Bayesian Decision Process for Budget-efficient Crowdsourced Clustering

Xiaozhou Wang[1] , **Xi Chen**[2] , **Qihang Lin**[3] and **Weidong Liu**[1,4]
[1]School of Mathematical Sciences, Shanghai Jiao Tong University, China
[2]Stern School of Business, New York University, USA
[3]Tippie College of Business, University of Iowa, USA
[4]MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China
wangxiaozhou@sjtu.edu.cn, xchen3@stern.nyu.edu, qihang-lin@uiowa.edu, weidongl@sjtu.edu.cn

Abstract

The performance of clustering depends on an appropriately defined similarity between two items. When the similarity is measured based on human perception, human workers are often employed to estimate a similarity score between items in order to support clustering, leading to a procedure called crowdsourced clustering. Assuming a monetary reward is paid to a worker for each similarity score and assuming the similarities between pairs and workers' reliability have a large diversity, when the budget is limited, it is critical to wisely assign pairs of items to different workers to optimize the clustering result. We model this budget allocation problem as a Markov decision process where item pairs are dynamically assigned to workers based on the historical similarity scores they provided. We propose an optimistic knowledge gradient policy where the assignment of items in each stage is based on the minimum-weight K-cut defined on a similarity graph. We provide simulation studies and real data analysis to demonstrate the performance of the proposed method.

1 Introduction

Clustering is one of the most important tasks in unsupervised machine learning with a wide range of applications. The goal of clustering is to group similar items together based on an appropriately chosen similarity or dissimilarity (distance) measure between pairs of items. Popular clustering methods include connectivity-based clustering [Nocetti *et al.*, 2003], centroid-based clustering [Zhong, 2005] and distribution-based clustering [Xu *et al.*, 1998]. When each item can be represented by a feature vector, a common practice is to define a similarity/dissimilarity measure (e.g., cosine similarity or Euclidean distance) between pairs of items based on their feature vectors. However, there exist many scenarios where such feature vectors either are difficult to construct or do not well reflect the similarity between items, especially when the similarity needs to be consistent with human's perception, e.g., the similarity between paintings, songs, and videos. In order to facilitate clustering tasks under these situations, human workers are often employed to exam pairs of items and manually determine whether each pair is similar or not. The clustering procedure based on human-provided similarity judgment is called *crowdsourced clustering*.

Crowdsourcing is an effective approach to integrate human's intelligence by decomposing a main task (e.g. clustering) into many micro tasks (e.g. measuring similarity between pairs) which can be completed by online workers in parallel. Crowdsourcing has been successfully utilized in various tasks, such as clustering [Mazumdar and Saha, 2016; Luo *et al.*, 2018], classification [Tran-Thanh *et al.*, 2013; Bragg *et al.*, 2013], ranking [Chen *et al.*, 2013] and entity resolution [Whang *et al.*, 2013; Vesdapunt *et al.*, 2014].

In spite of its popularity, the information collected from crowdsourcing can be very noisy because of the different backgrounds of the workers, especially when each micro task requires a worker's subjective decision, e.g., when a worker is asked to judge whether two pictures are similar or not. Moreover, information provided by careless or unreliable workers will also have a low quality. Hence, a common strategy in crowdsourcing is to assign the same micro task to multiple workers, hoping that the majority of workers are reliable and different opinions can be integrated into a correct final conclusion, e.g. by majority vote (see, e.g., [Tao *et al.*, 2019; Zhang and Wu, 2018]). However, this strategy significantly increases the cost of crowdsourcing since a worker will receive a monetary reward for each completed micro task. When the owner of the tasks has only a small amount of budget, it is important to dynamically allocate the budget over micro tasks and workers such that the budget will shift towards more challenging micro tasks and more reliable workers, maximizing the quality of the final output before the budget runs out. This problem is called *the budget allocation problem in crowdsourcing*. A few budget allocation strategies have been developed for classification and ranking in crowdsourcing [Karger *et al.*, 2011; Chen *et al.*, 2015; Chen *et al.*, 2016].

In the setting of the crowdsourced clustering problem, a micro task requires a worker to assign a binary label to each pair of items, indicating if they are similar or not. In this case, the two key factors affecting budget allocation are (1) the similarity between each pair of items and (2) the reliability of each worker. For a pair of highly similar/dissimilar items, the labels provided by workers are consistent and the true similarity can be easily estimated with a little budget.

On the contrary, when the similarity is ambiguous, the labels become inconsistent and thus more workers' opinions are needed. However, these two factors are unknown initially and can only be estimated after collecting some labels from workers. This motivates us to develop a multi-stage budget allocation strategy where these two factors are dynamically estimated based on the labels collected in earlier stages and used to guide the budget allocation in the remaining stages.

To do so, we model the similarity as a score between zero and one, and view the clustering problem as the minimum K-cut problem on a graph where each node is an item and each edge is weighted by the similarity between the nodes [Goldschmidt and Hochbaum, 1988]. We first consider the setting where all workers are fully reliable. In this case, we introduce a prior distribution for the similarity between each pair of items. Then we formulate the crowdsourced clustering problem into a finite-horizon Bayesian Markov decision process (MDP) [Puterman, 2014] where, in each stage, the state variables are posterior distributions of similarity scores given the collected labels, and the action is to select an edge (a pair of items) and send it to a random worker for labeling. Because solving a MDP is computationally challenging due to the curse of dimensionality, effective approximate policies, such as knowledge gradient (KG) [Gupta and Miescke, 1996; Frazier *et al.*, 2008] and optimistic knowledge gradient (Opt-KG) [Chen *et al.*, 2015], have been proposed. Our policy is related to the Opt-KG policy where the pair is selected based on the possible reduction of the minimum K-cut value when the label for the selected pair is returned from workers. After that, we further extend the model and the policy to the case where the workers are unreliable and have different reliability. In the latter case, the action in each stage is to select not only the pair but also the worker to label this pair.

The rest of the paper is organized as follows. In Section 2, we first introduce the problem setup and then provide the Bayesian decision process with reliable workers. In Section 3, we extend the proposed method to the case of unreliable workers. Section 4 provides the simulation studies with both reliable and unreliable workers. In Section 5, we present numerical results on real data. Conclusions and further works are given in Section 6.

2 Bayesian Decision Process with Reliable Workers

We first investigate the problem under the setting with only fully responsible workers. Here, being fully reliable means that the workers judge the similarity between items to the best of their knowledge. However, due to the subjective nature of human perception-based similarity, the labels provided by workers are not necessarily consistent (see Eq. (1) below for a more precise description on our modeling of fully reliable workers). We will extend our study to the setting with unreliable workers (see Section 3).

Suppose there are N items that need to be grouped into K clusters based on their pairwise similarity. We model the true similarity between items i and j ($1 \le i < j \le N$) by a latent similarity parameter $\theta_{ij} \in [0, 1]$ and a larger θ_{ij} represents a higher similarity between them. Assuming that the budget for clustering is T, one unit of budget is paid to a worker for each similarity label he or she provides. We model the crowdsourced clustering problem as a multi-stage decision making problem with T stages. In each stage, the decision maker chooses a pair of items (i, j) and assigns them to a worker randomly chosen from the crowd to request a similarity label. We denote by l_{ij} the binary similarity label provided by the worker with $l_{ij} = 1$ if the worker thinks i and j are similar and with $l_{ij} = 0$ otherwise. Furthermore, we assume that the number of workers is large enough and θ_{ij} equals the proportion of the workers who think items i and j are similar. In other words, the label l_{ij} returned by a random reliable worker has the following Bernoulli distribution

$$\Pr(l_{ij} = 1|\theta_{ij}) = \theta_{ij}, \ \Pr(l_{ij} = 0|\theta_{ij}) = 1 - \theta_{ij}. \quad (1)$$

We assume that θ_{ij} follows a Beta prior Beta(a^0_{ij}, b^0_{ij}) for $1 \le i < j \le N$. Suppose, at stage t, Beta(a^t_{ij},b^t_{ij}) is the posterior distribution for θ_{ij} and a pair (i_t, j_t) is assigned to a random worker who returns a similarity label $l_{i_t j_t}$. According to (1), the posterior distribution of θ_{ij} will still be a Beta distribution $\mathrm{Beta}(a^{t+1}_{ij}, b^{t+1}_{ij})$ where for $1 \le i < j \le N$,

$$(a^{t+1}_{ij}, b^{t+1}_{ij}) = \begin{cases} (a^t_{ij} + 1, b^t_{ij}) & \text{if } (i,j) = (i_t, j_t), l_{i_t j_t} = 1; \\ (a^t_{ij}, b^t_{ij} + 1) & \text{if } (i,j) = (i_t, j_t), l_{i_t j_t} = 0; \\ (a^t_{ij}, b^t_{ij}) & \text{if } (i,j) \neq (i_t, j_t). \end{cases} \quad (2)$$

Next, we will model this crowdsourced labeling process as a Markov decision process (MDP). To do so, we define the state variable at stage t by $S_t = \{(a^t_{ij}, b^t_{ij})\}_{1 \le i < j \le N}$. Note that the conditional distribution of $l_{i_t j_t}$ conditioning on S_t is

$$\begin{aligned} \Pr(l_{i_t j_t} = 1|S_t) &= \mathbb{E}(\theta_{i_t j_t}|S_t) = \frac{a^t_{i_t j_t}}{a^t_{i_t j_t} + b^t_{i_t j_t}}, \\ \Pr(l_{i_t j_t} = 0|S_t) &= \mathbb{E}(1 - \theta_{i_t j_t}|S_t) = \frac{b^t_{i_t j_t}}{a^t_{i_t j_t} + b^t_{i_t j_t}}. \end{aligned} \quad (3)$$

After $l_{i_t j_t}$ is revealed, S_t is updated to S_{t+1} and S_{t+1} is different from S_t only at $(a_{i_t j_t}, b_{i_t j_t})$ according to (2). Combined with (2), (3) can be viewed as the state transition probability at state S_t after a decision (i_t, j_t). This also means that $\{S_t\}_{t \ge 0}$ is Markovian because S_{t+1} is fully determined by S_t and $l_{i_t j_t}$. Given this property, it suffices to consider a policy that only depends on S_t to choose (i_t, j_t). We denote such a policy by a sequence of mappings $(\pi_1, ..., \pi_T)$ with $\pi_t(S_t) \in \{(i,j)|1 \le i < j \le N\}$.

Given the limited budget, a good policy must choose (i_t, j_t) to collect sufficient information and quickly improve the quality of clustering. To measure the quality of clustering, we construct a *similarity graph*, which is a weighted, complete, undirected graph $G = (V, E, \{\theta_{ij}\})$ where the set of nodes V is the set of items for clustering, the set of edges E consists of all pairs of items, and the edge between i and j is weighted by θ_{ij}. Given this graph, a classical approach for clustering the nodes is to partition the graph into K disconnected components by removing the edges with as small total weight as possible. This is known as the minimum-weight K-cut problem:

$$\text{MinCut}(\{\theta_{ij}\}) \triangleq \min_{\{C_1, ..., C_K\}} \sum_{p=1}^{K-1} \sum_{q=p+1}^{K} \sum_{\substack{i \in C_p \\ j \in C_q}} \theta_{ij}, \quad (4)$$

where $\{C_1, C_2, ..., C_K\}$ forms a partition of V and $\theta_{ij} = \theta_{ji}$ if $j < i$. Since $\{\theta_{ij}\}$ is unknown, the clustering at stage t needs to be performed based on $\mathbb{E}(\theta_{ij}|S_t)$ by solving MinCut($\{\mathbb{E}(\theta_{ij}|S_t)\}$). We then use MinCut($\{\mathbb{E}(\theta_{ij}|S_t)\}$) to measure the quality of clustering at stage t with a smaller value representing better clustering. To obtain a good clustering result after T stages, the budget allocation problem is then formulated as the following dynamic optimization

$$\min_{\pi_1,...,\pi_T} \mathbb{E}[\text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_T)\})]. \quad (5)$$

Problem (5) can be equivalently reformulated as

$$\max_{\pi_1,...,\pi_T} -\text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_0)\}) + \sum_{t=0}^{T-1} \mathbb{E}[R(S_t, i_t, j_t, l_{i_t j_t})] \quad (6)$$

where $R(S_t, i_t, j_t, l_{i_t j_t})$

$$\triangleq \text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_t)\}) - \text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_{t+1})\}) \quad (7)$$

represents the expected reduction of the total weight of the minimum K-cut after updating the posterior distribution of $\theta_{i_t j_t}$ according to $l_{i_t j_t}$. Note that S_{t+1} depends on $l_{i_t j_t}$. Eq. (7) can be also interpreted as the expected improvement of the quality of clustering. Hence, (6) is an MDP where the stage-wise reward is $R(S_t, i_t, j_t, l_{i_t j_t})$.

However, (6) is difficult to solve optimally due to the curse of dimensionality. Therefore, we aim at a computationally efficient policy with a good performance in practice. A popular class of online learning policy is the *knowledge gradient* (KG) [Gupta and Miescke, 1996; Frazier *et al.*, 2008], which chooses (i_t, j_t) to maximize $\mathbb{E}[R(S_t, i_t, j_t, l_{i_t j_t})|S_t]$, or equivalently, to minimize $\mathbb{E}[\text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_{t+1})\})|S_t]$.

According to (2) and (3), we have $\{\mathbb{E}(\theta_{ij}|S_{t+1})\} = U^t(i_t, j_t, l_{i_t j_t})$ where $U^t(i_t, j_t, l_{i_t j_t}) \triangleq \{u_{ij}^t\}$ such that

$$u_{ij}^t \triangleq \begin{cases} (a_{ij}^t + 1)/(a_{ij}^t + b_{ij}^t + 1) & \text{if } (i,j) = (i_t, j_t); \\ a_{ij}^t/(a_{ij}^t + b_{ij}^t) & \text{otherwise,} \end{cases}$$

when $l_{i_t j_t} = 1$ and

$$u_{ij}^t \triangleq \begin{cases} a_{ij}^t/(a_{ij}^t + b_{ij}^t + 1) & \text{if } (i,j) = (i_t, j_t); \\ a_{ij}^t/(a_{ij}^t + b_{ij}^t) & \text{otherwise,} \end{cases}$$

when $l_{i_t j_t} = 0$. According to (3), the KG policy can be described as

$$\begin{aligned} &(i_t, j_t) = \arg\max_{i<j} \mathbb{E}[R(S_t, i, j, l_{ij})|S_t] \\ &= \arg\min_{i<j} \mathbb{E}[\text{MinCut}(\{\mathbb{E}(\theta_{ij}|S_{t+1})\})|S_t] \\ &= \arg\min_{i<j} \left(\frac{a_{ij}^t \text{MinCut}(U^t(i,j,1))}{a_{ij}^t + b_{ij}^t} + \frac{b_{ij}^t \text{MinCut}(U^t(i,j,0))}{a_{ij}^t + b_{ij}^t} \right). \end{aligned}$$

However, as shown in [Chen *et al.*, 2015] for the crowdsourced classification problem, the KG policy is myopic and may keep labeling a few pairs without exploring others. To address this issue, [Chen *et al.*, 2015] proposed the *optimistic knowledge gradient* (Opt-KG) policy which chooses (i_t, j_t) to maximize the optimistic outcome of stage-wise reward rather than the expected reward. It is shown by [Chen *et al.*, 2015] that the Opt-KG policy outperformed the KG policy because it balances exploration and exploitation better.

Algorithm 1 The Opt-KG policy for crowdsourced clustering with reliable workers

Input: Parameters of prior Beta distributions $S_0 = \{(a_{ij}^0, b_{ij}^0)\}_{1 \le i < j \le N}$, the budget T, and the number of clusters K.

1: **for** $t = 1, ..., T$ **do**
2: Compute $R_t^+(i,j)$ for $1 \le i < j \le N$ based on (7) and (8).
3: Select the pair of items (i_t, j_t) according to:

$$\text{Opt-KG: } (i_t, j_t) = \arg\max_{i<j} R_t^+(i,j),$$

and send it to a random worker.
4: Acquire the similarity label $l_{i_t j_t} \in \{1, 0\}$.
5: Update S_t to S_{t+1} according to (2).
6: **end for**

Output: The K clusters $\{C_1, ..., C_K\}$ that solve MinCut($\{\mathbb{E}(\theta_{ij}|S_T)\}$) defined by (4) and (3).

Motivated by [Chen *et al.*, 2015], we define the optimistic reward as

$$R_t^+(i,j) \triangleq \max_{l=0,1} R(S_t, i, j, l) \quad (8)$$

and propose the Opt-KG policy for crowdsourced clustering which chooses

$$(i_t, j_t) = \arg\max_{i<j} R_t^+(i,j) = \arg\min_{i<j} \Big(\min_{l=0,1} \text{MinCut}(U^t(i,j,l)) \Big),$$

where the second equality is from (7) and the fact that MinCut($\{\mathbb{E}(\theta_{ij}|S_t)\}$) does not depend on $l_{i_t j_t}$. We formally state the Opt-KG policy in Algorithm 1.

In practice, we may make the Opt-KG policy more time-efficient by utilizing the capability of parallel processing of crowdsourcing. Indeed, in each stage, we can choose the B pairs with the largest $R_t^+(i,j)$ and send them to different random workers simultaneously. After receiving the similarity labels from them all, we update the parameters S_{t+1} for the posterior distribution of each θ_{ij} in a way similar to (2).

3 Bayesian Decision Process with Unreliable Workers

The method proposed in Section 2 requires that all workers are fully reliable. However, unreliable workers do exist in practice who provide noisy labels because of, e.g., not comparing items carefully. It is critical to accurately assess the reliability of each worker in an early stage so that items and budget can flow to more reliable workers. In this section, we show that this can be done by extending our Opt-KG policy.

We assume that there are W heterogeneous workers and introduce a parameter $\rho_w \in [0,1]$ to represent the w-th worker's reliability. We define ρ_w as the probability of worker w labeling a pair of items in the same way as a randomly selected fully reliable worker (i.e. labeling to the best of the worker's knowledge). Specifically, if we assign a pair (i,j) to worker w, the label returned, denoted by l_{ijw}, has the distribution:

$$\begin{aligned} &\Pr(l_{ijw} = 1|\theta_{ij}, \rho_w) = \rho_w \theta_{ij} + (1 - \rho_w)(1 - \theta_{ij}), \\ &\Pr(l_{ijw} = 0|\theta_{ij}, \rho_w) = \rho_w (1 - \theta_{ij}) + (1 - \rho_w)\theta_{ij}. \end{aligned} \quad (9)$$

Here, ρ_w can be also interpreted as the probability that worker w provides the label by flipping the label from a reliable worker. When worker w is fully reliable, we have $\rho_w = 1$ and (9) reduces to (1), namely, the randomness of labels is only because of the subjective judgment of worker. Similarly, $\rho_w = 0.5$ means worker w labels the pair randomly.

Similar to θ_{ij}, we assume that each ρ_w is independently drawn from a Beta prior distribution Beta(c_w^0, d_w^0). Then, given the similarity label $l_{ijw} = l \in \{0, 1\}$, the posterior distribution of θ_{ij} and ρ_w is

$$\begin{aligned} &p(\theta_{ij}, \rho_w | a_{ij}^0, b_{ij}^0, c_w^0, d_w^0, l_{ijw} = l) \qquad (10)\\ &\sim \Pr(l_{ijw} = l | \theta_{ij}, \rho_w) \text{Beta}(\theta_{ij} | a_{ij}^0, b_{ij}^0) \text{Beta}(\rho_w | c_w^0, d_w^0). \end{aligned}$$

Unfortunately, the posterior joint distribution for θ_{ij} and ρ_w in (10) is no longer the product of Beta distributions. Therefore, after receiving a new label, we cannot update the posterior distribution as a state variable by updating the parameters a_{ij}^0, b_{ij}^0, c_w^0, and d_w^0 as in (2). To address this issue, we apply variational approximation based on the moment matching to approximate the posterior distributions of θ_{ij} and ρ_w as independent Beta distributions.

Suppose, at stage t, θ_{ij} and ρ_w are independent and their posterior distributions are Beta(a_{ij}^t, b_{ij}^t) and Beta(c_w^t, d_w^t), respectively. (This is true for $t = 0$.) Let $S_t = \{(a_{ij}^t, b_{ij}^t)\}_{1 \le i < j \le N}$ and $\Theta_t = \{(c_w^t, d_w^t)\}_{1 \le w \le W}$ be the state variables for MDP. After sending (i_t, j_t) to worker w_t, we receive a label $l_{i_t j_t w_t}$ and approximate the posterior distributions for $\theta_{i_t j_t}$ and ρ_{w_t} as

$$\begin{aligned} &p(\theta_{ij}, \rho_w | a_{ij}^t, b_{ij}^t, c_w^t, d_w^t, l_{ijw})\\ &\sim \Pr(l_{ijw} | \theta_{ij}, \rho_w) \text{Beta}(\theta_{ij} | a_{ij}^t, b_{ij}^t) \text{Beta}(\rho_w | c_w^t, d_w^t)\\ &\approx \text{Beta}(\theta_{ij} | \tilde{a}_{ij}(l), \tilde{b}_{ij}(l)) \times \text{Beta}(\rho_w | \tilde{c}_w(l), \tilde{d}_w(l)), \qquad (11) \end{aligned}$$

for $(i, j, w) = (i_t, j_t, w_t)$ and $l = l_{i_t j_t w_t}$. Here, the parameters $\tilde{a}_{ij}(l)$, $\tilde{b}_{ij}(l)$, $\tilde{c}_w(l)$, and $\tilde{d}_w(l)$ are defined as the values that make the distributions of θ_{ij} and ρ_w on both sides of (11) have the same first and second moments, known as the moment matching technique. In particular, the moments of the right hand side are functions of $\tilde{a}_{ij}(l)$, $\tilde{b}_{ij}(l)$, $\tilde{c}_w(l)$, and $\tilde{d}_w(l)$, while the moments of the left hand side are functions of a_{ij}^t, b_{ij}^t, c_w^t, d_w^t, and l_{ijw}, leading to a system of four equations and four unknown parameters. Hence, $\tilde{a}_{ij}(l)$, $\tilde{b}_{ij}(l)$, $\tilde{c}_w(l)$, and $\tilde{d}_w(l)$ can be uniquely solved from that system for $(i, j, w) = (i_t, j_t, w_t)$ according to $l_{ijw} = l = 1$ or 0. For $(i, j, w) \neq (i_t, j_t, w_t)$, the posteriors of θ_{ij} and ρ_w are unchanged.

Applying this approximation in each stage, we are able to represent the state variables for our MDP as $S_{t+1} = \{(a_{ij}^{t+1}, b_{ij}^{t+1})\}_{1 \le i < j \le N}$ and $\Theta_{t+1} = \{(c_w^{t+1}, d_w^{t+1})\}_{1 \le w \le W}$ for stage $t + 1$, where

$$(a_{ij}^{t+1}, b_{ij}^{t+1}) = \begin{cases} (\tilde{a}_{ij}(l), \tilde{b}_{ij}(l)) & \text{if } (i, j) = (i_t, j_t);\\ (a_{ij}^t, b_{ij}^t) & \text{if } (i, j) \neq (i_t, j_t), \end{cases} \qquad (12)$$

for $1 \le i < j \le N$, and

$$(c_w^{t+1}, d_w^{t+1}) = \begin{cases} (\tilde{c}_w(l), \tilde{d}_w(l)) & \text{if } w = w_t;\\ (c_w^t, d_w^t) & \text{if } w \neq w_t, \end{cases} \qquad (13)$$

for $1 \le w \le W$.

Algorithm 2 Bayesian decision process with unreliable workers based on the Opt-KG policy

Input: Parameters of prior Beta distributions $\{a_{ij}^0, b_{ij}^0\}_{1 \le i < j \le N}$ and $\{c_w^0, d_w^0\}_{1 \le w \le W}$, the budget T, the number of clusters K.

1: **for** $t = 1, ..., T$ **do**

2: Compute $R_t^+(i, j, w)$ for $1 \le i < j \le N$ and $1 \le w \le W$ based on (15) and (16).

3: Select the pair of items (i_t, j_t) and the worker w_t according to:

$$\text{Opt-KG: } (i_t, j_t, w_t) = \arg\max_{i<j, w} R_t^+(i, j, w)$$

and send items (i_t, j_t) to worker w_t.

4: Acquire the similarity label $l_{i_t j_t w_t} \in \{1, 0\}$.

5: Update S_t to S_{t+1} according to (12) and update Θ_t to Θ_{t+1} according to (13) .

6: **end for**

Output: The K clusters $\{C_1, ..., C_K\}$ that solve MinCut$(\{\mathbb{E}(\theta_{ij} | S_T)\})$ defined by (4) and (3).

Defining a policy as a sequence of mappings $(\pi_1, ..., \pi_T)$ with $\pi_t(S_t, \Theta_t) \in \{(i, j, w) | 1 \le i < j \le N, 1 \le w \le W\}$, the crowdsourced clustering problem with unreliable workers can be approximated by the following MDP similar to (6):

$$\begin{aligned} \max_{\pi_1, ..., \pi_T} &- \text{MinCut}(\{\mathbb{E}(\theta_{ij} | S_0, \Theta_0)\}) \qquad (14)\\ &+ \sum_{t=0}^{T-1} \mathbb{E}[R(S_t, \Theta_t, i_t, j_t, w_t, l_{i_t j_t w_t})] \end{aligned}$$

where $R(S_t, \Theta_t, i_t, j_t, w_t, l_{i_t j_t w_t})$

$$\triangleq \text{MinCut}(\{\mathbb{E}(\theta_{ij} | S_t)\}) - \text{MinCut}(\{\mathbb{E}(\theta_{ij} | S_{t+1})\}). \qquad (15)$$

Recall (12), (13) and (3). We have $\{\mathbb{E}(\theta_{ij} | S_{t+1})\} = U^t(i_t, j_t, w_t, l_{i_t j_t w_t}) \triangleq \{u_{ij}^t(l)\}$ where $l = l_{i_t j_t w_t}$ and

$$u_{ij}^t(l) = \begin{cases} \tilde{a}_{ij}(l) / (\tilde{a}_{ij}(l) + \tilde{b}_{ij}(l)) & \text{if } (i, j) = (i_t, j_t);\\ a_{ij}^t / (a_{ij}^t + b_{ij}^t) & \text{otherwise.} \end{cases}$$

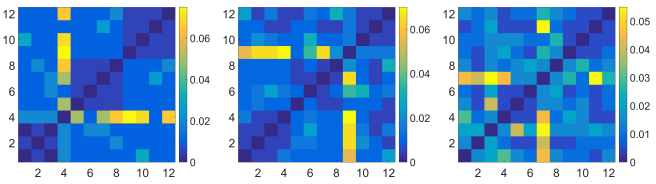
Similar to (8), we define the optimistic reward as

$$R_t^+(i, j, w) \triangleq \max_{l=0,1} R(S_t, \Theta_t, i_t, j_t, w_t, l) \qquad (16)$$

and propose the Opt-KG policy for the setting with unreliable workers as

$$\begin{aligned} (i_t, j_t, w_t) &= \arg\max_{i<j, w} R_t^+(i, j, w)\\ &= \arg\min_{i<j, w} \Big(\min_{l=0,1} \text{MinCut}(U^t(i, j, w, l)) \Big), \end{aligned}$$

where the second equality is from (15). We formally state the Opt-KG policy in Algorithm 2. Similar to Algorithm 1, in Algorithm 2, we also can choose B different (i, j, w)'s with the largest $R_t^+(i, j, w)$ and receive B labels in each stage in order to improve the time efficiency.



(a) Beta(10,1) (b) Beta(4,1) (c) Beta(5,2)

Figure 1: The heat maps of labeling frequencies for different distributions of the similarity within clusters.

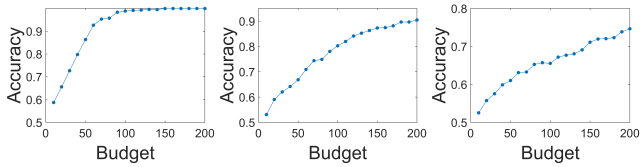
4 Simulation Studies

In this section, we conduct the numerical experiments on the simulated data to show the performance of the proposed budget allocation policies in terms of their (a) labeling frequency and (b) clustering accuracy. Here, (a) means the distribution of the budget over item pairs (and workers). To measure (b), we generate the data in K clusters, denoted by $C_1, \ldots, C_K$. And the clustering accuracy is defined as $\max_\sigma \sum_{k=1}^{K} \frac{r_{k\sigma(k)}}{N}$, where r_{kl} is the number of items in C_k assigned to cluster l and $\sigma : \{1, 2, \ldots, K\} \to \{1, 2, \ldots, K\}$ is a permutation of $\{1, 2, \ldots, K\}$. Besides, we also use the normalized mutual information (NMI) score to evaluate the clustering performance. We adopt the uniform prior Beta$(1, 1)$ for each θ_{ij} and the prior Beta$(4, 1)$ for each ρ_w unless otherwise specified. We utilize the graph partitioning algorithm [Hespanha, 2004] based on spectral factorization to solve the Min-K-Cut problem. We compare our method (Opt-KG) with the knowledge gradient (KG) and random assignment (Random).

4.1 Simulation with Reliable Workers

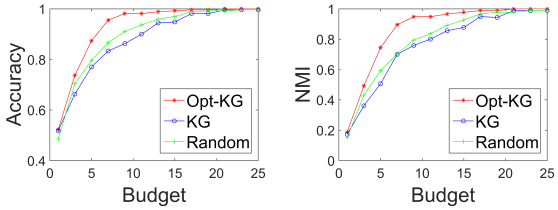
In this section, we assume all workers are fully reliable. We first investigate the labeling frequency. We generate 12 items that form three clusters $\{C_1, C_2, C_3\}$. More specifically, the first four items are from C_1, the next four items are from C_2, and the last four items are from C_3. We assume that the total budget T is 200 and the similarity parameter θ_{ij} between items in the same cluster is generated from Beta(a_s, b_s), and θ_{ij} between items in different clusters is generated from Beta(a_d, b_d), where $(a_d, b_d) = (b_s, a_s)$. We consider three settings: $(a_s, b_s) = (10, 1), (4, 1), (5, 2)$. The labeling frequency for each setting is plotted in Figure 1. We note that each box represents one pair of items and the warmer color correponds to a higher percentage of the budget spent on labeling that pair. According to our setting, the difficulty of clustering increases from Figure 1(a) to 1(b) or 1(c) because the expectation of θ_{ij} within clusters (i.e. $a_s/(a_s + b_s)$) become closer to the expectation value of θ_{ij} between clusters (i.e. $b_s/(a_s + b_s)$). We can observe from Figure 1 that, when a_s and b_s are very different (clustering is easy), the labeling frequency is higher on the pairs from different clusters, and when a_s and b_s are close (clustering is difficult), the labeling frequency of the pairs from the same cluster becomes higher.

Next, we investigate the robustness of the proposed method with the uniform prior distribution for θ_{ij} by varying the true generating distribution of θ_{ij}. Here, we use the same set of 12 items as described above. We plot the clustering accuracy for different levels of budget $T = 10, 20, ..., 200$ in Figure 2. For better visualization, we omit the standard deviation on the



(a) Beta(10,1) (b) Beta(4,1) (c) Beta(5,2)

Figure 2: The clustering accuracies for different distributions of the similarity within clusters.



(a) Accuracy (b) NMI

Figure 3: The clustering accuracy and NMI by different policies with reliable workers.

figures as the standard deviations are relatively small. As we can see, when a_s and b_s become closer, the proposed method needs more budget to obtain a high clustering accuracy, which is consistent with the intuition.

Next, we compare the performance of the proposed Opt-KG policy with the KG policy and the random sampling policy (select item pairs randomly at every stage). We choose the simulated data with a larger size and conduct each policy with a mini-batch of size $B = 100$ and a total budget of $T = 25$. We simulate 30 items in three clusters and generate θ_{ij} between the items in the same cluster from Beta$(5, 2)$, and generate θ_{ij} between the items in different clusters from Beta$(2, 5)$. We report the clustering accuracy and NMI obtained by each policy for $T = 1, 3, 5, ..., 25$ in Figure 3. This figure shows that our Opt-KG policy increases both performance metrics more rapidly as the budget is consumed.

4.2 Simulation with Unreliable Workers

In this section, we consider the simulation studies with unreliable workers. First, we investigate the sensitivity of the proposed method to the workers' reliability. We fix ten workers with reliability $\boldsymbol{\rho} = (\rho_1, \rho_2, ..., \rho_{10}) = (0.55, 0.60, ..., 1)$. We simulate 12 items in 3 clusters and generate θ_{ij} within clusters from Beta(4,1) and θ_{ij} between clusters from Beta(1,4). We choose the total budget $T = 400$. Figure 4 reports the average labeling frequency over these ten workers, which shows that our policy will detect the reliable workers and the workers with higher reliability are assigned with more items to label. Such assignments efficiently increase the performance of clustering and save the budget.

Next, we investigate the robustness of the proposed method with the prior Beta$(4, 1)$ for ρ_w by varying the true generating distribution of ρ_w. We generate θ_{ij} and items as above and consider five workers with ρ_k generated from Beta(8,1), Beta(11,2) or Beta(3,1). Figure 5 displays how the clustering accuracy varies with the level of budget ($T = 60, 120, ..., 600$) in our Opt-KG policy when using Beta$(4, 1)$

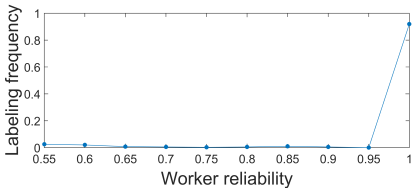
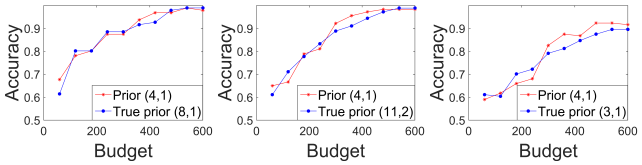
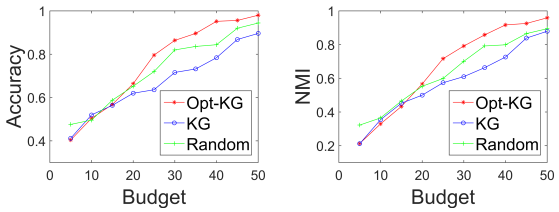


Figure 4: The labeling frequency over workers with different reliability.



(a) Beta(8,1) (b) Beta(11,2) (c) Beta(3,1)

Figure 5: Comparison of clustering accuracies using the prior Beta$(4,1)$ and the true priors on the reliability parameter.



(a) Accuracy (b) NMI

Figure 6: Performance comparison of clustering accuracy and NMI under the unreliable workers setting.

as the prior for ρ_w and when using the true generating distributions of ρ_w as the prior. As we can see, the prior Beta$(4,1)$ leads to a better clustering result than the true distributions.

We compare the performance of the proposed Opt-KG policy with the KG policy and the random sampling policy with a total budget of $T = 50$ and a batch size $B = 300$. We simulate 50 items in 5 clusters and generate θ_{ij} within clusters from Beta$(5,2)$ and θ_{ij} between clusters from Beta$(2,5)$. We consider five workers with reliability $\boldsymbol{\rho} = (\rho_1, \rho_2, ..., \rho_5) = (0.70, 0.75, ..., 0.90)$. We report the clustering accuracy and NMI obtained by each policy for $T = 5, 10, ..., 50$ in Figure 6. According to Figure 6, our Opt-KG policy increases the clustering quality fastest as the budget is consumed.

5 Experiment with Real Data

We compare different policies for clustering four datasets [Dua and Graff, 2017; Bagnall *et al.*, 2019]: soybean, olive oil, meat and iris. In each dataset, we calculate the Euclidean distance between each pair of instances and obtain θ_{ij} through a linear mapping based on the principle that pairs of instances with smaller distances have higher similarity. For instances i and j from the same cluster, θ_{ij} ranges from 0.7 to 0.9. For instances from different clusters, θ_{ij} ranges from 0.1 to 0.3. We first consider the reliable workers setting. We generate the labels from a random worker using (1) with the true θ_{ij}. The uniform prior Beta(1,1) is used in the Opt-KG policy with the total budget $T = 10$, the batch size $B = 200$ for soybean and olive oil and $B = 300$ for meat and iris. We run each policy 10 times and report the averaged clustering accuracies under different budget levels ($T = 1, 4, 7, 10$) in Table 1. As we can see, our Opt-KG policy has the best performance. After only seven stages, the proposed method based on the Opt-KG policy can attain a high clustering accuracy.

Dataset (N,K)	Policy	Reliable workers				Unreliable workers			
		T=1	T=4	T=7	T=10	T=1	T=4	T=7	T=10
Soybean $(47,4)$	Opt-KG	0.47	0.85	0.96	0.98	0.47	0.61	0.83	0.87
	KG	0.48	0.71	0.92	0.97	0.46	0.57	0.59	0.68
	Random	0.46	0.67	0.87	0.94	0.47	0.59	0.74	0.84
Olive oil $(60,4)$	Opt-KG	0.39	0.70	0.88	0.92	0.42	0.55	0.75	0.81
	KG	0.47	0.60	0.64	0.63	0.41	0.53	0.54	0.63
	Random	0.38	0.71	0.82	0.83	0.43	0.58	0.72	0.76
Meat $(120,3)$	Opt-KG	0.42	0.76	0.92	1.00	0.41	0.77	0.94	0.97
	KG	0.41	0.67	0.80	0.90	0.39	0.69	0.69	0.71
	Random	0.42	0.71	0.93	0.97	0.42	0.79	0.92	0.91
Iris $(150,3)$	Opt-KG	0.41	0.76	0.89	0.91	0.69	0.62	0.86	0.96
	KG	0.39	0.58	0.63	0.61	0.59	0.64	0.65	0.61
	Random	0.37	0.58	0.87	0.89	0.42	0.49	0.75	0.84

Table 1: Performance comparison on four datasets.

Next, we consider the setting with unreliable workers. We consider five workers with reliability $\boldsymbol{\rho} = (\rho_1, \rho_2, ..., \rho_5) = (0.70, 0.75, ..., 0.90)$ and generate labels according to (9). We choose the uniform prior Beta$(1,1)$ for θ_{ij} and choose Beta$(4,1)$ as the prior for ρ_w in the Opt-KG policy with $T = 10$, $B = 400$ for soybean and olive oil and $B = 1000$ for meat and iris. From Table 1 we can see that the Opt-KG policy is superior to the KG policy and the random sampling policy under the setting with unreliable workers.

6 Conclusions

In this paper, we propose an online policy for the budget allocation problem in crowdsourced clustering. We transform the clustering problem into a graph partition problem based on the minimum-weight K-cut problem. We introduce the priors of the similarity parameters for item pairs and workers' reliability and then model the problem as a Bayesian Markov decision process. We develop a computationally efficient Opt-KG policy to approximately solve the MDP for both cases of reliable and unreliable workers. The experimental studies show that the proposed method achieves a good performance.

This paper uses Min-K-Cut function to solve the graph partition problem. In future work, it will be interesting to study the budget allocation problem through other graph partition methods. When only one label (or a batch of labels) is obtained at one stage, how to solve the optimization problem at the next stage locally and faster is challenging. Another interesting extension is to consider the dynamic pricing strategy. A novel pricing scheme instead of equal cost can motivate workers to provide labels with higher quality.

Acknowledgments

Xiaozhou Wang and Weidong Liu are supported by National Program on Key Basic Research Project (973 Program, 2018AAA0100704), NSFC Grant No. 11825104. Xi Chen is supported by the NSF Grant via IIS-1845444 and the Bloomberg Data Science Research Grant.

References

[Bagnall *et al.*, 2019] Anthony Bagnall, Jason Lines, William Vickers, and Eamonn Keogh. The UEA & UCR time series classification repository, 2019.

[Bragg *et al.*, 2013] Jonathan Bragg, Mausam, and Daniel S Weld. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI Conference on Human Computation and Crowdsourcing*, pages 25–33, 2013.

[Chen *et al.*, 2013] Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 193–202. ACM, 2013.

[Chen *et al.*, 2015] Xi Chen, Qihang Lin, and Dengyong Zhou. Statistical decision making for optimal budget allocation in crowd labeling. *The Journal of Machine Learning Research*, 16(1):1–46, 2015.

[Chen *et al.*, 2016] Xi Chen, Kevin Jiao, and Qihang Lin. Bayesian decision process for cost-efficient dynamic ranking via crowdsourcing. *The Journal of Machine Learning Research*, 17(1):7617–7656, 2016.

[Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[Frazier *et al.*, 2008] Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

[Goldschmidt and Hochbaum, 1988] Olivier Goldschmidt and Dorit S Hochbaum. Polynomial algorithm for the k-cut problem. In *Proceedings of 29th Annual Symposium on Foundations of Computer Science*, pages 444–451. IEEE, 1988.

[Gupta and Miescke, 1996] Shanti S Gupta and Klaus J Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2):229–244, 1996.

[Hespanha, 2004] Joao P Hespanha. An efficient matlab algorithm for graph partitioning. *Santa Barbara, CA, USA: University of California*, 2004.

[Karger *et al.*, 2011] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 284–291. IEEE, 2011.

[Luo *et al.*, 2018] Yucen Luo, Tian Tian, Jiaxin Shi, Jun Zhu, and Bo Zhang. Semi-crowdsourced clustering with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3212–3222, 2018.

[Mazumdar and Saha, 2016] Arya Mazumdar and Barna Saha. Clustering via crowdsourcing. *arXiv preprint arXiv:1604.01839*, 2016.

[Nocetti *et al.*, 2003] Fabian Garcia Nocetti, Julio Solano Gonzalez, and Ivan Stojmenovic. Connectivity based k-hop clustering in wireless networks. *Telecommunication Systems*, 22(1-4):205–220, 2003.

[Puterman, 2014] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[Tao *et al.*, 2019] D. Tao, J. Cheng, Z. Yu, K. Yue, and L. Wang. Domain-weighted majority voting for crowdsourcing. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1):163–174, 2019.

[Tran-Thanh *et al.*, 2013] Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 901–908. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[Vesdapunt *et al.*, 2014] Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

[Whang *et al.*, 2013] Steven Euijong Whang, Peter Lofgren, and Hector Garcia-Molina. Question selection for crowd entity resolution. *Proceedings of the VLDB Endowment*, 6(6):349–360, 2013.

[Xu *et al.*, 1998] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings 14th International Conference on Data Engineering*, pages 324–331. IEEE, 1998.

[Zhang and Wu, 2018] Jing Zhang and Xindong Wu. Multi-label inference for crowdsourcing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2738–2747. ACM, 2018.

[Zhong, 2005] Shi Zhong. Efficient online spherical k-means clustering. In *Proceedings of International Joint Conference on Neural Networks*, volume 5, pages 3180–3185. IEEE, 2005.