CrossMark

Robust Plackett–Luce model for k-ary crowdsourced preferences

Bo Han[1] [iD] · **Yuangang Pan[1]** · **Ivor W. Tsang[1]**

Abstract The aggregation of k-ary preferences is an emerging ranking problem, which plays an important role in several aspects of our daily life, such as ordinal peer grading and online product recommendation. At the same time, crowdsourcing has become a trendy way to provide a plethora of k-ary preferences for this ranking problem, due to convenient platforms and low costs. However, k-ary preferences from crowdsourced workers are often noisy, which inevitably degenerates the performance of traditional aggregation models. To address this challenge, in this paper, we present a RObust PlAckett–Luce (ROPAL) model. Specifically, to ensure the robustness, ROPAL integrates the Plackett–Luce model with a denoising vector. Based on the Kendall-tau distance, this vector corrects k-ary crowdsourced preferences with a certain probability. In addition, we propose an online Bayesian inference to make ROPAL scalable to large-scale preferences. We conduct comprehensive experiments on simulated and real-world datasets. Empirical results on "massive synthetic" and "real-world" datasets show that ROPAL with online Bayesian inference achieves substantial improvements in robustness and noisy worker detection over current approaches.

Keywords Ranking · k-Ary crowdsourced preferences · Robust Plackett–Luce model · Online Bayesian inference

---

✉ Ivor W. Tsang
Ivor.Tsang@uts.edu.au

Bo Han
Bo.Han@student.uts.edu.au

Yuangang Pan
Yuangang.Pan@student.uts.edu.au

[1] Centre for Artificial Intelligence (CAI), University of Technology Sydney, Sydney, Australia

1 Introduction

Many daily applications, such as meta-search, online product recommendation and ordinal peer grading, face the aggregation of k-ary preferences (Liu 2009; Volkovs et al. 2012; Raman and Joachims 2014). Namely, multiple k-ary preferences over objects are aggregated as a L-ary preference ($k \ll L$). Meanwhile, crowdsourcing has become a trendy way to provide a plethora of k-ary preferences for this aggregation problem (Venanzi et al. 2014) due to two reasons: first, it is convenient to collect preferences on crowdsourcing platforms; second, the cost of crowdsourced preferences is very low.

However, k-ary preferences from crowdsourced workers are often noisy (Sheng et al. 2008; Snow et al. 2008; Ok et al. 2016), which adversely affects the robustness of traditional aggregation models (i.e., permutation-based or score-based models). Although the state-of-the-art CrowdBT (Chen et al. 2013) aggregates pairwise crowdsourced preferences effectively, it cannot model the characteristics of k-ary preferences seamlessly. Specifically, CrowdBT splits each k-ary preference into multiple pairwise preferences before the aggregation, which may lead to cyclical inconsistency ($a > b$ and $b > a$ or $a > b$, $b > c$ and $c > a$).[1] In addition, traditional inferences make aggregation models impractical for large-scale preferences. For example, when the number of preferences is large (e.g., 500,000 preferences), the Gibbs sampling approach is too slow to infer the permutation-based model easily (Lu and Boutilier 2011). Therefore, for large-scale k-ary crowdsourced preferences, can we build a robust aggregation model with an efficient inference under the single ground-truth assumption?[2]

In this paper, we present a RObust PlAckett–Luce (ROPAL) model coupled with an efficient online inference, which is tailor-made for large-scale k-ary crowdsourced preferences. To ensure the robustness, ROPAL integrates the traditional Plackett–Luce model with a denoising vector. Based on the Kendall-tau distance, this vector corrects k-ary crowdsourced preferences with a certain probability. Specifically, each denoising vector is actually a distribution that matches with an unique worker quality. For an expert, the first entry of her denoising vector, denoting the probability that the preferences accord with the ground truth, is close to 1. That is to say, we trust all her preferences. For an amateur, since she only makes mistakes on comparable objects during the annotation process, the first two entries of her denoising vector dominate the distribution (the denoising vector). Therefore, ROPAL corrects her preferences with the probability revealed by the first or second entry of her denoising vector. For a spammer, as she annotates the preferences carelessly, each entry of her denoising vector is nearly $\frac{1}{C_k^2+1}$. Therefore, the likelihood of the preferences from each spammer is a constant, which means ROPAL discards all her preferences. For a malicious worker, ROPAL learns her denoising vector by exploring the pattern she corrupts the preferences. For example, if a malicious worker annotates the preferences in exactly an opposite order, the last entry of her denoising vector will be close to 1. To sum up, ROPAL reveals the ground truth of her preferences according to each entry of her estimated denoising vector.

---

[1] There are two kinds of cyclical inconsistencies: first, $a > b$ and $b > a$ yielded between two k-ary preferences from the same worker; second, $a > b$, $b > c$ and $c > a$ yielded within single k-ary preference. The first type is caused by the inconsistent annotation of a worker when she annotates different preferences. The second type is actually the invalid preferences yielded from the possible inconsistent combination ($a > b$ and $b > a$ or $a > b$, $b > c$ and $c > a$) of sub-preferences, where these sub-preferences are the partial preferences split from the original k-ary preference. The first type of cyclical inconsistencies arises during the process of data collection, which is hard for model designers to control. Usually, we assume the crowdsourced preferences annotated by the same worker are consistent to avoid the first type of cyclical inconsistencies. In this paper, we mainly deal with the second type of cyclical inconsistencies.

[2] In this paper, we assume that the full preference has the single ground truth ranking shared by all workers, which is a fundamental assumption in many preference aggregation models.

To ensure the scalability of ROPAL, we adopt the online learning strategy for the Bayesian extension of ROPAL and resort to moment-matching methods to update its hyperparameters incrementally. All hyperparameters are updated by closed-form solutions, which makes ROPAL scalable to large-scale preferences. Our contributions are as follows.

- We present a RObust PlAckett–Luce (ROPAL) model to aggregate k-ary preferences directly, which avoids the issue of cyclical inconsistency.
- We introduce a denoising vector in ROPAL to deal with crowdsourced (noisy) preferences and detect noisy workers simultaneously. We design this vector based on Kendall-tau distance, which avoids the curse of permutation-space partition.
- We propose an online Bayesian inference, which makes ROPAL scalable to large-scale preferences.
- We conduct comprehensive experiments on "large-scale simulated" and "real-world" datasets. Empirical results demonstrate that ROPAL with the online Bayesian inference achieves substantial improvements in robustness and noisy worker detection over current approaches.

The remainder of this paper is organized as follows. In Sect. 2, the related work is presented. Section 3 is the core of this paper. Initially, we introduce the k-ary crowdsourced preferences setting. Then, for this setting, we propose a RObust PlAckett–Luce (ROPAL) model, which integrates the Plackett–Luce model (for k-ary preferences) with a denoising vector (for robustness). Lastly, we present an online Bayesian inference to make ROPAL scalable to large-scale preferences. In Sect. 4, we provide the experiment setup and empirical results related to simulated and real-world experiments. The conclusive remarks and future works are given in Sect. 5.

2 Related work

Our work is partially related to a novel ranking problem, namely, aggregating multiple k-ary preferences into a consensus L-ary preference, where $k \ll L$. This ranking problem has a wide range of practical applications. For example, in ordinal peer grading (Raman and Joachims 2014), each grader returns an ranking over her partial assignments, and her ranking is regarded as an ordinal feedback. All ordinal feedbacks from different graders are aggregated to yield the final grades over full assignments. Meta-search with the goal of merging multiple partial rankings from different sources to output a single full ranking, is another example (Volkovs et al. 2012). For online product recommendation (Lv et al. 2011), the system in eBay or Amazon recommends the most related products to each customer according to her browsing history. Namely, multiple ordinal records are aggregated as a single consensus recommendation list. Our work is also related to crowdsourcing (Deng et al. 2016; Krishna et al. 2017), which provides sufficient preferences for this ranking problem.

When aggregating k-ary crowdsourced preferences, two crucial challenges should be noted: preference noise and computational cost. To be specific, in crowdsourcing, online workers can be roughly categorized as domain experts and amateurs according to their performance. Due to the low reward, most crowdsourced preferences are from amateurs rather than domain experts. Therefore, crowdsourced preferences are often noisy (Kazai et al. 2011; Vuurens et al. 2011), which adversely affects the robustness of conventional aggregation models (i.e, score-based and permutation-based). In addition, traditional inferences, such as expectation-maximization (EM) and Gibbs sampling approaches, need multiple iterations to converge (Bishop 2006). Therefore, traditional inferences render aggregation models

impractical for large-scale datasets in the real world. In the following, we list representative examples to explain these two challenges.

Among the score-based models, the Plackett–Luce model (Luce 1959; Plackett 1975) is suitable to aggregate k-ary preferences directly. However, k-ary crowdsourced preferences are full of noise, which very likely results in the remarkable degeneration of robustness in the Plackett–Luce model. Although CrowdBT (Chen et al. 2013) extends the Bradley and Terry (1952) model to aggregate pairwise crowdsourced preferences robustly, it is unable to deal with k-ary crowdsourced preferences directly. Specifically, CrowdBT splits each k-ary preference into multiple pairwise preferences before the aggregation, which leads to the issue of cyclical inconsistency. Among permutation-based models, the Mallows model (1957) tends to model the distribution of a full L-ary preference directly, but the exact inference for this distribution tends to be time-consuming or even intractable. Recent work has extended the Mallows model to define distributions of multiple k-ary preferences (Lu and Boutilier 2011, 2014). However, when the number of multiple k-ary preferences is large, the sampling approach leveraged by permutation-based models is typically slow, which makes these models impractical for large-scale preferences.

To sum up, in this paper, our target consists of two aspects. The first aspect is to propose a robust ranking model that aggregates k-ary crowdsourced preferences effectively. The second aspect is to present an efficient inference that makes the proposed model scalable to large-scale preferences.

3 Aggregating with k-ary crowdsourced preferences

We begin this section with k-ary crowdsourced preferences setting. Then, we introduce the Plackett–Luce (PL) model and the crowdsourced Bradley–Terry (CrowdBT) model, respectively. Meanwhile, we discuss their deficiencies on the aggregation of k-ary crowdsourced preferences. Furthermore, we present our RObust PlAckett–Luce (ROPAL) model, which avoids the above deficiencies. Specifically, ROPAL introduces a denoising vector to effectively deal with crowdsourced (noisy) preferences and detect noisy workers. Lastly, we propose an online Bayesian inference to make ROPAL scalable to large-scale preferences.

3.1 k-Ary crowdsourced preferences setting

In Table 1, we first illustrate the common mathematical notations that are used later. Then we compare traditional crowdsourced preference setting with k-ary crowdsourced preferences setting in Fig. 1. Specifically, in traditional preference setting, a finite set of objects $\Omega = \{o_1, o_2, \ldots, o_L\}$ is assigned to W workers. Each worker w ranks the whole set Ω independently to yield only one preference O_w[3] (the full preference in the upper panel of Fig. 1) according to a certain criterion, such as personal preferences or attitudes.

However, the total number of objects L tends to be very large in many real-world applications (Shah et al. 2013; Luaces et al. 2015). Therefore, the annotation process cannot be carried out completely every time. Normally, worker w only annotates her most-confident l ($< L$) objects and leaves the remaining $L - l$ objects undefined (the partial preference O_w in the upper panel of Fig. 1). Numerous researchers are working on this (partial) ranking aggregation problem (Dwork et al. 2001; Schalekamp and Zuylen 2009; Kolde et al. 2012).

---

[3] In our k-ary crowdsourced preferences setting, each worker w provides N_w preferences in total. We use $O_{n,w}$ to show the difference, which denotes the nth preference annotated by worker w.

Table 1 Common mathematical notations

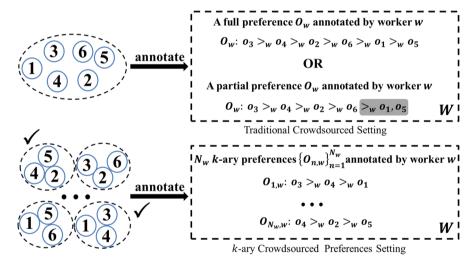| Notation	Explanation
Ω	Set of objects, $\Omega = \{o_1, o_2, \ldots, o_L\}$
ξ	Subset of fixed size k of Ω, $\xi \subseteq \Omega$
L	Total number of objects
W	Total number of workers
D	Collection of all k-ary preferences
D_w	Collection of k-ary preferences annotated by worker w
N_w	$|D_w|$, number of k-ary preferences annotated by worker w
$O_{n,w}$	The nth preference annotated by worker w
S_j	Score for object j
$\boldsymbol{\eta}_w$	A vector, denoising vector for worker w used in ROPAL
η_w	A number, worker quality for worker w used in CrowdBT
$N(\mu_j, \sigma_j^2)$	Prior Normal distribution of S_j
$Dir(\boldsymbol{\alpha}_w)$	Prior Dirichlet distribution of $\boldsymbol{\eta}_w$

Fig. 1 Comparison between two settings. Upper panel: In traditional crowdsourced setting, worker w annotates the whole set to yield a full or partial preference. The gray area denotes 2 remaining objects (o_1, o_5), which are assumed to be ranked lower. Lower panel: In k-ary crowdsourced preferences setting, worker w annotates multiple subsets to yield several partial preferences. $O_{n,w}$ means the nth k-ary preference annotated by worker w, and worker w provides N_w partial preferences in total. Note that: (1) the notation W in the corner means W workers complete the annotation process independently and repeatedly; (2) the subset with "$\sqrt{}$" is selected by worker w

Most of their work assumes the remaining $L - l$ objects are tacitly assumed to be ranked lower (Guiver and Snelson 2009; Mollica and Tardella 2017). Therefore, the partial preferences can be modelled by the classical Plackett–Luce models. However, the traditional (partial) ranking aggregation problem still faces the following problems: first, it is unrealistic to assume the

rare objects are less important, especially for the large L ranking problem; second, lots of workers tend to annotate their most-confident l ($< L$) objects to ensure their accuracy, which will undoubtedly results in a lack of information contained in the collected preferences.

Ordinal peer grading, where students assess each other, is a promising approach to tackle the large-scale evaluation problem (Freeman and Parks 2010; Shah et al. 2013; Luaces et al. 2015; Kulkarni et al. 2015; Prpić et al. 2015). Motivated by ordinal peer grading, we propose our k-ary crowdsourced preferences setting, which overcomes the limitation when L is large (Alfaro and Shavlovsky 2014). Specifically, a large set of objects Ω is broken into C_L^k distinct tasks $\{\xi_i\}_{i=1}^{C_L^k}$ with $|\xi_i| \equiv k$, where each task ξ_i is a subset of Ω. Then N_w of C_L^k tasks are assigned randomly to worker w to annotate, where $\forall w \in \{1, 2, \ldots, W\}$ (the lower panel of Fig. 1). Taking into account the task difficulty, in real applications, the size of each task should be much smaller than the total number of objects (e.g., $|\xi_i| \equiv k < 7 \ll |\Omega| = L$). In the following, we aim to aggregate the k-ary preferences annotated by W workers into a full preference over all L objects.

3.2 Deficiency of classical models

Given a k-ary preference $O: o_1 > o_2 > \cdots > o_k$,[4] it is suitable to be modeled by Classical Plackett–Luce model f_{PL} as follows:

$$f_{PL}(O|\mathbf{S}) = \prod_{i=1}^{k-1} \frac{e^{S_{o_i}}}{\sum_{t=i}^{k} e^{S_{o_t}}}, \tag{1}$$

where $\mathbf{S} = \{S_1, S_2, \ldots, S_L\}$ denotes implicit scores associated with each object. Based on Eq. (1), for the collection of all k-ary crowdsourced preferences D, the likelihood function $\Pr(D|\mathbf{S})$ can be represented as:

$$\Pr(D|\mathbf{S}) = \prod_{w=1}^{W} \Pr(D_w|\mathbf{S}) = \prod_{w=1}^{W} \prod_{n=1}^{N_w} f_{PL}(O_{n,w}|\mathbf{S}), \tag{2}$$

where D_w is the collection of multiple k-ary preferences annotated by worker w. $O_{n,w}$ is the nth preference annotated by worker w. More intuitive explanations can be found in the lower panel of Fig. 1. Note that, the traditional Plackett–Luce model [Eq. (2)] does not consider worker quality. Therefore, this model treats preferences from different workers equally. However, when preferences are derived from crowdsourcing platforms, these preferences annotated by amateurs are often noisy, which inevitably degrades the aggregation performance of the traditional Plackett–Luce model.

Chen et al. (2013) proposed a pairwise model (CrowdBT) to aggregate pairwise crowdsourced references robustly, since CrowdBT considers the worker quality. However, the pairwise model handles k-ary preferences clumsily for two reasons. First, pairwise models cannot model the characteristics of k-ary preferences seamlessly. Second, pairwise models splits each k-ary preference into multiple pairwise preferences before the aggregation, which may lead to the second type of cyclical inconsistency. Assume that worker w has a unique worker quality η_w, namely the probability that the pairwise preference $o_a >_w o_b$ annotated by worker w accords with the true preference $o_a > o_b$: $\eta_w = \Pr(o_a >_w o_b \mid o_a > o_b)$. Therefore, according to the law of total probability, we have:

---

[4] O denotes a general preference. O_w denotes the single preference annotated by worker w in traditional crowdsourced setting, while $O_{n,w}$ denotes the nth k-ary preference annotated by worker w in k-ary crowdsourced setting.

$$\Pr(o_a >_w o_b) = \Pr(o_a >_w o_b \mid o_a > o_b)\Pr(o_a > o_b)$$
$$+ \Pr(o_a >_w o_b \mid o_a < o_b)\Pr(o_a < o_b)$$
$$= \eta_w \Pr(o_a > o_b) + (1 - \eta_w)\Pr(o_a < o_b).$$

Assume that we have a k-ary subset $(o_1, o_2, \ldots, o_k)$, which is annotated as $O_{n,w}: o_1 >_w o_2 >_w \cdots >_w o_k$ by worker w. A pairwise model will first split this k-ary preference into C_k^2 pairwise preferences $o_i >_w o_j$, where $(i, j) \subset \{1, 2, \ldots, k\}$ and $i \neq j$. Then, the model aggregates these pairwise preferences using worker quality η_w. Here, we take $k = 3$ as an example. The 3-ary preference can be modeled as:

$$\Pr(o_a >_w o_b >_w o_c) = \Pr(o_a >_w o_b)\Pr(o_a >_w o_c)\Pr(o_b >_w o_c)$$
$$= \prod_{(i,j)} \left\{ \eta_w \Pr(o_i > o_j) + (1 - \eta_w)\Pr(o_i < o_j) \right\},$$

where $(i, j) \in \{(a, b), (a, c), (b, c)\}$. In this case, we identify two cyclical inconsistencies, namely $\Pr(o_a > o_b)\Pr(o_c > o_a)\Pr(o_b > o_c)$ and $\Pr(o_b > o_a)\Pr(o_a > o_c)\Pr(o_c > o_b)$. This leads to the deficiency of pairwise aggregation models. Specifically, if the worker quality η_w is 0.8, the coefficients $\eta_w^2(1 - \eta_w)$ and $\eta_w(1 - \eta_w)^2$ of these two cyclical inconsistencies are 12.8 and 3.2%, respectively. According to Theorems 1 and 2, the cyclical inconsistency becomes more serious with the increase of k. In Theorem 1, we first prove that the number of cyclical inconsistencies increases with k.

Theorem 1 *Assume that k-ary preferences are aggregated by a pairwise model, then the number of cyclical inconsistencies caused by a k-ary preference is equal to $2^{C_k^2} - k!$.*

Remark 1 The number of cyclical inconsistencies caused by a 6-ary preference is 32,048.

However, Theorem 1 does not imply that the probability of occurrence of cyclical inconsistencies increases with k in reality. Thus, in Theorem 2, we should prove that the proportion of the probability that cyclical inconsistencies happen in a k-ary preference increases along with k.

Theorem 2 *Under the same assumption of Theorem 1, the likelihood $\Pr(o_1 >_w o_2 >_w \cdots >_w o_k)$ can be decomposed into $2^{C_k^2}$ distinct combinations of pairwise preferences. Let $\Pr(o_1 >_w o_2 >_w \cdots >_w o_k) = \Pr_k(\mathbb{I}) + \Pr_k(\mathbb{C})$, where $\Pr_k(\mathbb{I})$ is the probability of occurrence of inconsistent combinations, and $\Pr_k(\mathbb{C})$ is the probability of occurrence of consistent combinations. Then $\frac{\Pr_{k+1}(\mathbb{I})}{\Pr(o_1 >_w o_2 >_w \cdots >_w o_{k+1})} > \frac{\Pr_k(\mathbb{I})}{\Pr(o_1 >_w o_2 >_w \cdots >_w o_k)}$, which means the proportion of the probability that the cyclical inconsistencies happen in a k-ary preference increases along with k.*

The proofs related to Theorems 1 and 2 can be found in the "Appendix". The above analysis motivates us to integrate the Plackett–Luce model with the worker quality, which not only avoids the second type of cyclical inconsistencies in nature, but also improves the robustness for k-ary crowdsourced preferences setting.

3.3 Robust Plackett–Luce model

A k-ary preference $O_{n,w}$ has $k!$ distinct permutations $\{g_i(O_{n,w})\}_{i=1}^{k!}$, which constitutes a finite partition of the entire sample space (see Fig. 2). Each permutation $g_i(O_{n,w})$ corresponds to one potential ground truth of $O_{n,w}$. Inspired by CrowdBT, it is intuitive to leverage the original partition (see the left panel of Fig. 2) directly, and assign a denoising vector $\boldsymbol{\eta_w}$ for each

worker to define the $k!$ conditional probabilities of $O_{n,w}$ given permutation $g_i(O_{n,w})$ $\forall i \in \{1, 2, \ldots, k!\}$. To be specific, let $\eta_w^i = \Pr\left(O_{n,w} \mid g_i(O_{n,w})\right)$ $\forall i \in \{1, 2, \ldots, k!\}$, denoting the conditional probability of the observation $O_{n,w}$ given permutation $g_i(O_{n,w})$ being the ground truth. Therefore, according to the law of total probability, we have:

$$
\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}) = \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) = \sum_{i=1}^{k!} \Pr\left(O_{n,w} \mid g_i(O_{n,w})\right) f_{PL}\left(g_i(O_{n,w}) \mid \mathbf{S}\right)
$$

$$
= \sum_{i=1}^{k!} \eta_w^i f_{PL}\left(g_i(O_{n,w}) \mid \mathbf{S}\right), \tag{3}
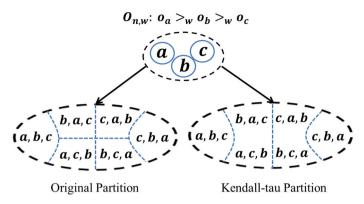$$

where $\boldsymbol{\eta}$ denotes $\{\boldsymbol{\eta}_w\}_{w=1}^W$, and $\boldsymbol{\eta}_w = [\eta_w^1, \eta_w^2, \ldots, \eta_w^{k!}]$ represents the denoising vector for worker w. $\mathbf{S} = \{S_1, S_2, \ldots, S_L\}$ denotes implicit scores associated with all objects. The first equation holds because the preference $O_{n,w}$ only depends on worker w's denoising vector $\boldsymbol{\eta}_w$. We use f_{PL} to represent the Plackett–Luce model [Eq. (1)].

However, the original partition used in Eq. (3) leads to the *curse of permutation-space (p-space) partition*. To be specific, since the number of permutations $k!$ increases dramatically with the length of preference k, the approach [Eq. (3)] requires the estimation of parameter $\boldsymbol{\eta_w}$ with length $k!$ for each worker w. Each entry η_w^i corresponds to each permutation transformation function $g_i()$. The high dimension of the denoising vector results in the heavy burden of parameter estimation. Therefore, we need to design a clever and practical partition to avoid the curse of p-space partition. Let us further consider the different levels of workers. For an expert, most of her preferences are in accord with the ground truth and we can ignore other possible permutations in the sample space. For an amateur, she usually makes mistakes on comparable objects, which are ranked at abutting positions in her preferences. Therefore, it is intuitive to measure the performance of each worker by the mistakes that she makes on comparable objects, instead of disparate objects. This approach is consistent with the Kendall-tau distance (Kendall 1948; Diaconis 1988), which counts the number of pairwise disagreements between two preferences. Furthermore, for a given preference $O_{n,w}$, we observe that: (1) not all possible permutations are equally important; (2) the larger the Kendall-tau distance of a permutation to the preference $O_{n,w}$, the lower the probability that this permutation becomes the ground truth. Overall, we consider the partition of the sample space with the Kendall-tau distance (see the right panel of Fig. 2). Specifically, Let $\eta_w^r = \Pr\left(O_{n,w} \mid G_r(O_{n,w})\right)$ $\forall r \in \{0, 1, \ldots, C_k^2\}$, where $G_r(O_{n,w}) = \{g_i(O_{n,w}) \mid Kt\left(g_i(O_{n,w}), O_{n,w}\right) = r\}, i = 1, 2, \ldots, k!\}$ denotes the set of possible permutations whose Kendall-tau distance $Kt(,)$ to preference $O_{n,w}$ is r. For example, when $r = 0$, we have $G_0(O_{n,w}) = O_{n,w}$, denoting that preference $O_{n,w}$ accords with the ground truth. Benefiting from this approach, the length of the denoising vector $\boldsymbol{\eta}_w$ reduces from $k!$ [Eq. (3)] to $C_k^2 + 1$ [Eq. (4)]. Finally, our proposed ROPAL for k-ary preference aggregation can be represented as follows:

$$
\max_{\mathbf{S}, \boldsymbol{\eta}} \prod_{w=1}^W \prod_{n=1}^{N_w} \sum_{r=0}^{C_k^2} \eta_w^r f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right), \tag{4}
$$

where $\boldsymbol{\eta} \equiv \{\boldsymbol{\eta}_w\}_{w=1}^W$ and $\boldsymbol{\eta}_w = [\eta_w^0, \eta_w^1, \ldots, \eta_w^{C_k^2}]$ represents the denoising vector for worker w. $f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)$ is the likelihood of a set of permutations $g_i(O_{n,w})$ $\forall i \in \{1, 2, \ldots, k!\}$ with the Kendall-tau distance to the preference $O_{n,w}$ being r. For example, we have

$$O_{n,w}: o_a >_w o_b >_w o_c$$

Fig. 2 Different partitions of sample space. Given a preference $O_{n,w}: o_a >_w o_b >_w o_c$, its 6 distinct permutations constitute the entire sample space. Left Panel: The intuitive partition inspired by CrowdBT. Right Panel: Kendall-tau based partition. For notational simplicity, we use a, b, c to represent $o_a > o_b > o_c$

$f_{PL}\big(G_0(O_{n,w}) \mid \mathbf{S}\big) = f_{PL}(O_{n,w} \mid \mathbf{S})$, which can be modelled by the Plackett–Luce model [Eq. (1)].

According to $\eta_w^r = \Pr\big(O_{n,w} \mid G_r(O_{n,w})\big) \ \forall r \in \{0, 1, \ldots, C_k^2\}$, we have: (1) the larger the r, the more dissimilar the preference and its corresponding ground truth; (2) the larger the η_w^r, the higher probability the permutation, whose Kendall-tau distance to preference $O_{n,w}$ is r, happens to be the ground truth.

From the perspective of workers, for an expert w whose preferences accord with the ground truth, we set $\eta_w^0 \approx 1$ and $\eta_w^r \approx 0 \ \forall r \geq 1$, which means we trust all preferences annotated by worker w. For an amateur w, who only makes mistakes on comparable objects, we let $\eta_w^0 + \eta_w^1 \approx 1$ and $\eta_w^r \approx 0 \ \forall r \geq 2$. Thus, ROPAL will trust worker w with the probability based on η_w^0, or corrects her preferences with the probability revealed by η_w^1. For a spammer w, she annotates the preferences carelessly, hence we define $\eta_w^r = \frac{1}{C_k^2+1} \ \forall r \in \{0, 1, \ldots, C_k^2\}$ and the likelihood of any preference from worker w is a constant ($\frac{1}{C_k^2+1}$). That is to say, our model discards all the preferences annotated by worker w. For a malicious worker w who provides incorrect preferences deliberately, we observe the pattern that she corrupts the preferences and assign a suitable value for η_w. To be specific, if she always provide inverted preferences, we set $\eta_w^{C_k^2} \approx 1$ and ROPAL reveals the ground truth of these preferences.

From the perspective of the denoising vector, large η_w^0 indicates worker w is considered as an expert, who often provides the correct preferences. For large η_w^1, ROPAL treats worker w as an amateur, who makes wrong decisions on comparable objects occasionally. Furthermore, when $\eta_w^r \ (r \geq 2)$ is large, she is most likely to be a malicious worker, who makes mistakes on easy tasks intentionally. In general, ROPAL will explore the worker quality for each worker w, and choose to trust her preferences based on η_w^0 or reveal the possible ground truth according to $\eta_w^r \forall r \geq 1$. Therefore, ROPAL takes the worker quality into consideration, and becomes more robust than traditional PL models. According to our analysis of $\boldsymbol{\eta}_w$, ROPAL can identify noisy (non-expert) workers whose η_w^0 is small.

In a nutshell, the parameter of worker quality is extended from a real number in CrowdBT to a real vector in our proposed ROPAL. In term of worker categorization, ROPAL has a better advantage in distinguishing worker quality. Specifically, CrowdBT can only distinguish noisy workers from experts; while ROPAL can make a finer-grained categorization of noisy (non-

expert) workers into amateur, spammer and malicious workers. As a result, ROPAL achieves the better performance in robustness and noisy worker detection.

3.4 Online Bayesian inference

To estimate the parameters in ROPAL efficiently [Eq. (4)], we resort to an online learning strategy. First, we extend ROPAL to a Bayesian framework and adopt the moment-matching approach to update the hyperparameters incrementally. For simplicity, we assume independent prior distributions for each parameter. Specifically, we assume that $\mathbf{S} = (S_1, S_2, \ldots, S_L)$ is a random vector representing the scores for each object, and each entry of $\mathbf{S}$ is independent. Then, we introduce the standard Gaussian prior for each score $S_l \sim N(0, 1)$ $\forall l = 1, \ldots, L$. Meanwhile, we introduce a prior for the denoising vector $\boldsymbol{\eta}_w$, namely $\boldsymbol{\eta}_w \sim Dir(\boldsymbol{\alpha}_w)$ with $\boldsymbol{\alpha}_w = (\alpha_w^0, \alpha_w^1, \ldots, \alpha_w^{C_k^2})$ (Bishop 2006). How to set the prior hyperparameters $\{\boldsymbol{\alpha}_w\}_{w=1}^W$ is explained in the experiment section. Here, the Bayesian version of ROPAL shares many similar characteristics with the Thurstonian model (Thurstone 1927; Maydeu-Olivares 1999), as they all adopt the single ground-truth assumption. However, there is one essential difference that the scores of objects estimated by ROPAL are completely independent of workers, while the Thurstonian model learns a worker-specific variance $\sigma_{l,w}^2$ for each worker w.

Given a k-ary preference $O_{n,w}$ annotated by worker w, according to the Bayesian Theorem, the intermediate[5] posterior can be represented by the likelihood and the intermediate prior:

$$
\begin{aligned}
\Pr(\mathbf{S}, \boldsymbol{\eta} \mid O_{n,w}) &= \frac{\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \Pr(\mathbf{S}) \Pr(\boldsymbol{\eta})}{\Pr(O_{n,w})} \\
&= \frac{\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{l=1}^L N(S_l; \mu_l, \sigma_l^2) \prod_{v=1}^W Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v)}{\Pr(O_{n,w})}.
\end{aligned}
\tag{5}
$$

The first equation holds because preference $O_{n,w}$ only depends on worker w's denoising vector $\boldsymbol{\eta}_w$. Therefore, we have the equation $\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}) \equiv \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w)$.

According to Eq. (3), the posterior density $\Pr(\mathbf{S}, \boldsymbol{\eta} \mid O_{n,w})$ is complicated or even intractable due to complex likelihood function $\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta})$. To make our online learning approach feasible, we adopt the mean-field assumption to approximate the posterior density in Eq. (5), namely:

$$
\Pr(\mathbf{S}, \boldsymbol{\eta} \mid O_{n,w}) \approx q(\mathbf{S})q(\boldsymbol{\eta}) = \prod_{l=1}^L N(S_l; \mu_l^{new}, \sigma_l^{2new}) \prod_{v=1}^W Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v^{new}).
$$

where we use the superscript ^{new} to denote the posterior parameters.

However, due to the complex likelihood function in Eq. (3), the moments (μ, σ^2) of score $\mathbf{S}$ cannot be computed from a closed-form solution even with the mean-field variational approximation. Inspired by the Bayesian approximation method proposed in Weng and Lin (2011), we can estimate the variational parameters (μ, σ^2) of approximate posterior $q(\mathbf{S})$ by differential operation instead of integral operation. Therefore, we extend our method to more general likelihood function satisfying twice differentiable.

Lemma 1 (Corollary 2 in Weng and Lin 2011) *Let $\mathbf{z} = (z_1, z_2, \ldots, z_L)$ be a random vector, where each entry is independent and $z_l \sim N(0, 1)$, $l = 1, 2, \ldots, L$. Suppose that $f(O \mid \mathbf{z})$*

---

[5] Due to the online learning strategy, each time we only need one k-ary preference to update the hyperparameters. Therefore, we use the word "intermediate" to emphasize that the corresponding posterior is with respect to a single k-ary preference $O_{n,w}$ not the whole dataset D. The same is true for the "intermediate prior".

is the likelihood function of observation O and f is almost twice differentiable. Then the mean and the variance of the posterior density $\Pr(\mathbf{z} \mid O)$ given the observation O can be approximated as

$$E[\mathbf{z}] \approx E\left[\frac{\nabla_{\mathbf{z}} f(O \mid \mathbf{z})}{f(O \mid \mathbf{z})}\right], \tag{6a}$$

$$E[z_p z_q] \approx \mathbf{1}_{pq} + E\left[\frac{\nabla_{\mathbf{z}}^2 f(O \mid \mathbf{z})}{f(O \mid \mathbf{z})}\right]_{pq}, \quad p, q = 1, \ldots, L \tag{6b}$$

where $\mathbf{1}_{pq} = 1$ if $p = q$ and 0 otherwise.

Remark 2 Suppose that each object l in a ranking system is parametrized by the score S_l, and assume that the prior distribution of S_l is $N(\mu_l, \sigma_l^2)$. Upon the completion of one update, let O denote a k-ary preference and $\mathbf{z} = (z_1, z_2, \ldots, z_L)$, where $z_l = \frac{S_l - \mu_l}{\sigma_l} \sim N(0, 1) \ \forall l = 1, \ldots, L$, and L is the number of objects. $l(\mathbf{z})$ is the likelihood function of preference O and twice differentiable. Thus, the posterior density of z_l given the k-ary preference O is

$$E[\mu_l] \approx \mu_l + \sigma_l E[z_l] \approx \mu_l + \sigma_l E\left[\frac{\partial l(\mathbf{z})/\partial z_l}{l(\mathbf{z})}\right], \tag{7a}$$

$$E[\sigma_l^2] \approx \sigma_l^2 Var[z_l] = \sigma_l^2 \left(E(z_l^2) - E^2(z_l)\right) \approx \sigma_l^2$$
$$\left(1 + E\left[\frac{\partial^2 l(\mathbf{z})/\partial^2 z_l}{l(\mathbf{z})}\right] - E\left[\frac{\partial l(\mathbf{z})/\partial z_l}{l(\mathbf{z})}\right]^2\right), \tag{7b}$$

where $l = 1, 2, \ldots, L$.

For a k-ary preference $O_{n,w} : o_1 >_w o_2 >_w \cdots >_w o_k$ annotated by worker w, we first update the moments of score $\mathbf{S}$, then that of denoising vector $\boldsymbol{\eta}_w$. To update $\mathbf{S}$, we integrate $\boldsymbol{\eta}$ out to obtain the intermediate marginalized likelihood $l(\mathbf{S})$:

$$l(\mathbf{S}) = \int \cdots \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{v=1}^{W} Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v) d\boldsymbol{\eta}_1 \ldots d\boldsymbol{\eta}_W$$
$$= \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w. \tag{8}$$

The equality holds due to our assumption that each worker completes the annotation process independently. Therefore, we can integrate out another denoising vector $\boldsymbol{\eta}_v (v \neq w)$. According to the definition of PL model in Eq. (1) and likelihood in Eq. (3), we have:

$$l(\mathbf{S}) = \sum_{r=0}^{C_k^2} E_{Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)}\left[\eta_w^r\right] f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right) = \sum_{r=0}^{C_k^2} \frac{\alpha_w^r}{\sum_{t=0}^{C_k^2} \alpha_w^t} f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right), \tag{9}$$

where $G_r(O_{n,w}) = \{g_i(O_{n,w}) \mid Kt\left(g_i(O_{n,w}), O_{n,w}\right) = r\}, i = 1, 2, \ldots, k!\}$, and f_{PL} represents the Plackett–Luce model [Eq. (1)].

Let $\mathbf{z} = (z_1, z_2, \ldots, z_L)$, where $z_l = \frac{S_l - \mu_l}{\sigma_l} \sim N(0, 1) \ \forall l = 1, \ldots, L$, and L is the number of objects. It is notable in Eq. (9) that $l(\mathbf{S})$ only depends on the score of objects in preference $O_{n,w}$ during each update. Therefore, based on our mean-field assumption, only the moments of objects in preference $O_{n,w}$ need to be updated. According to Eq. (7a) in Remark 2, we update $\mu_{o_i} \ \forall o_i \in \{o_1, o_2, \ldots, o_k\}$ as follows:

$$\mu_{o_i}^{new} \approx \mu_{o_i} + \sigma_{o_i} E\left[\frac{\partial l(\mathbf{S})/\partial z_{o_i}}{l(\mathbf{S})}\right] \approx \mu_{o_i} + \sigma_{o_i}^2 \frac{\partial l(\mathbf{z})/\partial z_{o_i}}{l(\mathbf{z})}|_{\mathbf{z}=\mathbf{0}}, \tag{10}$$

where we set $\mathbf{z} = \mathbf{0}$ so that $\mathbf{S}$ is replaced by $\boldsymbol{\mu}$. Such an approximation is reasonable as we expect that the posterior density of $\mathbf{S}$ to be concentrated on $\boldsymbol{\mu}$ (Weng and Lin 2011).

According to Eq. (7b) in Remark 2, we update $\sigma_{o_i}^2 \ \forall o_i \in \{o_1, o_2, \ldots, o_k\}$ as follows:

$$(\sigma_{o_i}^2)^{new} \approx \sigma_{o_i}^2 \left(1 + E\left[\frac{\partial^2 l(\mathbf{S})/\partial^2 z_{o_i}}{l(\mathbf{S})}\right] - E\left[\frac{\partial l(\mathbf{S})/\partial z_{o_i}}{l(\mathbf{S})}\right]^2\right)$$

$$\approx \sigma_{o_i}^2 \max\left(1 + \frac{\partial}{\partial z_{o_i}}\left(\frac{\partial l(\mathbf{S})/\partial z_{o_i}}{l(\mathbf{S})}\right)|_{\mathbf{z}=\mathbf{0}}, \kappa_1\right), \tag{11}$$

where κ_1 is a small positive value to ensure a positive $\sigma_{o_i}^2$.

We now update the moments of denoising vector $\boldsymbol{\eta}_w$, and integrate $\mathbf{S}$ out to obtain the intermediate marginalized likelihood $l(\boldsymbol{\eta})$:

$$l(\boldsymbol{\eta}) = \int \cdots \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{l=1}^{L} N(S_l; \mu_l, \sigma_l^2) dS_1 \ldots dS_L$$

$$= \int \cdots \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{i=1}^{k} N(S_{o_i}; \mu_{o_i}, \sigma_{o_i}^2) dS_{o_1} \ldots dS_{o_k}.$$

The equality holds due to our independent assumption about the score $\mathbf{S}$. According to the definitions of likelihood in Eq. (3), we have:

$$l(\boldsymbol{\eta}) = E_{N(S_{o_1}, S_{o_2}, \ldots, S_{o_k})} \left[\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w)\right] = \sum_{r=0}^{C_k^2} \eta_w^r R_r, \tag{12}$$

where $R_r = E\left[f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)\right]$. $f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)$ is the likelihood of the permutations $g_i(O_{n,w}) \ \forall i \in \{1, 2, \ldots, k!\}$ with the Kendall-tau distance to the preference $O_{n,w}$ being r. We use f_{PL} to represent the Plackett–Luce model [Eq. (1)].

Furthermore, R_r can be approximated with the second-order Taylor approximation of $f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)$ at $\mu_{o_i} \ \forall o_i \in \{o_1, o_2, \ldots, o_k\}$. Specifically,

$$R_r = E\left[f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)\right]$$

$$\approx f_{PL}\left(G_r(O_{n,w}) \mid \boldsymbol{\mu}\right) + \frac{1}{2}\left(\sum_{t=0}^{C_k^2} \sigma_{o_t}^2 \frac{\partial^2}{\partial S_{o_t}^2}\left(f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)\right)\right)|_{\mathbf{S}=\boldsymbol{\mu}},$$

Then we do the following normalized nonnegative transformation to R_r,

$$R_r = \frac{max(R_r, \kappa_2)}{\sum_{t=0}^{C_k^2} max(R_t, \kappa_2)} \qquad r = 0, 1, \ldots, C_k^2,$$

where κ_2 is a small positive value to ensure a positive R_r.

Let $R = \Pr(O_{n,w})$ denote the likelihood of k-ary preference, we have:

$$
R = \Pr(O_{n,w}) = \int \cdots \int l(\boldsymbol{\eta}) \prod_{v=1}^{W} Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v) d\boldsymbol{\eta}_1 \ldots d\boldsymbol{\eta}_W
$$

$$
= \int l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w
$$

$$
= R_0 E_{Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)} \left[\eta_w^0 \right] + R_1 E_{Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)} \left[\eta_w^1 \right] + \cdots + R_{C_k^2} E_{Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)} \left[\eta_w^{C_k^2} \right]
$$

$$
= \frac{R_0 \alpha_w^0 + R_1 \alpha_w^1 + \cdots + R_{C_k^2} \alpha_w^{C_k^2}}{\sum_{t=0}^{C_k^2} \alpha_w^t}. \tag{13}
$$

As can be seen in Eq. (12), $l(\boldsymbol{\eta})$ is only associated with $\boldsymbol{\eta}_w$. Thus, $l(\boldsymbol{\eta})$ degenerates to $l(\boldsymbol{\eta}_w)$ for the third equation in Eq. (13). According to the Bayesian theorem, we have:

$$
\Pr(\boldsymbol{\eta_w} \mid O_{n,w}) = \frac{l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)}{\Pr(O_{n,w})} = \frac{l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)}{R}.
$$

Above all, only the moments of denoising vector $\boldsymbol{\eta}_w$ need to be updated during the current update. We compute $E\left[\eta_w^r \right]$ and $E\left[(\eta_w^r)^2 \right]$, where $r = 0, \ldots, C_k^2$ as follows:

$$
E\left[\eta_w^r \right] = \int \eta_w^r \Pr(\boldsymbol{\eta}_w \mid O_{n,w}) d\boldsymbol{\eta}_w = \frac{1}{R} \int \eta_w^r l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w
$$

$$
= \frac{\alpha_w^r \left(\sum_{p=0}^{C_k^2}(R_p \alpha_p) + R_r \right)}{R \left(\sum_{q=0}^{C_k^2} \alpha_w^q + 1 \right) \left(\sum_{q=0}^{C_k^2} \alpha_w^q \right)}, \tag{14a}
$$

$$
E\left[(\eta_w^r)^2 \right] = \int (\eta_w^r)^2 \Pr(\boldsymbol{\eta}_w \mid O_{n,w}) d\boldsymbol{\eta}_w = \frac{1}{R} \int (\eta_w^r)^2 l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w
$$

$$
= \frac{\alpha_w^r (\alpha_w^r + 1) \left(\sum_{p=0}^{C_k^2}(R_p \alpha_p) + 2R_r \right)}{R \left(\sum_{q=0}^{C_k^2} \alpha_w^q + 2 \right) \left(\sum_{q=0}^{C_k^2} \alpha_w^q + 1 \right) \left(\sum_{q=0}^{C_k^2} \alpha_w^q \right)}. \tag{14b}
$$

We then update the hyperparameters α_w^r for denoising vector $\boldsymbol{\eta}_w$:

$$
(\alpha_w^r)^{new} = \frac{\left(E\left[\eta_w^r \right] - E\left[(\eta_w^r)^2 \right] \right) E\left[\eta_w^r \right]}{E\left[(\eta_w^r)^2 \right] - \left(E\left[\eta_w^r \right] \right)^2}. \tag{15}
$$

As a result of the closed-form update solutions, our online Bayesian inference can handle massive k-ary preferences. Our online Bayesian inference for scalable ROPAL is given in Algorithm 1.

4 Experiments

In this section, we perform experiments on large-scale simulated datasets to validate the robustness of ROPAL preliminarily (Sect. 4.1). Then, we conduct experiments on two real-world datasets, namely "PeerGrading" and "BabyFace", to show the superiority of ROPAL

Algorithm 1 Online Bayesian Inference for Scalable ROPAL

Input: augmented k-ary crowdsourced preferences D with extra worker information. Each augmented subset
$(w, O_{n,w})$ represents the n-th preference $O_{n,w} : o_1 >_w o_2 >_w \cdots >_w o_k$ provided by a worker w.
$\{\mu_l\}_{l=1}^{L}$, $\{\sigma_l^2\}_{l=1}^{L}$ and $\{\boldsymbol{\alpha}_v\}_{v=1}^{W}$ are hyperparameters, where L and W are the number of objects and
workers, respectively. T is a positive integer, which means the number of times that we run through
the data. $|D|$ denotes the number of all preferences.
for $t = 1, 2, \ldots, T$ **do**
    Shuffle: the sequence of augmented k-ary crowdsourced preferences randomly.
    for $iter = 1, 2, \ldots, |D|$ **do**
        Fetch: an augmented k-ary preference from $D(iter)$.
        Assign: $(w, O_{n,w}) \leftarrow D(iter)$.
        Score Update: $\mu_{o_i}, \sigma_{o_i}^2$ by Eq. (10),(11) $\forall o_i \in \{o_1, o_2, \ldots, o_k\}$.
        Worker Quality Update: α_w^r where $r = 0, \ldots, C_k^2$ by Eq. (15).
    end
end
Output: The full preference over objects by ranking $\boldsymbol{\mu}$

---

in robustness and noisy worker detection (Sect. 4.2). Note that, for k-ary crowdsourced
preferences setting, we reveal the deficiency of pairwise CrowdBT in Sect. 3.2. Meanwhile,
we propose (general) k-ary ROPAL to solve this deficiency in Sect. 3.3. In this paper, we
showcase the performance of ROPAL by assigning $k = 3$. Theoretical details related to $k = 3$
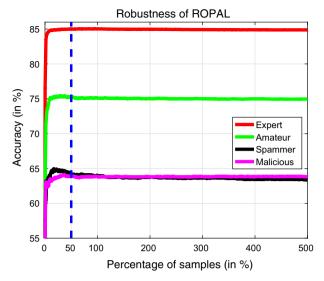can be found in "Appendix".

4.1 Simulated experiments

Datasets&metrics We generate simulated datasets in a similar way as the description in
CrowdBT (Chen et al. 2013). Assume that there are L objects with ground-truth scores
from 1 to L. Each k-ary subset, composed of k ($k \ll L$) objects randomly, is labeled
by W workers with different worker qualities $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$ following a Dirichlet distribution
$Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)$. According to the probability theory, when all the worker qualities are sim-
ulated by the same prior distribution, the average quality of workers can be estimated by
hyperparameters $\boldsymbol{\alpha}_0$ while retaining diversity among workers. Therefore, we control the qual-
ity of the data by choosing the proper hyperparameters $\boldsymbol{\alpha}_0$. To compare with other models,
we adopt the Wilcoxon–Mann–Whitney statistics (Yan et al. 2003) to evaluate the accu-
racy $\frac{\sum_{i,j} \mathbf{1}(r_i > r_j) \wedge \mathbf{1}(\mu_i > \mu_j)}{\sum_{i,j} \mathbf{1}(r_i > r_j)}$ of different models, where $r_i > r_j$ represents the ground truth
preference between object i, j and μ_i is the expected value of the score for the ith object.

Parameter initialization For the hyperparameters (μ_l, σ_l^2) of score S_l, we assign a standard
normal prior for S_l, i.e. $S_l \sim N(0, 1) \forall l \in \{1, 2, \ldots, L\}$ in all experiments. For the hyperpa-
rameter $\boldsymbol{\alpha}_w$ of uncertainty vector $\boldsymbol{\eta}_w$ $\forall w \in \{1, 2, \ldots, W\}$ in ROPAL, we initialize each $\boldsymbol{\alpha}_w$
with 7 gold tasks with known true preferences. The same initialization method also applies
to the hyperparameter (α_w, β_w) of worker quality η_w $\forall w \in \{1, 2, \ldots, W\}$ in CrowdBT. We
set $L = 1000$, $W = 600$, and assign each worker $N_w \equiv 1000$ tasks. The number of gener-
ated preferences reaches $W \times N_w = 6 \times 10^5$, which cannot be tackled by general models
with a batch learning strategy. In addition, we set $T = 5$ in Algorithm 1 for all simulated
experiments.

Robustness First, we showcase the robustness of ROPAL on four simulated datasets. Specif-
ically, according to our analysis for four kinds of workers, hyperparameter $\boldsymbol{\alpha}_0$ is set to
$(24, 3, 2, 1)$, $(5, 4, 3, 2)$, $(5, 5, 4, 4)$ and $(2, 5, 4, 2)$, which yields "Expert", "Amateur",
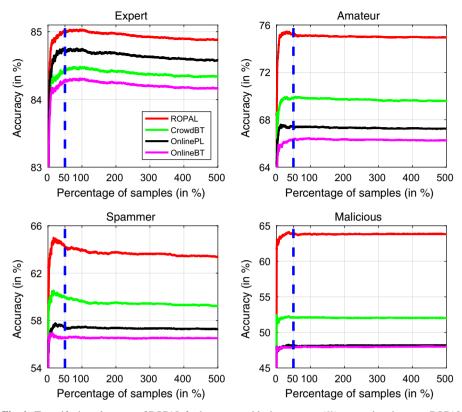
Fig. 3 To showcase the *robustness* of ROPAL preliminarily, we provide the accuracy (%) on large-scale simulated datasets, which include "Expert", "Amateur", "Spammer" and "Malicious" settings, respectively. 50% means we run the experiment with half of the data, while 500% denotes that we run the experiment on the same dataset five times with a random order

"Spammer" and "Malicious" settings, respectively. Furthermore, hyperparameter $\alpha_0 = (24, 3, 2, 1)$ denotes the probability that preferences from each worker will be in accord with the ground truth is almost 80%. The four different settings also ensure the descending order of the average quality $E(\eta_w^0)$ of four simulated datasets. From Fig. 3, we make the following observations: (1) under the Expert setting, ROPAL has the best accuracy (above 84.5%) due to the highest quality of the dataset; (2) the accuracy of ROPAL decreases by 10% on the Amateur setting and further decreases to below 65% on the Spammer setting, denoting that the quality of dataset affects the accuracy of ROPAL; (3) it is interesting to note that ROPAL on the Malicious setting obtains comparable or even better accuracy than that on the Spammer setting. This is because ROPAL tends to discard the preferences from spammers, Whereas for malicious workers, ROPAL tries to explore the patterns that they corrupt the preferences and thereby gather more information from the dataset to inference; and (4) it is notable that the accuracy on all settings reaches a high level with less than 50% of data, then the accuracy fluctuates slightly with subsequent data. Therefore, for the large dataset, the accuracy will not be improved by running multiple passes through the dataset.

Second, we further verify the robustness of ROPAL by comparing it with well-known models: (1) Bradley–Terry (BT), (2) Plackett–Luce (PL), and (3) CrowdBT (Chen et al. 2013). We make modifications to the above models for a fair comparison. (1) BT and CrowdBT cannot model the k-ary preference aggregation directly. Therefore, we split each k-ary preference as $k(k-1)/2$ (full-pair) pairwise preferences. (2) Since our model is updated with an online strategy, we implement both BT and PL with an online strategy, which is much more efficient than a batch learning strategy.

As we can see in Fig. 4, (1) on all settings, ROPAL achieves the highest accuracies over other baselines; (2) on the Expert setting, all models achieve accuracy above 84%, benefiting from the high quality of the dataset; (3) on the Amateur, Spammer and Malicious settings, the advantage of ROPAL and CrowdBT become noticeable. Since ROPAL and CrowdBT

Fig. 4 To verify the *robustness* of ROPAL further, we provide the accuracy (%) comparison between ROPAL and well-known models: OnlineBT, OnlinePL and CrowdBT. 50% means we run the experiment with half of the data, while 500% denotes that we run the experiment on the same dataset five times with a random order. Note that, on the malicious setting, the accuracy of OnlinePL and OnlineBT are almost the same, thus their figures overlap

consider worker quality, they are more robust to crowdsourced preferences; (4) all PL-based models show minor improvements over corresponding BT-based models, because BT-based models tends to split *k*-ary crowdsourced preferences and cyclical inconsistency occurs; (5) on all settings, the accuracy of all models reaches their highest values with about 50% of the data, and then remains stable or decreases slowly with subsequent data. As the metric that we adopt to evaluate the accuracy is independent of all models, the decrease in accuracy indicates the four models are overfitting on the Expert and Spammer settings. Therefore, for small dataset, the performance may improve by running multiple passes through the data (as verified by our real experiment), while for large datasets, we only run through the dataset once to avoid overfitting.

Computational cost Furthermore, we theoretically compare the computational costs of the four models in Table 2, which is verified by the empirical results on the Expert setting later (see Fig. 5). Note that: (1) the computational costs of all models are independent of the settings; and (2) since we implement the inference in an online-updating strategy, all models can be updated with one *k*-ary preference each time. From Table 2, we clearly see that two major processes contribute to the overall computational cost: namely split-
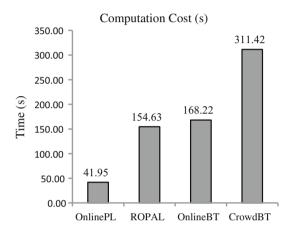
Table 2 Computational cost of ROPAL and other baselines

	Splitting process (t_1)	Updating process (t_2)	Computational cost
OnlinePL	0	$2k$	$2kt_2$
ROPAL	0	$4C_k^2 + 2k + 5$	$(4C_k^2 + 2k + 5)t_2$
OnlineBT	C_k^2	$4C_k^2$	$C_k^2 t_1 + 4C_k^2 t_2$
CrowdBT	C_k^2	$11C_k^2$	$C_k^2 t_1 + 11C_k^2 t_2$

We implement all the methods with an online strategy. We assume that t_1 is the cost of the splitting process (extracting a binary preference from a k-ary preference), and t_2 is the cost of the updating process (completing an update in Algorithm 1)

Fig. 5 To verify the *efficiency* of ROPAL, we measure the time taken by the four models when we conduct the experiment on the Expert setting. The number of preferences reaches $W \times N_w \times T = 3 \times 10^6$. Experiments are implemented on a Linux cluster with a 3.47 GHz CPU and 11.7 GB memory

ting and updating. The theoretical analysis in Table 2 is consistent with our observations in Fig. 5: (1) the computational cost of CrowdBT is much higher than that of the other models. Table 2 provides two reasons. First, CrowdBT splits k-ary preferences as C_k^2 pairwise preferences for input. Second, CrowdBT needs to update worker quality compared to OnlineBT; and (2) the computational cost of OnlineBT is slightly higher than that of ROPAL. From Table 2, we see that although the updating process dominates the overall computational cost, the splitting process in OnlineBT still affects the overall computational cost.

4.2 Real-world experiments

In this section, we first conduct experiments on the "PeerGrading" dataset to verify the robustness of ROPAL in real-world situations. More importantly, we then perform experiments on the "BabyFace" dataset to show the superiority of ROPAL in terms of robustness and noisy worker detection.

4.2.1 PeerGrading

Datasets We employ two peer grading datasets introduced by Raman and Joachims (2014), which are uniformly called "PeerGrading". The PeerGrading dataset was collected as part of a senior-undergraduate and masters-level class. It consists of the Poster (PO) dataset and the Final Report (FR) dataset. To be specific, there are 42 assignments (objects), 148

Table 3 To preliminarily verify the *robustness* of ROPAL in real-world situations, we provide the accuracy (%) on the PeerGrading dataset (PO and FR datasets)

Dataset	ROPAL	CrowdBT	OnlinePL	OnlineBT
PO(T=1)	83.11 ± 1.42	81.33 ± 1.19	76.80 ± 2.62	79.76 ± 1.14
PO(T=10)	83.67 ± 1.28	82.04 ± 1.34	77.16 ± 2.55	79.60 ± 0.98
FR(T=1)	73.64 ± 1.37	73.96 ± 1.61	73.78 ± 1.99	72.22 ± 1.57
FR(T=10)	76.86 ± 0.41	74.25 ± 0.42	75.64 ± 0.96	72.26 ± 0.62

The accuracy is represented by the mean with the standard deviation. Note that $T = 1$ denotes we run the algorithm on the dataset only once, the same is true for $T = 10$

students (workers) and 7 TAs participated in the PO dataset. For the FR dataset, we have 44 assignments, 153 students and 9 TAs. This size of class is attractive, since it is large enough for collecting a substantial number of peer grades, meanwhile, it allows teaching assistant (TA) grading to serve as the ground truth.

Parameter initialization For the hyperparameter $\boldsymbol{\alpha}_w$ in ROPAL, we have no access to the gold preferences because the average number of preferences annotated by each worker is too small (6 for PO and 2 for FR). However, as demonstrated in Raman and Joachims (2014), most students are high-quality (expert) workers in PO and FR datasets. According to our analysis in Sect. 3.3, for an expert worker w, the first entry of her denoising vector dominates the distribution (the denoising vector $\boldsymbol{\alpha}_w$). Therefore, we have $E[\eta_w^0] = \frac{\alpha_w^0}{\sum_r \alpha_w^r}$, large η_w^0 indicates worker w is an expert, who often provides the correct preferences, and vice versa. As for the hyperparameter (α_w, β_w) in CrowdBT, we have $E[\eta_w] = \frac{\alpha_w}{\alpha_w + \beta_w}$ for worker w. That is to say, the large α_w represents the high accuracy of the preferences annotated by worker w, and vice versa. For a fair comparison, we set $\boldsymbol{\alpha}_w = [24, 3, 2, 1]$ in ROPAL and $\alpha_w = 4, \beta_w = 1 \; \forall w \in \{1, 2, \ldots, W\}$ in CrowdBT, namely $E[\eta_w^0] = 0.8$ and $E[\eta_w] = 0.8$ for all workers in ROPAL and CrowdBT, respectively. This parameter initialization is consistent with our assumption that most students are expert workers in two peer grading datasets.

Robustness We explore the robustness of ROPAL preliminarily in real-world situations. As the PeerGrading dataset is small compared with our synthetic datasets, we first explore the effect of T[6] on the PeerGrading dataset. Specifically, we set T to 1 and 10 in Algorithm 1, respectively. Then, we repeat the experiment 10^3 times to collect the accuracy (mean and standard deviation) of ROPAL and other baselines in Table 3.

From Table 3, we observe that: (1) ROPAL achieves higher or comparable accuracy on all experiments; (2) on PO dataset, the accuracy of PORAL and other baseline remains stable when we run the experiment multiple times ($T = 10$). As PO dataset is a large dataset, this is consistent with our results in our synthetic experiment; and (3) since FR dataset is smaller compared to PO dataset, the accuracy of PL-based models increase obviously when we run through the FR dataset more times ($T = 10$) on FR dataset. While those of BT-based models do not enjoy such benefit, Since BT-based models tend to split each k-ary preference into C_k^2 binary preferences, BT-based models are more stable on small dataset.

---

[6] Running the experiment on the PeerGrading dataset T times with a random order.

Fig. 6 Images of child's facial microexpressions with 18 levels from anger to happy. We build the "BabyFace" dataset based on these images

4.2.2 BabyFace

Datasets We build a real-world dataset called "BabyFace". The BabyFace dataset is based on images of child's facial microexpressions, and includes 18 levels from happy to angry (see Fig. 6). This number of levels is attractive, since it is large enough to collect a substantial number of ordinal facial preferences, while allowing the authors to acquire a credible full preference to serve as a baseline. The BabyFace dataset consists of 816 distinct subsets, with each subset including three different microexpressions. We submit them to Amazon Mechanical Turk and require each worker to provide her preference from happy to angry. We receive preferences from 105 workers. We only consider workers who labelled at least 60 subsets, which yields a k-ary crowdsourced preferences setting containing 3074 preferences labelled by 41 workers. Furthermore, we ask 7 adults independent of our data-collecting process to provide a unanimous (full) preference of the microexpressions. Although we have no access to the ground-truth preference, the unanimous preference can serve as the ground-truth preference for three reasons: (1) the size (18) of levels is appropriate for a worker to provide a full preference; (2) the unanimous preference shared by 7 workers is consistent with our single ground-truth assumption; and (3) the 7 workers have access to all microexpressions at the same time. There is no time limitation for the annotation process, which ensures the credibility of the unanimous preference.

Parameter initialization The initialization of crowdsourced preferences in the real dataset is consistent with that in the simulated dataset. To be specific, for hyperparameter (μ_i, σ_i) of score S_i, we assign a standard normal prior for S_i, i.e. $S_i \sim N(0, 1) \forall i \in \{1, 2, \ldots, 18\}$. For hyperparameter $\boldsymbol{\alpha}_w$, we initialize each $\boldsymbol{\alpha}_w$ by 7 gold subsets with known true preferences. The initialization method also applies to the hyperparameters of CrowdBT and other baselines.

Robustness We further explore the robustness of ROPAL on the BabyFace dataset. Following the experiment on the PeerGrading dataset, we first set T to 1 and 10 to explore the effect of

Table 4 To further verify the *robustness* of ROPAL in the real world, we provide the accuracy (%) on the BabyFace dataset with T being 1 and 10, respectively

Dataset	ROPAL	CrowdBT	OnlinePL	OnlineBT
BabyFace (T = 1)	89.28 ± 0.87	88.85 ± 2.74	87.96 ± 2.01	84.36 ± 0.2
BabyFace (T = 10)	91.50 ± 0.33	89.54 ± 0.61	88.24 ± 1.48	$84.38 \pm 1.56 \times 10^{-15}$
Cleaned BabyFace	92.09 ± 0.19	91.54 ± 0.18	91.31 ± 0.32	$90.65 \pm 6.25 \times 10^{-16}$

To verify the efficacy of ROPAL and CrowdBT on *noisy worker detection*, we provide the accuracy (%) on the cleaned BabyFace dataset. The accuracy is represented by the mean with the standard deviation. Note that $T = 1$ denotes that we run the algorithm on the dataset only once, and $T = 10$ denotes that we run the algorithm ten times

T on the BabyFace dataset. Then, we repeat the experiment 10^3 times to collect the accuracy (mean and standard deviation) of all models in Table 4.

From the accuracy of the four models on the BabyFace dataset (the second and third lines in Table 4), we can observe that: (1) compared with $T = 1$, the accuracy of all algorithms except OnlineBT improves and the variances of all algorithms decrease for $T = 10$, which denotes that the results benefit from running the algorithms on the small dataset multiple times; (2) it is interesting to note that the accuracy of OnlineBT remains stable when we run the experiment multiple times. Since OnlineBT generates more preferences by splitting each k-ary preference into C_k^2 binary preferences, OnlineBT is more stable on small datasets. However, CrowdBT does not enjoy the benefit from splitting, due to the issue of cyclical inconsistency; (3) the accuracy of ROPAL is higher than that of CrowdBT and other baselines on the BabyFace dataset with T being 1 and 10, respectively; (4) ROPAL and CrowdBT outperform OnlinePL and OnlineBT, which shows the superiority of ROPAL and CrowdBT for considering worker quality. Both OnlinePL and OnlineBT assume each worker has the expertise for all tasks. Therefore, their performance is easily affected by crowdsourced preferences; and (5) it is noted that OnlinePL has a comparable performance to CrowdBT, which is consistent with the theoretic analysis in Theorems 1 and 2, the reason being that CrowdBT splits each k-ary preference into C_k^2 binary preferences before the aggregation, which leads to the issue of cyclical inconsistency.

Noisy worker detection ROPAL introduces a denoising vector $\boldsymbol{\eta}_w$ for each worker w. Specifically, each entry η_w^r ($r = 0, 1, \ldots, 3$) denotes the probability that the corresponding permutation, whose Kendall-tau distance to preference $O_{n,w}$ is r, happens to be the ground truth. Therefore, we can leverage $E\big[\eta_w^0\big] = \frac{\alpha_w^0}{\sum_r \alpha_w^r}$ to denote the accuracy that the preferences from worker w accord with the ground truth. Likewise, CrowdBT introduces worker quality η_w, which also denotes the annotation accuracy of worker w, where $E[\eta_w] = \frac{\alpha_w}{\alpha_w + \beta_w}$. Here, we leverage these two values as the worker quality indicator to detect noisy workers.

Figure 7 shows two groups of workers. Both groups comprise the 6 lowest worker qualities detected by ROPAL and CrowdBT, respectively. We make the following observations. (1) According to the estimated $\boldsymbol{\eta}_w$ of ROPAL, the first two entries of $worker\,8$ and $worker\,17$'s denoising vectors dominate the distribution. This indicates they are amateurs, who makes wrong decisions on comparable objects. Meanwhile, the third entry η_w^2 of $worker\,1$, $worker\,5$ and $worker\,12$ is large, since they are most likely to be malicious workers, who annotate k-ary subsets casually or even maliciously. (2) According to the 6 lowest estimated η of CrowdBT, 5 of these workers overlap with workers detected by ROPAL. However, there is no doubt that these workers will be categorized as experts by CrowdBT, since their esti-
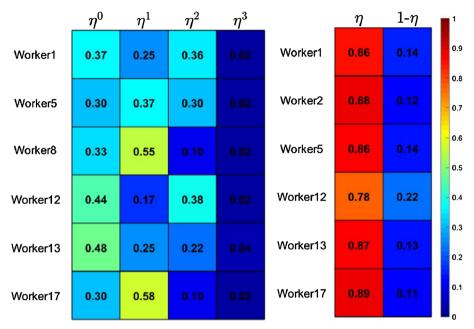
Fig. 7 To verify the effectiveness of ROPAL and CrowdBT on *noisy worker detection*, we provide the maps of estimated η related to noisy workers. Left Panel: The estimated η of 6 noisy workers detected by ROPAL. Right Panel: The estimated η of 6 noisy workers detected by CrowdBT. Each row represents the distribution of worker quality for each worker. The column in the left panel from left to right represents the probability of the preference from correct to inverted respectively, while the column in the right panel from left to right represents the probability that the worker provides right answers and wrong answers respectively

mated η are very large. Therefore, ROPAL shows superiority in noisy worker detection over CrowdBT.

To verify the efficacy of ROPAL and CrowdBT on noisy worker detection, we conduct the following experiments. For the BabyFace dataset, we remove the preferences annotated by the 5 noisy workers detected by ROPAL and CrowdBT simultaneously. Then, we rerun the four models on the cleaned BabyFace dataset. We repeat the experiment 10^3 times to measure the accuracy (as shown in mean and standard deviation) (the bottom line in Table 4). All crowdsourced preferences are consistent with the experiments on the original BabyFace dataset. Compared to the results on the BabyFace dataset in Table 4, we make the following observations according to the accuracy (%) on the cleaned BabyFace dataset in Table 4: (1) the accuracy of ROPAL stabilises around 92%, which means that ROPAL is robust enough to the preferences provided by noisy workers; (2) the accuracy of OnlinePL and OnlineBT increases dramatically on the cleaned BabyFace dataset. It means that both ROPAL and CrowdBT detect the noisy (non-expert) workers in the dataset. They can be used to improve the quality of the original dataset by detecting the preferences annotated by noisy workers; (3) the accuracy of CrowdBT also improves significantly on the cleaned BabyFace dataset. Although CrowdBT can identify noisy (non-expert) workers, CrowdBT seems not be robust enough to the crowdsourced preferences from these workers. Therefore, CrowdBT is less competitive than ROPAL in terms of robustness; and (4) the standard deviation of all models reduce since the quality of the dataset improves.

5 Conclusion

In this paper, we reveal the deficiency of pairwise models for aggregating k-ary crowd-sourced preferences, which motivates us to propose the RObust PlAckett–Luce (ROPAL) model. Specifically, ROPAL introduces a denoising vector to model worker quality, which corrects k-ary crowdsourced preferences with a certain probability. In addition, we design an online Bayesian inference, which makes ROPAL scalable to large-scale preferences. Comprehensive experiments on simulated and real-world datasets show that ROPAL is more robust than other well-known approaches. Moreover, ROPAL shows superiorities over CrowdBT in noisy worker detection. In future, we can extend our work in the following aspects. (1) How to choose k in practice is a meaningful direction. (2) Mixed membership models for preferences aggregation (Wang et al. 2015; Zhao et al. 2016) is a promising direction. Combining the mixed membership assumption along with the worker quality will be a new topic under the crowdsourcing setting. With these extensions, ROPAL can be applied to more complex situations.

Appendix

Proof of Theorem 1

Theorem 1 *Assume that k-ary preferences are aggregated by a pairwise model while considering work quality, then the number of cyclical inconsistencies caused by a k-ary preference is equal to $2^{C_k^2} - k!$.*

Proof When k-ary preferences are aggregated by a pairwise model, the pairwise model first splits each k-ary preference into C_k^2 pairwise preferences. On this basis, if the pairwise model takes worker quality into consideration, we have $2^{C_k^2}$ distinct combinations of pairwise preferences. Therefore, the number of cyclical inconsistencies caused by a k-ary preference is equal to $2^{C_k^2} - k!$, since a k-ary subset corresponds to $k!$ possible ground-truth preferences. □

Proof of Theorem 2

Theorem 2 *Under the same assumption of Theorem 1, the likelihood $P(O_1 >_w O_2 >_w \cdots >_w O_k)$ can be decomposed into $2^{C_k^2}$ distinct combinations of pairwise preferences. Let $P(O_1 >_w O_2 >_w \cdots >_w O_k) = P_k(\mathbb{I}) + P_k(\mathbb{C})$, where $P_k(\mathbb{I})$ is the probability of the occurrence of inconsistent combinations, and $P_k(\mathbb{C})$ is the probability of the occurrence of consistent combinations. Then $\frac{P_{k+1}(\mathbb{I})}{P(O_1>_w O_2>_w\cdots>_w O_{k+1})} > \frac{P_k(\mathbb{I})}{P(O_1>_w O_2>_w\cdots>_w O_k)}$, which means the ratio of the probability that cyclical inconsistencies happen in a k-ary preference increases with k.*

Proof

$$
\begin{aligned}
& P(O_1 >_w O_2 >_w \cdots >_w O_{k+1}) \\
&= \prod_{(i,j) \subseteq [k+1], i \neq j} \{\eta_w P(O_i > O_j) + (1 - \eta_w) P(O_i < O_j)\} \\
&= P(O_1 >_p O_2 >_p \cdots >_p O_k) \prod_{l \in [k]} \{\eta_w P(O_l > O_{k+1}) + (1 - \eta_w) P(O_l < O_{k+1})\} \\
&= [P_k(\mathbb{I}) + P_k(\mathbb{C})] \prod_{l \in [k]} \{\eta_w P(O_l > O_{k+1}) + (1 - \eta_w) P(O_l < O_{k+1})\}, \quad (16)
\end{aligned}
$$

where $[k] = \{1, 2, \ldots, k\}$. It is notable that, all derivatives of inconsistent combinations in one k-ary preference are still inconsistent combinations in the corresponding $(k + 1)$-ary preference. Furthermore, according to Theorem 1, we have

$$
2^{C_{k+1}^2} - (k+1)! = 2^{\frac{(k+1)k}{2}} - (k+1)k! = 2^k 2^{C_k^2} - (k+1)k! \geq 2^k (2^{C_k^2} - k!). \quad (17)
$$

The equality in Eq. (17) holds if and only if $k = 1$. That is to say, when $k \geq 2$, there exists inconsistent combinations in one $(k + 1)$-ary preference deriving from consistent combinations in the corresponding k-ary preference. Then we have:

$$
\begin{aligned}
& \frac{P_{k+1}(\mathbb{I})}{P(O_1 >_w O_2 >_w \cdots >_w O_{k+1})} \\
&\geq \frac{P_k(\mathbb{I}) \prod_{l \in [k]} \{\eta_w P(O_l > O_{k+1}) + (1 - \eta_w) P(O_l < O_{k+1})\}}{P(O_1 >_w O_2 >_w \cdots >_w O_{k+1})} \\
&= \frac{P_k(\mathbb{I})}{P(O_1 >_w O_2 >_w \cdots >_w O_k)}. \quad (18)
\end{aligned}
$$

The inequality in Eq. (18). □

Online Bayesian inference for $k = 3$

Under the 3-ary crowdsourced preferences setting, the proposed ROPAL [Eq. (4)] can be represented as:

$$
\max_{\mathbf{S}, \boldsymbol{\eta}} \prod_{w=1}^{W} \prod_{n=1}^{N_w} \sum_{r=0}^{3} \eta_w^r f_{PL}\Big(G_r(O_{n,w}) \mid \mathbf{S}\Big), \quad (19)
$$

where $\boldsymbol{\eta} \equiv \{\boldsymbol{\eta}_w\}_{w=1}^{W}$ represents the denoising vectors for all workers, and $\mathbf{S} = \{S_1, S_2, \ldots, S_L\}$ denotes the implicit scores associated with each object. We use f_{PL} to represent the Plackett–Luce model [Eq. (1)].

For a 3-ary preference $O_{n,w} : o_a >_w o_b >_w o_c$ annotated by worker w, according to our online Bayesian inference, we first update the moments of score $\mathbf{S}$, then that of denoising vector $\boldsymbol{\eta}_w$. To update $\mathbf{S}$, we integrate $\boldsymbol{\eta}$ out to obtain the intermediate marginalized likelihood $l(\mathbf{S})$:

$$
\begin{aligned}
l(\mathbf{S}) &= \int \cdots \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{v=1}^{W} Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v) d\boldsymbol{\eta}_1 \ldots d\boldsymbol{\eta}_W \\
&= \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w.
\end{aligned}
$$

$$= \sum_{r=0}^{3} E_{Dir(\eta_w|\alpha_w)} \left[\eta_w^r\right] f_{PL}\left(G_r(O_{n,w}) \mid \mathbf{S}\right)$$

$$= \frac{\alpha_w^0}{\sum_{t=0}^{3} \alpha_w^t} f_{PL}(o_a > o_b > o_c) + \frac{\alpha_w^1}{\sum_{t=0}^{3} \alpha_w^t} \left[f_{PL}(o_a > o_c > o_b) + f_{PL}(o_b > o_a > o_c)\right]$$

$$+ \frac{\alpha_w^2}{\sum_{t=0}^{3} \alpha_w^t} \left[f_{PL}(o_b > o_c > o_a) + f_{PL}(o_c > o_a > o_b)\right] + \frac{\alpha_w^3}{\sum_{t=0}^{3} \alpha_w^t} f_{PL}(o_c > o_b > o_a)$$

$$= \frac{(\alpha_w^0 e^{S_{o_b}} + \alpha_w^1 e^{S_{o_c}}) e^{S_{o_a}} \Delta_{ab} \Delta_{ac} + (\alpha_w^1 e^{S_{o_a}} + \alpha_w^2 e^{S_{o_c}}) e^{S_{o_b}} \Delta_{ab} \Delta_{bc} + (\alpha_w^2 e^{S_{o_a}} + \alpha_w^3 e^{S_{o_b}}) e^{S_{o_c}} \Delta_{ac} \Delta_{bc}}{(\sum_{t=0}^{3} \alpha_w^t)(e^{S_{o_a}} + e^{S_{o_b}} + e^{S_{o_c}}) \Delta_{ab} \Delta_{ac} \Delta_{bc}}$$

$$= \frac{A + B + C}{(\sum_{t=0}^{3} \alpha_w^t) \Delta \Delta_{ab} \Delta_{ac} \Delta_{bc}}, \tag{20}$$

where $\Delta_{pq} = e^{S_{o_p}} + e^{S_{o_q}}$, $p, q \in \{a, b, c\}$ and $\Delta = e^{S_{o_a}} + e^{S_{o_b}} + e^{S_{o_c}}$. We use the following notations in Eq. (20) to simplify our presentation:

$$A = \left(\alpha_w^0 e^{S_{o_b}} + \alpha_w^1 e^{S_{o_c}}\right) e^{S_{o_a}} \Delta_{ab} \Delta_{ac}, \, B = \left(\alpha_w^1 e^{S_{o_a}} + \alpha_w^2 e^{S_{o_c}}\right) e^{S_{o_b}} \Delta_{ab} \Delta_{bc},$$

$$C = \left(\alpha_w^2 e^{S_{o_a}} + \alpha_w^3 e^{S_{o_b}}\right) e^{S_{o_c}} \Delta_{ac} \Delta_{bc}, \, \Omega = A + B + C.$$

Let $\mathbf{z} = (z_1, z_2, \ldots, z_L)$, where $z_l = \frac{S_l - \mu_l}{\sigma_l} \sim N(0, 1) \, \forall l = 1, \ldots, L$, and L is the number of objects. Based on our mean-field assumption, only the moments of objects in preference $O_{n,w}$ need to be updated. According to Eq. (7a) in Remark 2, we update $\mu_{o_a}, \mu_{o_b}, \mu_{o_c}$ as follows:

$$\mu_{o_a}^{new} \approx \mu_{o_a} + \sigma_{o_a} E\left[\frac{\partial l(\mathbf{S})/\partial z_{o_a}}{l(\mathbf{S})}\right] \approx \mu_{o_a} + \sigma_{o_a}^2 \frac{\partial l(\mathbf{z})/\partial z_{o_a}}{l(\mathbf{z})}|_{\mathbf{z}=0} = \mu_{o_a}$$
$$+ \sigma_{o_a}^2 \left(\frac{A}{\Omega} - \frac{e^{\mu_{o_a}}}{\Delta}\right), \tag{21}$$

where we set $\mathbf{z} = \mathbf{0}$ so that $\mathbf{S}$ is replaced by $\boldsymbol{\mu}$. Such an approximation is reasonable as we expect that the posterior density of $\mathbf{S}$ to be concentrated on $\boldsymbol{\mu}$ (Weng and Lin 2011). Similarly, we have:

$$\mu_{o_b}^{new} \approx \mu_{o_b} + \sigma_{o_b}^2 \left(\frac{B}{\Omega} - \frac{e^{\mu_{o_b}}}{\Delta}\right), \tag{22}$$

$$\mu_{o_c}^{new} \approx \mu_{o_c} + \sigma_{o_c}^2 \left(\frac{C}{\Omega} - \frac{e^{\mu_{o_c}}}{\Delta}\right). \tag{23}$$

According to Eq. (7b) in Remark 2, we update $\sigma_{o_a}^2, \sigma_{o_b}^2, \sigma_{o_c}^2$ as follows:

$$(\sigma_{o_a}^2)^{new} \approx \sigma_{o_a}^2 \left(1 + E\left[\frac{\partial^2 l(\mathbf{S})/\partial^2 z_{o_a}}{l(\mathbf{S})}\right] - E\left[\frac{\partial l(\mathbf{S})/\partial z_{o_a}}{l(\mathbf{S})}\right]^2\right)$$

$$\approx \sigma_{o_a}^2 \left(1 + \frac{\partial}{\partial z_{o_a}}\left(\frac{\partial l(\mathbf{S})/\partial z_{o_a}}{l(\mathbf{S})}\right)|_{\mathbf{z}=0}\right)$$

$$\approx \sigma_{o_a}^2 \max\left(1 + \sigma_{o_a}^2 \frac{A(\Omega - A)}{\Omega^2} - \frac{e^{\mu_{o_a}}(\Delta - e^{\mu_{o_a}})}{\Delta^2}, \kappa_1\right), \tag{24}$$

where κ_1 is a small positive value to ensure a positive $\sigma_{o_a}^2$. Similarly, we have:

$$(\sigma_{o_b}^2)^{new} \approx \sigma_{o_b}^2 \left(1 + \frac{\partial}{\partial z_{o_b}} \left(\frac{\partial l(\mathbf{S})/\partial z_{o_b}}{l(\mathbf{S})} \right) |_{\mathbf{z=0}} \right)$$

$$\approx \sigma_{o_b}^2 \max \left(1 + \sigma_{o_b}^2 \frac{A(\Omega - A)}{\Omega^2} - \frac{e^{\mu_{o_b}}(\Delta - e^{\mu_{o_b}})}{\Delta^2}, \kappa_1 \right), \qquad (25)$$

$$(\sigma_{o_c}^2)^{new} \approx \sigma_{o_c}^2 \left(1 + \frac{\partial}{\partial z_{o_c}} \left(\frac{\partial l(\mathbf{S})/\partial z_{o_c}}{l(\mathbf{S})} \right) |_{\mathbf{z=0}} \right)$$

$$\approx \sigma_{o_c}^2 \max \left(1 + \sigma_{o_c}^2 \frac{A(\Omega - A)}{\Omega^2} - \frac{e^{\mu_{o_c}}(\Delta - e^{\mu_{o_c}})}{\Delta^2}, \kappa_1 \right). \qquad (26)$$

We now update the moments of denoising vector $\boldsymbol{\eta}_w$, and integrate $\mathbf{S}$ out to obtain the intermediate marginalized likelihood $l(\boldsymbol{\eta})$:

$$l(\boldsymbol{\eta}) = \int \cdots \int \Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \prod_{l=1}^{L} N(S_l; \mu_l, \sigma_l^2) dS_1 \ldots dS_L$$

$$= E_{N(S_{o_a}, S_{o_b}, S_{o_c})} \left[\Pr(O_{n,w} \mid \mathbf{S}, \boldsymbol{\eta}_w) \right]$$

$$= \eta_w^0 R_0 + \eta_w^1 R_1 + \eta_w^2 R_2 + \eta_w^3 R_3,$$

where

$$R_0 = E\left[f_{PL}(o_a > o_b > o_c) \right], \quad R_1 = E\left[f_{PL}(o_a > o_c > o_b) + f_{PL}(o_b > o_a > o_c) \right],$$

$$R_2 = E\left[f_{PL}(o_b > o_c > o_a) + f_{PL}(o_c > o_a > o_b) \right], \quad R_3 = E\left[f_{PL}(o_c > o_b > o_a) \right].$$

We calculate R_0 using the second-order Taylor approximation of $f_{PL}(o_a > o_b > o_c)$ at $(\mu_{o_a}, \mu_{o_b}, \mu_{o_c})$, and R_1, R_2, R_3 can be calculated in the same way.

$$R_0 = E\left[f_{PL}(o_a > o_b > o_c) \right]$$

$$\approx \frac{e^{\mu_{o_a}} e^{\mu_{o_a}}}{\Delta \Delta_{bc}}$$

$$+ \frac{1}{2} \left(\sigma_{o_a}^2 \frac{\partial^2}{\partial S_{o_a}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) + \sigma_{o_b}^2 \frac{\partial^2}{\partial S_{o_b}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) + \sigma_{o_c}^2 \frac{\partial^2}{\partial S_{o_c}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) \right) |_{\mathbf{S}=\mu},$$

where

$$\frac{\partial^2}{\partial S_{o_a}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) = \frac{e^{S_{o_a}+S_{o_b}}(\Delta - 2e^{S_{o_a}})}{\Delta^3},$$

$$\frac{\partial^2}{\partial S_{o_b}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) = \frac{e^{S_{o_a}+S_{o_b}} \Delta_{ac} \left(e^{S_{o_c}} \Delta - 2e^{S_{o_b}} \Delta_{bc} \right)}{\Delta^3 \Delta_{bc}^2} - \frac{e^{S_{o_a}+2S_{o_b}}(2e^{S_{o_c}} \Delta - e^{S_{o_b}} \Delta_{bc})}{\Delta^2 \Delta_{bc}^3},$$

$$\frac{\partial^2}{\partial S_{o_c}^2} \left(\frac{e^{S_{o_a}+S_{o_b}}}{\Delta \Delta_{bc}} \right) = \frac{2e^{S_{o_a}+S_{o_b}+2S_{o_c}}}{\Delta \Delta_{bc}} \left(\frac{1}{\Delta_{bc}^2} + \frac{1}{\Delta^2} \right) - \frac{e^{S_{o_a}+S_{o_b}+S_{o_c}} \left(\Delta_{ab} + e^{S_{o_b}} \right)}{\Delta^2 \Delta_{bc}^2}.$$

Then we do the following normalized nonnegative transformation to R_i,

$$R_i = \frac{max(R_i, \kappa_2)}{\sum_{t=0}^{3} max(R_t, \kappa_2)} \quad i = 0, 1, \ldots, 3 hyper,$$

where κ_2 is a small positive value to ensure a positive R_i.

Let $R = \Pr(O_{n,w})$ denote the likelihood of k-ary preference, we have:

$$R = \Pr(O_{n,w}) = \int \cdots \int l(\boldsymbol{\eta}) \prod_{v=1}^{W} Dir(\boldsymbol{\eta}_v \mid \boldsymbol{\alpha}_v) d\boldsymbol{\eta}_1 \ldots d\boldsymbol{\eta}_W$$

$$= \int l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w$$

$$= R_0 E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)} \left[\eta_w^0 \right] + R_1 E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)} \left[\eta_w^1 \right] + \cdots + R_3 E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)} \left[\eta_w^3 \right]$$

$$= \frac{R_0 \alpha_w^0 + R_1 \alpha_w^1 + R_2 \alpha_w^2 + R_3 \alpha_w^3}{\sum_{r=0}^{3} \alpha_w^t}. \tag{27}$$

According to the Bayesian theorem, we have:

$$\Pr(\boldsymbol{\eta_w} \mid O_{n,w}) = \frac{l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)}{\Pr(O_{n,w})} = \frac{l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w)}{R}.$$

Most importantly, only the moments of denoising vector $\boldsymbol{\eta}_w$ need to be updated during the current update. We compute $E\left[\eta_w^r\right]$ and $E\left[(\eta_w^r)^2\right]$, where $r = 0, \ldots, 3$ as follows:

$$E\left[\eta_w^r\right] = \int \eta_w^r \Pr(\boldsymbol{\eta}_w \mid O_{n,w}) d\boldsymbol{\eta}_w = \frac{1}{R} \int \eta_w^r l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w$$

$$= \frac{\alpha_w^r \left(\sum_{p=0}^{3} (R_p \alpha_p) + R_r \right)}{R \left(\sum_{q=0}^{3} \alpha_w^q + 1 \right) \left(\sum_{q=0}^{3} \alpha_w^q \right)}, \tag{28a}$$

$$E\left[(\eta_w^r)^2\right] = \int (\eta_w^r)^2 \Pr(\boldsymbol{\eta}_w \mid O_{n,w}) d\boldsymbol{\eta}_w = \frac{1}{R} \int (\eta_w^r)^2 l(\boldsymbol{\eta}_w) Dir(\boldsymbol{\eta}_w \mid \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w$$

$$= \frac{\alpha_w^r (\alpha_w^r + 1) \left(\sum_{p=0}^{3} (R_p \alpha_p) + 2R_r \right)}{R \left(\sum_{q=0}^{3} \alpha_w^q + 2 \right) \left(\sum_{q=0}^{3} \alpha_w^q + 1 \right) \left(\sum_{q=0}^{3} \alpha_w^q \right)}. \tag{28b}$$

We then update the hyperparameters α_w^r for denoising vector $\boldsymbol{\eta}_w$:

$$(\alpha_w^r)^{new} = \frac{\left(E\left[\eta_w^r\right] - E\left[(\eta_w^r)^2\right] \right) E\left[\eta_w^r\right]}{E\left[(\eta_w^r)^2\right] - \left(E\left[\eta_w^r\right] \right)^2}. \tag{29}$$

Finally, we get the Score Update: $\mu_{o_a}, \mu_{o_b}, \mu_{o_c}, \sigma_{o_a}^2, \sigma_{o_b}^2, \sigma_{o_c}^2$ by Eqs. (21), (22), (23), (24), (25), (26), and (2) Worker Quality Update: α_w^r where $r = 0, \ldots, 3$ by Eq. (29). Substituting the Score Update and Worker Quality Update in Algorithm 1 with our new results, we obtain the Online Bayesian Inference for $k = 3$.

References

Alfaro, L., & Shavlovsky, M. (2014). Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *ACM technical symposium on computer science education* (pp. 415–420). ACM.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Bradley, R., & Terry, M. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, *39*, 324–345.

Chen, X., Bennett, P. N., Collins-Thompson, K., & Horvitz, E. (2013). Pairwise ranking aggregation in a crowdsourced setting. In *WSDM* (pp. 193–202). ACM.

Deng, J., Krause, J., Stark, M., & Fei-Fei, L. (2016). Leveraging the wisdom of the crowd for fine-grained recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(4), 666–676.

Diaconis, P. (1988). *Group representations in probability and statistics*. Lecture Notes-Monograph Series (Vol. 11, pp. i–192). Hayward, CA: Institute of Mathematical Statistics.

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In *WWW* (pp. 613–622). ACM.

Freeman, S., & Parks, J. W. (2010). How accurate is peer grading? *CBE-Life Sciences Education*, *9*(4), 482–488.

Guiver, J., & Snelson, E. (2009). Bayesian inference for Plackett–Luce ranking models. In *ICML* (pp. 377–384). ACM.

Kazai, G., Kamps, J., Koolen, M., & Milic-Frayling, N. (2011). Crowdsourcing for book search evaluation: Impact of hit design on comparative system ranking. In *SIGIR conference on research and development in information retrieval* (pp. 205–214). ACM.

Kendall, M. G. (1948). *Rank correlation methods*. London: Charles Griffin & Company Limited.

Kolde, R., Laur, S., Adler, P., & Vilo, J. (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, *28*(4), 573–580.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L. J., Shamma, D. A., Bernstein, M. S., & Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision, 123*, 1–42.

Kulkarni, C., Wei, K., Le, H., Chia, D., Papadopoulos, K., Cheng, J., et al. (2015). Peer and self assessment in massive online classes. In *Design thinking research* (pp. 131–168). Berlin: Springer.

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

Lu, T., & Boutilier, C. (2011). Learning Mallows models with pairwise preferences. In *ICML* (pp. 145–152).

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

Luaces, O., Díez, J., Alonso-Betanzos, A., Troncoso, A., & Bahamonde, A. (2015). A factorization approach to evaluate open-response assignments in MOOCs using preference learning on peer assessments. *Knowledge-Based Systems*, *85*, 322–328.

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York: Wiley.

Lv, Y., Moon, T., Kolari, P., Zheng, Z., Wang, X., & Chang, Y. (2011). Learning to model relatedness for news recommendation. In *WWW* (pp. 57–66). ACM.

Mallows, C. L. (1957). Non-null ranking models. I. *Biometrika*, *44*(1/2), 114–130.

Maydeu-Olivares, A. (1999). Thurstonian modeling of ranking data via mean and covariance structure analysis. *Psychometrika*, *64*(3), 325–340.

Mollica, C., & Tardella, L. (2017). Bayesian Plackett–Luce mixture models for partially ranked data. *Psychometrika, 82*(2):442–458. doi:10.1007/s11336-016-9530-0.

Ok, J., Oh, S., Shin, J., & Yi, Y. (2016). Optimality of belief propagation for crowdsourced classification. In *ICML* (pp. 535–544).

Plackett, R. L. (1975). The analysis of permutations. *Applied Statistics, 24*, 193–202.

Prpić, J., Melton, J., Taeihagh, A., & Anderson, T. (2015). *MOOCs and crowdsourcing: Massive courses and massive resources*.

Raman, K., & Joachims, T. (2014). Methods for ordinal peer grading. In *KDD* (pp. 1037–1046). ACM.

Schalekamp, F., & Zuylen, A. (2009). Rank aggregation: Together we're strong. In *Workshop on algorithm engineering and experiments* (pp. 38–51). Society for Industrial and Applied Mathematics.

Shah, N., Bradley, J., Parekh, A., Wainwright, M., & Ramchandran, K. (2013). A case for ordinal peer-evaluation in MOOCs. In *NIPS workshop on data driven education*.

Sheng, V., Provost, F., & Ipeirotis, P. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *KDD* (pp. 614–622). ACM.

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *EMNLP* (pp. 254–263). Association for Computational Linguistics.

Thurstone, L. (1927). The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, *21*(4), 384–400.

Venanzi, M., Guiver, J., Kazai, G., Kohli, P., & Shokouhi, M. (2014). Community-based Bayesian aggregation models for crowdsourcing. In *WWW* (pp. 155–164). ACM.

Volkovs, M., Larochelle, H., & Zemel, R. (2012). Learning to rank by aggregating expert preferences. In *CIKM* (pp. 843–851). ACM.

Vuurens, J., Vries, A., & Eickhoff, C. (2011). How much spam can you take? An analysis of crowdsourcing results to increase accuracy. In *SIGIR workshop on crowdsourcing for information retrieval* (pp. 21–26).

Wang, Y. S., Matsueda, R., & Erosheva, E. A. (2015). *A variational EM method for mixed membership models with multivariate rank data: An analysis of public policy preferences*. arXiv preprint arXiv:1512.08731

Weng, R. C., & Lin, C. J. (2011). A Bayesian approximation method for online ranking. *Journal of Machine Learning Research*, *12*(Jan), 267–300.

Yan, L., Dodier, R., Mozer, M., & Wolniewicz, R. (2003). Optimizing classifier performance via an approximation to the Wilcoxon–Mann–Whitney statistic. In *ICML* (pp. 848–855).

Zhao, Z., Piech, P., & Xia, L. (2016). Learning mixtures of Plackett–Luce models. In *ICML* (pp. 2906–2914).