- [26] A. Moayedikia, W. Yeoh, K.-L. Ong, Y.L. Boo, Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach, *Decis. Support. Syst.* 122 (2019) 113065.
- [27] A.G. Neiat, A. Bouguettaya, S. Mistry, Incentive-based crowdsourcing of hotspot services, *ACM Transac. Internet Technol. (TOIT)* 19 (2019) 1–24.
- [28] R. Paap, What are the advantages of MCMC based inference in latent variable models? *Statistica Neerlandica* 56 (2002) 2–22.
- [29] T. Pfeiffer, X.A. Gao, Y. Chen, A. Mao, D.G. Rand, Adaptive polling for information aggregation, Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
- [30] V.C. Raykar, S. Yu, Eliminating spammers and ranking annotators for crowdsourced labeling tasks, *J. Mach. Learn. Res.* 13 (2012) 491–518.
- [31] V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, *J. Mach. Learn. Res.* (2010) 11.
- [32] F. Rodrigues, F. Pereira, B. Ribeiro, Sequence labeling with multiple annotators, *Mach. Learn.* 95 (2014) 165–181.
- [33] M. Rostami, D. Huber, T.-C. Lu, A crowdsourcing triage algorithm for geopolitical event forecasting, 2018, Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 377–381.
- [34] V.S. Sheng, F. Provost, P.G. Ipeirotis, Get another label? Improving data quality and data mining using multiple, noisy labelers, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 614–622.
- [35] J. Song, H. Wang, Y. Gao, B. An, Active learning with confidence-based answers for crowdsourcing labeling tasks, *Knowl.-Based Syst.* 159 (2018) 244–258.
- [36] Y. Suzuki, Y. Matsuda, S. Nakamura, Additional operations of simple HITs on microtask crowdsourcing for worker quality prediction, *J. Inform. Process.* 27 (2019) 51–60.
- [37] A. Tarasov, S.J. Delany, B. Mac Namee, Dynamic estimation of worker reliability in crowdsourcing for regression tasks: making it work, *Expert Syst. Appl.* 41 (2014) 6190–6210.
- [38] D. Van Ravenzwaaij, P. Cassey, S.D. Brown, A simple introduction to Markov chain Monte-Carlo sampling, *Psychon. Bull. Rev.* 25 (2018) 143–154.
- [39] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, M. Shokouhi, Community-based bayesian aggregation models for crowdsourcing, Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 155–164.
- [40] J. Whitehill, T.-F. Wu, J. Bergsma, J.R. Movellan, P.L. Ruvolo, Whose vote should count more: optimal integration of labels from labelers of unknown expertise, *Adv. Neural Inf. Proces. Syst.* (2009) 2035–2043.
- [41] P. Wu, E.W. Ngai, Y. Wu, Toward a real-time and budget-aware task package allocation in spatial crowdsourcing, *Decis. Support. Syst.* 110 (2018) 107–117.
- [42] H. Yu, C. Miao, Y. Chen, S. Fauvel, X. Li, V.R. Lesser, Algorithmic management for improving collective productivity in crowdsourcing, *Sci. Rep.* 7 (2017) 1–11.
- [43] J. Zhang, X. Wu, V.S. Sheng, Active learning with imbalanced multiple noisy labeling, *IEEE Transac. Cybernetics* 45 (2014) 1095–1107.

Alireza Moayedikia is a lecturer in information systems and business analytics at Faculty of Business and Law, Swinburne University of Technology. He received his PhD from Deakin University Australia. His research interests include machine learning, data analytics and their applications in crowdsourcing and recommender systems. His research works have appeared in reputable journals such as Decision Support Systems, Engineering Applications of AI, and Neurocomputing.

Hadi Ghaderi is a senior lecturer at Swinburne Business School, Swinburne University of Technology, Australia. He has led several industry-engaged research projects in the area of supply chain digitalization and transformation. His research interest is focused around supply chain digitalization, digital business optimization and data analytics. He is also the Program Leader for Supply Chain Analytics at Swinburne Data Science Research Institute, where the focus is on building supply chain capability by sensing various data sources and providing advanced analytics for smarter decision making.

William Yeoh is an associate professor at Deakin University's Department of Information Systems and Business Analytics. His research is supported by various funding bodies and has appeared in high-tier journals. He has received several awards including the ICT Educator of the Year Gold Award, internationally competitive IBM Faculty Awards, Deakin's Vice-Chancellor Award for Value Innovation, and Deakin's Faculty Excellence in Research Award. His coached team was crowned the World Champion at the 2016 IBM Watson Analytics Global Competition held in Las Vegas.