Journal of Machine Learning Research 17 (2016) 1-40 Submitted 2/16; Published 11/16

Bayesian Decision Process for Cost-Efficient Dynamic Ranking via Crowdsourcing

Xi Chen* XCHEN3@STERN.NYU.EDU
Stern School of Business
New York University
New York, New York, 10012, USA

Kevin Jiao JJIAO@STERN.NYU.EDU
Stern School of Business
New York University
New York, New York, 10012, USA

Qihang Lin QIHANG-LIN@UIOWA.EDU
Tippie College of Business
University of Iowa
Iowa City, Iowa, 52242, USA

Editor: Qiang Liu

Abstract

Rank aggregation based on pairwise comparisons over a set of items has a wide range of applications. Although considerable research has been devoted to the development of rank aggregation algorithms, one basic question is how to efficiently collect a large amount of high-quality pairwise comparisons for the ranking purpose. Because of the advent of many crowdsourcing services, a crowd of workers are often hired to conduct pairwise comparisons with a small monetary reward for each pair they compare. Since different workers have different levels of reliability and different pairs have different levels of ambiguity, it is desirable to wisely allocate the limited budget for comparisons among the pairs of items and workers so that the global ranking can be accurately inferred from the comparison results. To this end, we model the active sampling problem in *crowdsourced ranking* as a Bayesian Markov decision process, which dynamically selects item pairs and workers to improve the ranking accuracy under a budget constraint. We further develop a computationally efficient sampling policy based on knowledge gradient as well as a moment matching technique for posterior approximation. Experimental evaluations on both synthetic and real data show that the proposed policy achieves high ranking accuracy with a lower labeling cost.

Keywords: crowdsourced ranking, Bayesian, Markov decision process, dynamic programming, knowledge gradient, moment matching

1. Introduction

Inferring the ranking over a set of items, such as documents, images, movies, or URL links, is an important learning problem with many applications in areas like web search, recommendation systems, online games, etc. An interesting problem related to rank inference is estimating a score for each item based on a certain criterion that the items can be ranked,

*. The authors are listed in alphabetical order

such as the score of relevance or the score of quality. Typically, both the ranking and the scores of items can be inferred from a collection of high-quality labels on the items. There are mainly two different types of labels. The label of the first type is associated with each individual item in order to characterize the property of the item itself, for example, a binary or an ordinal score (e.g., 5-point grade). The label of the second type is instead associated with a subset of items that reveal their relative properties, for example, a partial ranking that covers only this subset. Labels of both types can be obtained by soliciting the knowledge of human workers, depending on whether the worker is employed to evaluate a single item or to compare a subset of items according to a given criterion. In practice, a binary score usually cannot fully distinguish all items and ordinal scores from different workers are often inconsistent due to the difference in their understandings of the grades in the ordinal scoring scheme. Therefore, the second type of labels has been more widely adopted, which can effectively reduce the impact of misunderstanding among workers and is more appropriate for ranking fine-grained items with a large number of graduations (e.g., in our real data experiment on accessing reading difficulty of an article into one of twelve American grade levels). Moreover, empirical evidences show that the ranking accuracy of a human worker typically decreases when he or she has to compare many items at a time. For this reason, in this paper, we only consider the relative comparisons over *pairs* of items and the label from a human worker indicates which item is preferred to the other.

The traditional approach of conducting pairwise comparisons by a small group of experts is usually time consuming and expensive. It fails to meet the growing need of labeled data for ranking tasks. Because of the advent of online crowdsourcing services (Howe, 2006) such as Amazon Mechanical Turk, a more efficient and more economic approach has emerged: a large amount of unlabeled pairs of items are posted to a crowdsourcing platform, where a crowd of workers are hired to perform pairwise comparisons and provide labels of the assigned pairs. Given the labels from crowd workers, we can infer a global ranking over all items. We refer to the process of collecting pairwise labels and ranking items as *crowdsourced ranking*.

Despite its availability and scalability, challenges remain in crowdsourced ranking. A certain amount of monetary reward is paid to a worker for each pair of items he or she compares while there is usually only a fixed amount of budget available, limiting the total number of pairwise labels we can collect. Hence, there is a need for a budget-efficient decision process for allocating the budget over item pairs and workers. In particular, on crowdsourcing platforms, there are unreliable workers who submit their answers quickly but carelessly in order to obtain more monetary reward with less effort. Hence, the comparison results provided by crowd workers often contain non-negligible noise. As a remedy, multiple workers are hired to compare the same pair of items independently in the hope that the correct ranking can be recovered, and that the unreliable workers can be identified by comparing their answers with the rest of workers. However, each pairwise comparison will incur a pre-specified monetary cost. Without a careful control, such a repetitive labeling strategy often results in too many labels on the same pair by different workers, leading to a high cost. Furthermore, because of the diversity of their backgrounds and expertise, workers do not always agree with each other in the results of pairwise comparisons, especially when the two items in comparison are competitive to each other. We refer to such a competitive pair as an *ambiguous pair* since the ordering of them is more difficult to be determined.

Presumably, a greater budget should be spent on ambiguous pairs, but identifying ambiguous pairs under the budget constraint itself is a challenging problem, which requires some effective learning scheme. Given the trade-off between the labeling cost and the quality of ranking results, there are two fundamental challenges in crowdsourced ranking:

1. Given the inconsistent pairwise labels from crowd workers with different reliability, how to aggregate these labels into a global ranking over items.

2. With both unreliable workers and ambiguous pairs initially unidentified, how to incorporate a learning scheme with an efficient sampling procedure (over both pairs of items and workers) under the budget constraint to achieve the highest ranking accuracy.

To address these challenges, we need to first model the reliability of workers and the ambiguity of item pairs and analyze how they influence the pairwise label. To this end, we adopt a combination of the Bradley-Terry-Luce ranking model (Bradley and Terry, 1952; Luce, 1959) for modeling the comparison results and the Dawid-Skene model (Dawid and Skene, 1979) for workers' reliability. The reason why we adopt the Bradley-Terry-Luce model is that learning such a model will not only provide a ranking over items but also give a score to each item, which can be useful in many applications (e.g., providing player's rating in chess games). We measure the quality of the ranking inferred from the collected labels using the *Kendall's tau rank correlation coefficient* (Kendall's tau for short) with respect to the underlying true ranking.

Under such a model and a quality measure, we propose a dynamic sampling and ranking procedure which addresses the aforementioned two challenges in a unified framework. In particular, we first introduce the priors for items' latent true scores and workers' reliability and formulate the crowdsourced ranking problem into a finite-horizon Bayesian *Markov decision problem* (MDP), whose state variables correspond to the posterior distributions given the observed labels. Here, the number of stages is determined by the total budget, i.e., the total number of pairs that can be requested for labeling. As the budget level increases, the size of the state space grows at an exponential rate, which makes the exact solving of such a MDP problem intractable. To address the computational difficulty, we propose an efficient sampling strategy called *approximated knowledge gradient* (AKG) policy based on the popular knowledge gradient policy (Powell, 2010; Frazier, 2009; Frazier et al., 2008; Ryzhov et al., 2012). The proposed policy dynamically chooses the next pair of items and the worker that together lead to a maximum expected improvement in Kendall's tau rank correlation coefficient. Finally, to determine the global ranking that maximizes the expected Kendall's tau, one needs to solve a maximum linear ordering problem (Grötschel et al., 1984), which is a NP-hard problem (and in fact, APX-hard (approximable-hard) (Mishra and Sikdar, 2004)). To address this challenge, we propose a moment matching technique to approximate the posteriors in parametric forms so that the linear ordering problem under the approximated posterior can be easily solved by a simple sorting procedure.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we introduce the model and the proposed policy under the simplified case where all workers are homogeneous and perfectly reliable. In Section 4, we extend our policy to the case where the crowd workers have heterogeneous reliability. In Section 5, we

present numerical results on both simulated and real datasets, followed by conclusions in Section 6. The detailed proofs and derivations are provided in the appendix.

2. Related Work

The dataset of partial rankings over items can be generated from a variety of sources including crowdsourcing services (Shah et al., 2016b), online competition games (e.g., Microsoft's TrueSkill system (Herbrich et al., 2007)), and online users' activities such as browsing, clicking and transactions that reveal certain preferences. Learning a global ranking of a large set of items by aggregating a collection of partial rankings/preferences has been an active research area for the past ten years (see, e.g., Gleich and Lim (2011); Negahban et al. (2012); Yi et al. (2013); Shah et al. (2016a,b); Rajkumar and Agarwal (2014); Lu and Boutilier (2014); Volkovs and Zemel (2014)). However, most work on rank aggregation considers a static estimation problem — inferring a global ranking based on a pre-existing dataset. The problem we consider here is related to but significantly different from these works because we model crowdsourced ranking as a dynamic procedure where the inference of ranking and collection of data proceed concurrently and influence each other.

The crowdsourced ranking problem we considered has a close connection with the dynamic sorting problem using noisy pairwise comparisons, which has been studied by several authors (Ailon, 2012; Braverman and Mossel, 2008; Radinsky and Ailon, 2011; Wauthier et al., 2013; Jamieson and Nowak, 2011). However, these papers assume the noise of pairwise comparison results has the same distribution for all pairs, which is not reasonable in crowdsourced ranking because workers usually rank significantly different items more correctly than they do for similar items. The approaches proposed by Pfeiffer et al. (2012) and Qian et al. (2015) assume that the labeling noise depends on the latent qualities or features of the items. However, their approaches do not model the reliability of workers in the decision process. In contrast, our approach allows a label's noise to depend not only on the items themselves, but also on the reliability of the worker who provides the label. The ranking model adopted in this paper, which combines the Bradley-Terry-Luce model and the Dawid-Skene model, was originally proposed in (Chen et al., 2013), which also considers a similar problem of Bayesian statistical decision-making for crowdsourced ranking. However, the sampling strategy developed in Chen et al. (2013), which prioritizes the pair of items and the worker with the highest information gain, is a simple heuristic without a well-defined objective function to be optimized. In contrast, our work chooses the expected Kendall's tau as the objective function to maximize, which guides the development of the knowledge gradient policy.

In addition to crowdsourced ranking, the problem of crowdsourced categorical labeling/classification has been extensively studied in the past five years. Most work aims at solving a static problem, which infers the categorical labels and workers' reliability based on a static problem (see, e.g., Dawid and Skene (1979); Raykar et al. (2010); Welinder et al. (2010); Whitehill et al. (2009); Liu et al. (2012); Gao and Zhou (2013); Zhang et al. (2014)). Recently, some research has been devoted to dynamic sampling in crowdsourced classification (Karger et al., 2013b,a; Bachrach et al., 2012; Ertekin et al., 2012; Kamar et al., 2012; Ho et al., 2013; Chen et al., 2015). In particular, both Kamar et al. (2012) and Chen et al. (2015) utilized the Markov decision process to model the budget allocation (i.e.,

sampling over items and workers) process. Since we also adopt a Bayesian Markov decision process with a variant of knowledge gradient policy, the spirit of our method is similar to that in Chen et al. (2015). However, since the statistical model for a ranking problem is fundamentally different from that of a classification problem, the Markov decision process in this paper is significantly different from the one introduced by Chen et al. (2015) in many aspects such as the objective function, stage-wise rewards, transition probabilities, optimal policy, etc. For example, the policy by Chen et al. (2015) is designed to maximize the expected classification accuracy while our policy aims at maximizing the expected Kendall's tau with respective to the true ranking. In fact, even for a static problem with a given set of collected data, inferring the ranking with the maximum expected Kendall's tau is equivalent to a NP-hard maximum linear ordering problem while classifying items with a maximum expected accuracy can be done in closed-form by Bayesian decision rule. In this paper, we avoid this computational challenge by exploiting the structure of the expected Kendall's tau and approximating the posteriors using moment matching. We also note that, although one can view the problem of ranking K items as a problem of classifying $K(K-1)/2$ pairs (each pair is treated as an item in Chen et al. (2015)), such an approach increases the size of the problem and ignores the dependency between pairwise labels.

In addition, it is worth to note that the problem we consider here is different from the typical tasks in machine-learned ranking or learning to rank (Liu, 2009; Acharyya, 2013) where some feature information is available for each item and training data is used to calibrate some statistical models for ranking new items. In contrast to these problems, the feature information is not necessary in our crowdsourced ranking problem. Moreover, besides being applied to ranking items directly, our methods can be utilized to collect training labels for learning to rank problems. According to the type of training data utilized, statistical ranking methods can be classified into three categories (Liu, 2009; Acharyya, 2013): pointwise method, pairwise method and listwise method. The pointwise methods (Li et al., 2008; Cooper et al., 1992; Crammer and Singer, 2001) learn a ranking model based on the data of scores or ratings of items. The pairwise methods (Freund et al., 2003; Burges et al., 2005; Zheng et al., 2008; Cao et al., 2006) and the listwise methods (Xu and Li, 2007; Cao et al., 2007; Taylor et al., 2008; Kuo et al., 2009) learn a ranking model using pairwise comparison results or partial rankings over a subset of items. For the pairwise or listwise methods, the crowdsourced ranking technique we proposed can be used as an upstream procedure that provides high-quality pairwise/listwise comparison data which helps increase the accuracy of the models in the aforementioned papers.

3. Crowdsourced Ranking by Homogeneous Workers

In this section, we first consider a simplified setting where workers are homogeneous (we will clarify the meaning of "homogeneous workers" shortly). In Section 4, we further extend the developed method for homogeneous workers to heterogeneous workers with different levels of reliability.

3.1 Model Setup

We assume that there are K items (denoted by $\{1, \ldots, K\}$) to be ranked and each item i has an *unknown* latent score $\theta_i > 0$ for $i = 1, 2, \ldots, K$. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)^T$, where each

latent score θ_i models the intensity of preference to item i under some criterion. A *ranking* over K items $\{1, 2, \ldots, K\}$ is a permutation/one-to-one mapping $\pi : \{1, 2, \ldots, K\} \to \{1, 2, \ldots, K\}$ and $\pi(i)$ is the *rank* of item i under π. We follow the convention that $\theta_i > \theta_j$ means item i is preferred to item j and thus item i should have a higher rank than item j. Therefore, the underlying *true ranking* π^* over K items is determined by the ranking of their latent scores, i.e.,

$$\pi^*(i) > \pi^*(j) \quad \text{if and only if} \quad \theta_i > \theta_j. \tag{1}$$

We note that the latent scores naturally provide a characterization of *ambiguity for a pair of items*: when the values of θ_i and θ_j are closer, the pair of item i and j is more ambiguous in the sense that the true ordering of them is less obvious.

The way we explore the ranking of θ_i's is through the collection of workers' preferences on different pairs of items. Specifically, we will present only two items at a time to a worker, who will be asked to compare these two items according to the given ranking criterion. Each worker will not be asked to compare the same pair more than once. The results of comparisons will be collected over time and become our historical data, based on which, our task is to infer the true ranking π^*.

In this section, we consider a basic setup where the crowd workers are assumed to be *homogeneous*, meaning that the probabilistic outcomes of their comparisons are only affected by the ambiguities of pairs. More specifically, suppose a worker is randomly selected from the crowd to compare a pair of items i and j with $i < j$ and the comparison result is denoted by a random variable Y_{ij}:

$$Y_{ij} = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by the randomly selected worker} \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by the randomly selected worker.} \end{cases} \tag{2}$$

The setting of homogeneous workers means the probability distribution of Y_{ij} takes the following form

$$\Pr(Y_{ij} = 1) = \frac{\theta_i}{\theta_i + \theta_j} \quad \text{and} \quad \Pr(Y_{ij} = -1) = \frac{\theta_j}{\theta_i + \theta_j} \quad \text{for } i, j = 1, 2, \ldots, K. \tag{3}$$

The probabilistic model we used in (3) is the well-known Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959). We choose this model for the distribution of Y_{ij} because it admits a simple structure and well fits our framework of dynamic sampling. Furthermore, our method developed for the BTL model can be easily extended to the case of heterogeneous workers which will be studied in Section 4.

It is worthwhile to mention that other comparison models can potentially be implemented here. Considering a simplified version of the Thurstone model (Thurstone, 1927) in which each object i has a score following $N(\theta_i, 1)$, then we have

$$\Pr(Y_{ij} = 1) = \Phi\left(\frac{\theta_i - \theta_j}{\sqrt{2}}\right) \quad \text{and} \quad \Pr(Y_{ij} = -1) = \Phi\left(\frac{\theta_j - \theta_i}{\sqrt{2}}\right).$$

The problem can still be formulated using a Bayesian decision process framework. However, there are several reasons why the BTL model is favored in this paper. First of all, moment

matching under the Thurstone model does not have closed-form solutions and hence we must rely on numerical scheme to compute the first and second moments of the posterior. Second, using moment matching approach, because the posterior is an n-dimensional multivariate Gaussian distribution, we need to update $n(n+1)/2$ parameters (the number of mean parameters plus the number of off-diagonal elements of the covariance matrix) during each iteration of the algorithm whereas with Dirichlet posterior there are only n parameters. Last but not least, with Thurstone model the ranking is no longer a simple sorting of parameters, which is a feature of the BTL model as shown in Theorem 2.

Since each worker can compare the same pair at most once, we assume the size of the crowd workers is large enough so that the distribution of Y_{ij} stays the same after sampling workers without replacement. Note that we can assume $\sum_{i=1}^{K} \theta_i = 1$ without loss of generality since the distribution of Y_{ij} in (3) remains unchanged if we multiply each θ_i by the same positive constant. The probability $\frac{\theta_i}{\theta_i+\theta_j}$ in (3) can also be interpreted as the percentage of workers in the crowd who prefer item i to item j.

Since the probabilistic model (3) does not incorporate or reveal the quality of each worker in the comparison result, in the subsequent study of this section, we only need to focus on how to dynamically select pairs of items to compare. The worker will be selected randomly from the crowd. A dynamic choice over workers will be incorporated into our method in Section 4 where the performance of workers is modeled heterogeneously.

3.2 Bayesian Decision Process

In a typical crowdsourcing marketplace, a monetary cost must be paid to a worker every time this worker completes a task such as comparing a pair of items. We assume the cost for each comparison is one unit and the total budget available is T units so that at most T pairs (repetition allowed) can be compared in total. Since comparing different pairs will generate different historical data and reveal different information about the true ranking, it is critical to dynamically determine the right sequence of pairs to compare in order to maximize the final ranking accuracy, especially when the budget T is small.

In the traditional offline setting, one needs to determine T pairs at a time beforehand and request the comparisons on those pairs in a batch. The potential problem of such a static approach is that the budget T is not spent in an efficient way to discover the true ranking. In fact, the distribution in (3) implies that, when two items have similar latent scores, workers will provide highly inconsistent preferences and it is hard to reach an agreement on such a pair. In this case, the comparison results will be very noisy and one needs to spend more budget on this pair in order to rank them correctly. In contrast, when two items have significantly different latent scores, workers will provide consistent answers so that the additional information we can obtain is little from repeatedly comparing the same two items. In this case, one might want to reduce the budget on such a pair. Unfortunately, without any prior knowledge of the latent scores, it is impossible to decide how much budget should be spent on each pair before observing some comparison results.

In order to efficiently allocate the limited total budget over all pairs, we consider a dynamic crowdsourced ranking policy (Algorithm 1) where only one pair of items is selected and presented to a worker at each time based on historical comparison results. This online

method allows the budget to be adaptively shifted towards the ambiguous pairs so that the final ranking accuracy can be improved.

In particular, given the total budget T, the dynamic decision process consists of T stages and, in stage $t = 0, 1, \ldots, T-1$, a pair of items (i_t, j_t) with $i_t < j_t$ is presented to a randomly selected worker and we receive the comparison result $Y_{i_t j_t}$ defined in (2) and (3). The historical comparison results up to stage t can be summarized by a $K \times K$ matrix M^t with its entry[1] M^t_{ij} equal to the number of times item i is preferred to item j up to stage t. For each stage t where the pair (i_t, j_t) is compared, we define Δ^t to be a sparse $K \times K$ matrix with only one non-zero element: $\Delta^t_{i_t j_t} = 1$ if $Y_{i_t j_t} = 1$ and $\Delta^t_{j_t i_t} = 1$ if $Y_{i_t j_t} = -1$. By its definition, M^t can be updated iteratively as follows

$$M^0 = \mathbf{0}, \quad M^{t+1} = M^t + \Delta^t \quad \text{for } t = 0, 1, \ldots, T-1, \tag{4}$$

where $\mathbf{0}$ denotes the $K \times K$ all-zero matrix.

We denote an *adaptive dynamic budget allocation/sampling policy* by $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\ldots,T-1}$ where $(i_t, j_t) = (i_t(M^t), j_t(M^t))$ depends on the previous comparison results through M^t. Our goal is to find the best $\mathcal{A}$ so that the inferred ranking based on all the historical comparisons (represented by M^T) achieves the highest accuracy.

To measure the accuracy of an inferred ranking π, we adopt the popular evaluation criterion — normalized *Kendall's tau rank correlation coefficient* (Kendall, 1938) between π and π^* (Kendall's tau for short):

$$\begin{aligned} \tau(\pi, \pi^*) &\equiv \frac{\left|\{(i,j) : i < j,\ (\pi(i) - \pi(j))\,(\pi^*(i) - \pi^*(j)) > 0\}\right|}{K(K-1)/2} \\ &= \frac{2}{K(K-1)} \sum_{i \neq j} \mathbf{1}_{\{\pi(i) > \pi(j)\}} \mathbf{1}_{\{\theta_i > \theta_j\}}, \end{aligned} \tag{5}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function. Here, the numerator counts the number of pairs that π and π^* agree with each other and the denominator is the total number of pairs over K items. Hence, $\tau(\pi, \pi^*) \in [0, 1]$ and represents the percentage of agreements between π and π^*. The ranking accuracy of π is higher when $\tau(\pi, \pi^*)$ is closer to one and $\pi = \pi^*$ if and only if $\tau(\pi, \pi^*) = 1$.

However, we cannot infer a ranking based on the collected data by directly maximizing $\tau(\pi, \pi^*)$ because π^* and $\boldsymbol{\theta}$ are unknown. To address this challenge, we adopt a Bayesian framework by proposing a prior distribution on $\boldsymbol{\theta}$ and infer a ranking π that maximizes the posterior expectation of $\tau(\pi, \pi^*)$. Recall that the vector of latent scores $\boldsymbol{\theta}$ is assumed to lie in the simplex

$$\Delta \equiv \left\{ \boldsymbol{\theta} \in \mathbb{R}^k \,\middle|\, \sum_{i=1}^K \theta_i = 1, \theta_i > 0 \right\}. \tag{6}$$

It is natural to assume that $\boldsymbol{\theta}$ is drawn from a *Dirichlet prior distribution* parameterized by $\boldsymbol{\alpha}^0 = (\alpha^0_1, \ldots, \alpha^0_K)^T$ with $\alpha^0_i > 0$ for all i (note that Dirichlet distribution of order K is supported on Δ). Namely,

$$\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^0) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^0)} \prod_{i=1}^K \theta_i^{\alpha_i - 1},$$

1. In this paper, the notation A_{ij} represents the entry in the i-th row and j-th column of matrix A.

where $\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K}\Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}$ and $\Gamma(x) \equiv \int_0^\infty \lambda^{x-1}e^{-\lambda}\mathrm{d}\lambda$ is the gamma function. Given the comparison data M^t up to stage t and the probability distribution of each comparison result in (3), the density function of the posterior distribution of $\boldsymbol{\theta}$ takes the following form,

$$p(\boldsymbol{\theta}|M^t,\boldsymbol{\alpha}^0) = \frac{1}{\mathrm{H}(M^t,\boldsymbol{\alpha}^0)}\prod_{i\neq j}\left(\frac{\theta_i}{\theta_i+\theta_j}\right)^{M_{ij}^t}\prod_i \theta_i^{\alpha_i^0-1} = \frac{1}{\mathrm{H}(M^t,\boldsymbol{\alpha}^0)}\frac{\prod_{i=1}^{K}\theta_i^{\beta_i^t+\alpha_i^0-1}}{\prod_{i<j}(\theta_i+\theta_j)^{M_{ij}^t+M_{ji}^t}}, \quad (7)$$

where $\boldsymbol{\beta}^t = (\beta_1^t,\beta_2^t,\ldots,\beta_K^t)^T$ with $\beta_i^t \equiv \sum_{j\neq i} M_{ij}^t$, i.e., the number of times item i is preferred to another item up to stage t, and

$$\mathrm{H}(M^t,\boldsymbol{\alpha}^0) \equiv \int_\Delta \frac{\prod_{i=1}^{K}\theta_i^{\beta_i^t+\alpha_i^0-1}}{\prod_{i<j}(\theta_i+\theta_j)^{M_{ij}^t+M_{ji}^t}}\mathrm{d}\boldsymbol{\theta},$$

is the normalization constant.

With this posterior distribution in place and with M^t at any stage t, we can infer a ranking $\widehat{\pi}_t$ to maximize the posterior expected ranking accuracy measured by its Kendall's tau with respect to π^*, namely, to find

$$\widehat{\pi}_t \quad \in \quad \arg\max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|M^t,\boldsymbol{\alpha}^0\right], \quad (8)$$

where the expectation is taken with respect to the posterior distribution $p(\boldsymbol{\theta}|M^t,\boldsymbol{\alpha}^0)$ in (7). We denote the corresponding maximum posterior expected accuracy by $h(M^t)$, i.e.,

$$h(M^t) \quad \equiv \quad \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|M^t,\boldsymbol{\alpha}^0\right], \quad (9)$$

where the dependence of h on the prior $\boldsymbol{\alpha}^0$ is suppressed for notational simplicity. We are interested in finding a dynamic budget allocation policy $\mathcal{A} = \{(i_t,j_t)\}_{t=0,1,\ldots,T-1}$ that maximizes $h(M^T)$, i.e., the final expected ranking accuracy when the budget is exhausted. This problem can be stated as

$$\max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}}\left[h(M^T)|\boldsymbol{\alpha}^0\right], \quad (10)$$

where $\mathbb{E}^{\mathcal{A}}$ represents the expectation over the sample paths (i.e., the sampled pairs and outcomes) generated by the policy $\mathcal{A}$.

The maximization problem in (10) can be formulated as a T-stage Bayesian Markov decision process (MDP), where the *state variable* is the posterior distribution in (7) or simply the matrix M^t. The *state space* at each stage t denoted by $\mathcal{S}^t$ takes the form of

$$\mathcal{S}^t = \left\{M^t \in \mathbb{Z}_{\geq 0}^{K\times K} : \sum_{i,j} M_{ij}^t = t\right\}, \quad (11)$$

where $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers. The state variable makes a transition according to (4) given the observed comparison result $Y_{i_tj_t}$, where the sampled pair (i_t,j_t) is determined by the policy $\mathcal{A}$. The expected transition probabilities take the form of,

$$\mathbb{E}\left[\Pr(Y_{ij}=1)|M^t,\boldsymbol{\alpha}^0\right] \quad = \quad \mathbb{E}\left[\frac{\theta_i}{\theta_i+\theta_j}\Big|M^t,\boldsymbol{\alpha}^0\right] \quad (12)$$

$$\mathbb{E}\left[\Pr(Y_{ij}=-1)|M^t,\boldsymbol{\alpha}^0\right] \quad = \quad \mathbb{E}\left[\frac{\theta_j}{\theta_i+\theta_j}\Big|M^t,\boldsymbol{\alpha}^0\right] \quad (13)$$

for $1 \leq i < j \leq K$ and the expectation is taken over the posterior of $\boldsymbol{\theta}$ in (7). To complete the definition of our Bayesian MDP for crowdsourced ranking, we still need to define the *stage-wise reward*. To this end, we rewrite $h(M^T)$ in (10) as a telescopic sum,

$$h(M^T) = \sum_{t=0,1,\ldots,T-1} R(M^t, i_t, j_t, Y_{i_t j_t}); \qquad R(M^t, i_t, j_t, Y_{i_t j_t}) \equiv h(M^{t+1}) - h(M^t), \quad (14)$$

and note that $R(M^t, i_t, j_t, Y_{i_t j_t}) = h(M^{t+1}) - h(M^t)$ only depends on M^t, i_t, j_t, $Y_{i_t j_t}$. Given (14), the maximization problem (10) is equivalent to

$$\begin{aligned} & \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[h(M^0) + \sum_{t=0}^{T-1} R(M^t, i_t, j_t, Y_{i_t j_t}) \middle| \boldsymbol{\alpha}^0 \right] \qquad (15) \\ = \quad & h(M^0) + \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[\sum_{t=0}^{T-1} \mathbb{E}\left[R(M^t, i_t, j_t, Y_{i_t j_t}) \middle| M^t, \boldsymbol{\alpha}^0 \right] \middle| \boldsymbol{\alpha}^0 \right]. \end{aligned}$$

From (15), it is clear that $R(M^t, i_t, j_t, Y_{i_t j_t})$ is the *stage-wise reward*, which can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result $Y_{i_t j_t}$ at stage t for $t = 0, 1, \ldots, T-1$.

Given the Bayesian MDP in place, we can apply the dynamic programming (DP) algorithm (a.k.a. backward induction) (Puterman, 2005) to compute the optimal policy. Although DP finds the optimal policy, its computation is intractable because:

1. The sophisticated form of the posterior distribution in (7) makes it difficult to evaluate the posterior expected ranking accuracy $\mathbb{E}\left[\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0\right]$ in (9) and the expected transition probabilities in (12) and (13).

2. The maximization problem (9) for solving the optimal posterior expected ranking accuracy is essentially a linear ordering problem (Grötschel et al., 1984), which is NP-hard in general (see Section 3.3 for more details).

3. The size of the state space $\mathcal{S}^t$ grows exponentially in t according to (11), which is known as the curse of dimensionality that prevents us from solving (15) exactly with the standard techniques such value iteration, policy iteration and linear programming.

To address these challenges, we propose an approximated knowledge gradient policy (AKG) in the next Section.

3.3 Approximated Knowledge Gradient Policy

In this section, we describe an approximated policy to solve (10), which is computationally efficient and still provides an inferred ranking with high quality. The proposed approximation policy belongs to the family of *knowledge gradient* (KG) policies (Gupta and Miescke, 1996; Frazier et al., 2008; Powell, 2010; Ryzhov et al., 2012), which is essentially a single-step look-ahead policy. In our problem, the KG policy will sample the next pair of items with the highest expected stage-wise reward in each stage, i.e., choosing the pair (i_t, j_t) such that

$$
\begin{aligned}
(i_t, j_t) \quad &\in \arg\max_{i<j} \quad \mathbb{E}\left[R(M^t, i_t, j_t, Y_{i_t j_t})\middle| M^t, \boldsymbol{\alpha}^0\right] \\
&= \arg\max_{i<j} \quad \mathbb{E}\left[\Pr(Y_{ij}=1)|M^t, \boldsymbol{\alpha}^0\right] R(M^t, i_t, j_t, 1) \\
&\qquad\quad +\mathbb{E}\left[\Pr(Y_{ij}=-1)|M^t, \boldsymbol{\alpha}^0\right] R(M^t, i_t, j_t, -1).
\end{aligned} \tag{16}
$$

Despite its simplicity and wide applicability, the implementation of the KG policy for our problem in (16) is still computationally intractable since we have to evaluate the expected stage-wise reward $\mathbb{E}\left[R(M^t, i_t, j_t, Y_{i_t j_t})\middle| M^t, \boldsymbol{\alpha}^0\right]$, where two main challenges will arise.

First, we have to evaluate the transition probabilities (12) and (13) as well as the ranking accuracy (9), which can be written as

$$
\begin{aligned}
h(M^t) &= \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0\right] \\
&= \max_{\pi} \frac{2\sum_{i\neq j} \mathbb{E}\left[\mathbf{1}_{\{\pi(i)>\pi(j)\}}\mathbf{1}_{\{\theta_i>\theta_j\}}|M^t, \boldsymbol{\alpha}^0\right]}{K(K-1)} \\
&= \max_{\pi} \frac{2\sum_{i\neq j} \mathbf{1}_{\pi(i)>\pi(j)}\Pr\left(\theta_i>\theta_j|M^t, \boldsymbol{\alpha}^0\right)}{K(K-1)}.
\end{aligned} \tag{17}
$$

However, due to the complicated structure of the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ in (7), the expected transition probabilities (12) and (13) and the posterior probability $\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$ in (17) do not admit a closed form so that one needs to use multidimensional numerical integral or sampling techniques to compute their values. Note that for each stage t, we need to evaluate (12), (13) and $\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$ for all $K(K-1)/2$ pairs. When these quantities cannot be easily computed, the overall computational cost will be extremely expensive.

Second, even if the posterior probabilities $\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$ for all pairs are given, the maximization problem (17) with respect to a global ranking π is still very challenging. In fact, this problem is equivalent to the *maximum linear ordering problem (MAX-LOP)* described as follows. Let $G = (V, E, w)$ be a completed directed graph defined on a set V of K nodes, where the edge set E contains the directed arcs between all pairs of nodes and $w(i, j)$ refers to the weight associated with the arc from node i to node j. A tournament D is a sub-graph of G such that, for any pair of nodes i and j, D contains either the arc from i to j or the arc from j to i but not both. The MAX-LOP aims to find an acyclic tournament D with a maximum total weight on its arcs. If we interpret the arc from node i and node j as the preference of node i to node j under a ranking criterion, each acyclic tournament in G corresponds one-to-one to a global ranking of the nodes. Hence, MAX-LOP is equivalent to finding a ranking π such that the total weight $\sum_{\pi(i)>\pi(j)} w(i, j)$ is maximized. In problem (17), the nodes correspond to the K items and the weight $w(i, j) = \Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$. Unfortunately, the MAX-LOP is known to be a NP-hard problem and in fact, APX (approximable)-complete and thus no PTAS (Polynomial Time Approximation Scheme) under P $\neq$ NP (Mishra and Sikdar, 2004).

Given these two challenges, evaluating $\mathbb{E}\left[R(M^t, i_t, j_t, Y_{i_t j_t})\middle| M^t, \boldsymbol{\alpha}^0\right]$ and solving (16) repeatedly at each stage are computationally intractable. To address this problem, we propose an *approximated knowledge gradient* (AKG) policy, which first replaces the stage-wise reward (14) by an approximated but computable reward and then chooses the pair that

maximizes this approximated reward. Our approximation scheme starts with approximating the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ in (7) recursively using a sequence of Dirichlet distributions $\text{Dir}(\boldsymbol{\alpha}^t)$ for $t = 1, 2, \ldots, T$ based on *moment matching*. One key benefit of such an approximation is that, at each stage t, the approximated posterior distribution of $\boldsymbol{\theta}$ is still a Dirichlet distribution so that the NP-hard MAX-LOP problem in (17) will admit a simple solution via a sorting procedure (see Theorem 2).

Although there exist other methods for posterior approximation, these methods cannot be implemented as efficiently as moment matching in our application. For example, some methods such as variational inference (e.g., Beal, 2003; Paisley et al., 2012) minimize the KL-divergence between the exact posterior and the variational posterior, which requires an iterative optimization algorithm as a subroutine. Other methods like Gibbs sampler are computationally expensive in our case because the full conditional distribution does not have a closed form to allow easy sampling. In contrast, the proposed (algorithmic) moment matching admits a closed-form solution for approximating the posterior, which is computationally very efficient, and further provides a Dirichlet distribution as the approximated posterior, which facilitates solving the MAX-LOP. We note that the close-form update is critical for online crowdsourcing applications to reduce the computation time between two stages. In practice, since the crowd workers want to maximize their return in a short period of time, they may quit the current task if we let them wait for too long before we determine the next pair. Finally, we note that, although providing the theoretical guarantee for such an iterative approximation is hard in the Bayesian setup, we empirically show that the resulting AKG policy will generate a final ranking of a high accuracy with the limited budget.

Now we formally introduce the posterior approximation and AKG policy. Suppose $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$ for some parameters $\boldsymbol{\alpha} \in \mathbb{R}^K$. We consider a basic case where only one comparison result Y_{ij} for a pair (i, j) with $i < j$ has been observed. In this case, we approximate the posterior $p(\boldsymbol{\theta}|Y_{ij}, \boldsymbol{\alpha})$ by another Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha}')$ such that

$$\mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')\right] = \mathbb{E}[\theta_k|Y_{ij}, \boldsymbol{\alpha}] \text{ for } k = 1, 2, \ldots, K \tag{18}$$

$$\mathbb{E}\left[\sum_{k=1}^K \theta_k^2|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')\right] = \mathbb{E}\left[\sum_{k=1}^K \theta_k^2|Y_{ij}, \boldsymbol{\alpha}\right]. \tag{19}$$

This system of equations has the following explicit characterization.

Proposition 1 *Suppose $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and Y_{ij} is the only comparison result for $i < j$. Let $\alpha_0 = \sum_{k=1}^K \alpha_k$ and $\alpha_0' = \sum_{k=1}^K \alpha_k'$. The equations (18) and (19) can be represented as*

$$\begin{cases} \frac{\alpha_i'}{\alpha_0'} = \frac{\left(\alpha_i + \frac{1+Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha_j'}{\alpha_0'} = \frac{\left(\alpha_j + \frac{1-Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha_k'}{\alpha_0'} = \frac{\alpha_k}{\alpha_0} \quad \text{for} \quad k \neq i, j \\ \sum_{k=1}^K \frac{\alpha_k'(\alpha_k'+1)}{\alpha_0'(\alpha_0'+1)} = \frac{\left(\alpha_i + \frac{1+Y_{ij}}{2}\right)\left(\alpha_i + \frac{3+Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \\ \qquad + \frac{\left(\alpha_j + \frac{1-Y_{ij}}{2}\right)\left(\alpha_j + \frac{3-Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + \sum_{k\neq i,j} \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}. \end{cases} \tag{20}$$

The proof of Proposition 1 is provided in the Appendix. We denote any $\boldsymbol{\alpha}'$ that satisfies (18) and (19), and thus (20), by

$$\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij}). \tag{21}$$

Note that, given $\boldsymbol{\alpha}$, i, j and Y_{ij}, the right-hand sides of (20) are all constants so that we can solve $\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij})$ in a closed form. In fact, we denote the constants on the right hand sides of (20) as C_i, C_j, C_k (for $k \neq i, j$) and D, respectively. It is easy to show that $\sum_{k=1}^{K} C_k = 1$. The first three equalities in (20) imply that $\alpha'_k = C_k \alpha'_0$ for $k = 1, 2, \ldots, K$ so that the fourth equality in (20) can be represented as $\sum_{k=1}^{K} C_k(C_k\alpha'_0 + 1) = D(\alpha'_0 + 1)$. Solving α'_0 from this equation leads to a closed-form for $\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij})$ as follows

$$\alpha'_0 = \frac{D-1}{\sum_{k=1}^{K} C_k^2 - D} \quad \text{and} \quad \alpha'_k = C_k \alpha'_0 \text{ for } k = 1, 2, \ldots, K. \tag{22}$$

Although the above approximation scheme is established for only one comparison result, it produces a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ which has the same type as the prior distribution $\mathrm{Dir}(\boldsymbol{\alpha})$. Therefore, as more comparison results are generated sequentially, we can apply this approximation scheme iteratively after each comparison result. In particular, given a policy $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\ldots,T-1}$ with $i_t < j_t$ and the comparison results $\{Y_{i_t j_t}\}_{t=0,1,\ldots,T-1}$, we define $\boldsymbol{\alpha}^t$ recursively as

$$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}(\boldsymbol{\alpha}^t, i_t, j_t, Y_{i_t j_t}) \tag{23}$$

for $t = 1, 2, \ldots, T$. By doing so, we approximate the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ by the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ for $t = 1, 2, \ldots, T$.

With $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ approximated by $\mathrm{Dir}(\boldsymbol{\alpha}^t)$, we can mitigate the two challenges mentioned at the beginning of this subsection. First, we can approximate (12) and (13) as

$$\mathbb{E}\left[\Pr(Y_{ij} = 1)\middle|M^t, \boldsymbol{\alpha}^0\right] \approx \mathbb{E}\left[\frac{\theta_i}{\theta_i + \theta_j}\middle|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right] = \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tag{24}$$

$$\mathbb{E}\left[\Pr(Y_{ij} = -1)\middle|M^t, \boldsymbol{\alpha}^0\right] \approx \mathbb{E}\left[\frac{\theta_i}{\theta_i + \theta_j}\middle|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right] = \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \tag{25}$$

and approximate $\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$ in (7) as

$$\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right) \approx \Pr\left(\theta_i > \theta_j|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right) = \int_{\frac{1}{2}}^{1} t^{\alpha_i^t - 1}(1-t)^{\alpha_j^t - 1}\mathrm{d}t = I_{\frac{1}{2}}(\alpha_j^t, \alpha_i^t), \tag{26}$$

where $I_x(a, b) = \dfrac{\mathrm{B}(x; a, b)}{\mathrm{B}(a, b)}$ is known as the *regularized incomplete beta function* with $\mathrm{B}(x; a, b) = \int_0^x \lambda^{a-1}(1-\lambda)^{b-1}\,\mathrm{d}\lambda$ and $\mathrm{B}(a, b) = \int_0^1 \lambda^{a-1}(1-\lambda)^{b-1}\,\mathrm{d}\lambda$. Note that the approximated quantities in (24), (25) and (26) are much easier to compute than the original ones.

More importantly, the approximation (26) simplifies the NP-hard MAX-LOP in (17):

$$\max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0\right] \approx \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right].$$

The right-hand side is still a MAX-LOP but has a special structure so that it can be solved easily by a simple sorting procedure. In particular, the following theorem shows that when $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$, the optimal ranking in (16) can be obtained by sorting the components of $\boldsymbol{\alpha}$.

Theorem 2 *Suppose* $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$. *We have*

$$\begin{aligned}\Pi_{\boldsymbol{\alpha}} &\equiv \{\pi | \pi \textit{ is a ranking of } \{1,2,\ldots,K\} \textit{ such that } \pi(i) > \pi(j) \textit{ only if } \alpha_i \geq \alpha_j \textit{ for all } i,j\} \\ &= \arg\max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta} \sim \textit{Dir}(\boldsymbol{\alpha})\right] \end{aligned} \tag{27}$$

Proof We first show that $\arg\max_\pi \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right] \subset \Pi_{\boldsymbol{\alpha}}$. Suppose $\hat{\pi}$ is the optimal solution of (27) where $\hat{\pi}(j) > \hat{\pi}(i)$ for a pair i and j with $\alpha_i > \alpha_j$. We put all items in a row with their ranks given by $\hat{\pi}$ decreasing from the left to the right and obtain a pattern like

$$X \cdots X j \underbrace{X \cdots X}_{S} i X \cdots X,$$

where X represents some item different from i and j and S represents the set of items ranked between i and j. We will show that the objective value of (27) can be increased by switching the ranks of i and j.

Recall that the expected accuracy of $\hat{\pi}$ can be represented as

$$\begin{aligned}\mathbb{E}\left[\tau(\hat{\pi},\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right] &= \frac{2}{K(K-1)} \sum_{i' \neq j'} \mathbf{1}_{\hat{\pi}(i') > \hat{\pi}(j')} \Pr\left(\theta_{i'} > \theta_{j'} | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right) \\ &= \frac{2}{K(K-1)} \left[I_{\frac{1}{2}}(\alpha_i,\alpha_j) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_j) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_i,\alpha_s) + C \right], \end{aligned} \tag{28}$$

where C is the summation of the remaining terms like $I_{\frac{1}{2}}(\alpha_{i'},\alpha_{j'})$ which have either at least one of i' and j' not in $S \cup \{i,j\}$ or both i' and j' in S.

Note that switching the ranks of i and j does not change the values of the terms in C. In fact, after such a switch, we obtain a new ranking $\hat{\pi}'$ whose objective value in (27) is

$$\mathbb{E}\left[\tau(\hat{\pi}',\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right] = \frac{2}{K(K-1)} \left[I_{\frac{1}{2}}(\alpha_j,\alpha_i) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_j,\alpha_s) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_i) + C \right].$$

Using the fact that $I_{\frac{1}{2}}(a,b)$ is monotonically decreasing in a and monotonically increasing in b and noticing that $\alpha_i > \alpha_j$, we have

$$I_{\frac{1}{2}}(\alpha_j,\alpha_i) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_j,\alpha_s) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_i) > I_{\frac{1}{2}}(\alpha_i,\alpha_j) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_j) + \sum_{s\in S} I_{\frac{1}{2}}(\alpha_i,\alpha_s),$$

which implies $\mathbb{E}\left[\tau(\hat{\pi}',\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right] > \mathbb{E}\left[\tau(\hat{\pi},\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right]$, contradicting with the optimality of $\hat{\pi}$. Hence, we can have $\hat{\pi}(i) > \hat{\pi}(j)$ only if $\alpha_i \geq \alpha_j$, meaning that $\hat{\pi} \in \Pi_{\boldsymbol{\alpha}}$.

We then show $\arg\max_\pi \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right] = \Pi_{\boldsymbol{\alpha}}$ by showing that $\mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right]$ has the same value for any $\pi \in \Pi_{\boldsymbol{\alpha}}$. Suppose $\hat{\pi}$ and $\hat{\pi}'$ both belong to $\Pi_{\boldsymbol{\alpha}}$ and there exists a pair i and j with $i \neq j$ such that $\hat{\pi}(i) > \hat{\pi}(j)$ and $\hat{\pi}'(j) > \hat{\pi}'(i)$. By the definition of $\Pi_{\boldsymbol{\alpha}}$, we have $\alpha_i = \alpha_j$ so that

$$\Pr\left(\theta_i > \theta_j | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right) = I_{\frac{1}{2}}(\alpha_j,\alpha_i) = \frac{1}{2} = I_{\frac{1}{2}}(\alpha_i,\alpha_j) = \Pr\left(\theta_j > \theta_i | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})\right).$$

This means

$$\begin{aligned} & \mathbf{1}_{\hat{\pi}(i)>\hat{\pi}(j)}\Pr\left(\theta_i > \theta_j | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right) + \mathbf{1}_{\hat{\pi}(j)>\hat{\pi}(i)}\Pr\left(\theta_j > \theta_i | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right) \\ = \quad & \mathbf{1}_{\hat{\pi}'(i)>\hat{\pi}'(j)}\Pr\left(\theta_i > \theta_j | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right) + \mathbf{1}_{\hat{\pi}'(j)>\hat{\pi}'(i)}\Pr\left(\theta_j > \theta_i | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right) \end{aligned}$$

for any pair i and j so that $\mathbb{E}\left[\tau(\hat{\pi}, \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right] = \mathbb{E}\left[\tau(\hat{\pi}', \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right]$ by the formulation (28), which completes the proof. ■

Given a parameter vector $\boldsymbol{\alpha}$, we denote any ranking in $\Pi_{\boldsymbol{\alpha}}$ by $\pi_{\boldsymbol{\alpha}}$. Using moment matching and Theorem 2, we can approximate the stage-wise reward $R(M^t, i, j, Y_{ij})$ by

$$\begin{aligned} R(M^t, i, j, Y_{ij}) &= h(M^{t+1}) - h(M^t) \\ &= \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*) | M^{t+1}, \boldsymbol{\alpha}^0\right] - \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0\right] \\ &\approx \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right] \\ &= \mathbb{E}\left[\tau(\pi_{\hat{\boldsymbol{\alpha}}}, \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \mathbb{E}\left[\tau(\pi_{\hat{\boldsymbol{\alpha}}}, \pi^*) | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right] \\ &= \frac{2}{K(K-1)} \left(\sum_{i', j' \,:\, \pi_{\hat{\boldsymbol{\alpha}}}(i') > \pi_{\hat{\boldsymbol{\alpha}}}(j')} I_{\frac{1}{2}}(\hat{\alpha}_{j'}, \hat{\alpha}_{i'}) - \sum_{i', j' \,:\, \pi_{\boldsymbol{\alpha}^t}(i') > \pi_{\boldsymbol{\alpha}^t}(j')} I_{\frac{1}{2}}(\alpha^t_{j'}, \alpha^t_{i'}) \right) \\ &\equiv \tilde{R}(\boldsymbol{\alpha}^t, i, j, Y_{ij}) \end{aligned} \tag{29}$$

where $\hat{\boldsymbol{\alpha}} = \mathbf{MM}(\boldsymbol{\alpha}^t, i, j, Y_{ij})$, the third equality is from Theorem 2 and the fourth equality is due to (26). Putting (16), (24), (25), and (29) together, we can approximate the expected stage-wise reward $\mathbb{E}\left[R(M^t, i, j, Y_{ij}) \middle| M^t, \boldsymbol{\alpha}^0\right]$ as

$$\begin{aligned} & \mathbb{E}\left[R(M^t, i, j, Y_{ij}) \middle| M^t, \boldsymbol{\alpha}^0\right] \\ = \quad & \mathbb{E}\left[\Pr(Y_{ij} = 1) | M^t, \boldsymbol{\alpha}^0\right] R(\boldsymbol{\alpha}^t, i, j, 1) + \mathbb{E}\left[\Pr(Y_{ij} = -1) | M^t, \boldsymbol{\alpha}^0\right] R(M^t, i, j, -1) \\ \approx \quad & \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, -1). \end{aligned} \tag{30}$$

The proposed AKG policy will choose the pair (i_t, j_t) that maximizes the approximated expected stage-wise reward in (30). As a summary, we describe the AKG policy as Algorithm 1.

It is noteworthy that it is easy to implement a *batch version* of Algorithm 1. In fact, the AKG policy in Algorithm 1 is known as an *index policy* where the right-hand side of (31), which calculates the marginal improvement on the ranking accuracy, can be treated as the index for each pair of items. The AKG policy selects the pair with the highest index at each stage. In the batch version, instead of selecting only one pair, one heuristics is to select the top B pairs and distribute to workers simultaneously, where B is a pre-defined batch size. Such a batch implementation can reduce the waiting time of crowd workers and thus accelerate the ranking procedure. Moreover, the AKG policy can be combined with some other batch optimization techniques (Wu and Frazier, 2016) to determine the optimal set of pairs to evaluate next.

Algorithm 1 Approximated Knowledge Gradient Policy with Homogeneous Workers

Initialization: Choose $\boldsymbol{\alpha}^0$ for the prior distribution. Let M^0 be a $K \times K$ all-zero matrix.
For $t = 0, \ldots, T-1$ **do**

1: For each pair (i, j) with $i < j$, compute $\tilde{R}(\boldsymbol{\alpha}^t, i, j, 1)$ and $\tilde{R}(\boldsymbol{\alpha}^t, i, j, -1)$ according to (29).

2: Select (i_t, j_t) such that

$$(i_t, j_t) \in \arg\max_{i<j} \left[\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, -1) \right] \tag{31}$$

and present item i_t and item j_t to a randomly selected worker and receive the comparison result $Y_{i_t j_t}$.

3: According to (21) and (22), compute

$$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}(\boldsymbol{\alpha}^t, i_t, j_t, Y_{i_t j_t}) \tag{32}$$

End For
Return: The aggregated ranking $\pi_{\boldsymbol{\alpha}^T}$ obtained by sorting the components of $\boldsymbol{\alpha}^T$.

4. Crowdsourced Ranking by Heterogeneous Workers

In the previous section, we considered the setting of homogeneous workers, where the comparison results are determined only by the intrinsic latent scores of items but not by the characteristics of workers. However, on crowdsourcing platforms, the quality of the workers varies a lot. Some workers are less reliable or lack of the domain knowledge; some workers are spammers, who either do not actually take a look at the assigned pairs or are robots pretending to be human workers, and thus provide random comparison results in order to quickly receive payment; some workers may be poorly informed (or even malicious), misunderstand the ranking criteria and thus always flip the comparison results. To identify the reliability of a worker, one can assign the same pair of items to multiple workers and hope to identify the unreliable ones whose labels are often different from the majority. However, the abuse of this strategy will result in hiring too many workers and lead to a quick growth of the monetary cost. In order to maximize the accuracy of the final ranking under the limited amount of budget, it is critical to balance the budget spent on estimating the reliability of the workers and learning the true ranking of the items. To formalize such trade-off, we incorporate the reliability of each worker to our previous Bayesian MDP and generalize the AKG policy to the heterogeneity of workers.

4.1 Model Setup

Similar to the previous setting, we assume that each item i has an unknown latent score $\theta_i > 0$ for $i = 1, 2, \ldots, K$ which determines its true ranking π^* (see (1)) and $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^0)$. In the setting of heterogeneous workers, we assume that there are M crowd workers in total, denoted by $w = 1, 2, \ldots, M$. If a pair of items i and j with $i < j$ is presented to the worker

w, we denote the returned comparison result by a random variable Y_{ij}^w such that

$$Y_{ij}^w = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by worker } w \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by worker } w. \end{cases} \tag{33}$$

To model the reliability for workers, we introduce M latent parameters $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_M)^T$ of reliability with $\rho_w \in [0,1]$ for worker w and assume Y_{ij}^w has the following distribution

$$\Pr(Y_{ij}^w = 1) = \rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i + \theta_j} \tag{34}$$

$$\Pr(Y_{ij}^w = -1) = \rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_i}{\theta_i + \theta_j} \tag{35}$$

for $1 \le i < j \le K$ and $w = 1, 2, \ldots, M$. This model can be viewed as a combination of Dawid-Skene model for categorical labeling tasks (Dawid and Skene, 1979; Raykar et al., 2010; Karger et al., 2013a) and Bradley-Terry-Luce (BTL) model, which was first introduced in Chen et al. (2013). Such a mixture of BTL model is flexible and capable of modeling various types of workers. When $\rho_w = 1$, the distribution in (34) and (35) reduces to (3), and we refer to worker w with $\rho_w = 1$ as a "fully reliable" worker[2]. Therefore, the reliability parameter ρ_w can be interpreted as the probability that worker w behaves as a random fully reliable workers in the previous section, namely, the one whose preference over a pair i and j follows a distribution in accordance with the BTL model (3). The worker with ρ_w closer to 1 is considered to be more reliable while a worker with ρ_w closer to 0 tends to be a poorly informed (or malicious) one who intentionally gives answers oppositive to the majority (truth). Also, a worker is known as a spammer if the associated ρ_w is near 0.5 since this worker prefers i or j in any pair i and j with an equal probability regardless of their latent scores.

The reliability of each worker is unknown for the ranking task, which needs to be gradually identified during the comparison process. In the Bayesian framework, since the reliability parameter ρ_w is supported on $[0,1]$, it can be naturally modeled to follow a Beta prior distribution, i.e., $\rho_w \sim \text{Beta}(\mu_w^0, \nu_w^0)$, for $w = 1, 2, \ldots, M$, where $\boldsymbol{\mu}^0 = (\mu_1^0, \mu_2^0, \ldots, \mu_M^0)$ and $\boldsymbol{\nu}^0 = (\nu_1^0, \nu_2^0, \ldots, \nu_M^0)$ are positive parameters.

4.2 Bayesian Decision Process

In this section, we model the sequential decision problem with a finite budget of T in the setting of heterogeneous workers. Since the workers now have different levels of reliability, we can no longer randomly select a worker from the crowd in each stage. Instead, we need to adaptively determine not only which pair of items to be compared but also who should perform this comparison task according to the historical results so that the budget can be gradually shifted towards more reliable workers.

2. We note that the full reliability does not imply that the worker is capable of identifying the latent scores of items and always give the correct comparison result, i.e., preferring the item with a higher latent score. Instead, being fully reliable only means the worker tries her best to provide the preference after a careful consideration, and the inconsistency of comparisons among workers is mainly because the intrinsic ambiguity of the pair of items.

Suppose a pair of items (i_t, j_t) with $i_t < j_t$ is compared by a worker w_t in stage t and the comparison result is $Y_{i_t j_t}^{w_t}$ defined in (33). The historical comparison results up to stage t can be summarized by a $K \times K \times M$ tensor $\mathbf{M}^t$, which is updated iteratively as follows. In particular, at each stage t, we define $\mathbf{\Delta}^t$ to be a sparse $K \times K \times M$ tensor with only non-zero element: if $Y_{i_t j_t}^{w_t} = 1$, $\mathbf{\Delta}^t_{i_t j_t w_t} = 1$ and if $Y_{i_t j_t}^{w_t} = -1$, $\mathbf{\Delta}^t_{j_t i_t w_t} = 1$. Let

$$\mathbf{M}^0 = \mathbf{0}, \quad \mathbf{M}^{t+1} = \mathbf{M}^t + \mathbf{\Delta}^t \quad \text{for } t = 0, 1, \ldots, T-1, \tag{36}$$

where $\mathbf{0}$ is a $K \times K \times M$ all-zero tensor. In contrast to the matrix M^t in (4), each element in the tensor $\mathbf{M}^t$ takes the value either zero or one because each worker is not allowed to compare the same pair more than once. The dynamic budget allocation policy is denoted by $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\ldots,T-1}$ where $(i_t, j_t, w_t) = (i_t(\mathbf{M}^t), j_t(\mathbf{M}^t), w_t(\mathbf{M}^t))$ depends on the previous comparison results through $\mathbf{M}^t$. The posterior distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ in stage t are denoted by $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$, respectively.

Similar to the homogeneous worker setup, we adopt the Kendall's tau (5) to measure the ranking accuracy. At each stage t, we denote the maximum posterior expected ranking accuracy by (with a slight abuse of notation)

$$\begin{aligned} h(\mathbf{M}^t) &\equiv \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \\ &= \max_{\pi} \frac{2\sum_{i\neq j} \mathbf{1}_{\pi(i)>\pi(j)} \Pr\left(\theta_i > \theta_j|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right)}{K(K-1)}. \end{aligned} \tag{37}$$

The maximizer in (37) is the optimal ranking inferred from the historical comparison results up to the stage t. Our goal is to search for the optimal policy $\mathcal{A}$ that maximizes the final expected ranking accuracy $h(\mathbf{M}^T)$, i.e.,

$$\max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}}\left[h(\mathbf{M}^T)|\boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right]. \tag{38}$$

This maximization problem can be further reformulated in a telescopic sum

$$h(\mathbf{M}^0) + \max_{\mathcal{A}} \mathbb{E}\left[\sum_{t=0}^{T-1} \mathbb{E}\left[R(\mathbf{M}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t})\big|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right]\Big|\boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right], \tag{39}$$

where

$$R(\mathbf{M}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) \equiv h(\mathbf{M}^{t+1}) - h(\mathbf{M}^t), \tag{40}$$

is the *stage-wise reward* depending on $\mathbf{M}^t$,i_t,j_t,w_t and $Y_{i_t j_t}^{w_t}$. It can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result at stage t. The *state variable* of the MDP (38) or (39) is the tensor $\mathbf{M}^t$ which evolves according to (36) and the state space at each t is

$$\mathcal{S}^t = \Big\{\mathbf{M} \in \{0,1\}^{K\times K\times M} : \sum_{i,j,w} \mathbf{M}_{ijw} = t\Big\}.$$

The *expected transition probabilities* of MDP (38) are

$$\mathbb{E}\left[\Pr(Y_{ij}^w = 1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] = \mathbb{E}\left[\rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i + \theta_j}\Big|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \quad (41)$$

$$\mathbb{E}\left[\Pr(Y_{ij}^w = -1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] = \mathbb{E}\left[\rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_i}{\theta_i + \theta_j}\Big|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \quad (42)$$

for $i, j = 1, 2, \ldots, K$ and $w = 1, 2, \ldots, M$. So far, we have modeled the sequential budget allocation in the heterogeneous worker setting as a Bayesian MDP. Due to the similar reasons that have been explained in Section 3.2, although the dynamic programming can be directly applied to solve the Bayesian MDP and obtain the optimal policy, it is computationally intractable. In fact, the Bayesian MDP (39) is even more challenging to solve than that for the homogeneous worker setting due to a much larger state space after introducing the reliability of workers. In the next subsection, we will propose a computationally efficient approximated knowledge gradient policy for (39).

4.3 Approximated Knowledge Gradient Policy

To solve the Bayesian MDP (39), we still consider the family of knowledge gradient (KG) policies. In our problem, the KG policy will select the pair of items and the worker that together give the highest expected stage-wise reward. In particular, at the t-stage, the KG policy for (39) will choose the pair (i_t, j_t) and the worker w_t such that

$$\begin{aligned}(i_t, j_t, w_t) &\in \arg\max_{i<j,w} \mathbb{E}\left[R(\mathbf{M}^t, i, j, w, Y_{ij}^w)\big|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \quad (43)\\ &= \arg\max_{i<j,w} \Big\{\mathbb{E}\left[\Pr(Y_{ij}^w = 1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] R(\mathbf{M}^t, i, j, w, 1)\\ &\qquad + \mathbb{E}\left[\Pr(Y_{ij}^w = -1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] R(\mathbf{M}^t, i, j, w, -1)\Big\}.\end{aligned}$$

To implement the KG policy (43), we encounter the same difficulties as when we implemented (16). Specifically, since the posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ are sophisticated and the MAX-LOP problem (37) is NP-hard, we cannot efficiently evaluate the stage-wise reward (40) and the transition probabilities (41) and (42). To obtain a computationally efficient policy, we follow the techniques in Section 3.3 to approximate the posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\rho_w|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ recursively using a sequence of Dirichlet distributions $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ and a sequence of beta distributions $\mathrm{Beta}(\mu_w^t, \nu_w^t)$, respectively, for $w = 1, 2, \ldots, M$ and $t = 1, 2, \ldots, T$. The parameters $\boldsymbol{\alpha}^t(\alpha_1^t, \alpha_2^t, \ldots, \alpha_K^t)$, $\boldsymbol{\mu}^t = (\mu_1^t, \mu_2^t, \ldots, \mu_M^t)$ and $\boldsymbol{\nu}^t = (\nu_1^t, \nu_2^t, \ldots, \nu_M^t)$ will be chosen recursively based on moment matching.

Suppose $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ for some parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ and $\rho_w \sim \mathrm{Beta}(\mu_w, \nu_w)$ for each w with $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_M)$. We consider a basic scenario where only one comparison result Y_{ij}^w from worker w for a pair (i, j) has been observed. We can approximate $p(\boldsymbol{\theta}|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ and $p(\rho_w|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by

a Beta distribution $\text{Beta}(\mu'_w, \nu'_w)$ such that

$$\begin{aligned}
\mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')\right] &= \mathbb{E}[\theta_k|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}] \text{ for } k = 1, 2, \ldots, K \quad (44)\\
\mathbb{E}\left[\sum_{k=1}^K \theta_k^2|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')\right] &= \mathbb{E}\left[\sum_{k=1}^K \theta_k^2|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}\right] \quad (45)\\
\mathbb{E}\left[\rho_w|\rho_w \sim \text{Beta}(\mu'_w, \nu'_w)\right] &= \mathbb{E}\left[\rho_w|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}\right] \quad (46)\\
\mathbb{E}\left[\rho_w^2 + (1-\rho_w)^2|\rho_w \sim \text{Beta}(\mu'_w, \nu'_w)\right] &= \mathbb{E}\left[\rho_w^2 + (1-\rho_w)^2|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}\right]. \quad (47)
\end{aligned}$$

Note that we do not need to approximate $p(\rho_{w'}|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ for $w' \neq w$ since the worker w' has not performed any comparison so that $p(\rho_{w'}|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ is still the prior distribution $\text{Beta}(\mu_{w'}, \nu_{w'})$. This system of equations has the following explicit characterization.

Proposition 3 *Suppose $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and $\rho_w \sim Beta(\mu_w, \nu_w)$ for worker w and Y_{ij}^w is the only comparison result. Let $\alpha_0 = \sum_{k=1}^K \alpha_k$ and $\alpha'_0 = \sum_{k=1}^K \alpha'_k$. The equations (44), (45), (46) and (47) can be represented as*

$$\left\{\begin{aligned}
\tfrac{\alpha'_i}{\alpha'_0} &= \eta_{ijw}\tfrac{(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1-\eta_{ijw})\tfrac{\alpha_i(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}\\
\tfrac{\alpha'_j}{\alpha'_0} &= \eta_{ijw}\tfrac{\alpha_j(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1-\eta_{ijw})\tfrac{(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}\\
\tfrac{\alpha'_k}{\alpha'_0} &= \tfrac{\alpha_k}{\alpha_0} \quad \textit{for } k \neq i, j\\
\textstyle\sum_{k=1}^K \tfrac{\alpha'_k(\alpha'_k+1)}{\alpha'_0(\alpha'_0+1)} &= \eta_{ijw}\tfrac{(\alpha_i+1)(\alpha_i+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + (1-\eta_{ijw})\tfrac{\alpha_i(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}\\
&\quad +\eta_{ijw}\tfrac{\alpha_j(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + (1-\eta_{ijw})\tfrac{(\alpha_j+1)(\alpha_j+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}\\
&\quad +\textstyle\sum_{k\neq i,j}\tfrac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}\\
\tfrac{\mu'_w}{\mu'_w+\nu'_w} &= \eta_{ijw}\tfrac{\mu_w+(1+Y_{ij}^w)/2}{\mu_w+\nu_w+1} + (1-\eta_{ijw})\tfrac{\mu_w+(1-Y_{ij}^w)/2}{\mu_w+\nu_w+1}\\
\tfrac{\mu'_w(\mu'_w+1)+\nu'_w(\nu'_w+1)}{(\mu'_w+\nu'_w)(\mu'_w+\nu'_w+1)} &= \eta_{ijw}\tfrac{(\mu_w+(1+Y_{ij}^w)/2)(\mu_w+(3+Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
&\quad +(1-\eta_{ijw})\tfrac{(\mu_w+(1-Y_{ij}^w)/2)(\mu_w+(3-Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
&\quad +\eta_{ijw}\tfrac{(\nu_w+(1-Y_{ij}^w)/2)(\nu_w+(3-Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
&\quad +(1-\eta_{ijw})\tfrac{(\nu_w+(1+Y_{ij}^w)/2)(\nu_w+(3+Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}.
\end{aligned}\right. \quad (48)$$

where $\eta_{ijw} = \frac{[(1+Y_{ij}^w)\mu_w+(1-Y_{ij}^w)\nu_w]\alpha_i}{[(1+Y_{ij}^w)\mu_w+(1-Y_{ij}^w)\nu_w]\alpha_i+[(1+Y_{ij}^w)\nu_w+(1-Y_{ij}^w)\mu_w]\alpha_j}$.

The proof of Proposition 3 is given in Appendix. We denote any $\boldsymbol{\alpha}'$, μ'_w and ν'_w that satisfy (44), (45), (46) and (47), and thus (48), by

$$\boldsymbol{\alpha}' = \mathbf{MM}_\alpha(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w) \quad \text{and} \quad (\mu'_w, \nu'_w) = \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w). \quad (49)$$

Although the equations in Proposition 3 are more complicated than those in Proposition 1, the right-hand sides of (48) are still constants for any given i, j, w, Y_{ij}^w, $\boldsymbol{\alpha}$, μ_w and ν_w so that both $\boldsymbol{\alpha}' = \mathbf{MM}_\alpha(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w)$ and $(\mu'_w, \nu'_w) = \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w)$ can be solved in a closed form. In fact, we denote the constants on the right-hand sides of (20) as C_i, C_j, C_k (for $k \neq i, j$), D, E and F, respectively. It is easy to see that $\sum_{k=1}^K C_k = 1$. By the

same derivation for (22), we obtain the following closed form for $\boldsymbol{\alpha}' = \mathbf{MM}_{\alpha}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^{w})$

$$\alpha_0' = \frac{D-1}{\sum_{k=1}^{K} C_k^2 - D} \quad \text{and} \quad \alpha_k' = C_k \alpha_0' \text{ for } k = 1, 2, \ldots, K, \tag{50}$$

which takes the same form as (22) but with the constants C_k for $k = 1, 2, \ldots, K$ defined differently (which involve the information of worker w, i.e., μ_w and ν_w). Similarly, solving μ_w' and ν_w' from the last two equations in (48), we obtain the following closed form for $(\mu_w', \nu_w') = \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^{w})$

$$\mu_w' = \frac{(F-1)E}{E^2 + (1-E)^2 - F} \quad \text{and} \quad \nu_w' = \frac{(F-1)(1-E)}{E^2 + (1-E)^2 - F}. \tag{51}$$

Although the approximate scheme above is derived when there is only one comparison result, it generates a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ for $\boldsymbol{\theta}$ and a Beta distribution $\mathrm{Beta}(\mu_w', \nu_w')$ for ρ_w and does not change the Beta distribution $\mathrm{Beta}(\mu_{w'}, \nu_{w'})$ for $w' \neq w$. The fact that the approximated posteriors take the same form as the priors suggests that we can apply this approximation scheme iteratively to approximate $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ for any given policy $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\ldots,T-1}$. In particular, let $\boldsymbol{\alpha}^t$, $\boldsymbol{\mu}^t$ and $\boldsymbol{\nu}^t$ be the sequences of parameters generated recursively as follows

$$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}_{\alpha}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) \tag{52}$$

$$(\mu_w^{t+1}, \nu_w^{t+1}) = \begin{cases} \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) & \text{if } w = w_t \\ (\mu_w^t, \nu_w^t) & \text{if } w \neq w_t \end{cases} \tag{53}$$

for $t = 1, 2, \ldots, T$. The posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ can be approximated by $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ and $\Pi_{w=1,\ldots,M}\mathrm{Beta}(\mu_w^t, \nu_w^t)$, respectively.

Following the same strategy as in (24) and (25), we can approximate (41) and (42) as

$$\begin{aligned} & \mathbb{E}\left[\Pr(Y_{ij}^{w} = 1) \middle| \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \\ \approx\ & \mathbb{E}\left[\rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i + \theta_j} \middle| \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t), \rho_w \sim \mathrm{Beta}(\mu_w^t, \nu_w^t)\right] \\ =\ & \frac{\mu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \end{aligned} \tag{54}$$

and

$$\begin{aligned} & \mathbb{E}\left[\Pr(Y_{ij}^{w} = -1) \middle| \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right] \\ \approx\ & \mathbb{E}\left[\rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_i}{\theta_i + \theta_j} \middle| \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t), \rho_w \sim \mathrm{Beta}(\mu_w^t, \nu_w^t)\right] \\ =\ & \frac{\mu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \end{aligned} \tag{55}$$

and approximate $\Pr\left(\theta_i > \theta_j | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right)$ in (37) as

$$\Pr\left(\theta_i > \theta_j | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right) \approx \Pr\left(\theta_i > \theta_j | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right) = I_{\frac{1}{2}}(\alpha_j^t, \alpha_i^t). \tag{56}$$

The approximation (56) helps to simplify the NP-hard MAX-LOP in (37) as

$$\max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] \approx \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right],$$

where the right-hand side can be solved easily by sorting of the components of $\boldsymbol{\alpha}^t$ according to Theorem 2.

Similar to (29), the stage-wise reward is approximated as

$$\begin{aligned}
R(\mathbf{M}^t,i,j,w,Y_{ij}^w) &= \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\mathbf{M}^{t+1},\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] - \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\mathbf{M}^{t},\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] \\
&\approx \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \max_{\pi} \mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right] \\
&= \frac{2}{K(K-1)}\left(\sum_{\pi_{\hat{\boldsymbol{\alpha}}}(i')>\pi_{\hat{\boldsymbol{\alpha}}}(j')} I_{\frac{1}{2}}(\hat{\alpha}_{j'},\hat{\alpha}_{i'}) - \sum_{\pi_{\boldsymbol{\alpha}^t}(i')>\pi_{\boldsymbol{\alpha}^t}(j')} I_{\frac{1}{2}}(\alpha^t_{j'},\alpha^t_{i'})\right) \\
&\equiv \tilde{R}(\boldsymbol{\alpha}^t,i,j,w,Y_{ij}^w) \qquad (57)
\end{aligned}$$

where $\hat{\boldsymbol{\alpha}} = \mathbf{MM}_{\alpha}(\boldsymbol{\alpha}^t,i,j,w,Y_{ij}^w)$. Putting (54), (55), (56) and (57) together, we can approximate the expected stage-wise reward $\mathbb{E}\left[R(\mathbf{M}^t,i,j,w,Y_{ij}^w)\big|M^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right]$ as

$$\begin{aligned}
&\mathbb{E}\left[R(\mathbf{M}^t,i,j,w,Y_{ij}^w)\big|M^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] \\
=\ &\mathbb{E}\left[\Pr(Y_{ij}^w=1)\big|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] R(\boldsymbol{\alpha}^t,i,j,w,1) \\
&+\mathbb{E}\left[\Pr(Y_{ij}^w=-1)\big|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right] R(\boldsymbol{\alpha}^t,i,j,w,-1) \\
\approx\ &\left(\frac{\mu_w^t}{\mu_w^t+\nu_w^t}\frac{\alpha_i^t}{\alpha_i^t+\alpha_j^t}+\frac{\nu_w^t}{\mu_w^t+\nu_w^t}\frac{\alpha_j^t}{\alpha_i^t+\alpha_j^t}\right)\tilde{R}(\boldsymbol{\alpha}^t,i,j,w,1) \\
&+\left(\frac{\mu_w^t}{\mu_w^t+\nu_w^t}\frac{\alpha_j^t}{\alpha_i^t+\alpha_j^t}+\frac{\nu_w^t}{\mu_w^t+\nu_w^t}\frac{\alpha_i^t}{\alpha_i^t+\alpha_j^t}\right)\tilde{R}(\boldsymbol{\alpha}^t,i,j,w,-1). \qquad (58)
\end{aligned}$$

When the workers have various levels of reliability, our AKG policy will choose the pair (i_t,j_t) and present it to worker w_t so that (58) is maximized. The AKG policy for the setting of heterogeneous workers is formally presented as Algorithm 2. Note that when $\rho_w = 1$ for all w, we do not need to solve (46) and (47) anymore and thus the rest of the problem reduces to the homogeneous setting.

5. Experiment

In this section, we conduct empirical studies using both simulated and real data. We compare the proposed AKG algorithms to some existing methods in terms of ranking accuracy versus different levels of budget as well as computation time. We also show some interesting properties of the proposed AKG policies, e.g., how budget will be allocated over pairs of items with different levels of ambiguity and workers with different levels of reliability. The ranking accuracy is evaluated using the Kendall's tau as defined in (5).

Algorithm 2 Approximated Knowledge Gradient Policy with Heterogeneous Workers

Initialization: Choose $\boldsymbol{\alpha}^0$, $\boldsymbol{\mu}^0$ and $\boldsymbol{\nu}^0$ for the prior distributions.

For $t = 0, \ldots, T-1$ **do**

1: For each pair (i, j) with $i < j$, compute $\tilde{R}(\boldsymbol{\alpha}^t, i, j, w, 1)$ and $\tilde{R}(\boldsymbol{\alpha}^t, i, j, w, -1)$ according to (57).

2: Select (i_t, j_t, w_t) such that

$$(i_t, j_t) \in \underset{i<j,\, w}{\arg\max} \left[\left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, 1) + \left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, -1) \right] \tag{59}$$

and present item i_t and item j_t to worker w_t and receive the comparison result $Y_{i_t j_t}^{w_t}$.

3: According to (49), (50) and (51), compute

$$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}_{\alpha}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) \tag{60}$$

$$(\mu_w^{t+1}, \nu_w^{t+1}) = \begin{cases} \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) & \text{if } w = w_t \\ (\mu_w^t, \nu_w^t) & \text{if } w \neq w_t \end{cases} \tag{61}$$

End For

Return: The aggregated ranking $\pi_{\boldsymbol{\alpha}^T}$ obtained by sorting the components of $\boldsymbol{\alpha}^T$.

5.1 Simulated Study under the Homogeneous Workers Setting

In this section, we assume that all workers are fully reliable and investigate the performance of the AKG policy (Algorithm 1). Two scenarios are designed: 10 items with a total budget of 100, and 100 items with a total budget of 1000. Each scenario consists of 100 independent trials and the average ranking accuracy is reported. For each trial, the latent item scores $\boldsymbol{\theta}$ is sampled uniformly from the simplex in (6), which determines the true ranking π^*. Given $\boldsymbol{\theta}$, the comparison results are generated according to the Bradley-Terry-Luce model (3). We compare several different methods, including the proposed AKG, random sampling (uniformly random sampling), distance-based sampling, adaptive polling (Pfeiffer et al., 2012) and rank centrality with uniform sampling or knowledge gradient sampling (Negahban et al., 2012). The details of the methods are provided as follows.

1. **AKG** (see Algorithm 1): We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex (i.e., $\boldsymbol{\alpha}^0$ is set to be an all-one vector).

2. **Random Sampling**: The random sampling algorithm is similar to Algorithm 1 in terms of the posterior approximation (by moment matching) and rank inference (by sorting the approximated posterior parameters $\boldsymbol{\alpha}^t$) after receiving each label. The only difference is that this algorithm replaces Step 2 of Algorithm 1 by a random sampling policy, which selects (i_t, j_t) randomly at each stage. We also choose the uniform distribution on the simplex as the prior.

3. **Distance-Based Sampling**: This algorithm is also the same as Algorithm 1 in terms of the posterior approximation. However, in the sampling phase, this algorithm simply selects the pair of items (i_t, j_t) with the closest posterior parameters α_i^t and α_j^t. We choose the uniform distribution on the simplex as the prior.

4. **Adaptive Polling**: This is a greedy policy proposed by Pfeiffer et al. (2012), which chooses the pair of items to maximize the KL-divergence between the posterior and prior. The initial $K \times K$ matrix M used in adaptive polling is set to 0 on the diagonal and 0.15 everywhere else.

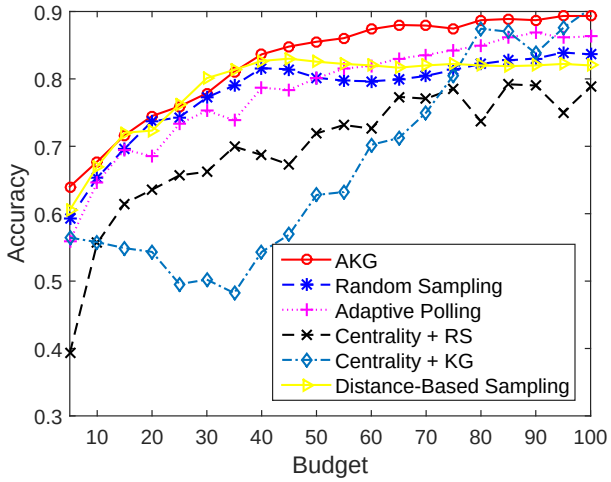
5. **Rank Centrality**: This is a static rank aggregation algorithm recently proposed by Negahban et al. (2012). We combine it with both the random sampling policy and the knowledge gradient policy. Specifically, for **Centrality + RS**, we randomly select a pair of items at each stage and infer the true ranking using rank centrality. For **Centrality + KG**, we select the next pair of items using AKG policy, but estimate the ranking using rank centrality.

It is worthwhile to point out that we are able to compute the optimal policy exactly only up to the 4-item case, which is not interesting from the ranking perspective and thus is left out from the experiment.

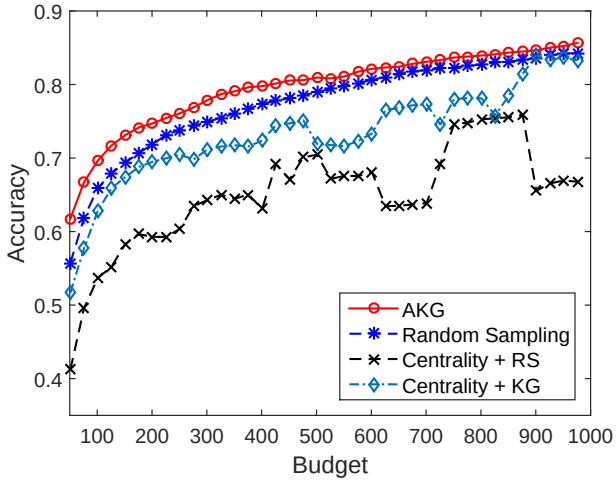
As we can see from Figure 1, the AKG policy has higher accuracy than other methods at all budget levels. Note that the average accuracy of AKG surpasses the level of 70% with only 20 pairs in the case of 10 items. In general, random sampling has similar performance as AKG at the beginning, but eventually AKG will outperform random sampling as it will spend more budget on the ambiguous pairs. This will be verified in the next experiment. Meanwhile, if we combine rank centrality with knowledge gradient sampling, the performance of the algorithm can be boosted significantly. Furthermore, the curves of ranking accuracy of AKG are in general monotonically increasing and have fewer "bumps" than other algorithms. This implies that the sequence of posterior parameters $\boldsymbol{\alpha}^t$ is quite stable when the budget level becomes larger. We also note that due to the high computational cost of adaptive polling, it takes extremely long time when the number of items is 100 and thus we omit its performance in Figure 1b.

It is worthwhile to note that AKG runs significantly faster than the adaptive polling method. It enjoys the advantage of closed-form updating rule during each iteration/stage without using a numerical algorithm as a subroutine, which is a good feature for online applications. In contrast, adaptive polling is much slower because it requires inverting a $K \times K$ matrix for all $O(K^2)$ possible pairs and all possible comparison results in each iteration. Table 1 gives the computation time of a *single iteration* for both AKG and adaptive polling. Note that in the 25-item case, the computation time for adaptive polling of a single iteration has already exceeded 40 minutes. Therefore, we omit to present the computation time of adaptive polling when the number of items is 100 in Table 1 since each iteration/stage would take hours to run.

Next, we study the allocation of labeling budget over pairs of items with different levels of ambiguity when using the AKG policy. Again, we consider two scenarios: $K = 10, T = 100$ and $K = 100, T = 1000$, each with 100 independent trials. We report the averaged labeling frequency of each pair. The results are presented in Figure 2 in the form of heat maps.



(a) 10 Items



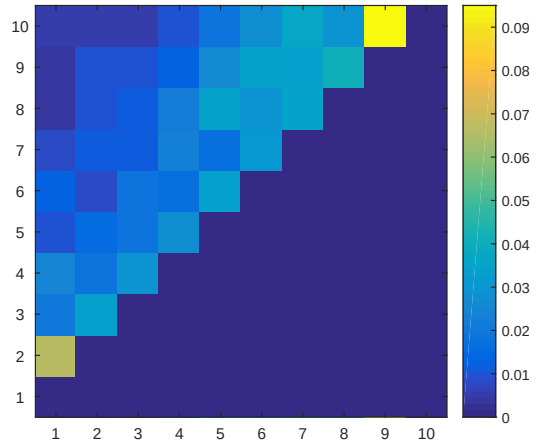
(b) 100 Items

Figure 1: Performance comparison under the homogeneous workers setting. The x-axis is the budget level and y-axis is the averaged ranking accuracy.

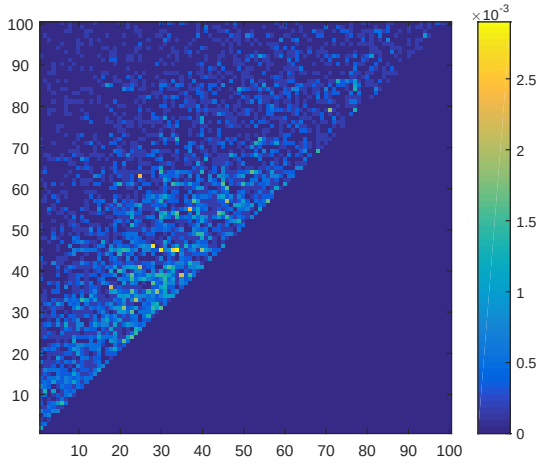
Table 1: Comparison in computation time under the homogeneous workers setting.

No. of Items	AKG	Adaptive Polling
10	0.023 sec	20 sec
25	0.75 sec	42 min
100	22 sec	-

In Figure 2, each small block represents a pair of items. Items are sorted based on their true latent scores, from lowest to highest along both y-axis and x-axis, so that the item pairs along the back-diagonal are more ambiguous than those around the corner. Figure



(a) 10 Items



(b) 100 Items

Figure 2: Heat map of labeling frequency for item pairs with different levels of ambiguity

2 presents the normalized number of comparisons over different pairs in total T stages. It can be seen from Figure 2 that the back-diagonal pairs in general have higher labeling frequency than other pairs. Some adjacent pairs are labeled 10 times more frequently than the distant pairs. To further demonstrate this property, we design a scenario in which out of 10 items, the two worst items and the two best items have very close true scores respectively. Although the main goal of the algorithm is to achieve higher ranking accuracy, we are still curious to see whether our policy can spend the budget on these two pairs. As we can see from Figure 3, it is clear that the algorithm concentrates on the 1-2 pair and the 9-10 pair. This implies that our policy can identify and explore more ambiguous pairs to improve the learning of the true ranks.

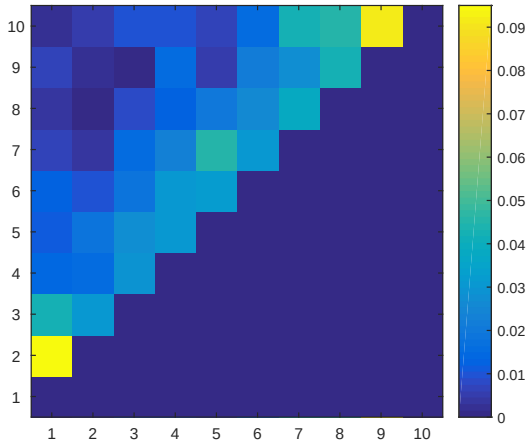


Figure 3: Heat map of labeling frequency for pairs with very close scores

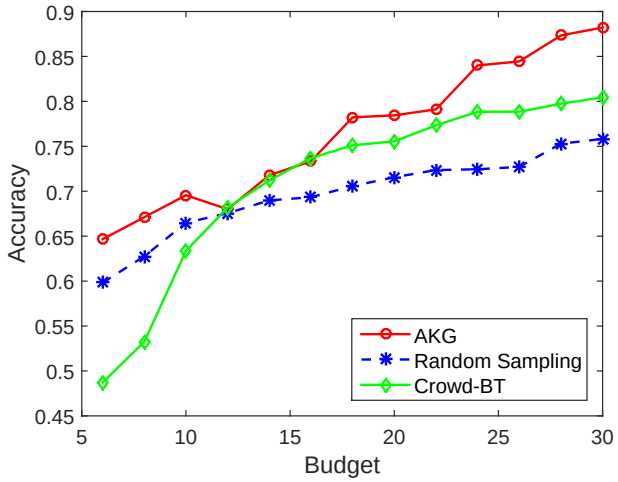
5.2 Simulated Study under the Heterogeneous Workers Setting

In this section we bring worker quality ρ_w into consideration, which is assumed to be drawn from the Beta(4,1) distribution. We choose the Beta(4,1) to generate ρ_w since the average reliability measure of workers in this case is $4/5 = 80\%$. This assumption is in line with the practice in that there are usually more reliable workers than unreliable ones. Similar to the homogeneous worker setting, we consider two scenarios: 10 items with 10 heterogeneous workers ($K = 10$, $M = 10$); 100 items with 50 heterogeneous workers ($K = 100$, $M = 50$) and we note that each worker is allowed to label any pair at most once. We compare the following three methods.

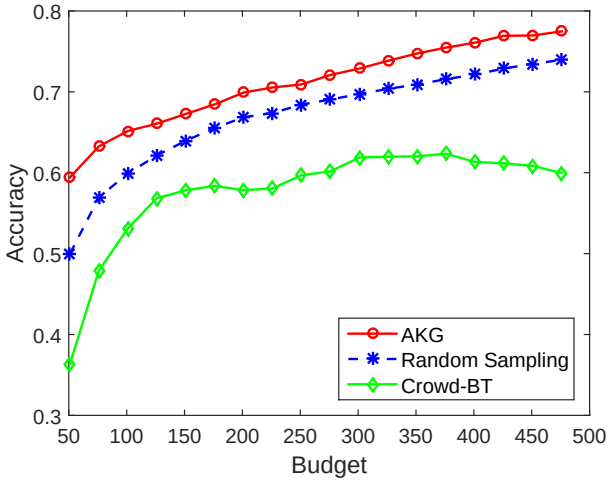
1. **AKG** (see Algorithm 2): We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex (i.e., $\boldsymbol{\alpha}^0$ is set to be an all-one vector) and choose $\mu_w^0 = 4, \nu_w^0 = 1$ for each worker $w = 1, 2, \ldots, M$.

2. **Random Sampling**: It is implemented simply by replacing Step 2 of Algorithm 2 by a random sampling policy, which selects a triplet {item i, item j, worker w} uniformly randomly at each stage. The choices of priors are the same as in AKG. Like the AKG method, the random sampling algorithm also maintains a Dirichlet distribution for the scores of items and a beta distribution for the reliability parameter of each worker using moment matching.

3. **Crowd-BT**: This is an adaptive algorithm recently proposed by Chen et al. (2013), which chooses the triplet {item i, item j, worker w} at each iteration to maximize the information gain. This can be viewed as an extension of the adaptive polling (Pfeiffer et al., 2012) by incorporating the workers' reliability. Unlike adaptive polling which computes the relative entropy for each pair exactly, Crowd-BT uses moment matching to approximate the posterior and hence runs significantly faster than adaptive polling. The parameter γ, which balances the exploitation-exploration trade-off in Chen et al. (2013), is set to 1 in this experiment.



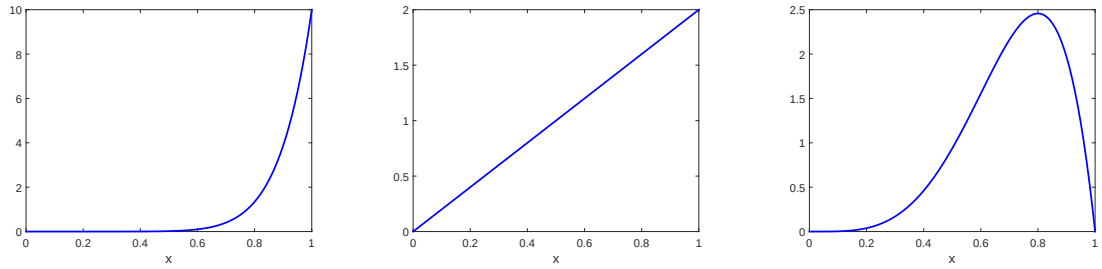
(a) 10 Items



(b) 100 Items

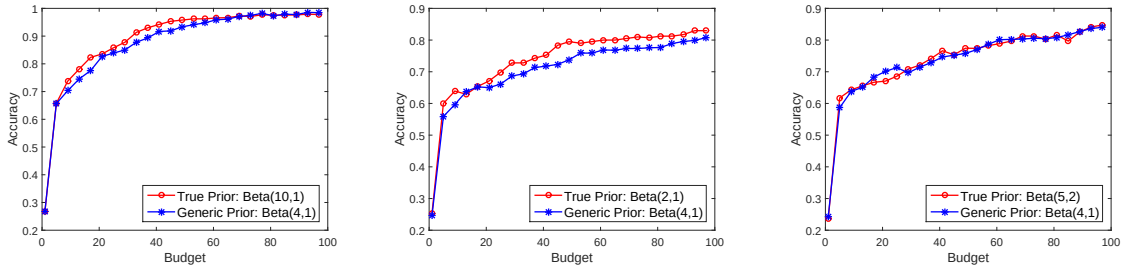
Figure 4: Performance comparison under the heterogeneous workers setting. The x-axis is the budget level and y-axis is the averaged ranking accuracy.

The comparison results are presented in Figure 4, where AKG outperforms the other two methods, especially when the budget level is low. The performance of random sampling is comparable to AKG at the beginning. As we gather more information, AKG can learn the reliability of workers so that the budget will be gradually shifted towards those reliable workers (as shown later in Figure 7). In fact, it can be seen from Figure 4 that the ranking accuracy of AKG increases more quickly than that of other methods. In this experiment, even if there is a small amount of budget (e.g. $T = K$), the AKG policy is still able to achieve reasonably good performance. We notice that in the 100-item case Crowd-BT is beaten by random sampling. The main reason is that when the reliability of workers varies and the pool is large, it is difficult to balance exploration and exploitation for Crowd-BT,



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

Figure 5: Density plots of different Beta distributions for generating ρ_w



(a) Beta$(10, 1)$ Prior (b) Beta$(2, 1)$ Prior (c) Beta$(5, 2)$ Prior

Figure 6: Comparisons between AKG using Beta(4,1) prior and AKG using the true generating distribution as prior.

Table 2: Computation time under the heterogeneous workers setting.

No. of Items	No. of Workers	AKG
10	10	0.038 sec
25	20	0.82 sec
100	50	41 sec

which has already been acknowledged in Chen et al. (2013). Similar to the previous setting, we also give the table of the computation time of a *single iteration* for AKG in Table 2. As we can see from the table, even with another dimension of uncertainty — the reliability of workers, AKG is still quite fast, and thus is suitable for online implementation.

In order to investigate how sensitive the prior for workers' reliability ρ_w is, we generate the workers' true reliability parameters from three different distributions, Beta$(10, 1)$, Beta$(2, 1)$, and Beta$(5, 2)$, and compare the performances of AKG between using the true generating distribution as the prior and using the generic Beta$(4, 1)$ as the prior. The results are plotted in Figure 6. As one can see from Figure 6, using the true generating distribution and generic Beta$(4, 1)$ prior lead to very similar performance in all three cases. Although there are some small differences between the two groups of curves, they are not significant as to the overall performance of the algorithm. This result shows that when there is no

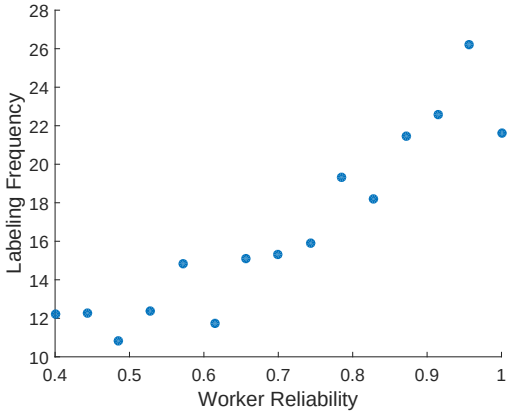


Figure 7: Averaged number of comparisons (a.k.a., labeling frequency) made by workers with different levels of reliability ρ_w.

exact information on the quality of all workers, Beta$(4, 1)$ is a reasonable prior for workers' reliability and the proposed AKG policy is quite robust to the prior distribution in use.

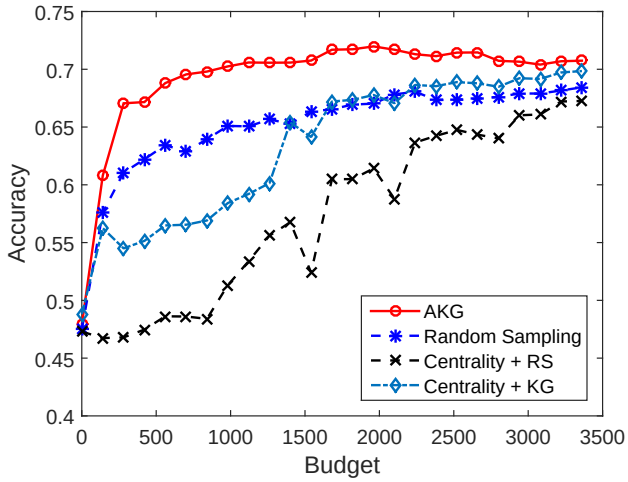
Finally, we investigate whether good workers are indeed assigned more comparison tasks by our AKG policy in the setting of heterogeneous workers. In particular, we consider $K = 10$ items and $M = 15$ workers with the workers' true reliability parameters $\rho_w, w = 1, 2, \ldots, M$ ranging from 0.4 to 1 with an equal space in between. This crowd of workers is fixed and the total budget in each trial $T = 250$. We report the averaged number of pairs assigned to workers with different levels of reliability in Figure 7. As one can see from Figure 7, there is a clear trend that more reliable workers receive more pairs on average.

5.3 Real Data Study

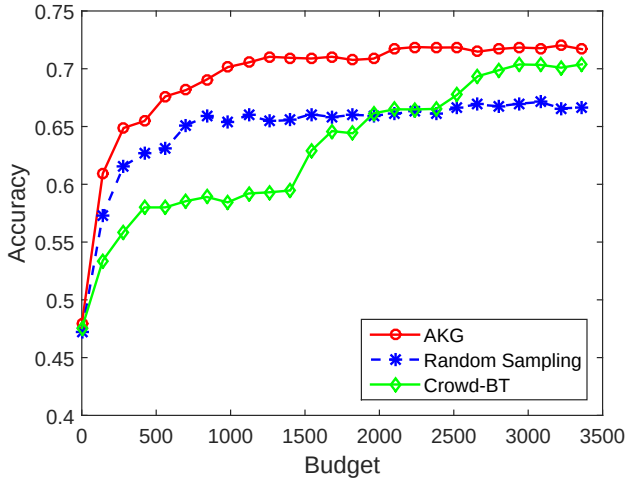
We now apply the proposed AKG policy (Algorithm 2) to a real dataset on reading difficulty levels (Collins-Thompson and Callan, 2004). The dataset comprises $K = 491$ different paragraphs, each assigned an integer-valued true reading difficulty score ranging from $1, 2, \ldots, 12$. Here, a higher score means the paragraph is more difficult to read. A total number of $M = 217$ different workers from Canada and the United States performed the comparison tasks on an online crowdsourcing platform called CrowdFlower[3]. Each worker was presented a pair of paragraphs every time and the worker identified which paragraph is more difficult to read. To overcome the issue of an imbalanced judgemental pool, each worker was allowed to compare at most 40 different pairs. There are 7,898 pairwise comparison results available in this dataset. Using these pairwise labels, we apply the AKG policy to recover the ranking by difficulty of these 491 paragraphs. We note that since the underlying truth is given as a difficulty level (1–12) for each paragraph (denoted by s_i for $i = 1, \ldots, K$) instead of a global ranking, we measure the accuracy of a ranking π as

$$\frac{2}{K(K-1)} \sum_{i \neq j} \mathbf{1}_{\{\pi(i) > \pi(j)\}} \mathbf{1}_{\{s_i \geq s_j\}}.$$

3. http://www.crowdflower.com/



(a) Homogeneous workers (fully reliable)



(b) Heterogeneous workers

Figure 8: Performance comparison on the real dataset

In the above definition of ranking accuracy, when two paragraphs have the same reading difficulty level, any ranking between this pair will be treated as correct. It is also worth noting that, in the knowledge gradient step in (59), it is possible that the selected triplet (i_t, j_t, w_t) does not exist in the dataset (i.e., the worker w_t did not compare i_t and j_t in this data). Hence, in our implementation of AKG, we select the triplet in the dataset that maximizes the right-hand side of (59). We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex. This dataset also comes with a rating for each worker which measures the long-run performance of this worker on CrowdFlower. A higher rating implies a higher reliability of the worker. This dataset shows the averaged workers' rating is above 0.75. Thus, we still use Beta(4,1) as the prior on workers' reliability.

We run experiments in two different settings. The first one assumes that all workers are homogeneous and fully reliable. In this setting, we only need to select the next pair of paragraphs to compare but can randomly choose a worker to perform the comparison task. In this case, four algorithms are implemented (AKG policy (Algorithm 1), random sampling, rank centrality with the random sampling policy, and rank centrality with the knowledge gradient policy) and we report the averaged accuracy over 100 independent trials in Figure 8a to minimize the sampling effect of randomly selecting the next worker. The second experiment incorporates the heterogeneous reliability of workers so that the algorithms have to select both the pair to compare and the worker to perform the comparison task. In this case, three algorithms, AKG policy (Algorithm 2), random sampling and Crowd-BT, are implemented and the result is shown in Figure 8b. As one can see from these two plots, AKG outperforms the other methods in both settings, especially when the amount of budget is relatively low. As the budget level increases, the performance of Crowd-BT and rank centrality will eventually improve and achieve a similar accuracy as AKG.

6. Conclusion

In this paper, we address the dynamic budget allocation problem in crowdsourced ranking. Using the Kendall's tau with respect to the true ranking as the measure of ranking accuracy, we formulate the problem of maximizing expected Kendall's tau by sequential comparisons into a Bayesian Markov decision process. To further address the computational challenges (especially, solving the NP-hard MAX-LOP) involved in the decision process, we propose an approximated knowledge gradient policy, which is not only computationally efficient but also achieves good performance as shown in the experimental sections.

We note that although this paper focuses on the Bradley-Terry-Luce model (Bradley and Terry, 1952; Luce, 1959), it will be interesting to study the dynamic sampling in crowdsourced ranking for other ranking models such as permutation-based models (e.g., Mallows (Mallows, 1957) and CPS (Qin et al., 2010) models) or stochastically transitive models (Fishburn, 1973; Shah et al., 2016b)). Meanwhile, theoretical bounds on posterior approximation errors are difficult to obtain and error propagation does exist during each iteration of the algorithm. In our future analysis we would like to quantify this error. Another interesting future direction is to incorporate the feature information of each item into the probabilistic model of the pairwise comparison results and develop a dynamic sampling policy that can further improve the ranking accuracy via modeling the feature information.

Acknowledgments

Xi Chen would like to acknowledge support for this project from the Google Faculty Research Award.

Appendix

In this section, we provide detailed proofs of some propositions in the paper.

Proof (**of Proposition 1**)

We will only show that (18) and (19) can be represented as (20) when $Y_{ij} = 1$. The proof for $Y_{ij} = -1$ is similar.

It is known that $\mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha'_k}{\alpha'_0}$ and $\mathbb{E}\left[\theta_k^2|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha'_k(\alpha'_k+1)}{\alpha'_0(\alpha'_0+1)}$ for $k = 1, 2, \ldots, K$, which characterize the left-hand sides of (18) and (19).

With elementary calculus, we can show

$$\Pr(Y_{ij} = 1|\boldsymbol{\alpha}) = \int_\Delta \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k - 1} \mathrm{d}\boldsymbol{\theta} = \frac{\alpha_i}{\alpha_i + \alpha_j} \tag{62}$$

so that

$$p(\boldsymbol{\theta}|Y_{ij} = 1, \boldsymbol{\alpha}) = \frac{p(\boldsymbol{\theta}, Y_{ij} = 1|\boldsymbol{\alpha})}{\Pr(Y_{ij} = 1|\boldsymbol{\alpha})} = \frac{\alpha_i + \alpha_j}{\alpha_i} \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\mathrm{B}(\alpha)} \prod_{k=1}^K \theta_k^{\alpha_k - 1}. \tag{63}$$

Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ with $\beta_i = \alpha_i + 1$ and $\beta_k = \alpha_k$ for $k \neq i$. Then, we can show that

$$\begin{aligned}
\mathbb{E}[\theta_i|Y_{ij} = 1, \boldsymbol{\alpha}] &= \frac{\alpha_i + \alpha_j}{\alpha_i} \left[\int_\Delta \frac{\theta_i^2}{\theta_i + \theta_j} \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k - 1} \mathrm{d}\boldsymbol{\theta} \right] \\
&= \frac{\alpha_i + \alpha_j}{\alpha_0} \left[\int_\Delta \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\mathrm{B}(\boldsymbol{\beta})} \prod_{k=1}^K \theta_k^{\beta_k - 1} \mathrm{d}\boldsymbol{\theta} \right] \\
&= \frac{\alpha_i + 1}{\alpha_0} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1},
\end{aligned} \tag{64}$$

where the first and the third equalities are due to (62) and (63) and the second equality is by the definition of $\boldsymbol{\beta}$ and the property $\Gamma(x + 1) = x\Gamma(x)$ of Gamma function. Using a similar argument, we can show that

$$\mathbb{E}[\theta_j|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_j}{\alpha_0} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1} \tag{65}$$

$$\mathbb{E}[\theta_k|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_k}{\alpha_0} \quad \text{for } k \neq i, j \tag{66}$$

$$\mathbb{E}\left[\theta_i^2|Y_{ij} = 1, \boldsymbol{\alpha}\right] = \frac{\alpha_i + 1}{\alpha_0} \frac{\alpha_i + 2}{\alpha_0 + 1} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \tag{67}$$

$$\mathbb{E}\left[\theta_j^2|Y_{ij} = 1, \boldsymbol{\alpha}\right] = \frac{\alpha_j}{\alpha_0} \frac{\alpha_j + 1}{\alpha_0 + 1} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \tag{68}$$

$$\mathbb{E}\left[\theta_k^2|Y_{ij} = 1, \boldsymbol{\alpha}\right] = \frac{\alpha_k}{\alpha_0} \frac{\alpha_k + 1}{\alpha_0 + 1} \quad \text{for } k \neq i, j. \tag{69}$$

Note that, when $Y_{ij} = 1$, the right-hand sides of (18) and (19) can be represented as the right-hand side of (20) using (64)∼(69) ■

Proof (**of Proposition 3**)

We will only show the conclusion when $Y_{ij}^w = 1$. The proof for $Y_{ij}^w = -1$ is similar.

When $Y_{ij}^w = 1$, we have $\eta_{ijw} = \frac{\mu_w \alpha_i}{\mu_w \alpha_i + \nu_w \alpha_j}$. We will first show (44) and (45) can be represented as the first four equations in (48). Since $\mathbb{E}\left[\theta_k | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha'_k}{\alpha'_0}$ and $\mathbb{E}\left[\theta_k^2 | \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha'_k(\alpha'_k+1)}{\alpha'_0(\alpha'_0+1)}$ for $k = 1, 2, \ldots, K$, the left-hand sides of the first four equations in (48) and those of (44) and (45) are identical.

With (62) and some basic properties of the Beta distribution, we can show

$$\begin{aligned} \Pr(Y_{ij}^w = 1 | \boldsymbol{\alpha}, \mu_w, \nu_w) &= \mathbb{E}\left[\rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j} \Big| \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}), \rho_w \sim \mathrm{Beta}(\mu_w, \nu_w)\right] \\ &= \frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j} \end{aligned} \tag{70}$$

so that

$$\begin{aligned} & p(\boldsymbol{\theta}, \rho_w | Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w) \\ = & \frac{\left(\rho_w \frac{\theta_i}{\theta_i+\theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\right) \frac{1}{\mathrm{B}(\boldsymbol{\alpha})\mathrm{B}(\mu_w,\nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k - 1} \rho_w^{\mu_w - 1}(1-\rho_w)^{\nu_w - 1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}. \end{aligned} \tag{71}$$

The equations (70) and (71), together with (64), imply

$$\begin{aligned} & \mathbb{E}[\theta_i | Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w] \\ = & \frac{\int_0^1 \int_\Delta \left(\rho_w \frac{\theta_i^2}{\theta_i+\theta_j} + (1-\rho_w)\frac{\theta_j\theta_i}{\theta_i+\theta_j}\right) \frac{1}{\mathrm{B}(\boldsymbol{\alpha})\mathrm{B}(\mu_w,\nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k-1} \rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1} \mathrm{d}\boldsymbol{\theta}\mathrm{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \frac{\int_\Delta \left(\frac{\mu_w}{\mu_w+\nu_w}\frac{\theta_i^2}{\theta_i+\theta_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\theta_j\theta_i}{\theta_i+\theta_j}\right) \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1} \mathrm{d}\boldsymbol{\theta}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_0}\frac{\alpha_i+1}{\alpha_i+\alpha_j+1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} + \frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_0}\frac{\alpha_j}{\alpha_i+\alpha_j+1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \eta_{ijw}\frac{(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1-\eta_{ijw})\frac{\alpha_i(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}. \end{aligned} \tag{72}$$

Using a similar argument, we can show that

$$\mathbb{E}[\theta_j|Y_{ij}^w=1,\boldsymbol{\alpha},\mu_w,\nu_w]=\eta_{ijw}\frac{\alpha_j(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}+(1-\eta_{ijw})\frac{(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \tag{73}$$

$$\mathbb{E}[\theta_k|Y_{ij}^w=1,\boldsymbol{\alpha},\mu_w,\nu_w]=\frac{\alpha_k}{\alpha_0}\quad \text{for } k\neq i,j \tag{74}$$

$$\mathbb{E}\left[\theta_i^2|Y_{ij}^w=1,\boldsymbol{\alpha},\mu_w,\nu_w\right]=\frac{\eta_{ijw}(\alpha_i+1)(\alpha_i+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}+\frac{(1-\eta_{ijw})\alpha_i(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \tag{75}$$

$$\mathbb{E}\left[\theta_j^2|Y_{ij}^w=1,\boldsymbol{\alpha},\mu_w,\nu_w\right]=\frac{\eta_{ijw}\alpha_j(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}+\frac{(1-\eta_{ijw})(\alpha_j+1)(\alpha_j+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \tag{76}$$

$$\mathbb{E}\left[\theta_k^2|Y_{ij}^w=1,\boldsymbol{\alpha},\mu_w,\nu_w\right]=\frac{\alpha_k}{\alpha_0}\frac{\alpha_k+1}{\alpha_0+1}\quad \text{for } k\neq i,j. \tag{77}$$

In the next, we will show (46) and (47) can be represented as the last two equations in (48). When $Y_{ij}^w=1$, the last two equations in (48) become

$$\begin{cases} \frac{\mu_w'}{\mu_w'+\nu_w'} &= \eta_{ijw}\frac{\mu_w+1}{\mu_w+\nu_w+1}+(1-\eta_{ijw})\frac{\mu_w}{\mu_w+\nu_w+1} \\ \frac{\mu_w'(\mu_w'+1)+\nu_w'(\nu_w'+1)}{(\mu_w'+\nu_w')(\mu_w'+\nu_w'+1)} &= \eta_{ijk}\frac{(\mu_w+1)(\mu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}+(1-\eta_{ijk})\frac{(\mu_w)(\mu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} \\ & \quad +\eta_{ijk}\frac{(\nu_w)(\nu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}+(1-\eta_{ijk})\frac{(\nu_w+1)(\nu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}. \end{cases} \tag{78}$$

It is known that $\mathbb{E}\left[\rho_w|\rho_w\sim\text{Beta}(\mu_w',\nu_w')\right]=\frac{\mu_w'}{\mu_w'+\nu_w'}$, $\mathbb{E}\left[\rho_w^2|\rho_w\sim\text{Beta}(\mu_w',\nu_w')\right]=\frac{\mu_w'(\mu_w'+1)}{(\mu_w'+\nu_w')(\mu_w'+\nu_w'+1)}$ and $\mathbb{E}\left[(1-\rho_w)^2|\rho_w\sim\text{Beta}(\mu_w',\nu_w')\right]=\frac{\nu_w'(\nu_w'+1)}{(\mu_w'+\nu_w')(\mu_w'+\nu_w'+1)}$, indicating that the left-hand sides of (46) and (47) match those of (78).

To characterize the right-hand sides of (46) and (47), we first derive from (71) that

$$\begin{aligned} & \mathbb{E}\left[\rho_w|Y_{ij}^w,\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\nu}\right] \\ = & \frac{\int_0^1\int_\Delta\left(\rho_w^2\frac{\theta_i}{\theta_i+\theta_j}+\rho_w(1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\right)\frac{1}{\text{B}(\boldsymbol{\alpha})\text{B}(\mu_w,\nu_w)}\prod_{k=1}^K\theta_k^{\alpha_k-1}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}\text{d}\boldsymbol{\theta}\text{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \frac{\int_0^1\left(\frac{\alpha_i}{\alpha_i+\alpha_j}\rho_w^2+\frac{\alpha_j}{\alpha_i+\alpha_j}\rho_w(1-\rho_w)\right)\frac{1}{\text{B}(\mu_w,\nu_w)}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}\text{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}+\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\nu_w}{\mu_w+\nu_w+1}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ = & \eta_{ijw}\frac{\mu_w+1}{\mu_w+\nu_w+1}+(1-\eta_{ijw})\frac{\mu_w}{\mu_w+\nu_w+1}. \end{aligned} \tag{79}$$

Following a similar procedure, we can show

$$\begin{aligned}
& \mathbb{E}[\rho_w^2 + (1-\rho_w)^2 | o_i \succ_w o_j, \theta \sim Dir(\alpha), \rho_w \sim Beta(\mu_w, \nu_w)] \\
= & \frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\mu_w+2}{\mu_w+\nu_w+2}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} + \frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\nu_w}{\mu_w+\nu_w+2}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\
& + \frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\nu_w+1}{\mu_w+\nu_w+1}\frac{\mu_w}{\mu_w+\nu_w+2}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} + \frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\nu_w+1}{\mu_w+\nu_w+1}\frac{\nu_w+2}{\mu_w+\nu_w+2}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\
= & \eta_{ijw}\frac{(\mu_w+1)(\mu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} + (1-\eta_{ijw})\frac{(\mu_w)(\mu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} \\
& + \eta_{ijw}\frac{(\nu_w)(\nu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} + (1-\eta_{ijw})\frac{(\nu_w+1)(\nu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}.
\end{aligned} \tag{80}$$

Putting (79) and (80) together, we have shown that the right-hand sides of (78) are exactly the right-hand sides of (70) and (71), which completes the proof. ■

References

S. Acharyya. *Learning to rank in supervised and unsupervised settings using convexity and monotonicity*. PhD thesis, Electrical and Computer Engineering, The University of Texas at Austin, 2013.

N. Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(1):137–164, 2012.

Y. Bachrach, T. Minka, J. Guiver, and T. Graepel. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *International Conference on Machine Learning (ICML)*, 2012.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

R. A. Bradley and M. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324345, 1952.

M. Braverman and E. Mossel. Noisy sorting without resampling. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.

Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (ICML)*, 2007.

X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2013.

X. Chen, Q. Lin, and D. Zhou. Statistical decision making for optimal budget allocation in crowd labelling. *Journal of Machine Learning Research*, 16:1–46, 2015.

K. Collins-Thompson and J. Callan. A language modeling approach to predicting reading difficulty. In *HLT*, 2004.

W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic retrieval based on staged logistic regression. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992.

K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C*, 28:20–28, 1979.

S. Ertekin, H. Hirsh, and C. Rudin. Wisely using a budget for crowdsourcing. Technical report, MIT, 2012.

P. C. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical Psychology*, 10(4):327 – 352, 1973.

P. Frazier. *Knowledge-Gradient Methods for Statistical Learning*. PhD thesis, Princeton University, 2009.

P. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(11):933–969, 2003.

C. Gao and D. Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. arXiv:1310.5764, 2013.

D. Gleich and L. h. Lim. Rank aggregation via nuclear norm minimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.

S. S. Gupta and K. 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.

R. Herbrich, T. Minka, and T. Graepel. Trueskill (TM): a bayesian skill rating system. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning (ICML)*, 2013.

J. Howe. The rise of crowdsourcing. *Wired*, 2006.

K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

E. Kamar, S. Hacker, and E. Horvitz. Combing human and machine intelligence in large-scale crowdsourcing. In *International Conference on Autonomous Agents and Multiagent System*, 2012.

D. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2013a.

D. R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):81–92, 2013b.

M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.

J.-W. Kuo, P.-J. Cheng, and H.-M. Wang. Learning to rank from Bayesian decision inference. In *ACM Conference on Information and Knowledge Management*, 2009.

P. Li, C. Burges, and Q. Wu. Learning to rank using classification and gradient boosting. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

Q. Liu, J. Peng, and A. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

T. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3:225–331, 2009.

T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *Journal of Machine Learning Research*, 15(1):3783–3829, 2014.

R. Luce. *Individual choice behavior: a theoretical analysis*. Wiley, 1959.

C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

S. Mishra and K. Sikdar. On approximability of linear ordering and related np-optimizationproblems ongraphs. *Discrete Applied Mathematics*, 136(2–3):249–269, 2004.

S. Negahban, S. Oh, and D. Shah. Rank centrality: ranking from pair-wise comparisons. arXiv:1209.1688, 2012.

J. Paisley, D. Blei, and M. Jordan. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning (ICML)*, 2012.

T. Pfeiffer, X. A. Gao, Y. Chen, A. Mao, and D. G. Rand. Adaptive polling for information aggregation. In *AAAI Conference on Artificial Intelligence*, 2012.

W. B. Powell. *The Knowledge Gradient for Optimal Learning.* Wiley Encyclopedia for Operations Research and Management Science, 2010.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 2005.

L. Qian, J. Gao, and H. V. Jagadish. Learning user preferences by adaptive pairwise comparison. *Proceedings of the VLDB Endowment*, 8(11):1322–1333, 2015.

T. Qin, X. Geng, and T. Y. Liu. A new probabilistic model for rank aggregation. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

K. Radinsky and N. Ailon. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *ACM International Conference on Web Search and Data Mining*, 2011.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning (ICML)*, 2014.

V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4):1297–1322, 2010.

I. O. Ryzhov, W. B. Powell, and P. I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17, 2016a.

N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning (ICML)*, 2016b.

M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: Optimising non-smooth rank metrics. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2008.

L. L. Thurstone. The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, 21:384–400, 1927.

M. N. Volkovs and R. S. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15(1):1135–1176, 2014.

F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning (ICML)*, 2013.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

J. Wu and P. I. Frazier. The parallel knowledge gradient method for batch bayesian optimization. arXiv:1606.04414, 2016.

J. Xu and H. Li. AdaRank: A boosting algorithm for information retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

J. Yi, R. Jin, S. Jain, and A. K. Jain. Inferring users' preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013.

Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In *Advances in Neural Information Processing Systems (NIPS)*. 2008.