A Cyber Physical System Crowdsourcing Inference Method Based on Tempering: An Advancement in Artificial Intelligence Algorithms

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6618980, 11 pages
https://doi.org/10.1155/2021/6618980

Research Article

A Cyber Physical System Crowdsourcing Inference Method Based on Tempering: An Advancement in Artificial Intelligence Algorithms

Jia Liu,[1] **Mingchu Li**,[1] **William C. Tang**,[2] **and Sardar M. N. Islam**[3]

[1]*School of Software Technology and Key Laboratory for Ubiquitous Network and Service Software, Dalian University of Technology, Dalian 116620, China*
[2]*Department of Biomedical Engineering, University of California, Irvine 92697-2715, USA*
[3]*Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne 14428, Australia*

Correspondence should be addressed to Jia Liu; jialiudlut@gmail.com

Received 19 December 2020; Revised 24 January 2021; Accepted 6 February 2021; Published 19 February 2021

Academic Editor: Wei Wang

Activity selection is critical for the smart environment and Cyber-Physical Systems (CPSs) that can provide timely and intelligent services, especially as the number of connected devices is increasing at an unprecedented speed. As it is important to collect labels by various agents in the CPSs, crowdsourcing inference algorithms are designed to help acquire accurate labels that involve high-level knowledge. However, there are some limitations in the algorithm in the existing literature such as incurring extra budget for the existing algorithms, inability to scale appropriately, requiring the knowledge of prior distribution, difficulties to implement these algorithms, or generating local optima. In this paper, we provide a crowdsourcing inference method with variational tempering that obtains ground truth as well as considers both the reliability of workers and the difficulty level of the tasks and ensure a local optimum. The numerical experiments of the real-world data indicate that our novel variational tempering inference algorithm performs better than the existing advancing algorithms. Therefore, this paper provides a new efficient algorithm in CPSs and machine learning, and thus, it makes a new contribution to the literature.

1. Introduction

1.1. Research Background. Crowdsourcing refers to the practice whereby a company or organization outsources the tasks that used to be performed by employees to the crowds of nonspecific Internet agents. The wide use of the Internet enables crowdsourcing to make use of the wisdom of crowds as a cheap, fast, and convenient method. Therefore, crowdsourcing is much more powerful than traditional methods, especially in the fields of computer vision, natural language processing, environmental protection [1, 2], etc.

In recent years, the development of embedded computing and wireless communications has enabled the CPSs to become an important research and industrial field. A CPS is often designed as a cooperative network, consisting of sensors, actuators, and controllers [3–6]. In many applications, such as object identification, traffic management, and smart health, CPSs need to extract information and process data in a large scale, a task that is difficult or unable to accomplish at one single device [7, 8]. Therefore, crowdsourcing is important for CPSs to combat the above limitations [9–11].

However, existing CPS crowdsourcing also has disadvantages: for instance, (1) sometimes, agents can be unreliable, or there may be attackers or spammers who aim to corrupt the system; therefore, it is intuitive to assign the same tasks to multiple workers; (2) the difficulty levels of the tasks can vary a lot that if the workers are not properly rewarded, they may choose only the easy tasks, thus degrading the system; (3) it is also critical to allocate tasks to the right kind of workers in order to improve the efficiency and quality of the answers. However, when there is no longer a need to improve the quality of the collected data, we need to improve the computing algorithm to better extract the useful information from the data. Therefore, it is very important to implement inference algorithms to extract information from the data and deal with the above three disadvantages, the procedure of which is called crowdsourcing inference problem.

1.2. Related Work. In many fields, such as human pose estimation and smart medical, crowdsourcing inference has become a useful and cost-effective method to denote a large quantity of data. The existing crowdsourcing inference algorithms can be classified into four categories, including the weighted majority voting (WMV) method, the statistical method, new kind of inference tools, and variational method.

The first way to obtain ground truth is by majority voting (MV) or WMV, assigning different workers' different weight according to their expertise or reliability [12–15]. The limitation of MV is that it regards all workers to be equally and easily subject to attacks. In contrast to MV, WMV is intuitively more efficient, and all other inference algorithms can be treated as a kind of WMV, so it is meaningless to discuss the limitations of WMV since the situation varies accordingly. In general, the weight of workers should be assigned according to their reliability; therefore, some researchers use qualification test or hidden test to determine the weight of workers [16]. However, this approach is highly dependent on a rich budget and the cooperation of workers.

Another way is to use a statistical method, for example, maximum likelihood estimation (MLE) and expectation maximization (EM). Expectation maximization (EM) is a well-established way to compute the hyperparameters and ground truth in the process of implementing MLE. EM first makes some initial guess on the hyperparameters, then computes expectations under these hyperparameters, and repeats this process until convergence is reached. This process can be treated as an iterative decoding process, so EM is convenient and easy to understand [16–18]. However, EM may result in local optima and is challenging to scale.

Therefore, some researchers turn to a new kind of inference tools, including the widely used back-propagation (BP) and mean filed method (MF) [19]. These algorithms can be effective, as well as can guarantee local optimality and sometimes can guarantee global optimality. However, standard BP requires the knowledge of prior distribution, and in reality, it is difficult to implement due to the process of passing messages in the form of sufficient statistics. MF is the approximation approach that maximizes energy functional over the approximate distribution, which is easy to compute. However, MF may lose valuable information and fail to capture the dependency property of posterior distribution.

An efficient way to avoid the above disadvantages is to use variational method. For example, Liu et al.'s [20] variational inference algorithm (VMP) performs extremely well with good worker distribution prior. In tasks of crowdsourcing inference, Hoffman et al. [21] have formulated it into a stochastic variational inference problem. Liu et al. [22] also formulated the crowdsourcing inference problem into a variational inference model. Cai et al. [23] proposed a crowdsourcing prediction algorithm using variation inference. In summary, variational inference (VI) translates inference problem into an optimization problem with multiple local optimum. However, using the tempering technique, we can curb this to some extent. Intuitively, VI objective uses high-entropy variational distribution to replace data-proper variational distribution. By tempering, we can punish low-entropy distributions and loose this restraint gradually to reach distributions that fit the data.

1.3. Motivation. In view of the problems of the existing literature in this area of CPS crowdsourcing inference algorithms discussed above, the main objectives of this paper are to model the CPS crowdsourcing inference problem into a message passing algorithm, i.e., our novel variational tempering inference algorithm (VTI), and to study the performance error bound of this algorithm.

As a popular sampling as well as computing method, Markov Chain Monte Carlo (MCMC) has proved its efficiency in the field of CPSs and big data, due to its ability to deal with large amount of data, and simplifies the gradient computation [24]. Variational tempering (VT) is an extension of MCMC. Therefore, in VTI, we introduce global temperature. We then implement the traditional VI method to compute the gradient of it in each iteration and further update it in the next iteration, until the algorithm reaches convergence, which is also similar to the way that we treat tasks' answers as a distribution as well as worker reliability.

After researching into the existing works, we borrow the idea of VT and solve the problem of crowdsourcing inference by considering worker reliability and task difficulty. There are few papers in the literature considering task difficulty level. The reason we are considering task difficulty level in this paper is that we can assign tasks in a more efficient way [25], infer the ground truth more accurately [26, 27], and reward workers by means of incentives [28] etc. Therefore, the main objective of this paper is to obtain task ground truth by making the best use of worker reliability and task difficulty. The specific tasks we will perform in this paper are the following: (1) Model the crowdsourcing inference problem into a message passing algorithm. (2) Study the performance error bound of our novel variational tempering inference algorithm (VTI). (3) Use real data to simulate the probability error of VTI.

1.4. Contributions. The main contribution of this paper is a new algorithm to CPS crowdsourcing inference. Based on variational analysis and the property of exponential family, a kind of probabilistic graphical model, we try to solve crowdsourcing inference problem in CPSs. First, we infer the ground truth by considering worker reliability. Being different from previous works, we then hypothesize that tasks have different difficulty levels and use this as a parameter to eliminate the same worker constraint.

Therefore, the contributions of the paper are listed as follows:

(1) Formulate the CPS crowdsourcing inference problem into a message passing algorithm

(2) Study the performance error bound of the proposed variational tempering inference algorithm (VTI)

(3) Use real data to simulate the probability error of VTI. In general cases, VTI performs better than other algorithms

In this paper, we first review CPS crowdsourcing inference and variational tempering in Section 1. We then define the background assumptions of the inference model in Section 2. Thereafter, we derive our algorithm stepwise and show the procedure of how we obtain the results in Sections 3 and 4. We show the results of our algorithm in comparison with others in Section 5. Our VTI algorithm requires less constraint on worker prior and task difficulty level and performs better than the existing inference algorithms. In the end, Section 6 presents the conclusion of this study.

2. Preliminaries

In this part, we describe the mathematical notations of VTI. Assuming in the crowdsourcing network, there are M workers and N tasks. Suppose that each task has the value of $\{\pm 1\}$, and we use z_i to denote the true label of it, i.e., $z_i \in \{\pm 1\}$. We then denote $\mathcal{N}_j$ as worker j's neighbor set, i.e., the indexes of tasks that worker j takes. Likewise, $\mathcal{M}_i$ task i's neighbor set, i.e., the indexes of workers that task i is assigned to. $L \in \{0, \pm 1\}^{N\times M}$ is the label matrix, and $L_{ij} \in \{\pm 1\}$ is worker j's answer or label to task i, and $L_{ij} = 0$ means there is no link between worker j and task i. The goal of our VTI is to get the ground truth and difficulty level of task i as well as the reliability of worker j.

We use $q_j = \text{prob}[L_{ij} = z_i], i \in \mathcal{N}_i$ to denote the reliability of worker j and $d_i = \text{prob}[L_{ij} = z_i], j \in \mathcal{M}_j$ to denote the difficulty level of task i. Note that q and d are two separate factors, so the calculation of q does not involve d, and the calculation of d does not involve q. However, they both influence the label of each task. In addition, when $q_j \approx 1$, worker j produces 100% right answers. $q_j \approx 1/2$ can be tricky since the worker may be a spammer or just submits random answers that contain no useful information. However, if $q_j \leq 1/2$, the worker's answers can be made use of by just reverting its answer matrix. Further, we assume each worker is independent, and its reliability is drawn from the same distribution with hyperparameter θ_1. We assume different tasks are also independent, drawn from the distribution with hyperparameter θ_2.

It is reasonable to study cases when there are fewer spammers and attackers in the system; therefore, we assume $E[q_j \mid \theta_1] \geq 1/2$. We use beta prior for worker reliability $p(q_j \mid \theta_1) \propto q_j^{\alpha_1 - 1}(1-q_j)^{\beta_1 - 1}$ [29].

In Liu et al.'s work [22], the authors treat the crowdsourcing issue as a graphical inference problem, of which they first use Bayesian theory to reduce the objective function and then use variational inference to compute

$$\begin{aligned} p(z, q, d \mid L, \theta) &\propto \prod_{j\in[M]} p\left(q_j \mid \theta_1\right) \prod_{i\in[N]} p(d_i \mid \theta_2) \prod_{i\in\mathcal{N}_j} p\left(L_{ij} \mid z_i, q_j, d_i\right) \\ &= \prod_{j\in[M]} p\left(q_j \mid \theta_1\right) q_j^{c_j} \left(1 - q_j\right)^{\gamma_j - c_j} \prod_{i\in[N]} p(d_i \mid \theta_2), \end{aligned} \tag{1}$$

where $\gamma_j = |\mathcal{N}_j|$ is the number of answers given by worker j and $c_j \coloneqq \sum_{i\in\mathcal{N}_j} \mathbb{I}[L_{ij} = z_i]$ is the number of correct answers worker j gives. It follows that the value of z with the minimum error probability is

$$\begin{aligned} \hat{z}_i &= \operatorname{argmax}_{z_i} p(z_i \mid L, \theta) \text{ where } p(z_i \mid L, \theta) \\ &= \sum_{z_{[N]\setminus i}} \int_d \int_q p(z, q, d \mid L, \theta) dq dd \end{aligned} \tag{2}$$

Note that variable z is discrete and q is continuous, and $\theta = \{\theta_1, \theta_2\}$. Because the joint probability depends on both q and z, it is intractable to calculate $p(z, q \mid L, \theta)$ directly. We can take a detour and take the integration over q_j, resulting in a marginal posterior over z.

$$\begin{aligned} p(z \mid L, \theta) &= \int_d \int_q p(z, q, d \mid L, \theta) dq dd \\ &= \prod_{j\in[M]} \int_0^1 \int_0^1 p\left(q_j \mid \theta_1\right) p(d_i \mid \theta_2) q_j^{c_j} \left(1 - q_j\right)^{\gamma_j - c_j} dq dd \\ &\overset{def}{=} \prod_{j\in[M]} \psi_j\left(z_{\mathcal{N}_j}\right) \prod_{i\in[N]} \varphi_i\left(z_{\mathcal{N}_i}\right) \end{aligned} \tag{3}$$

of which $\psi_j(z_{\mathcal{N}_j})$ is the local factor of worker j, i.e., the summation of all tasks taken by worker j; $\varphi_i(z_{\mathcal{N}_i})$ is the local factor of task i, i.e., the summation of all workers take task i. Here, we also assume that ψ_j on θ_1 and L are independent, and φ_i on θ_2 and L are independent.

Further, we use Markov random field assumption to model $p(z \mid L, \theta)$ and treat the task assignment graph as a factor graph, of which variable nodes are task nodes, and factor nodes are worker nodes [30, 31]. We use $m_{i\to j}$ to denote the messages passing from tasks to workers and $m_{j\to i}$ to denote the messages passing from workers to tasks. Messages are initialized as independent Gaussian distribution or other suitable distribution. We then update them iteratively until convergence. To put it in a simple way, the messages are equivalent to the ground truth combined with all relevant factors. Therefore, $b_i(z_i)$ is the belief of the tasks, i.e., the intermediate estimated ground truth of the tasks. Therefore, the updating equations can be formulated as follows:

$$
\begin{aligned}
&\text{From tasks to workers}: && m_{i\to j}^{t+1}(z_i) \propto \sum_{z_{\mathcal{N}_i\backslash j}} \varphi_i\left(z_{\mathcal{N}_i}\right) \prod_{i'\in\mathcal{N}_{i\backslash j}} m_{j'\to i}^{t}(z_i),\\
&\text{From workers to tasks}: && m_{j\to i}^{t+1}(z_i) \propto \sum_{z_{\mathcal{N}_j\backslash i}} \psi_j\left(z_{\mathcal{N}_j}\right) \prod_{i'\in\mathcal{N}_{j\backslash i}} m_{i'\to j}^{t+1}(z_{i'}),\\
&\text{Calculating the beliefs}: && b_i^{t+1}(z_i) \propto \prod_{j\in\mathcal{M}_i} m_{j\to i}^{t+1}(z_i).
\end{aligned}
\tag{4}
$$

Note that tasks can have binary or multiple dimension labels, and in this case, $m_{i\to j}$ or $m_{j\to i}$ can instead be vectors.

3. System Model

In this section, we introduce our VTI and then form its derivations. We use label aggregation problem as an example to study crowdsourcing problem, and since others can be transformed to a label model easily, we use label aggregation to show VTI algorithm. Intuitively, we use log error rate to indicate the performance of our algorithm and the comparison criteria between existing inference algorithms. Further, the convergence rate and speed are also important criteria for inference problems.

In VTI, we add the task difficulty d to improve inference efficiency. At first, we can use d to help compute worker reliability q and then use both d and q to predict ground truth z. In general, there are three advantages when task requestors take into account of the task difficulty level. First, worker reliability parameter will be more valid if task difficulty is included to compute it. The reason is simple: more difficult tasks require workers with more expertise [32, 33]. Second, task difficulty is useful when assigning tasks and can reduce the task time and budget to some extent. Third, it is useful to consider task difficulty when compensating workers, which in turn can sustain the crowdsourcing system. Surely, it is important to consider cases where workers have no incentive to complete a task as soon as possible and may always submit completed tasks in the last minute. Some workers might even submit a random label in a short time. Therefore, a good difficulty level indicator should be able to deal with those situations.

First, we denote worker j's reliability as $q = \{q_j : j \in [M]\}$, task difficulty as $d = \{d_i : i \in [N]\}$, and the true answer of tasks $z = \{z_i : i \in [N]\}$. Therefore, the joint probability of worker reliability, task difficulty, ground truth, and conditioned on the answer matrix L and θ is as Equation (1).

3.1. Assumptions and ELBO Formation. In VTI, we use a three-layered graphical model to show the causal relationship of the parameters. In this model, all data points share global variables, while each local hidden variable belongs to each data point. As defined above, we denote $L = L_{1:N}$ as observed variables (label vector of the tasks), $z = z_{1:N}$ as local hidden values, and θ as global hidden values. The joint density of the model is

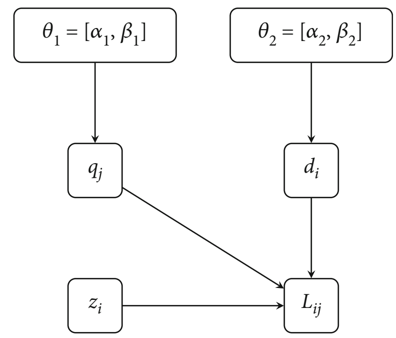

Figure 1: Probability graphical model of VTI.

$$
p(q, d, z, L) = p(q, d \mid \theta) \prod_{i=1}^{N} p(z_i, L_i \mid d, q) \tag{5}
$$

Figure 1 illustrates the graphical model of VTI.

Computing the posterior inference is the primary task of variational inference. Therefore, our main task is to estimate the ground truth of every task according to collected answers, which is $p(\theta, z \mid L)$, the conditional probability of hidden variables (ground truth), and global hidden variables (data distribution), given observations (collected answers from the crowdsourcing system). However, it is often the case that this conditional probability is difficult or computing-intensive to solve. So it is intuitive for us to resort to approximation methods.

VI is the method of using a family of distributions that is easy to compute to approximate the desired objective posterior distribution. The direct way is to minimize the KL divergence between the posterior distribution and the approximate distribution, which is also a function of the hyperparameters. Thus, the computing process is to solve the optimum of a functional, which is called variational analysis. In the domain of machine learning (ML), this is called the procedure of solving evidence lower bound (ELBO) in terms of a variational parameter.

$$
\mathscr{L}(\nu) = \mathbb{E}_q[\log p(q, d, z, L)] - \mathbb{E}_q[\log q(q, d, z \mid \nu)] \tag{6}
$$

of which $q = q(q, d, z \mid \nu)$ is the variational distribution we choose to approximate the true joint distribution $p(q, d, z \mid L)$. The detailed derivation of Equation (6) can be found in Appendix.

Assuming that the Bayesian network of our crowdsourcing inference model is fully factored, i.e., each worker and task are independent, and using the chain rule to extend the joint probability, we have

$$
q(q, d, z \mid \nu) = q(q \mid \lambda_1) \prod q(z_i \mid \phi_i) q(d \mid \lambda_2) \tag{7}
$$

of which ν is the general notation of the variational parameter, λ_1 is the global variational variable for q, λ_2 is the global variational variable for d, $\lambda = (\lambda_1, \lambda_2)$, and ϕ is local variational variable. VI optimizes Equation (2) using a gradient or coordinate ascent. To better find local optimum, we use variational inference with tempering.

As mentioned before, the procedure of tempering can promote the performance of the optimization process by curbing multioptimum. Therefore, we apply it in our VTI. Using $T \geq 1$ to denote the global temperature, the joint probability becomes

$$p(q, d, z, L \mid T) = \frac{p(z, L \mid q, d)^{1/T} p(q \mid \theta_1) p(d \mid \theta_2)}{c(T)} \tag{8}$$

of which

$$c(T) = \int p(z, L \mid q, d)^{1/T} p(q \mid \theta_1) p(d \mid \theta_2) dL dz \mathrm{d}q dd. \tag{9}$$

Applying Equation (8) in (6), we reach the tempered ELBO.

$$\begin{aligned}\mathscr{L}_A(\lambda, \phi; T) = {} & \mathbb{E}_q\left[\log p(q \mid \theta_1) + \mathbb{E}_q[\log p(d \mid \theta_2)]\right. \\ & - \mathbb{E}_q[\log q(q, d \mid \lambda)] \\ & + \sum_{i=1}^{N} \frac{\mathbb{E}_q[\log p(L_i, z_i \mid q, d)]}{T} \\ & - \mathbb{E}_q[\log p(z_i \mid \phi_i)].\end{aligned} \tag{10}$$

Note that the standard tempering method starts from high temperature and gradually reduces it, taking little account of the influence of data at first, reaching a high-entropy solution, and then slowly makes use of the data or evidence, and reaching a distribution more resembling the data.

Now, we discuss why tempering is effective. The 1st and 3rd terms of Equation (10) are the expected log prior and the log-likelihood. By maximizing these terms, it can result in hidden variables that can better explain the data. The 2nd and 4th terms combined are the variational distribution entropy. Since entropy is convex, it can act as a regularizer which distributes into a hidden variable's configuration. With smaller $1/T$, we obtain a small likelihood of ground truth and smooth entropy distribution. By decreasing T gradually, our variational distribution puts more and more weight on observed data. According to Mandt et al. [34], there are two types of tempering method, one method tempers both likelihood and prior, and the other method tempers only the likelihood. The latter has no problem of gradient stuck during iteration.

This paper discusses gradient-based tempering. As similar to a regular optimization problem, if the tempering is too slow, the algorithm will converge too fast and never reach the global optimum. If we temper too fast, on the other hand, we may skip the global optimum. Since it is difficult to find the proper schedule, we put our emphasis on learning data sequence adaptively. We use variational tempering, a method learning temperature schedule. In variational tempering, we introduce temperature as an auxiliary variable. Therefore, if the temperature is 1, it will degenerate to the original variational inference. We consider finite discrete temperature, for example, $1 = T_1 \leq T_2 \leq \cdots \leq T_M$, which allows us to compute partition function beforehand.

3.2. Objective Function. It is easy to derive that $p(L, z, q, d, y) = p(L, z, q, d \mid y) p(y)$. Given $p(y) = \prod_{m=1}^{M} \pi_m^{y_m}$, the tempered joint of task difficulty, worker reliability, ground truth and worker labels can be written as

$$p(L, z, q, d \mid y) = \frac{p(q, d)}{c(T_y)} \prod_{i=1}^{N} p(L_i, z_i \mid q, d)^{1/T_y} \tag{11}$$

of which T_y is a temperature related to y. Thus, the model becomes

$$p(L, z, q, d, y) = p(q, d) \prod_{m=1}^{M} \left(\frac{\pi_m}{c(T_m)} \prod_{i=1}^{N} p(L_i, z_i \mid q, d)^{1/T_m} \right)^{y_m}. \tag{12}$$

The objective of VTI is to maximize the above joint probability. By further using the chain rule of Bayesian theory, we obtain

$$p(z, q, d, y \mid \psi, \lambda, \gamma) = q(z \mid \psi) q(q, d \mid \lambda) q(y \mid \gamma) \tag{13}$$

of which r denotes the variational parameter for the temperature. Therefore, we obtain the tempered ELBO of VTI.

$$\begin{aligned}\mathscr{L}_T(\lambda, \phi; T) = {} & \mathbb{E}_q[\log p(q)] + \mathbb{E}_q[\log p(d)] + \mathbb{E}_q[\log p(y)] \\ & - \mathbb{E}_q[\log q(q)] - \mathbb{E}_q[\log q(d)] \\ & - \mathbb{E}_q\left[\frac{1}{T_y}\right] \sum_{i=1}^{N} \left(\mathbb{E}_q[\log p(L_i, z_i \mid q, d)]\right) \\ & - \mathbb{E}_q\left[\log C(T_y)\right] \\ & - \sum_{i=1}^{N} \mathbb{E}_q[\log q(z_i)] - \mathbb{E}_q[\log q(y)].\end{aligned} \tag{14}$$

Therefore, we can maximize this tempered ELBO, instead of optimizing the objective function directly, to obtain ground truth z and hidden parameters q, d.

3.3. Local Temperature. In real practice, often the data would arrive one at a time, or sometimes it is difficult to compute a huge amount of data point at once, so we use the incremental method, which computes each data point as soon as it arrives. This method can simplify the computation and show the influence of each data point to the whole network. Therefore, we can treat the potential or log value of each data point as the local factor. And instead of computing the global temperature at once, we can calculate their local temperature in a similar way. The algorithm is also more scalable to future data. Just like heat flows through the network with thermal conductivity, we can treat the potential or temperature of each local data that passes through the data network. Using

t_i to denote the local temperature, the joint probability of task labels, ground truth, hyperparameter, and temperature becomes

$$p(L, z, q, d, t) \propto p(q, d) \prod_{i=1}^{N} \left[p(L_i, z_i \mid q, d)^{1/t_i} p(t_i) \right]. \quad (15)$$

We can see from the above equation that adding the temperature parameter reduces the weight of the global hidden value and data, i.e., the weight of the ground truth and the task labels. In this way, the resulted distribution of the tasks will have higher entropy. Note that we no longer have to calculate the partition function since there is a $\propto$ in the equation. By adjusting the local temperature, we can obtain a more desired outcome. For instance, outliers can be best demonstrated by higher temperature, while lower temperature can give us the main situation of the data.

4. VTI Algorithms

In this section, we introduce the main part of VTI. We assume that the prior distributions, i.e., the worker prior and the task prior, and the distribution of parameters of the task prior and the worker prior are all from the exponential family [21], which has the following formulation.

$$p(q, d \mid \theta) = h(q, d) \exp \left\{ \theta^T t(q, d) - a_g(\theta) \right\},$$

$$p(z_i, L_i \mid q, d) = h(z_i, L_i) \exp \left\{ (q, d)^T t(z_i, L_i) - a_l(q, d) \right\}, \quad (16)$$

of which $t(q, d)$ and $t(z_i, L_i)$ are sufficient statistics of global and local data points, of which $a_g(\theta)$ and $a_l(q, d)$ are corresponding log normalizers.

Treating the objective function as a function of the global variational parameter, we obtain

$$\begin{aligned} \mathscr{L}_T(\lambda, \phi\,; T) &= \mathscr{L}_T(\lambda, \phi(\lambda, T)\,; T), \\ \phi(\lambda, T) &= \operatorname{argmax}_{\phi_{ij}} \mathscr{L}_T\left(\lambda, \phi_{ij}\,; T\right). \end{aligned} \quad (17)$$

According to Hoffman, the tempered ELBO is

$$\begin{aligned} \mathscr{L}_T(\lambda, \phi\,; T) = {} & \mathbb{E}_q \left[\frac{1}{T_y} \right] \mathbb{E}_q \left[\eta_g(L, z, \theta) \right]^T \nabla_\lambda a_g(\lambda) \\ & - \lambda^T \nabla_\lambda a_g(\lambda) + a_g(\lambda) \\ & - \mathbb{E}_q \left[\log C\left(T_y\right) \right] - \mathbb{E}_q [\log q(y)]. \end{aligned} \quad (18)$$

4.1. Take-Home Equation. According to Hoffman, the tempered ELBO's natural gradient with respect to the global variational parameter is

$$\begin{aligned} \nabla_\lambda \phi(\lambda, T) &= \mathbb{E}_q \left[\eta_g(L, z, \theta) \right] - \lambda, \\ \widehat{\lambda} &= \mathbb{E}_q \left[\eta_g(L, z, \theta) \right], \end{aligned} \quad (19)$$

of which $\eta_g(L, z, \theta) = (\alpha_1 + \sum_{i=1}^{N} t(z_i, L_i), \beta_1 + N)$.

$$\lambda_{t+1} = \lambda_t + \rho\left(\widehat{\lambda} - \lambda_t\right). \quad (20)$$

In the first step, we estimate $\widehat{\lambda}$ and then update it with λ_t with decreasing learning rate ρ_t. By dividing the expectation of the sufficient statistics, we can reduce the variance of the gradients.

We then optimize tempered ELBO. And the local variational update, i.e., the main body of VTI, is

$$\phi_{ij} = \frac{1}{T} \mathbb{E}_q \left[\eta_l\left(z_{i,-j}, L_i, q, d\right) \right], \quad (21)$$

of which, η_l is the local variational parameter and is also the natural parameter of the exponential family distribution $p(z_{ij} \mid L_i, z_{i,-j}, q, d)$. Therefore, we can compute η_l by constructing the exponential family distribution $p(z_{ij} \mid L_i, z_{i,-j}, q, d)$, since $L_i, z_{i,-j}$ is known, and q, d is initialized.

Therefore, VTI can be summarized in Algorithm 1.

```
Initialize λ_0, T, q_0, d_0, L, φ_ij, ρ_0
Output: z, q, d
for each label L_ij do
   Update local variational parameter:
   φ_ij = (1/T)E_q[η_l(z_{i,-j}, L_i, q, d)].
end for
Update global variational parameter:
λ̂ = E_q[η_g(L, z, θ)]
λ_{t+1} = λ_t + ρ(λ̂ − λ_t)
until convergence
Output: q_j = prob[L_ij = z_i], i ∈ 𝒩_j
   d_i = prob[L_ij = z_i], j ∈ ℳ_i
   z_i = sign [q_j L_ij], j ∈ ℳ_i
```

Algorithm 1. VTI.

4.2. Analysis. In this part, we analyze the performance guarantee of VTI. To allocate the tasks to different workers, we use a bipartite graph algorithm, of which the left degree of the graph l refers to how many workers are needed for one task, and the right degree of the graph r refers to how many tasks are allocated to one worker at one time period. Intuitively, the error rate e is dependent on l, r, q, and d. By the definition of q_j, we can see that the effective information provided by worker j can be measured by $g_j = |2q_j - 1\,|$; therefore, the expectation of effective information provided by one worker is $g_1 = \mathbb{E}[|2q_j - 1\,|]$. Let $\widehat{l} = l - 1$ and $\widehat{r} = r - 1$, u_1

$= \mathbb{E}[2q_j - 1]$, $u_2 = \mathbb{E}[2d_i - 1]$, $u = u_1/u_2$, $g_2 = \mathbb{E}[|2d_i - 1|]$, and $g = g_1/g_2$. Define $\rho_k^2 \equiv 2g/u^2(g^2 l\wedge r\wedge)^{k-1} + (3 + 1/g\widehat{lr})1 - (1/g^2 l\wedge r\wedge)^{k-1}/1 - (1/g^2\widehat{lr})$.

For $g^2\widehat{lr} > 1$, let $\rho_\infty^2 \equiv \lim_{k\to\infty}\rho_k^2$ such that

$$\rho_\infty^2 = \frac{\left(3 + \left(1/g\widehat{lr}\right)\right)g^2\widehat{lr}}{\left(g^2\widehat{lr} - 1\right)}. \tag{22}$$

We can then arrive at the following error bound.

Theorem 1. *Suppose n tasks are distributed to m workers, forming a l, r bipartite graph. If the worker reliability distribution and task difficulty distribution satisfy $u_1 > 0$, $u_2 > 0$, and $g > 1/(\widehat{lr})$, the k-th estimates of VTI can achieve*

$$\limsup_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} \mathbb{P}\left(z_i \neq \hat{z}_i\left(\left\{L_{ij}\right\}_{(i,j)\in E}\right)\right) \geq e^{-\lg/\left(2\sigma_k^2\right)}. \tag{23}$$

Theorem 2. *According to Theorem 1, we assume n tasks are distributed to m workers, forming a l, r bipartite graph. If the worker reliability distribution and task difficulty distribution satisfy $u_1 > 0$, $u_2 > 0$, and $g > 1/(\widehat{lr})$, the k-th estimates of VTI can achieve*

$$\limsup_{k\to\infty}\ \limsup_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} \mathbb{P}\left(z_i \neq \hat{z}_i\left(\left\{L_{ij}\right\}_{(i,j)\in E}\right)\right) \geq e^{-\lg/\left(2\rho_\infty^2\right)}. \tag{24}$$

The implication of this theorem is that the error rate is upper bounded by $e^{-\lg/(2\sigma_\infty^2)}$. Notice that the temperature parameter only affects the convergence and has no effect on error bound. We present the proof of Theorem 1 in Appendix.

4.3. Discussion. We developed a variational inference algorithm with tempering, which is scalable, and can analyze large amount of data by complex probabilistic graphical models. The main idea is to optimize the variational objective using stochastic optimization. By repeatedly sampling the data, it can estimate the natural gradient with noise. By introducing tempering into the objective function, we can simplify the computation complexity and improve convergence rate of the iteration. With VTI, we can easily apply label aggregation modeling to various kinds of pattern recognition and classification problem. Furthermore, VTI can be generalized to more settings as it opens the door for several research directions. Here, we analyze VTI's complexity and how to interpret its results.

First, VTI has the same runtime with majority voting, which is $O(ml)$.

Second, the assumption of conjugate priors is necessary. Conjugate prior plays an important role in variational inference. It enables the form of the posterior probability to resemble the prior probability, which simplifies Bayesian analysis to a large extent. Taking VTI for example, the worker and task prior are Gaussian, and their posteriors are also Gaussian. Using conjugate prior assumption, the inference model can be easily extended to different kinds of distribution that belong to the exponential family.

Third, the mean-field assumption is also necessary. By approximating the true parameters of the prior distribution with a distribution that is not restricted, we can reduce computation and solve complex probabilistic graphical models.

Fourth, our VTI can establish good performance under different distributions of q_j and d_i. Furthermore, VTI is robust to different initialization, i.e., almost every starting point can result in a unique solution. The reason for this robustness is that VTI computes the energy of the objective function and will surely decrease with each iteration.

5. Experiments and Results

In this section, we will assess seven advancing algorithms on real datasets for the experiments of our algorithm. First, the experimental setup is introduced, and then, the convergence performance of the collected crowdsourcing inference algorithm is analyzed. Finally, we make comparison of the error rate of VTI with the existing algorithms.

5.1. Experimental Setup. We use Amazon Mechanical Turk datasets (Data: https://github.com/musegoduci/variational-inference-for-crowdsourcing) to show the comparative results of the inference algorithms. The Amazon Mechanical Turk is a binary dataset, of which 0 means a worker identifies no target in a task, and 1 means the worker identifies target in a task. Note that the data need to be centralized before processing. We use bipartite assignment graph algorithm to generate the initial answer matrix for the simulation. Some notations are listed in Table 1. It should be noted that KOS (Karger, Oh, and Shah's iterative learning algorithm for crowdsourcing inference [29]) is a message passing-based algorithm, and we first identify the reliability of each worker by some data sample and then implement MLE to estimate the test sample's ground truth. However, BP, VMP, NMP (Liu et al.'s novel crowdsourcing inference method [22]), and VTI assume a probability distribution of worker prior and task prior and then solve the hyperparameters of the distribution, leading to a MLE estimation of the ground truth.

5.2. Convergence. In this part, we analyze the convergence behavior of VTI. Note that the overall executing time for an inference algorithm depends on how long it takes to iterate once, as well as on how many times it will iterate. Since the normalizing constants can be precomputed in advance, the executing time of one iterate depends only on the calculation of the natural gradient, which gives faster convergence than others. Therefore, VTI can converge in seconds. Furthermore, we use subsampling to obtain the natural gradient of the data, which is computationally economical in each iteration. Therefore, VTI possesses advantageous computational performance.

Table 1: Notations and settings.

Parameter	Commentate	Value
l	Degree of tasks	5
γ	Degree of workers	5
N	Number of tasks	1000
M	Number of workers	1000
L_{ij}	Worker j's label toward task i	
q_j	Worker j's reliability	
d_i	Task i's difficulty	
z_i	Task i's ground truth	
θ_1	Hyperparameter of worker reliability's distribution	
θ_2	Hyperparameter of task difficulty's distribution	
$[\alpha_1, \beta_1]$	Worker prior	[0.5,1]
$[\alpha_2, \beta_2]$	Task prior	[0.4,0.6]
ε	Convergence tolerance	10^{-6}

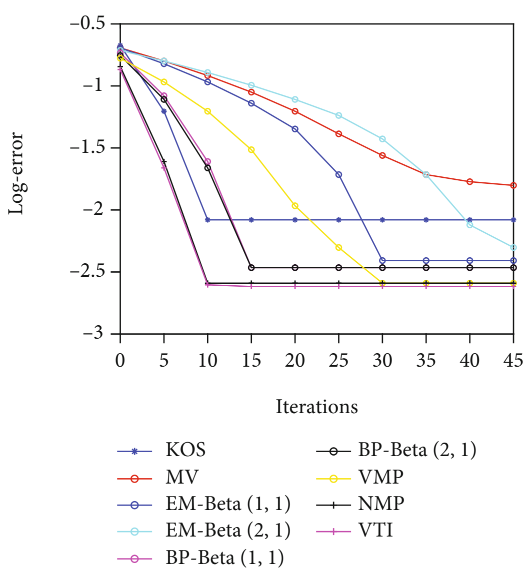

Figure 2: Convergence rate comparison.

From Figure 2, we can show that our VTI has a faster converge rate than KOS, MV, EM, and VMP. It is straight forward to know that this phenomenon is inherent. The reason is that VTI computes both the task difficulty and worker reliability simultaneously and reduces error in both directions. Furthermore, how fast KOS can converge depends on whether and how the ground truth is centered. If the degree of the bipartite graph is low, BP may not even converge. VMP and NMP all have rather good convergence performance because they, like VTI, are both based on the mean-field variational method.

5.3. Error Rate. In this research, we make comparison of VTI with MV, EM, KOS, VMP, and NMP.

In Figure 3, we simulate the performance of VTI with fixed right degree. Intuitively, the log error rate is positively correlated to the left degree, i.e., the quantity of tasks each worker takes. From the figure, we find that VTI has better performance under these settings.

In Figure 4, we simulate the performance of VTI with fixed left degree. Intuitively, the log error rate is positively correlated to the right degree, i.e., the quantity of workers each task takes. From the figure, we find that VTI has better performance under these settings. In Figure 5, we simulate the performance of VTI with fixed left and right degrees. Intuitively, the log error rate is positively correlated to both the degrees. From the figure, we find that VTI has better performance under these settings.

To sum up, it can be seen from Figures 3–5 that our VTI can achieve a slightly higher error rate, which proves that the inference algorithm often performs better when considering both the reliability of workers and task difficulty. And this algorithm can also ensure finding a local optimum.

6. Conclusion

In this paper, we solve the problem of cyber physical system crowdsourcing inference using variational inference with tempering. Our VTI considers not only the worker reliability, but also the task difficulty, which makes VTI adaptable to more complex probabilistic graphical model. With the tempering procedure, the iteration process can reach a smoother local optimum and better demonstrate the relationship of balancing the influence of single data point and global variational parameter. The results are promising in reality and insightful in the field of statistical inference. Therefore, the paper contributes to the research of finding efficient

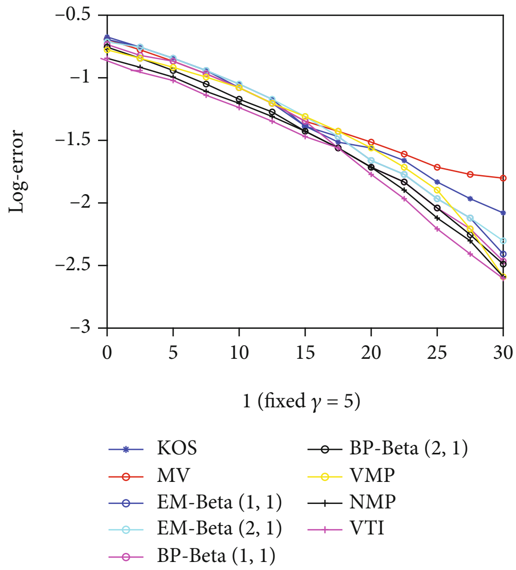

Figure 3: Comparison with fixed number of tasks per worker.

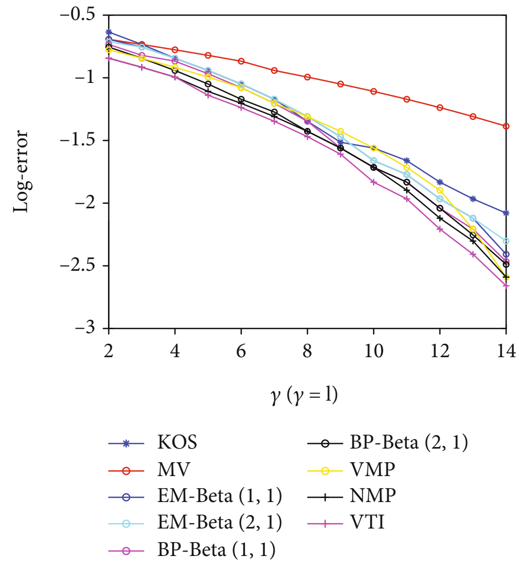

Figure 5: Comparison with the same worker and task degree.

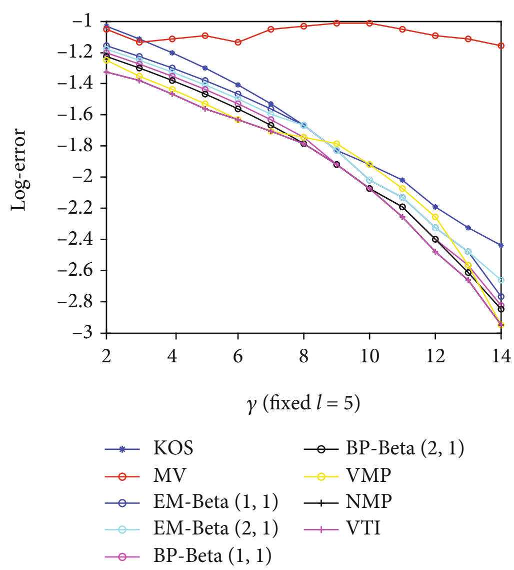

Figure 4: Comparison with fixed number of workers per task.

algorithms in CPSs. Promising directions in future include formulating the inference problem in a more general background and exploring the connection of variational inference with all other inference and estimation algorithms.

7. Limitations of VTI

VTI suffers from a few limitations just like other variational inference algorithms. First, VTI relies on the assumption of the worker prior and difficulty level prior. It is applicable only to CCEF model. Second, VTI assumes that each worker has the same reliability towards different kinds of questions. Although it is capable of dealing with cases when different tasks are categorized into different classes, VTI can become extremely complex doing so. Third, VTI has the same cold start issue with other variational inference algorithms. In this paper, we initialize worker reliability and task difficulty uniformly and compute it batch by batch. However, it is also difficult to achieve a computable bipartite graph if in practice, the answers of different tasks arrive at different rate.

Appendix

A. Derivation of Equation (6)

In this part, we show how to obtain Equation (6). Jensen's inequality is used in the 3rd step.

$$\begin{aligned}\log p(L) &= \log \int p(q,d,z,L)\,dzdqdd \\ &= \log \int p(q,d,z,L)\frac{q(q,d,z\mid v)}{q(q,d,z\mid v)}\,dzdqdd \\ &= \log\left(\mathbb{E}_q\left[\frac{p(q,d,z,L)}{q(q,d,z\mid v)}\right]\right) \\ &\ge \mathbb{E}_q[\log p(q,d,z,L)] - \mathbb{E}_q[\log q(q,d,z\mid v)] \triangleq \mathscr{L}(v).\end{aligned} \tag{A.1}$$

B. Proof of Theorem 1

Proof. Because of the local nature of crowdsourcing, we can treat it as a repetition coding and analyze its bit error bound in a similar way. In this case, we repeat each bit r times. By symmetry, we can assume all t_i's are +1. For any of the m tasks, $(1/m)\sum_{i\in[m]}\mathbb{P}(t_i \ne \hat{t}_i) \le \mathbb{P}(z_i^k \le 0)$, where z_i^k is the estimated answer after k-th iteration. Modeling the probabilistic graphical model of VTI as a bipartite graph and each node with its neighbor can be treated as a regular tree. Intuitively, we have the following equality:

$$\lim_{m\to\infty} \mathbb{P}\left(z_i^k \leq 0\right) = \mathbb{P}\left(z\wedge^k \leq 0\right). \tag{B.1}$$

In order to analyze the performance guarantee of VTI, we can borrow the idea in Chernoff bound [35]. It follows that $P(x\wedge^k \geq 0) \leq \mathbb{E}[e^{\lambda x\wedge^k}]$; therefore, $1 - P(x\wedge^k \geq 0) \geq \mathbb{E}[e^{\lambda x\wedge^k}]$, i.e., $P(x\wedge^k \leq 0) \geq \mathbb{E}[e^{\lambda x\wedge^k}]$. Define $\sigma_k^2 \equiv 2(l\wedge r\wedge)^k + u^2 l\wedge^3 \hat{r}(3g\hat{r} + 1)((g^2 l\wedge r\wedge)^{2k-4})(1 - (1/g^2 l\wedge r\wedge)^{k-1})/(1 - (1/g^2 \hat{l}\hat{r}))$, and $\mu_k \equiv u\hat{l}(g^2 l\wedge r\wedge)^{k-1}$, and variable z with mean μ and variance σ will be a sub-Gaussian distribution, and $\mathbb{E}[e^\lambda z] \leq e^{\mu\lambda + (1/2)\sigma^2\lambda^2}$. Therefore,

$$P\left(x\wedge^k \leq 0\right) \leq \mathbb{E}\left[e^\lambda x\wedge^k\right] \leq e^{-l\mu_k^2/\left(2l\wedge\sigma_k^2\right)}. \tag{B.2}$$

This, finishes the proof.

Data Availability

The data is available at https://github.com/musegoduci/variational-inference-for-crowdsourcing.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61802097 and 61877007), the Fundamental Research Funds for the Central Universities (Grant No. DUT20GJ205), the Project of Qianjiang Talent (Grant No. QJD1802020), and the Key Research & Development Plan of Zhejiang Province (Grant No. 2019C01012).

References

[1] A. Moayedikia, W. Yeoh, K. L. Ong, and Y. L. Boo, "Framework and literature analysis for crowdsourcing's answer aggregation," *Journal of Computer Information Systems*, vol. 60, no. 1, pp. 49–60, 2020.

[2] A. J. Porter, P. Tuertscher, and M. Huysman, "Saving our oceans: scaling the impact of robust action through crowdsourcing," *Journal of Management Studies*, vol. 57, no. 2, pp. 246–286, 2020.

[3] X. Han, G. Shen, X. Yang, and X. Kong, "Congestion recognition for hybrid urban road systems via digraph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 121, p. 102877, 2020.

[4] Y. J. Hsu, "Crowdsourcing agents for smart IoT," in *HAI '14: Proceedings of the second international conference on Human-agent interaction*, Tsukuba Japan, 2014.

[5] W. Wang, J. Liu, Z. Yang, X. Kong, and F. Xia, "Sustainable collaborator recommendation based on conference closure," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 311–322, 2019.

[6] W. Wang, J. Ren, M. Alrashoud, F. Xia, M. Mao, and A. Tolba, "Early-stage reciprocity in sustainable scientific collaboration," *Journal of Informetrics*, vol. 14, no. 3, p. 101041, 2020.

[7] W. Wang, X. Zhao, Z. Gong, Z. Chen, and W. Wei, "An attention-based deep learning framework for trip destination prediction of sharing bike," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[8] K. Xiangjie, T. Shiqin, G. Haoran et al., "Mobile edge cooperation optimization for wearable internet of things: a network representation-based framework," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2020.

[9] M. Li and P. Li, "Crowdsourcing in cyber-physical systems: stochastic optimization with strong stability," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 218–231, 2013.

[10] A. Nieto, A. Acien, G. Fernandez, Q. He, C. Xiang, and G. Wei, "Crowdsourcing analysis in 5g IoT: cybersecurity threats and mitigation," *Science of the Total Environment*, vol. 24, no. 3, pp. 881–889, 2019.

[11] M. Strohmeier, M. Smith, M. Schfer, V. Lenders, and I. Martinovic, "Crowdsourcing security for wireless air traffic communications," in *2017 9th International Conference on Cyber Conflict (CyCon)*, Tallinn, Estonia, 2017.

[12] B. M. Booth, K. Mundnich, and S. Narayanan, "Fusing annotations with majority vote triplet embeddings," in *AVEC'18: Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, pp. 83–89, Seoul Republic of Korea, 2018.

[13] Y. Chen, M. Guizani, Y. Zhang et al., "When traffic flow prediction and wireless big data analytics meet," *IEEE Network*, vol. 33, no. 3, pp. 161–167, 2019.

[14] P. Germain, A. Lacasse, F. Laviolette, M. Marchand, and J. F. Roy, "Risk bounds for the majority vote: from a PAC-Bayesian analysis to a learning algorithm," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 787–860, 2015.

[15] A. Hotaling and J. P. Bagrow, "Accurate inference of crowdsourcing properties when using efficient allocation strategies," 2019, https://arxiv.org/abs/1903.03104.

[16] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?," *Proceedings of the VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[17] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, pp. 355–368, Springer, 1998.

[18] X. Wang, C. Li, J. Zhang, Q. Zhang, and J. Hu, "Variational inference-based em for quantized fir system parameter identification," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pp. 636–640, Anchorage, AK, USA, 2018.

[19] S. Hao and X. Yang, "A globally convergent mean-field inference method in dense Markov random fields," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Wuhan, China, 2017.

[20] Q. Liu, J. Peng, and A. T. Ihler, "Variational inference for crowdsourcing," *Advances in neural information processing systems*, vol. 1, pp. 692–700, 2012.

[21] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[22] J. Liu, W. C. Tang, Y. Chen, M. Li, and M. Guizani, "A novel crowd-sourcing inference method," in *2019 15th International*

Wireless Communications and Mobile Computing Conference (IWCMC), Tangier, Morocco, Morocco, 2019.

[23] D. Cai, D. T. Nguyen, L. S. Hong, and L. Wynter, "Variational Bayesian inference for crowdsourcing predictions," 2020, https://arxiv.org/abs/2006.00778.

[24] A. Buchholz, F. Wenzel, and S. Mandt, "Quasi-monte carlo variational inference," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholmsmässan, Stockholm Sweden, 2018.

[25] M. Yusoff, M. N. S. B. M. Ikram, and N. Janom, "Task assignment optimization for crowdsourcing using genetic algorithm," *Advanced Science Letters*, vol. 24, no. 11, pp. 8205–8208, 2018.

[26] A. Moayedikia, W. Yeoh, K. L. Ong, and Y. L. Boo, "Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach," *Decision Support Systems*, vol. 122, article 113065, 2019.

[27] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3068–3078, 2018.

[28] Y. Xing, L. Wang, Z. Li, and Y. Zhan, "Multi-attribute crowdsourcing task assignment with stability and satisfactory," *IEEE Access*, vol. 7, pp. 133351–133361, 2019.

[29] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," *Advances in Neural Information Processing Systems*, pp. 1953–1961, 2011.

[30] L. E. Ortiz and Z. Gong, "Correlated equilibria for approximate variational inference in MRFs," 2016, https://arxiv.org/abs/1604.02737.

[31] Y. Yang, X. Cong, K. Long, Y. Luo, W. Xie, and Q. Wan, "MRF model-based joint interrupted SAR imaging and coherent change detection via variational Bayesian inference," *Signal Processing*, vol. 151, pp. 144–154, 2018.

[32] M. Venanzi, J. Guiver, P. Kohli, and N. R. Jennings, "Time-sensitive Bayesian information aggregation for crowdsourcing systems," *Journal of Artificial Intelligence Research*, vol. 56, pp. 517–545, 2016.

[33] Y. Yang, Q. Bai, and Q. Liu, "Modeling random guessing and task difficulty for truth inference in crowdsourcing," in *AAMAS '19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2288–2290, Montreal, Canada, 2019.

[34] S. Mandt, J. Mcinerney, F. Abrol, R. Ranganath, and D. Blei, "Variational tempering," *Computer Science*, 2014.

[35] D. Arjo, "Statistical models: theory and practice," *Technometrics*, vol. 48, no. 2, pp. 315–315, 2009.