Mach Learn (2018) 107:1333–1361
https://doi.org/10.1007/s10994-018-5716-2

Stagewise learning for noisy *k*-ary preferences

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

Received: 1 October 2017 / Accepted: 2 May 2018 / Published online: 11 May 2018

Abstract The aggregation of k-ary preferences is a novel ranking problem that plays an important role in several aspects of daily life, such as ordinal peer grading, online image-rating, meta-search and online product recommendation. Meanwhile, crowdsourcing is increasingly emerging as a way to provide a plethora of k-ary preferences for these types of ranking problems, due to the convenience of the platforms and the lower costs. However, preferences from crowd workers are often noisy, which inevitably degenerates the reliability of conventional aggregation models. In addition, traditional inferences usually lead to massive computational costs, which limits the scalability of aggregation models. To address both of these challenges, we propose a reliable CrowdsOUrced Plackett–LucE (COUPLE) model combined with an efficient Bayesian learning technique. To ensure reliability, we introduce an uncertainty vector for each crowd worker in COUPLE, which recovers the ground truth of the noisy preferences with a certain probability. Furthermore, we propose an Online Generalized Bayesian Moment Matching (OnlineGBMM) algorithm, which ensures that COUPLE is scalable to large-scale datasets. Comprehensive experiments on four large-scale synthetic datasets and three real-world datasets show that, COUPLE with OnlineGBMM achieves substantial improvements in reliability and noisy worker detection over other well-known approaches.

Keywords Noisy preferences · Rank aggregation · Reliable Plackett–Luce model · Online ranking · Bayesian moment matching

Editors: Jesse Davis, Elisa Fromont, Derek Greene, and Bjorn Bringmaan.

✉ Ivor W. Tsang
Ivor.Tsang@uts.edu.au

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

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

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

1 Introduction

Originally formulated in social choice theory (Lijphart 1994; Saari 1999), ranking aggregation has been the subject of a renewed interest in the machine learning community. The array of fields ranking aggregation has been applied to include ordinal peer grading (Raman and Joachims 2014), online image-rating (Knight and Keith 2005), meta-search (Desarkar et al. 2016) and online product recommendation (Liu 2009). The goal of ranking aggregation is to aggregate a set of k-ary preferences into an unanimous global preference[1] over a large (finite) set of objects, where each k-ary preference[2] is a totally ordered subset with k elements. With the increasing popularity of crowdsourcing, aggregation problems can now draw on a plethora of available k-ary preferences for two reasons. First, crowdsourcing is a convenient way to collect massive numbers of annotations (Deng et al. 2009). Second, annotations can be procured on these platforms at very low cost (Richard 2013). However, previous aggregation models face two obvious challenges when used in real-world applications: reliability and scalability.

In terms of reliability, most of conventional models are challenged by the noisy preferences crowd workers annotate, since these models fail to consider the quality of the preferences sources (Kazai et al. 2011; Vuurens et al. 2011). Due to the limited expertise of many crowd workers, their preferences may be noisy and inconsistent with the ground truth. Recently, Chen et al. (2013) proposed CrowdBT to reliably aggregate noisy pairwise preferences by explicitly modelling the quality of the worker. A simple method of generalizing CrowdBT into k-ary preferences is to break each k-ary preference into a set of pairwise preferences, then model each pairwise preference independently with CrowdBT. However, such a simple rank-breaking method would ignoring the dependencies between k-ary preference, which would, in turn, lead to inconsistent estimates (Soufiani et al. 2014; Khetan and Oh 2016). It is worth noting that consistency is a desired statistical property in any proposed model because, as the number of preferences used increases, the output of the model estimator should converge in probability to the true parameters.

In terms of scalability, traditional inferences for aggregation models, usually lead to a massive computational cost (Wainwright and Jordan 2008). For example, with a large number of preferences (e.g., 5×10^5 preferences), both the expectation-maximization (EM) and Gibbs sampling algorithms are too slow to infer the model's parameters. Therefore, traditional methods of inference make aggregation models impractical for large-scale datasets in the real-world challenges.

Each of these challenges raises a question: Can we build a reliable aggregation model for noisy k-ary preferences? Can we propose an efficient method of inference to ensure the models scale to large-scale datasets?

To address both these challenges at once, we propose a reliable CrowdsOUrced Plackett–LucE (COUPLE) model, combined with an efficient Bayesian learning technique. COUPLE models a k-ary preference as a series of sequential comparison stages. In each stage, one object from a number of alternatives is selected preferentially as a "local winner" without replacement. Because many crowd workers have limited expertise, a simple stagewise strategy can be easily confounded by the unreliable decisions crowd workers have made. Hence, to ensure the reliability of the estimated ranking, we propose a robust learning paradigm, called stagewise learning. This paradigm considers the indecision in crowd workers' choices

[1] In this paper, we assume that the global preference has the single ground-truth ranking, which is a fundamental assumption in many aggregation models. An in-depth discussion can be found in Sect. 3.3.

[2] k is different for different subsets. An intuitive explanation can be found in Fig. 1.

when selecting a local winner. In each stage, we identify several potential local winners according to an estimate of each crowd worker's expertise. Once all stages are complete, this robust learning strategy recovers the ground truth from the noisy preferences with a certain level of probability. To ensure COUPLE's scalability, we propose an efficient online Bayesian moment matching method, which ensures COUPLE scales to large-scale datasets. Specifically, we design analytic rules to efficiently update the posterior of COUPLE after each observation, which naturally leads to an online update facility. The contributions of this paper are summarized as follows:

1. We present a CrowdsOUrced Plackett–LucE (COUPLE) model to directly aggregate noisy k-ary preferences, which avoids the statistical inconsistencies caused by rank-breaking.
2. To ensure the reliability, we introduce an uncertainty vector to model the quality of each worker, which recovers the ground truth from their noisy preferences with a certain level of probability.
3. To ensure the scalability of COUPLE, we propose Online Generalized Bayesian Moment Matching (OnlineGBMM) algorithm to update the posterior of COUPLE analytically, which ensures COUPLE is scalable to large-scale datasets.
4. We conduct comprehensive experiments on four large-scale synthetic and three real-world datasets. Empirical results show that COUPLE with the proposed OnlineGBMM algorithm delivers substantial improvements in reliability over current approaches.

The remainder of this paper is organized as follows. Section 2 discusses the related work. In Sect. 3, we summarize the crowdsourced k-ary preference settings, followed by a detailed outline of the COUPLE model. We also discuss COUPLE in the context of classical models. Section 4 presents the OnlineGBMM algorithm for COUPLE. Section 5 demonstrates the superiority of COUPLE through empirical results on both synthetic and real-world datasets. Section 6 concludes the paper and envisions the future work.

2 Related work

This section contains a review of the literature related to our topic.

Probabilistic Ranking Models These models deal with learning probability distributions over permutations (i.e., rankings or preferences over objects). They solely concern preferences, paying little attention to features. There are two main paradigms: permutation-based and score-based ranking models. Permutation-based models are based on the notion of distances (Mallows 1957; Fligner and Verducci 1986), which express the probability of a permutation in terms of its distance to a reference permutation. Score-based models express the probability of a permutation in terms of element-specific scores. In addition, score-based models can be further divided into (1) the Bradley–Terry model (Bradley and Terry 1952), which is based on pairwise preferences; (2) the Thurstonian model (Thurstone 1927), which is also based on pairwise preferences, but adopts a different probability function; and (3) the Plackett–Luce model (Luce 1959; Plackett 1975), which is based on k-ary preferences.

Ranking Aggregation (Dwork et al. 2001; Guiver and Snelson 2009) Ranking aggregation aggregates preferences from different sources into one unanimous global preference. This technique is used in applications such as (1) ordinal peer grading (Raman and Joachims 2014), where ordinal feedbacks from different graders are aggregated to yield the final grades; (2) preference aggregation (Volkovs and Zemel 2012), e.g., online image-rating,[3] which

[3] https://en.wikipedia.org/wiki/Hot_or_Not.

aggregates ratings for the attractiveness of photos provided by different users (Knight and Keith 2005; Zagel et al. 2018); and (3) meta-search (Desarkar et al. 2016), which combines the search results of multiple search engines.

However, most existing methods fail to consider the quality of preference sources (Tsiporkova and Boeva 2006) and reliability suffers as a result of problems with the crowdsourced settings (Vuurens et al. 2011); crowdsourced k-ary preferences are often noisy (Liu et al. 2012; Ok et al. 2016). Vitelli et al. (2014) used a Bayesian framework to deal with the preference uncertainty, modelling the worker quality implicitly. Raman and Joachims (2014) proposed PeerGrader as a way of dealing with random noise. However, this method does not generalize well on real-world applications as they fail to model the human annotation noise. CrowdBT (Chen et al. 2013), specifically designed for pairwise preferences, was proposed to model the worker quality explicitly and aggregate the crowdsourced preferences reliably. However, CrowdBT needs to break each k-ary preference into independent pairwise preferences before aggregation, which has been recently shown to introduce inconsistency (Soufiani et al. 2013, 2014; Khetan and Oh 2016). In this work, we directly integrate worker quality into the Plackett–Luce model, which can aggregate the noisy preferences reliably and naturally circumvent the inconsistency caused by rank-breaking.

Online Ranking Online ranking is a practical technique for handle large-scale datasets. In these methods, the global rankings are updated with streaming preferences. Elo (1978) and Glickman (1999) are famous online ranking systems. However, they were designed for pairwise preferences, which limits their applications to more general cases that involve multiple object comparisons. Herbrich et al. (2007) developed TrueSkill, which constructs a graphical model and performs inference using an approximate message passing. Weng and Lin (2011) introduced a Bayesian approximation method to derive simple analytic rules for inference in k-ary preferences aggregation. Their methods, OnlineBT and OnlinePL, achieve competitive accuracy with the TrueSkill system but are much faster, as they both rely on analytical update rules rather than the iterative procedures in TrueSkill. The above techniques were originally designed for clean preferences. However, given that the preferences from crowd workers are often noisy, these online ranking techniques suffer from reliability under the crowdsourced k-ary preferences setting.

3 Towards the robust aggregation of noisy preferences

In Sect. 3.1, we introduce the crowdsourced k-ary preferences setting, followed by a discussion on the deficiencies of classical models in Sect. 3.2.1. Inspired by the stagewise interpretation of Plackett–Luce model, our COUPLE model is presented in Sect. 3.2.2, and its reliability is investigated in Sect. 3.2.3. Finally, the connections between COUPLE, Plackett–Luce, CrowdBT and other classical models are explored in Sect. 3.3.

3.1 Crowdsourced k-ary preferences setting

Before delving into the crowdsourced k-ary preferences setting, some common notations are explained in Table 1.

In traditional ordinal evaluation problems, a full set Ω with L objects is presented to W workers. Each worker ranks the entire set Ω independently to yield a full preference according to a certain criterion, such as personal hobbies or attitudes. However, L tends to be large in many real-world applications (Shah et al. 2013; Luaces et al. 2015). Normally, each worker tends to rank the $l(< L)$ objects they are most confident with, leaving the remaining

Table 1 Common mathematical notation

Notation	Explanation
Ω	Set of all objects, $\Omega = \{O_1, O_2, \ldots, O_L\}$
ξ	Subset of Ω, $\xi \subseteq \Omega$
L	$\|\Omega\|$, total number of all objects
W	Number of crowd workers
D	Collection of all k-ary preferences
D_w	Collection of k-ary preferences annotated by crowd worker w
N_w	$\|D_w\|$, number of k-ary preferences annotated by crowd worker w
$\rho_{n,w}$	The nth k-ary preference annotated by crowd worker w
$l_{\rho_{n,w}}$	The length of preference $\rho_{n,w}$
$\max(\xi)$	The best object in subset ξ according to a criterion
$O_i > O_j$	The ground truth order between O_i and O_j
$O_i \succsim O_j$	The preference annotated by crowd worker

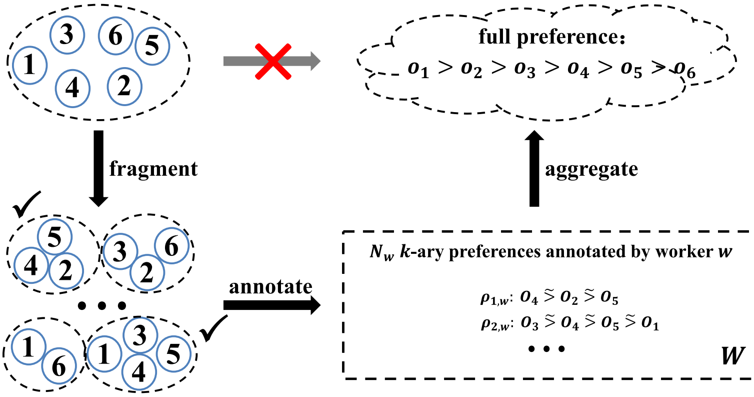
3
6
5
1
4
2
full preference:
$o_1 > o_2 > o_3 > o_4 > o_5 > o_6$
fragment
aggregate
5
4
2
3
6
2
annotate
1
6
3
1
5
4
N_w k-ary preferences annotated by worker w
$\rho_{1,w}$: $o_4 \succsim o_2 \succsim o_5$
$\rho_{2,w}$: $o_3 \succsim o_4 \succsim o_5 \succsim o_1$
• • •
W

Fig. 1 Crowdsourced k-ary Preferences Setting. Fragment: a large set of objects is broken into several subsets; Annotate: crowd workers annotate multiple (overlapped) subsets independently to yield k-ary preferences; Aggregation: aggregation rules are used to aggregate the noisy k-ary preferences from crowd workers into one unanimous global preference. Note that: (1) The tasks(subsets) with "✓" are assigned to the worker w. (2) The notation W in the corner denotes that W crowd workers complete the annotation independently

$L - l$ objects undefined. To model these partial preferences, many researchers assume that the remaining $L - l$ objects are tacitly to be ranked lower (Guiver and Snelson 2009; Mollica and Tardella 2016). However, assuming that rare objects are less important is unrealistic, especially for the large L ranking problem.

A promising approach for tackling large-scale evaluation problems originated in Massive Open Online Courses (MOOC)—peer grading (Raman and Joachims 2014; De Alfaro and Shavlovsky 2014; Kulkarni et al. 2013). In peer grading, student assignments are divided into small groups; then, each grader orders the assignments in terms of quality independently. Inspired by Peer Grading, we summarize a general setting: Crowdsourced k-ary Preferences Setting in Fig. 1. Here, a large set of objects Ω is randomly broken into several tasks $\{\xi_i\}_{i=1}^{T}$, where each task ξ_i is a subset of Ω and T is the number of tasks. Then $\{\xi_i\}_{i=1}^{T}$ are assigned randomly (with replacement) to W crowd workers to be annotated. We assume that each worker has their own beliefs as to the correct preferences for all L objects and will annotate each k-ary preference according to those beliefs. To control the difficulty of the task, the number of objects in each task is restricted to be much smaller than the total number of objects (e.g., $\max_i |\xi_i| \approx 7 \ll |\Omega| = L, \forall i \in \{1, \ldots, T\}$) (Raman and Joachims 2014).

The aim in this setting is to reliably and efficiently aggregate the noisy k-ary preferences annotated by W crowd workers into one unanimous global preference over all objects.

3.2 Reliability: from classical models to COUPLE

In this subsection, we first discuss the deficiency of classical models for noisy k-ary preferences. Then, we propose our COUPLE model and analyze its reliability and difficulty of optimization.

3.2.1 Intractability of classical models

For the k-ary preference (ordering of k objects), a classic model is the Plackett–Luce model. This model relies on Luce's axiom of choice (Luce 1959), i.e., the odds of choosing an object over another do not depend on the set of objects from which the choice is made. Suppose we have a set of k objects $\xi = \{O_1, O_2, \ldots, O_k\}$. Under Luce's axiom, the probability of selecting an object i from ξ is given by $\frac{e^{\lambda_i}}{\sum_{t=1}^{k} e^{\lambda_t}}$, where λ_i represents the **score** (real-value constant) associated with object i. The larger the score for λ_i, the higher position the object i locates in the preference. Considering a k-ary preference $\rho : O_1 > O_2 > \cdots > O_k$ as a sequence of choices: the top-ranked object is chosen first, followed by the second-ranked object from the remaining objects, and so on. It follows that the probability of the preference ρ is

$$f_{PL}(\rho|\boldsymbol{\lambda}) = \prod_{i=1}^{k} \frac{e^{\lambda_{\rho^{(i)}}}}{\sum_{t=i}^{k} e^{\lambda_{\rho^{(t)}}}}, \tag{1}$$

where $\rho^{(i)}$ is the ith-ranked object in ρ. The above model is also derived in Plackett (1975); hence the name the Plackett–Luce model. Given the assumption of a single ground truth, Eq. (1) actually defines the likelihood of each preference over the subset, based on the score (λs) for each object. The more a preference is consistent with the ground truth (over the subset), the larger the probability value of f_{PL}. However, since the parameter λs are unknown, f_{PL} itself cannot be viewed as an indicator to discriminate the high-quality preferences from the low-quality ones. Further, the parameter λs are usually estimated with a maximum likelihood estimation (MLE), which aims to finding the parameter λs (of f_{PL} model) that best fit the data, without distinction of the preference quality. As previously mentioned, preferences from crowdsourcing platforms are often noisy, and low-quality preferences could easily skew estimates of parameter λs. Therefore, the performance of a vanilla Plackett–Luce model suffers when aggregating crowdsourced k-ary preferences.

Chen et al. (2013) proposed CrowdBT to reliably aggregate crowdsourced pairwise preferences by introducing the worker quality η_w for crowd worker w, $\forall w \in \{1, 2, \ldots, W\}$. η_w represents the probability that the pairwise preference annotated by crowd worker w accords with the ground truth. Namely, $\eta_w = P(O_i \tilde{>} O_j | O_i > O_j)$, where $O_i \tilde{>} O_j$ denotes the pairwise preference annotated by crowd worker w, and $O_i > O_j$ is the ground truth between O_i and O_j. According to the law of total probability, we have $P(O_i \tilde{>} O_j) = \eta_w P(O_i > O_j) + (1 - \eta_w) P(O_i < O_j)$. However, CrowdBT was originally designed for pairwise preferences, so it cannot directly model crowdsourced k-ary preference. A simple practice used to generalize CrowdBT into crowdsourced k-ary preference is rank-breaking[4] (Soufiani et al. 2013; Shah et al. 2015; Negahban et al. 2016). Specifically, for a k-ary preference $\rho : O_1 \tilde{>} O_2 \tilde{>} \cdots \tilde{>} O_k$, full rank-breaking refers to the pair comparisons $\mathcal{A} = \left\{(O_i \tilde{>} O_j)|i, j \in \{1, 2, \ldots, k\}, i > j\right\}$; adjacent rank-breaking refers to the pair comparisons $\mathcal{B} = \{(O_i \tilde{>} O_{i+1})|i \in \{1, 2, \ldots, k-1\}\}$; and position-$i$ ($< k$) rank-breaking refers to the pair comparisons $\mathcal{C} = \left\{(O_i \tilde{>} O_j)|j \in \{1, 2, \ldots, k\}, j > i\right\}$. Therefore, we can break each k-ary preference into a set of pairwise preferences. Then, CrowdBT can model each pairwise comparison independently. Below, we take $k = 3$ as an example. The likelihood of a ternary preference with full rank-breaking can be expressed as:

$$P(O_a \tilde{>} O_b \tilde{>} O_c) = \prod_{(i,j)\in\mathcal{A}} \left\{\eta_w P(O_i > O_j) + (1-\eta_w) P(O_i < O_j)\right\},$$

where $\mathcal{A} = \{(a, b), (a, c), (b, c)\}$.

However, due to the ignored dependencies among the pairwise preferences, an inappropriate rank-breaking approach would result in inconsistent estimates according to Lemma 1. Further, the computational burden caused by the full rank-breaking would limit its application with large k preferences. See Sect. 4.4 for more details.

Lemma 1 (Corollary 1 in Soufiani et al. (2014)) *Given a k-ary preference, the only consistent rank-breaking for the Bradley–Terry model is the full rank-breaking.* □

The above analysis motivates us to directly integrate worker quality into the Plackett–Luce model. This not only avoids the inconsistency and computational burden caused by rank-breaking, but also reliably aggregates noisy k-ary preferences. Again, our aim is to reliably aggregate the noisy k-ary preferences annotated by crowd workers. Unlike the series of practices proposed by Raman and Joachims (2014) that try to reduce the adverse impact of noisy preferences, we aim to recover the ground truth by capturing each worker's annotation pattern from the noisy preferences. To compare k objects, there are $k!$ distinct permutations, which constitutes a finite partition of the entire permutation space. Therefore, we aim to traverse the entire permutation space of each k-ary preference and identify the ground truth.

Inspired by CrowdBT, it is intuitive to traverse the entire permutation space explicitly, where each permutation can be unique indexed based on its distance to the noisy preference (a similar indexing strategy to the permutation-based ranking model). Formally, let ρ denote the noisy preferences, and $\rho[i]$ be the permutation indexed as ith. Then we have $\eta_w^i = P(\rho|\rho[i])$, which denotes the conditional probability that we observe ρ selected given that the ith indexed permutation $\rho[i]$ is the ground truth.

However, this approach has the following drawbacks: (1) It requires that all preferences are the same length, and crowdsourced k-ary preferences may not satisfy this inflexible constraint. (2) The permutation-based indexing method is not scalable as k increases. For example, for moderate $k = 6$, the length of $\boldsymbol{\eta}_w$ reaches 6! (720), which is intractable for inference. Therefore, we need to design a clever and practical indexing method to traverse the permutation space.

3.2.2 The CrowdsOUrced Plackett–LucE (COUPLE) model

To assist in explaining COUPLE, it is helpful to revisit the Plackett–Luce model from the beginning. The Plackett–Luce model can be regarded as a stagewise model (Volkovs and

[4] Rank-breaking refers to the idea of splitting the observed preference into a set of pairwise comparisons and applying estimators tailored for pairwise preferences treating each piece of comparisons as independent.

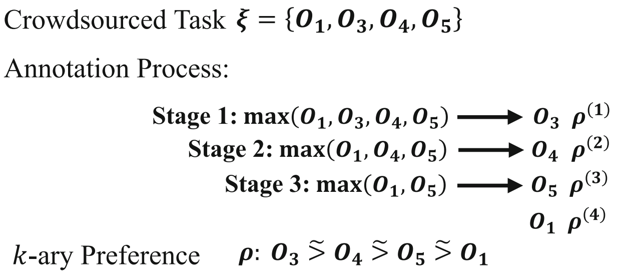
Crowdsourced Task $\boldsymbol{\xi} = \{O_1, O_3, O_4, O_5\}$
Annotation Process:
Stage 1: max(O_1, O_3, O_4, O_5) → O_3 $\rho^{(1)}$
Stage 2: max(O_1, O_4, O_5) → O_4 $\rho^{(2)}$
Stage 3: max(O_1, O_5) → O_5 $\rho^{(3)}$
O_1 $\rho^{(4)}$
k-ary Preference ρ: $O_3 \gtrsim O_4 \gtrsim O_5 \gtrsim O_1$

Fig. 2 Stagewise annotation process. In stage 1, object O_3 is selected as the "local winner" and ranked first. Then, O_3 is removed from the candidate set and another "local winners" O_4 is selected in stage 2; O_5 is selected in stage 3, and so on

Zemel 2012) that constructs a preference through a series of sequential stages. In each stage, compared to all the remaining alternatives, the object selected preferentially (without replacement) is regarded as the "local winner" (Fig. 2).

Inspired by the stagewise annotation process, we introduce the following stagewise learning strategy, namely the learning process is broken down into a number of sub-tasks that are completed in stages. The idea is to inject ranking information into the learning model gradually so as to focus on modelling the local winner in each stage, rather than modelling the complex ranking as a whole. Following the stagewise learning strategy, the Plackett-Luce model decomposes each k-ary preference into a series of sequential stages and models each stage independently. Therefore, the likelihood function for the k-ary preference ρ can also be expressed as follows:

$$P(\rho|\boldsymbol{\lambda}) = \prod_{i=1}^{k} P\left(X_i = \rho^{(i)}|\boldsymbol{\lambda}\right) = \prod_{i=1}^{k} \delta(\lambda_{\rho^{(i)}}), \tag{2}$$

where $X_i \triangleq \max(\rho^{(i)}, \rho^{(i+1)}, \ldots, \rho^{(k)})$ denotes the local winner in stage i. Further, the softmax function $\delta(\lambda_{\rho^{(i)}}) = \frac{e^{\lambda_{\rho^{(i)}}}}{\sum_{t=i}^{k} e^{\lambda_{\rho^{(t)}}}}$ is used to model the probability that object $\rho^{(i)}$ is selected as the local winner in stage i.

Remark 1 Given the assumption of a single ground truth, f_{PL} actually defines the likelihood of each preference over the subset. Hence, the preferences with a higher degree of consistency to the ground truth (over the subset) have a higher f_{PL} value. There are some rules that govern the behaviour characteristics of crowd workers and the likelihood (f_{PL}) of their corresponding annotations: (1) Expert workers have a clear understanding about the contrast among objects. Hence, their annotated preferences are usually fully consistent with the ground truth (the order of λs). Therefore, the likelihood f_{PL} of their preferences are usually the greatest, almost 1. (2) Amateur workers may mistakenly annotate the preferences due to their limited expertise with object contrast. In these cases, their annotated preferences are often slightly inconsistent with the ground truth (the order of λs), and the likelihood f_{PL} of the preference is usually smaller than the annotations by experts. Therefore, if we directly model the noisy k-ary preferences provided by amateur workers without making any distinctions about the quality of the preferences, the Plackett–Luce model's reliability would inevitably degrade. □

The stagewise learning strategy is a scalable approach for k-ary preferences in two aspects: (1) each stage can be further assumed independent, and can be updated in a distributed fashion.

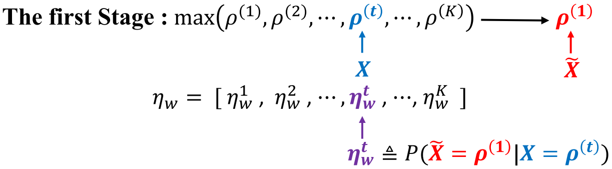
The first Stage : $\max(\rho^{(1)}, \rho^{(2)}, \cdots, \rho^{(t)}, \cdots, \rho^{(K)}) \longrightarrow \rho^{(1)}$
X
$\tilde{X}$
$\eta_w = [\eta_w^1, \eta_w^2, \cdots, \eta_w^t, \cdots, \eta_w^K]$
$\eta_w^t \triangleq P(\tilde{X} = \rho^{(1)} | X = \rho^{(t)})$

Fig. 3 An intuitive example for the first stage of our robust stagewise learning strategy

(2) Since the local winner in each stage is much easier to model than the entire preference, it is more flexible to introduce the latent variable with new functions (i.e., an indicator for worker quality). Due to crowd workers' limited expertise, a vanilla Plackett–Luce model (Eq. (2)) yields some deviations in modelling the noisy preferences. Hence, based on the introduced stagewise learning strategy, we first split each preference into a series of sequential stages and model each stage independently. Then we focus on recovering the ground truth in each stage, rather than directly identifying the ground truth of the whole preference.

For a K-ary preference,[5] at the first stage with K objects to be compared (See Fig. 3), we only need to identify the actual object that is ranked first among the candidate set of size K. Note that each candidate can be uniquely indexed based on the order in the original K-ary noisy preference. Formally, let ρ denote the noisy preferences, where $\rho^{(t)}$ is the tth ranked object (indexed as tth in the candidate set). We have omitted the subscripts n and w for brevity. Next, we introduce the uncertainty vector $\boldsymbol{\eta}_w$ for each crowd worker w to model the worker quality. The length of $\boldsymbol{\eta}_w$ for any crowd worker w is set to the maximal preference length K. Further, we assume $\boldsymbol{\eta}_w = [\eta_w^1, \eta_w^2, \ldots, \eta_w^K]$ with $\sum_{t=1}^K \eta_w^t = 1$, while each entry $\eta_w^t = P(\tilde{X} = \rho^{(1)} | X = \rho^{(t)})$ denotes the conditional probability that we observe $\rho^{(1)}$ selected given the tth indexed object $\rho^{(t)}$ being the ground truth. $\tilde{X}$ denotes the local winner selected by crowd worker w. X represents the object that should have been selected according to the ground truth. The strategy of stagewise learning overcomes the deficiency with a permutation-based approach (Sect. 3.2.1), which needs to enumerate all possible permutations, and significantly reduces the parameter space from $K!$ to K.

However, given a K-ary preference ρ, each stage will have a different number of objects to compare, which means different entries of the uncertainty vector will be active in each stage. Therefore, a single uncertainty vector is not suitable for processing all stages simultaneously. To avoid this issue, we propose the **renormalization trick**, namely normalizing the active entries in each stage, to populate the definition of the uncertainty vector to subsequent stages. Formally, in the ith stage, let $\bar{\eta}_w^{(t-i+1)} = \frac{\eta_w^{(t-i+1)}}{\sum_{v=1}^{K-i+1} \eta_w^v}$, where $t = i, i+1, \ldots, K$. Following the above rules, we can model the follow-up stages sequentially until there is only one candidate left (Fig. 4).

Specifically, only two objects are compared in stage $K-1$. However, the corresponding active entries $[\eta_w^1, \eta_w^2]$ do not constitute a valid distribution because $\eta_w^1 + \eta_w^2 \neq 1$. Therefore, we normalize the active entries to ensure at least one of the two objects is selected. Similarly, in the general stage i, we have $(K-i+1)$ candidates, which is less than the maximal preference length K. Only the top $(K-i+1)$ entries of $\boldsymbol{\eta}_w$ are active. Then, we apply the renormalization trick on the active entries $[\eta_w^1, \eta_w^2, \ldots, \eta_w^{(K-i+1)}]$, and generalize the definition of the uncertainty vector accordingly.

[5] K is the maximal preference length, where $K = \max_{n,w} l_{\rho_{n,w}}$, $n = 1, 2, \ldots, N_w$ and $w = 1, 2, \ldots, W$.

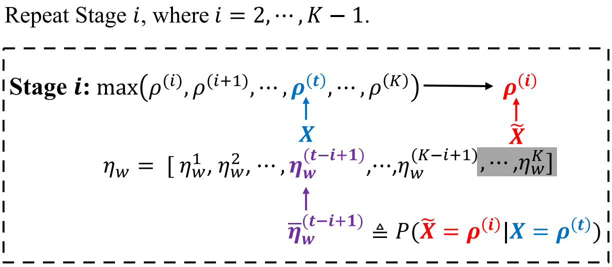
Repeat Stage i, where $i = 2, \cdots, K-1$.
Stage i: $\max(\rho^{(i)}, \rho^{(i+1)}, \cdots, \boldsymbol{\rho}^{(t)}, \cdots, \rho^{(K)})$
$\boldsymbol{\rho}^{(i)}$
$\boldsymbol{X}$
$\tilde{\boldsymbol{X}}$
$\eta_w = [\eta_w^1, \eta_w^2, \cdots, \boldsymbol{\eta}_w^{(t-i+1)}, \cdots, \eta_w^{(K-i+1)}, \cdots, \eta_w^K]$
$\bar{\boldsymbol{\eta}}_w^{(t-i+1)} \triangleq P(\tilde{\boldsymbol{X}} = \boldsymbol{\rho}^{(i)} | \boldsymbol{X} = \boldsymbol{\rho}^{(t)})$

Fig. 4 Robust stagewise learning strategy. $\tilde{X}$ denotes the local winner selected by crowd worker w. X represents the object that should have been selected according to the ground truth. For brevity, we have omitted the subscripts i which indicates the stage in which the object has been selected

Remark 2 According to our definition in Fig. 4, we can make the following observations: (1) For an expert worker w, η_w^t decreases exponentially with t, as experts have a clearer understanding about the contrast between the objects. (2) An amateur worker w may hesitate over comparable objects due to limited expertise. That is to say, η_w^1, denoting the conditional probability that the selected local winner accords with the ground truth, does not gain an absolute advantage over other entries $\eta_w^t (t \geq 2)$, especially η_w^2. □

Therefore, after integrating the Plackett–Luce model with the introduced uncertainty vector, the likelihood of the k-ary preference ρ in stage i can be represented as:

$$\begin{aligned} P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) &= \sum_{t=i}^{k} P\left(\tilde{X} = \rho^{(i)} | X = \rho^{(t)}\right) P\left(X = \rho^{(t)} | \boldsymbol{\lambda}\right) \\ &= \sum_{t=i}^{k} \bar{\eta}_w^{(t-i+1)} \delta\left(\lambda_{\rho^{(t)}}\right). \end{aligned} \tag{3}$$

Combining Eqs. (2) and (3), we propose our reliable COUPLE model for a collection of crowdsourced preferences D, which can be expressed as follows:

$$\begin{aligned} P\left(D | \boldsymbol{\lambda}, \{\boldsymbol{\eta}_w\}_{w=1}^{W}\right) &= \prod_{w=1}^{W} P(D_w | \boldsymbol{\lambda}, \boldsymbol{\eta}_w) = \prod_{w=1}^{W} \prod_{n=1}^{N_w} P(\rho_{n,w} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w) \\ &= \prod_{w=1}^{W} \prod_{n=1}^{N_w} \prod_{i=1}^{l_{\rho_{n,w}}} P\left(\tilde{X} = \rho_{n,w}^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) \\ &= \prod_{w=1}^{W} \prod_{n=1}^{N_w} \prod_{i=1}^{l_{\rho_{n,w}}} \sum_{t=i}^{l_{\rho_{n,w}}} \bar{\eta}_w^{(t-i+1)} \delta(\lambda_{\rho_{n,w}^{(t)}}), \end{aligned} \tag{4}$$

where $\boldsymbol{\eta}_w$ is the uncertainty vector for each crowd worker w. This uncertainty vector reveals crowd worker w's indecision to select the local winner in each stage. The optimization difficulty of the COUPLE model (Eq. 4) is discussed in Sect. 3.2.4. In particular, we resort to the Bayesian framework to infer the uncertainty vector $\boldsymbol{\eta}_w$, and choose a tailor-designed prior distribution to circumvent the need to directly optimize the normalization.

3.2.3 Reliability of COUPLE model

In this subsection, we characterize COUPLE's reliability through the score parameter $\boldsymbol{\lambda}$ and the quality parameter $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$.

Analysis of parameters $\boldsymbol{\lambda}$: Assume in stage i of a k-ary preference ρ, worker w selects object $\rho^{(i)}$ as the local winner: (1) In easy tasks, $\lambda_{\rho^{(i)}} \gg \{\lambda_{\rho^{(i+1)}}, \lambda_{\rho^{(i+2)}}, \ldots, \lambda_{\rho^{(k)}}\}$ denotes object $\rho^{(i)}$ exhibits significant advantages over the other objects. According to Eq. (3), we have $P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) \approx \eta_w^1$, namely, the likelihood function for stage i is mainly dependent on worker's expertise. (2) In a more difficult task, $\lambda_{\rho^{(i)}} \approx \cdots \approx \lambda_{\rho^{(i+m)}} \gg \{\lambda_{\rho^{(i+m+1)}}, \ldots, \lambda_{\rho^{(k)}}\}$, we have $P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) \approx \frac{1}{m+1}$, which means COUPLE cannot distinguish these $(m+1)$ objects $\{\rho^{(i)}, \rho^{(i+1)}, \ldots, \rho^{(i+m)}\}$ regardless of the worker's expertise.

Analysis of parameters $\boldsymbol{\eta}_w$: (1) If worker w is an expert, we have $\eta_w^1 \approx 1$ and $\eta_w^r \approx 0$ for $r \geq 2$, which means the ground truth object wound be selected in each stage with no hesitation. (2) Amateur workers tend to make more mistakes about similar objects, which means a choice needs to be made between m potential local winners at some stages. Fortunately, these m objects appear in abutting positions in a preference. Therefore, we have $\sum_{r=1}^{m} \eta_w^r \approx 1$ and usually $m = 2$. (3) If worker w is a spammer, we have $\eta_w^1 \approx \eta_w^2 \approx \cdots \approx \eta_w^K$. Thus, the likelihood for each stage equals to some constant, which means COUPLE cannot distinguish the objects and discard all the preferences D_w annotated by worker w. (4) Malicious workers intentionally select inferior objects in each stage. COUPLE places more weight on $\eta_w^r (r > 1)$ instead of η_w^1 to correct the order of the objects.

3.2.4 Optimization difficulty of COUPLE model

In this subsection, we discuss the optimization difficulty of the COUPLE model.

The objective of the proposed COUPLE model is formulated as a Maximum Likelihood Estimation (MLE) problem. The aim is simply to estimate the model parameters (λs and $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$) by maximizing Eq. (4). In principle, any solution strategies for MLE can be used as a candidate to solve this problem. What actually makes this problem difficult or even intractable for traditional MLE solutions lies in the introduced latent variable $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$ (a.k.a. uncertainty vector) and the renormalization trick required in each stage. In the following, we leverage three examples (Bishop 2006): Coordinate Gradient Descent (CGD) algorithm (Common practice for MLE), Expectation Maximization (EM) algorithm (Common practice for MLE with latent variable) and Markov Chain Monte Carlo (MCMC) method (Common practice for MLE with complex formula) to intuitively illustrate the difficulty of our proposed model.

In terms of CGD, we need to calculate the first order partial derivatives of the log likelihood (Eq. (4)) w.r.t. the parameters λs and $\boldsymbol{\eta}_w$, respectively. However, because the sum (integration over the latent variable $\boldsymbol{\eta}_w$) is inside of the product of Eq. (4), the partial derivatives w.r.t. the parameters λs and $\boldsymbol{\eta}_w$ become extremely complex. Further, since $\boldsymbol{\eta}_w$ is restricted to [0, 1], the box-constrained optimization would lead to an inaccurate and inefficient solution. Taking these two points into account, we shelved CGD and moved on to other possible candidates.

In terms of EM, it avoids calculating the derivative to the sum of the latent variables directly, and instead resorts to a surrogate lower bound for optimization. Therefore, EM, a silver bullet for MLE with latent variables, seems a promising approach for Eq. (4). However,

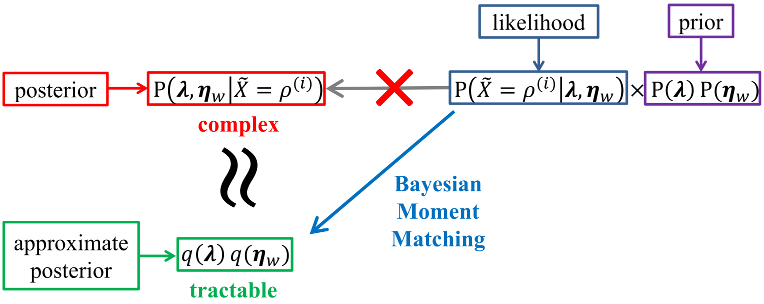
likelihood
prior
posterior
$P(\boldsymbol{\lambda}, \boldsymbol{\eta}_w | \tilde{X} = \rho^{(i)})$
$P(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w) \times P(\boldsymbol{\lambda})\, P(\boldsymbol{\eta}_w)$
complex
≈
Bayesian Moment Matching
approximate posterior
$q(\boldsymbol{\lambda})\, q(\boldsymbol{\eta}_w)$
tractable

Fig. 5 Bayesian Moment Matching: (1) define $q(\boldsymbol{\lambda})q(\boldsymbol{\eta}_w)$ in the same form with the prior (product of a Dirichlet with normals); (2) match the moments between $q(\boldsymbol{\eta}_w)$ and $P\left(\boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right)$; (3) match the moments between $q(\boldsymbol{\lambda})$ and $P\left(\boldsymbol{\lambda}|\tilde{X} = \rho^{(i)}\right)$

due to the introduced renormalization trick for $\boldsymbol{\eta}_w$ in each stage, we still need to calculate the derivative w.r.t. $\bar{\boldsymbol{\eta}}_w$ instead of $\boldsymbol{\eta}_w$ directly. Therefore, the renormalization trick makes the derivatives w.r.t. $\boldsymbol{\eta}_w$ remain complex. Moreover, we still need to conduct box-constrained optimization to $\boldsymbol{\eta}_w$ over the feasible region [0, 1]. In other words, EM does significantly simplify the optimization over parameter λs, but is still not able to complete the complex optimization over parameter $\boldsymbol{\eta}_w$.

In terms of MCMC, it is a competitive candidate for parameter estimation, especially for complex models. By constructing a Markov chain that has the desired distribution as its equilibrium distribution (i.e., the posterior distribution w.r.t. the model parameter λs and $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$), samples of the desired distribution can be obtained by observing the chain after a number of steps. Then we estimate the parameters of the posterior distribution by calculating the sufficient moments of the collected samples. According to the law of large numbers, the more samples collected, the more closely the moments of the sample should match the actual moments of desired distribution. However, due to the intrinsic properties of large-scale samples (large W and large N_w in Eq. (4)) and the high dimensionality of the parameters (large number of objects, large number of crowd workers) in our problem, MCMC's sampling process would become extremely inefficient. Therefore, MCMC is not a good option for Eq. (4).

The above difficulties prompted us to reject common practices and seek a tailor-made, but powerful, solution for our specific problem.

Bayesian moment matching (BMM) (Jaini et al. 2016) is a Bayesian approach used to estimate the model parameters. Specifically, it estimates the parameters of the approximate posterior by matching a set of sufficient moments of the exact complex posterior. Therefore, BMM can be viewed as an equivalent substitution of MCMC from the perspective of moment matching: BMM resorts to approximation to match the moments, while MCMC leverages the collected samples to match the moments. Under the independence assumption for samples, BMM can be further extended to the sequential update strategy, OnlineBMM (see Fig. 5). That is, the approximate posterior is updated after each sample instead of the whole dataset. Therefore, BMM has some inherent advantages over MCMC when dealing with large-scale datasets. In terms of the inefficiency of sampling-based methods (i.e. MCMC) for parameters with high dimensionality, the optimization-based methods (i.e. BMM) can naturally circumvent the curse of dimensionality. Further, based on the Bayesian theorem, BMM only needs to process the whole dataset once (see Fig. 6) and can be updated for new samples online. These advantages prompted us to consider BMM as a basic framework for Eq. (4). See Sect. 4 for more details.

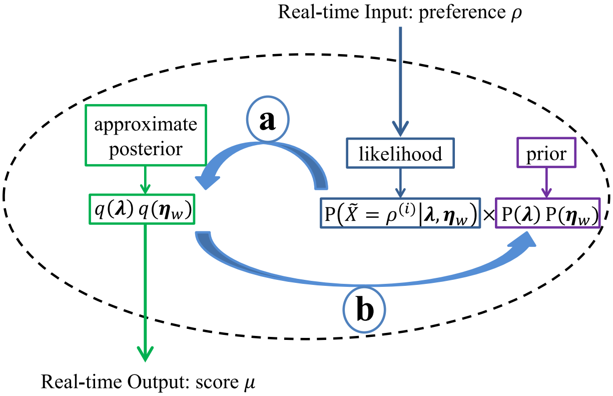
Real-time Input: preference ρ
approximate posterior
a
likelihood
prior
$q(\boldsymbol{\lambda})\, q(\boldsymbol{\eta}_w)$
$\mathrm{P}(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w) \times \mathrm{P}(\boldsymbol{\lambda})\, \mathrm{P}(\boldsymbol{\eta}_w)$
b
Real-time Output: score μ

Fig. 6 Online Generalized Bayesian moment matching (OnlineGBMM) for COUPLE: Step (**a**) estimate $q(\boldsymbol{\lambda})q(\boldsymbol{\eta}_w)$ with generalized Bayesian moment matching; Step (**b**) replace prior $P(\boldsymbol{\lambda})P(\boldsymbol{\eta}_w)$ with approximate posterior $q(\boldsymbol{\lambda})q(\boldsymbol{\eta}_w)$

3.3 Connection to related models

Connection to Plackett–Luce model If worker w is an expert, then we have $\eta_w^1 \approx 1$, which means worker w selects the local winner in each stage with no hesitation. That is, in a general stage i of a k-ary preference ρ, we have $P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) = \sum_{t=i}^{k} \bar{\eta}_w^{(t-i+1)} P\left(X = \rho^{(t)} | \boldsymbol{\lambda}\right) \approx P\left(X = \rho^{(t)} | \boldsymbol{\lambda}\right)$ for worker w. Therefore, COUPLE (4) degenerates into the vanilla Plackett–Luce model (1) when dealing with preferences from domain experts.

Connection to CrowdBT CrowdBT extends the Bradley–Terry model to aggregate pairwise preferences by considering the quality of the worker (Chen et al. 2013). The worker quality actually denotes the probability that worker w agrees with the ground truth; while COUPLE directly integrates worker quality into the Plackett–Luce model with an uncertainty vector $\boldsymbol{\eta}_w$ for each worker w. The uncertainty vector represents worker w's indecision about selecting the "local winner" in each stage. When COUPLE deals with pairwise preferences, we have $|\boldsymbol{\eta}_w| = K = \max_{w,n} l_{\rho_{n,w}} \equiv 2, \forall w \in \{1, 2, \ldots, W\}$ and $\forall n \in \{1, 2, \ldots, N_w\}$. According to the definition of $\boldsymbol{\eta}_w$ in Sect. 3.2.2, η_w^1 represents the conditional probability that the object ranked first according to the worker's belief also accords with the ground truth. Therefore, η_w^1 also reveals the accuracy of worker w. Overall, our COUPLE (4) degenerates into CrowdBT when dealing with pairwise preferences.

Connection to methods in Raman and Joachims (2014) COUPLE focuses on modelling the human annotation process and aims to recover the ground truth from the noisy preferences. Whereas, the methods in Raman and Joachims (2014) try to reduce the negative impact of noisy preferences. In other words, COUPLE: trusts high-quality preferences from expert workers; recovers the ground truth for low-quality preferences from amateur or malicious workers; and reduces the negative impact of random preferences from spammer workers. The methods proposed by Raman and Joachims (2014) trust the high-quality preferences from expert workers just as COUPLE does, but indiscriminately reduce the impact of low-quality preferences from non-expert workers. Therefore, benefiting from our fine-grained categorization of noisy workers, COUPLE can distil more useful information from the noisy preferences.

Connection to classical mixture models COUPLE operates on the assumption of a single ground truth, where the ground truth preference is a unanimous global preference shared by the vast majority of workers. Therefore, workers whose preferences are consistent with the ground truth preference are classified as experts; otherwise they are classified as noisy workers. Although the heterogeneity is a very common phenomenon in human annotation data, the proportions of the mixture components are distributed quite unevenly among the annotations (Turner and Miller 2012; Vitelli et al. 2014; Khare et al. 2015). Usually, most workers agree with the major component, while the remaining few workers agree with one of the other minor components. If we adopt the classical mixture formulation, the model is easy to underfit each minor component due to its insufficient number of supporting samples. In addition, most of the time, only the major component supported by the majority of the workers needs to be estimated. Therefore, we have assumed a single ground truth rather than the multiple ground truths assumed in the classical mixture formulation. Further, we have introduced a worker-specific uncertainty vector to weaken the influence of the minor components, This vector identifies the minority workers as the noisy workers and eliminates their preferences during the aggregation process. Specifically, for a crowd worker w who agrees with the major component (the ground-truth), the first entry η_w^1 of his/her uncertainty vector is close to 1, which denotes that he/she is an expert; while for a crowd worker w who agrees with one of the minor components, he/she would be classified as noisy worker, since η_w^1 does not dominate his/her uncertainty vector η_w. Section 3.2.3 contains some details on an even finer-grained categorization.

4 The online generalized Bayesian moment matching (OnlineGBMM) for COUPLE

Bayesian moment matching (Jaini et al. 2016) is a scalable technique for estimating a model's parameters. It estimates the approximate posterior by matching a set of sufficient moments of the exact complex posterior after each observation. However, due to the non-conjugate likelihood function (Plackett–Luce model), the moments for score $\boldsymbol{\lambda}$ have no closed-form integrations. To address this issue, we have combined COUPLE with a generalized Bayesian moment matching (GBMM) technique that helps to circumvent the need to compute some of the intractable moments. Based on the efficient posterior updating procedures, we introduce our OnlineGBMM algorithm, which makes COUPLE scale to large-scale datasets.

4.1 Main routine of Bayesian moment matching (BMM)

As shown in Fig. 5, we first extend COUPLE to its Bayesian version.[6] Specifically, we introduce a Normal prior $N(\lambda_r|\mu_r, \sigma_r^2)$ for each score $\lambda_r, r = 1, 2, \ldots, L$ and a Dirichlet prior $Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)$ for each uncertainty vector $\boldsymbol{\eta}_w, w = 1, 2, \ldots, W$.

Benefiting from the stagewise learning strategy, we can decompose a crowdsourced preference ρ into a series of sequential stages, and update one stage instead of the entire preference each time. Generally, the likelihood function for a general stage i of preference ρ is $P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}, \boldsymbol{\eta}_w\right)$ (See Eq. (3)). Accordingly, the posterior can be represented as follows,

[6] Here, we clarify that the Bayesian version of COUPLE is different from Thurstonian model (Maydeu-Olivares 1999). Although COUPLE and Thurstonian model all adopt the single ground-truth assumption, the hyperparameters σ_i^2 estimated by COUPLE are completely independent of workers, while Thurstonian model will learn a worker-specific variance $\sigma_{i,w}^2$ for each worker w.

$$P\left(\boldsymbol{\lambda}, \boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right) = \frac{P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}, \boldsymbol{\eta}_w\right) \prod_{r=1}^{L} N(\lambda_r|\mu_r, \sigma_r^2) Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}{P\left(\tilde{X} = \rho^{(i)}\right)}. \tag{5}$$

The main issue with Eq. (5) is that the posterior $P\left(\boldsymbol{\lambda}, \boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right)$ is hard to compute. To keep the computation tractable, we project the posterior into the same form with the prior (product of a Dirichlet with normals), by matching a set of sufficient moments of the approximate posterior with the exact posterior (See Fig. 5):

1. Matching the moments between $q(\boldsymbol{\eta}_w)$ and $P\left(\boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right)$. As $\boldsymbol{\eta}_w$ subjects to Dirichlet distribution, the approximate posterior $q(\boldsymbol{\eta}_w)$ needs to satisfy the moment constraints: $\int \eta_w^t q(\boldsymbol{\eta}_w)\, d\boldsymbol{\eta}_w = \int \eta_w^t P\left(\boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right) d\boldsymbol{\eta}_w$ and $\int (\eta_w^t)^2 q(\boldsymbol{\eta}_w)\, d\boldsymbol{\eta}_w = \int (\eta_w^t)^2 P\left(\boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right) d\boldsymbol{\eta}_w$, $t = 1, 2, \ldots, K$. Fortunately, we can solve the constraints with closed-form integrations (Rashwan et al. 2016), obtaining the posterior parameters $(\boldsymbol{\alpha}_w)^{new}$ accordingly.
2. Matching the moments between $q(\boldsymbol{\lambda})$ and $P\left(\boldsymbol{\lambda}|\tilde{X} = \rho^{(i)}\right)$. As $\boldsymbol{\lambda}$ subjects to Normal distribution, a set of sufficient moment constraints is: $\boldsymbol{\mu} = \int \boldsymbol{\lambda} q(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} = \int \boldsymbol{\lambda} P\left(\boldsymbol{\lambda}|\tilde{X} = \rho^{(i)}\right) d\boldsymbol{\lambda}$ and $\Sigma = \int (\boldsymbol{\lambda} - \boldsymbol{\mu})(\boldsymbol{\lambda} - \boldsymbol{\mu})^T q(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} = \int (\boldsymbol{\lambda} - \boldsymbol{\mu})(\boldsymbol{\lambda} - \boldsymbol{\mu})^T P\left(\boldsymbol{\lambda}|\tilde{X} = \rho^{(i)}\right) d\boldsymbol{\lambda}$.

However, due to the non-conjugacy between the likelihood[7] $P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}\right)$ (Eq. (3)) and normal prior $\prod_{r=1}^{L} N(\lambda_r|\mu_r, \sigma_r^2)$, the posterior $P\left(\boldsymbol{\lambda}|\tilde{X} = \rho^{(i)}\right)$ is too complex. Therefore, the posterior parameters $(\boldsymbol{\mu}^{new}, \Sigma^{new})$ cannot be computed analytically, because the integrations in the moment constraints are intractable.

4.2 Generalized Bayesian moment matching (GBMM)

In cases with a non-conjugate likelihood with a normal prior, we have followed the strategy introduced by Weng and Lin (2011). Weng and Lin (2011) proposed an efficient Bayesian approximation method based on Stein's Lemma (Woodroofe et al. 1989) to estimate the posterior parameters analytically.

Lemma 2 (Corollary 2 in Weng and Lin (2011)) *Let* $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_L)^T$ *be a random vector, where each entry is independent and* $Z_r \sim N(0, 1)$, $r = 1, 2, \ldots, L$. *Suppose that* $f(\mathbf{Z})$ *is the likelihood function and almost twice differentiable. Then, the mean and the variance of the posterior distribution can be approximated as*

$$E[\mathbf{Z}] = E\left[\frac{\nabla f(\mathbf{Z})}{f(\mathbf{Z})}\right], \tag{6a}$$

$$E[Z_p Z_q] = \mathbf{1}_{pq} + E\left[\frac{\nabla^2 f(\mathbf{Z})}{f(\mathbf{Z})}\right]_{pq}, \quad p, q = 1, \ldots, L \tag{6b}$$

where $\mathbf{1}_{pq} = 1$ *if* $p = q$ *and* 0 *otherwise, and* $[.]_{pq}$ *indicates the* (p, q) *component of a matrix.* □

[7] $P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}\right) = E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}\left[P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\lambda}, \boldsymbol{\eta}_w\right)\right]$.

In cases with a general normal prior, we instantiate Lemma 2 with COUPLE, and introduce Proposition 1 to deal with more general situations.

Proposition 1 *Let* $\mathbf{Z} = (Z_1, Z_2, \dots, Z_L)^T$*, where* $Z_r = \frac{\lambda_r - \mu_r}{\sigma_r} \sim N(0, 1)$*,* $r = 1, 2, \dots, L$*. Assume* $l(\mathbf{Z})$ *is the likelihood* $P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}\right)$ *and almost twice differentiable. Upon the completion of stage* i*, the posterior parameters* $(\mu_r^{new}, (\sigma_r^2)^{new})$ *of score* λ_r *can be estimated as:*

$$\mu_r^{new} = \mu_r + \sigma_r E\left[\frac{\partial l(\mathbf{Z})/\partial Z_r}{l(\mathbf{Z})}\right], \tag{7a}$$

$$(\sigma_r^2)^{new} = \sigma_r^2 \left(1 + E\left[\frac{\partial^2 l(\mathbf{Z})/\partial^2 Z_r}{l(\mathbf{Z})}\right]_{rr} - E\left[\frac{\partial l(\mathbf{Z})/\partial Z_r}{l(\mathbf{Z})}\right]^2\right), \tag{7b}$$

where $r = 1, 2, \dots, L$*.*

Proof sketch Substituting Z_r in Lemma 6a with general form $\lambda_r = \mu_r + \sigma_r * Z_r$ and replace the likelihood function, will result in Proposition 7a after simplifying. Similarly, we can result in Proposition 7b with the same procedure. □

According to the Bayesian approximation method introduced in Proposition 1, the posterior parameters $(\boldsymbol{\mu}^{new}, \Sigma^{new})$ of the approximate posterior $q(\boldsymbol{\lambda})$ can be estimated by a differential operation instead of an integral operation. Therefore, our GBMM can handle complex situations where the likelihood function is only required to be almost twice differentiable.

4.3 Posterior update

Given a crowdsourced k-ary preference ρ, we first decouple the complex likelihood into independent stages according to the stagewise learning strategy, and then update the hyperparameters in stages. In a general stage i, we first update the hyperparameters $\boldsymbol{\alpha}_w$, then update the hyperparameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.

4.3.1 Quality update for hyperparameters $\boldsymbol{\alpha}_w$

To update the hyperparameters $\boldsymbol{\alpha}_w$, we first integrate out $\boldsymbol{\lambda}$ to obtain the intermediate likelihood[8] $P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\eta}_w\right) = E_{N(\boldsymbol{\lambda}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)}\left[P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\lambda}, \boldsymbol{\eta}_w\right)\right] = \sum_{t=1}^{(k-i+1)} (\eta_w^t \times R_t)$, where $R_t = E_{N(\boldsymbol{\lambda}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)}[e^{\lambda_{\rho(i+t-1)}} / \sum_{m=i}^{k} e^{\lambda_{\rho m}}]$. Note that in a Bayesian framework, we do not directly conduct the renormalization on η_w, but rather choose a tailor-designed prior distribution, which yields the same effect. Further, R_t can be calculated by its 2nd-order Taylor approximation at $\boldsymbol{\mu}$. See the Appendix for more detailed explanations.

Let $R = P\left(\tilde{X} = \rho^{(i)}\right) = E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}\left[P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\eta}_w\right)\right]$ be the normalization constant, then the posterior distribution of $\boldsymbol{\eta}_w$ can be represented as:

$$P\left(\boldsymbol{\eta}_w | \tilde{X} = \rho^{(i)}\right) = \frac{P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\eta}_w\right) Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)}{P\left(\tilde{X} = \rho^{(i)}\right)} = \frac{P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\eta}_w\right) Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)}{R}. \tag{8}$$

[8] "Intermediate likelihood" denotes the corresponding likelihood respect to a single stage instead of the whole preference or the whole dataset.

Note that we only need to calculate the moments of the first $(k-i+1)$ entries of α_w, because the intermediate likelihood $P\left(\tilde{X}=\rho^{(i)}|\boldsymbol{\eta}_w\right)$ only depends on the first $(k-i+1)$ entries of η_w. According to Sect. 4.1, the sufficient moments $(E[\eta_w^t], E[(\eta_w^t)^2])$[9] for hyperparameter α_w^t can be calculated analytically (Bishop 2006) as follows:

$$\begin{aligned} E[\eta_w^t] &= \int \eta_w^t P\left(\boldsymbol{\eta}_w|\tilde{X}=\rho^{(i)}\right) d\boldsymbol{\eta}_w \\ &= \frac{\alpha_w^t\left(\sum_{v=1}^{(k-i+1)}(R_v \times \alpha_w^v)+R_t\right)}{R\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v+1\right)\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v\right)}, \qquad (9a) \\ E[(\eta_w^t)^2] &= \int (\eta_w^t)^2 P\left(\boldsymbol{\eta}_w|\tilde{X}=\rho^{(i)}\right) d\boldsymbol{\eta}_w \\ &= \frac{\alpha_w^t(\alpha_w^t+1)\left(\sum_{v=1}^{(k-i+1)}(R_v \times \alpha_w^v)+2R_t\right)}{R\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v+2\right)\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v+1\right)\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v\right)}. \qquad (9b) \end{aligned}$$

Then, we update the hyperparameter α_w^t of the Dirichlet distribution $Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)$ as follow:

$$(\alpha_w^t)^{new} = \frac{\left(E[\eta_w^t]-E[(\eta_w^t)^2]\right)E[\eta_w^t]}{E[(\eta_w^t)^2]-(E[\eta_w^t])^2}, \qquad (10)$$

where $t \in \{1, 2, \ldots, k-i+1\}$.

4.3.2 *Score update for hyperparameters* $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

To update the hyperparameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, we first integrate out $\boldsymbol{\eta}_w$ to obtain the intermediate likelihood $l(\boldsymbol{\lambda}) = E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}\left[P\left(\tilde{X}=\rho^{(i)}|\boldsymbol{\lambda},\boldsymbol{\eta}_w\right)\right] = \frac{\sum_{r=i}^{k}(\alpha_w^{(r-i+1)}\times e^{\lambda}{}_{\rho^{(r)}})}{(\sum_{v=1}^{(k-i+1)}\alpha_w^v)\times(\sum_{m=i}^{k}e^{\lambda}{}_{\rho^{(m)}})}$. Note that only the moments of the scores, which are involved in the intermediate likelihood $l(\boldsymbol{\lambda})$, will change during each stage update. Let $\boldsymbol{z} = z_{1:L}$, where $\boldsymbol{z} = \frac{\boldsymbol{\lambda}-\boldsymbol{\mu}}{\boldsymbol{\sigma}} \sim N(\mathbf{0}, \mathbf{1})$. According to Eq. (7a), we can directly calculate the posterior parameter $(\mu_{\rho^{(r)}})^{new}$ as follows:

$$(\mu_{\rho^{(r)}})^{new} \approx \mu_{\rho^{(r)}} + \sigma_{\rho^{(r)}}\frac{\partial l(\boldsymbol{z})/\partial z_{\rho^{(r)}}}{l(\boldsymbol{z})}\Big|_{\boldsymbol{z}=\mathbf{0}} = \mu_{\rho^{(r)}} + \sigma_{\rho^{(r)}}^2\left(\frac{\alpha_w^{(r-i+1)}\times e^{\mu_{\rho^{(r)}}}}{\Psi} - \frac{e^{\mu_{\rho^{(r)}}}}{\psi}\right), \qquad (11)$$

where $r \in \{i, i+1, \ldots, k\}$, $\psi = \sum_{m=i}^{k} e^{\mu_{\rho^{(m)}}}$ and $\Psi = \sum_{m=i}^{k}(\alpha_w^{m-i+1}\times e^{\mu_{\rho^{(m)}}})$. We set $\boldsymbol{z}=\mathbf{0}$, so that $\boldsymbol{\lambda}$ is replaced by $\boldsymbol{\mu}$. Such an approximation is reasonable as we expect that the posterior density of $\boldsymbol{\lambda}$ to be concentrated on $\boldsymbol{\mu}$ (Weng and Lin 2011). According to Eq. (7b), we can directly estimate the posterior parameter $(\sigma_{\rho^{(r)}}^2)^{new}$ as follows:

$$\begin{aligned} (\sigma_{\rho^{(r)}}^2)^{new} &\approx \sigma_{\rho^{(r)}}^2\left(1+\frac{\partial^2 l(\boldsymbol{z})/\partial^2 z_{\rho^{(r)}}}{l(\boldsymbol{z})}\Big|_{\boldsymbol{z}=\mathbf{0}} - \left(\frac{\partial l(\boldsymbol{z})/\partial z_{\rho^{(r)}}}{l(\boldsymbol{z})}\Big|_{\boldsymbol{z}=\mathbf{0}}\right)^2\right) \\ &\approx \sigma_{\rho^{(r)}}^2 \max\left(1+\sigma_{\rho^{(r)}}^2\left(\frac{\alpha_w^{(r-i+1)}\times e^{\mu_{\rho^{(r)}}}(\Psi-\alpha_w^{(r-i+1)}\times e^{\mu_{\rho^{(r)}}})}{\Psi^2}\right.\right. \end{aligned}$$

[9] See "Appendix" for detailed derivation.

$$-\frac{e^{\mu_{\rho(r)}}(\psi - e^{\mu_{\rho(r)}})}{\psi^2}\Bigg), \kappa\Bigg), \tag{12}$$

where $r \in \{i, i+1, \ldots, k\}$ and κ is a positive value to ensure a positive variance.

4.4 The OnlineGBMM algorithm

The OnlineGBMM for COUPLE model (Fig. 6) is summarized in Algorithm1 according to the above analysis. It is notable that the quality update and score update can both be completed with analytic solutions (Eqs. (10), (11), (12)). As a result of the efficient posterior updating procedure, OnlineGBMM allows COUPLE to inherently handle streaming preferences in real-time. Note that in Algorithm 1 we have introduced a reverse update strategy during the preference updates.

Algorithm 1 Online Generalized Bayesian Moment Matching (OnlineGBMM) for COUPLE

```
Initialization: Prior distribution parameters $\{\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \{\boldsymbol{\alpha}_w\}_{w=1}^{W}\}$.
Real-time Input: a $k$-ary preference $\rho$ along with worker index $w$.
    for $stage = k-1, 2, \ldots, 1$ do
        Quality Update: $\boldsymbol{\alpha_w}$ by Eq. (10).
        Score Update: $\boldsymbol{\mu}, \boldsymbol{\sigma}^2$ by Eq. (11), (12).
    end
Real-time Output:Ranking objects by sorting the obtained $\boldsymbol{\mu}$.
```

Remark 3 **Reverse Update Strategy:** Looking at Fig. 4, in stage 1, we have k potential candidates for the local winner for each k-ary preference, so the top k entries of the hyperparameter $\boldsymbol{\alpha}_w$ need to be updated; with the updates of stages, in stage i, we have only $(k-i+1)$ potential candidates, therefore, only the top $(k-i+1)$ entries of $\boldsymbol{\alpha}_w$ need to be updated. This means fewer entries of $\boldsymbol{\alpha}_w$ will be updated in later stages as there are fewer candidates. To better propagate the update information, we propose the reverse update strategy, which update from the highest stage $k-1$ to the lowest stage 1. That is, the top two entries of $\boldsymbol{\alpha}_w$ are updated in stage $k-1$. In stage i, $(k-i+1)$ entries need to be updated, so that the updating information from higher stages propagates to the fresh updated entries through the renormalization trick. In stage 1, all k objects compete to become the local winner, and the top k entries of $\boldsymbol{\alpha}_w$ will be updated. Although the fresh active entry α_w^k is only updated once during the preference updates, it assembles the update information from all former stages through renormalizing.

In Table 2, we compare the computational cost of COUPLE with three state-of-art online ranking models: (1) online Bradley–Terry (OnlineBT) (Weng and Lin 2011). (2) online Plackett–Luce (OnlinePL) (Weng and Lin 2011). (3) CrowdBT (Chen et al. 2013). First, we break each k-ary preference for the BT-based models into C_k^2 all possible pairwise preferences, then aggregate each pairwise preference independently. As all the methods are implemented with online learning, we only consider the computational cost of updating one k-ary preference. Note that we have not reported the computational cost for PeerGrader,[10] because it relies on SGD to estimate the model parameters and needs to process the whole dataset several times to converge. Therefore, as Bayesian methods only need to process the

[10] http://peergrading.org/.

Table 2 Computational cost of COUPLE and other models

	Split number	Score updated	Quality updated	Computational cost
OnlinePL	0	$2k$	0	$2kt_2$
COUPLE	0	$C_k^2 + k - 1$	$4C_k^2 + 5(k-1)$	$\big(5C_k^2 + 6(k-1)\big)t_2$
OnlineBT	C_k^2	2×2	0	$C_k^2 t_1 + 4C_k^2 t_2$
CrowdBT	C_k^2	2×2	7	$C_k^2 t_1 + 11C_k^2 t_2$

We assume that t_1 is the cost of extracting a pairwise preference from a k-ary preference, and t_2 is the cost of completing an update in Algorithm 1. The empirical verification of our analysis is presented in Fig. 8, Sect. 5.5

dataset once, SGD is inferior to the Bayesian online updating methods in terms of efficiency. Fig. 8 provides empirical verification of this comparison for reference.

It is obvious in Table 2 that, of all the methods, CrowdBT has the largest computation cost to update one k-ary preference, while OnlinePL has the lowest computation cost. We further compared the time cost of COUPLE and other models on one real dataset (Sect. 5.5), and the empirical results are consistent with our analysis in Table 2.

5 Experiments

In this section, we evaluate the reliability of COUPLE on four large-scale synthetic datasets, followed by experiments in two real-world applications—ordinal peer grading and online image-rating—to further verify the reliability of COUPLE in real-world situations.

5.1 Experiment setup

Datasets We generated *synthetic datasets* similar to the method described in CrowdBT (Chen et al. 2013). Assume that we have an object set $\Omega = \{O_1, O_2, \ldots, O_L\}$ with the ground-truth preference. Each task, composed of a subset selected randomly from Ω, was corrupted by W crowd workers with different uncertainly vectors $\{\boldsymbol{\eta}_w\}_{w=1}^{W}$ following a Dirichlet distribution $Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_0)$. We controlled the quality of dataset by choosing the proper hyperparameters $\boldsymbol{\alpha}_0$ while retaining diversity among crowd workers.

To verify the reliability of COUPLE in *ordinal peer grading*, we used two *PeerGrading* datasets (PO = Poster, FR = Final Report) introduced by Raman and Joachims (2014). They were collected as part of a senior-undergraduate and masters-level class. There are 42 assignments (objects), 148 students (crowd workers) and 7 TAs participated in the PO dataset. The FR dataset contains 44 assignments (objects), 153 students (crowd workers) and 9 TAs. More information can be found in the "Appendix". This size of class is appropriate, since it is large enough for collecting a substantial number of peer grades, meanwhile, it allows TA gradings to serve as the ground truth.

To further demonstrate the superiority of COUPLE in *online image-rating*, we built a facial image dataset (the *BabyFace* dataset) based on images of children's facial microexpressions with 18 levels from happy to angry. According to our crowdsourced k-ary preference setting, we divided 18 microexpressions into 816 distinct subsets, with each subset including three different microexpressions.[11] We submitted them to Amazon Mechanical Turk and collected the preferences from 105 crowd workers. We only considered workers who have at least 60

[11] We fixed the size of subsets for the convenience of comparing computational cost.

annotations, which yielded a collection of 3074 crowdsourced preferences annotated by 41 crowd workers. Further, we asked seven people to provide a credible unanimous (global) preference of the 18 microexpressions.

Baselines and metrics We compared COUPLE with three online rank aggregation models and two ordinal peer grading methods: (1) online Bradley–Terry (OnlineBT) (Weng and Lin 2011); (2) online Plackett–Luce (OnlinePL) (Weng and Lin 2011); (3) CrowdBT (Chen et al. 2013); and (4) PeerGrader (Raman and Joachims 2014). We adapted the Wilcoxon–Mann–Whitney statistics (Yan et al. 2003) to evaluate the accuracy $\frac{\sum_{i,j}\mathbf{1}(r_i > r_j)\wedge\mathbf{1}(\lambda_i > \lambda_j)}{\sum_{i,j}\mathbf{1}(r_i > r_j)}$, where $r_i > r_j$ represents the ground-truth preference between object i and object j with λ_i as the estimated score for object i.

Parameter initialization We assigned a standard normal prior $N(0, 1)$ for λ_i $\forall i \in \{1, 2, \ldots, L\}$ in all experiments. Inspired by Chen et al. (2013), we initialized each hyperparameter $\boldsymbol{\alpha}_w$ of the uncertainty vector $\boldsymbol{\eta}_w$ $\forall w \in \{1, 2, \ldots, W\}$ with 10 gold tasks with known ground-truth preferences in *synthetic simulations*. This method was also applied to the hyperparameter (α_w, β_w) of worker quality η_w $\forall w \in \{1, 2, \ldots, W\}$ in CrowdBT. The parameter initialization for the *BabyFace* dataset was consistent with the method used for the synthetic simulations.

There is no access to the gold preferences for the *PeerGrading* datasets, as the average number of preferences annotated by each worker is too small (six for PO and two for FR). Fortunately, Raman and Joachims (2014) demonstrated that most students are high-quality (expert) workers with the PO and FR datasets. According to our analysis in Sect. 3.2.2, η_w^i decreases exponentially with i for an expert worker w. Therefore, we assume $\boldsymbol{\alpha}_w = a_0 \times [a^{-1}a^{-2}\ldots a^{-K}]$, where $a_0 > 0$ and $a > 1$, resulting in $E[\eta_w^1] = \frac{1}{1+a^{-1}+\cdots+a^{-(K-1)}}$, where a large a denotes that worker w has a higher degree of confident when making decision.

In terms of the hyperparameter (α_w, β_w) for CrowdBT, we have $E[\eta_w] = \frac{\alpha_w}{\alpha_w+\beta_w}$ for worker w. That is to say, a large α_w represents highly accurate preferences annotated by worker w. For a fair comparison, we set $a_0 = 10$, $a = 6$ in COUPLE and $\alpha_w = 5$, $\beta_w = 1$ $\forall w \in \{1, 2, \ldots, W\}$ in CrowdBT, namely $E[\eta_w^1] \approx 0.83$ and $E[\eta_w] \approx 0.83$ for all workers in COUPLE and CrowdBT, respectively. This parameter initialization is consistent with our assumption that most students are expert workers in *PeerGrading* datasets (PO and FR).

5.2 Empirical results on large-scale synthetic datasets

First, we investigated the reliability of COUPLE on large-scale synthetic datasets. According to the analysis in Sect. 3.2.2, the hyperparameter $\boldsymbol{\alpha}_0$ was set to (5, 1, 0.1, 0.01), (5, 4, 1, 0.1), (5, 4, 3, 3) and (2, 5, 4, 1) to simulate datasets from expert, amateur, spammer and malicious workers, respectively. COUPLE can be applied to large k. We set $k \leq 4$ for better controlling the characteristics of synthetic datasets. We set $L = 1000$, $W = 500$, and assigned each worker $T = 900$ tasks. The number of generated preferences reaches $W \times T = 4.5 \times 10^5$. In addition, we ran the Algorithm 1 with a random sample sequence; the results are presented in Fig. 7. Note that PeerGrader takes too much time to produce a result (Fig. 8), so its accuracy on large-scale synthetic datasets could not be recorded.

Figure 7 shows that: (1) On all settings, COUPLE delivered a performance superior to other baselines; (2) On amateur, spammer and malicious settings, the advantage of COUPLE over CrowdBT became noticeable gradually, since COUPLE is able to correct mis-ordered objects in noisy preferences with a certain probability while CrowdBT discards noisy preferences directly; (3) It is clear that all PL-based models showed minor improvements over the corresponding BT-based models, since BT-based models must break each k-ary preference

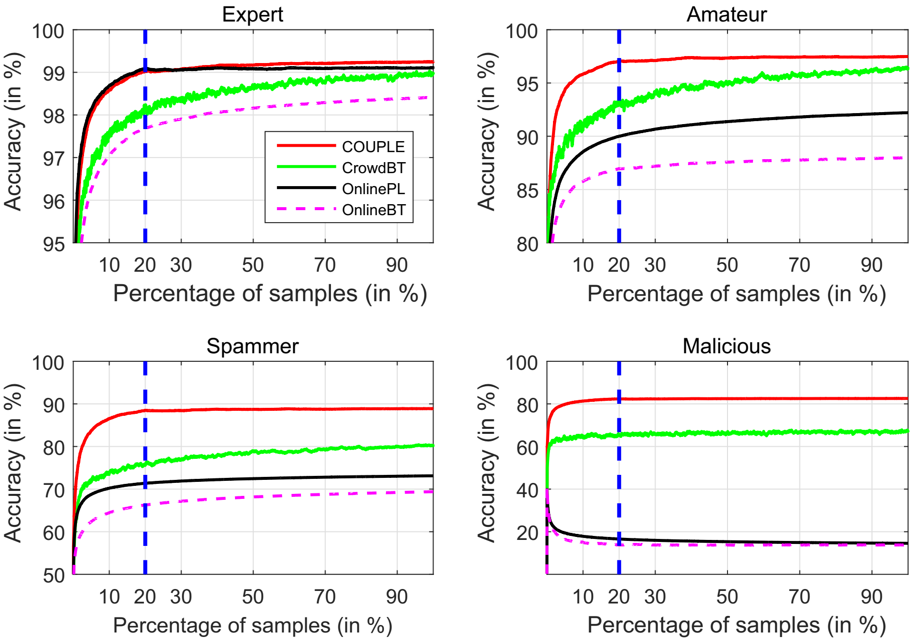

Fig. 7 To verify the *reliability* of COUPLE preliminarily, we provide the accuracy (%) with varying percentage of samples on large-scale synthetic datasets

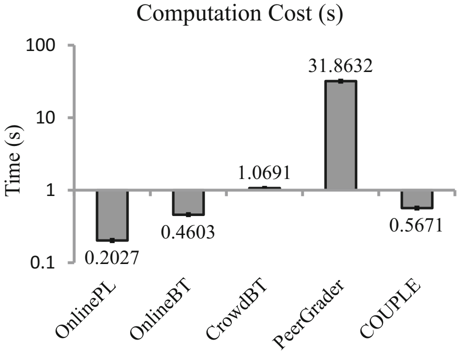

Fig. 8 To verify the *complexity analysis* in Table 2, we collect the time cost of the four models when we conduct the experiment on the *BabyFace* dataset. The time cost is represented by the mean with the standard deviation. As PeerGrader is SGD-based algorithms, we set the iteration number to one and collect the time cost for fair comparison. Empirical results were implemented in Matlab (2015b) with an Intel i5 processor (2.30 GHz) and 8 GB random-access memory (RAM)

into pairwise preferences before aggregation, which may introduce some biases; and (4) The accuracy of COUPLE reached stability at 20% of the samples on all settings. Therefore, COUPLE is able to produce reliable results by aggregating incomplete dataset.

5.3 Empirical results in ordinal peer grading

In this section, we explored the reliability of COUPLE on *PeerGrading* datasets (PO and FR). First, we duplicated the real datasets ten times to reduce the adverse effects of other

Table 3 To verify the *reliability* of COUPLE in ordinal peer grading, we provide the accuracy (%) on two ordinal peer grading datasets, namely PO and FR datasets

Dataset	COUPLE	CrowdBT	OnlinePL	OnlineBT	PeerGrader
PO	81.05 ± 0.99	78.03 ± 0.88	77.66 ± 1.89	73.64 ± 2.99	78.73
FR	78.73 ± 0.58	77.75 ± 0.32	76.68 ± 0.87	71.94 ± 0.91	70.35

Accuracy is represented by a mean with standard deviation. As PeerGrader is SGD-based algorithms, we iterated PeerGrader until convergence and only measured the accuracy only once

Table 4 Six workers with lowest work quality identified by COUPLE, CrowdBT, or PeerGrader on PO and FR datasets, respectively

Dataset	COUPLE	CrowdBT	PeerGrader
Poster (PO)	4, 75, 86, 103, 111, 124	30, 103, 118, 124, 157, 169	12, 52, 79, 103, 112, 124
Final report (FR)	30, 57, 82, 87, 125, 131	1, 57, 125, 131, 134, 141	2, 20, 30, 57, 87, 125

random factors and to ensure all models converge. Then, we ran the experiment 10^3 times to collect the results shown in Table 3.

Table 3 demonstrates that: (1) COUPLE showed obvious advantages over other baselines in terms of reliability; (2) COUPLE and CrowdBT consistently outperformed OnlinePL and OnlineBT, since they both consider worker quality; (3) The PL-based methods were more reliable than the BT-based methods because crowdsourced (noisy) preferences might magnify the effect of statistical inconsistencies, even though a full rank-breaking method was used; and (4) PeerGrader was more accurate than OnlinePL and OnlineBT on the PO dataset, and even higher than CrowdBT. However, on the FR dataset, the PeerGrader's accuracy was inferior to OnlinePL and OnlineBT. Because PeerGrader focuses on the random noise, it cannot accurately model the annotation noise introduced by humans. Hence, it is reasonable that PeerGrader may fail in some real-world applications.

Noisy worker detection According to our definition, COUPLE introduces an uncertainty vector $\boldsymbol{\eta}_w$ for each worker w, where $E\left[\eta_w^1\right] = \alpha_w^1 / \sum_{i=1}^K \alpha_w^i$ represents the probability that worker w selected the ground truth in each stage. Whereas CrowdBT introduces work quality η_w, which denotes the accuracy of the preferences annotated by worker w, where $E[\eta_w] = \frac{\alpha_w}{\alpha_w+\beta_w}$. Furthermore, PeerGrader also introduces a variable η_w, denoting the reliability of each crowd worker w (higher is better). Hereafter, we leverage these three values as indicators to detect noisy workers. Table 4 lists the six lowest-quality workers detected by COUPLE, CrowdBT, and PeerGrader.

It is worth noting that it is impossible to assess the reliability of the noisy workers identified by the three models because no ground truths for these workers are available.

Next, we conducted a series of experiments to verify the efficacy of COUPLE, CrowdBT and PeerGrader in terms of noisy worker detection. For the PO dataset, we first removed the preferences annotated by six noisy workers with the lowest indicator value detected by COUPLE, CrowdBT and PeerGrader. Then, we ran the five models on the cleaned PO dataset again. This process was repeated for FR dataset. We repeated the experiment 10^3 times and collected the results shown in Table 5. All parameters were consistent with the previous experiments on the PO and FR datasets.

In comparing to results for PO dataset in Table 3 to the cleaned PO datasets in Table 5, we can make the following observations: (1) The accuracy of COUPLE stabilized, but the

Table 5 To verify the efficacy of COUPLE and CrowdBT on *noisy worker detection*, we provide the accuracy (%) on cleaned PO and FR datasets

Cleaned dataset	COUPLE	CrowdBT	OnlinePL	OnlineBT	PeerGrader
PO (COUPLE)	81.29 ± 0.53	79.77 ± 0.40	78.59 ± 0.97	77.18 ± 0.86	79.96
PO (CrowdBT)	81.10 ± 0.79	78.72 ± 0.73	78.10 ± 1.29	76.38 ± 0.98	80.49
PO (PeerGrader)	81.69 ± 0.65	81.82 ± 0.9	81.58 ± 0.75	81.29 ± 0.64	79.44
FR (COUPLE)	78.83 ± 0.39	78.60 ± 0.21	78.23 ± 0.46	74.38 ± 0.42	70.50
FR (CrowdBT)	78.76 ± 0.41	78.06 ± 0.30	77.34 ± 0.54	73.00 ± 0.44	70.94
FR (PeerGrader)	76.51 ± 0.27	76.03 ± 0.31	76.18 ± 0.72	72.38 ± 0.35	68.73

The accuracy is represented by a mean with standard deviation. “PO (COUPLE)” denotes the cleaned PO dataset processed by COUPLE. This definition applies to other similar notations

Table 6 To verify the *reliability* of COUPLE in the real-world challenges, we provide the accuracy (%) on the *BabyFace* dataset

Dataset	COUPLE	CrowdBT	OnlinePL	OnlineBT	PeerGrader
BabyFace	92.49 ± 0.35	90.33 ± 0.33	88.24 ± 0.03	$88.24 \pm 3.48 \times 10^{-3}$	92.16
Cleaned *BabyFace*	92.30 ± 0.08	92.06 ± 0.61	92.80 ± 0.05	$92.10 \pm 1.20 \times 10^{-3}$	92.14

To verify the efficacy of COUPLE and other methods on *noisy worker detection*, we provide the accuracy (%) on the cleaned *BabyFace* dataset. The accuracy is represented by the mean with the standard deviation. As PeerGrader is SGD-based algorithms, we iterate PeerGrader until convergence and collect the accuracy only once

accuracy of CrowdBT increased. This means that COUPLE is more reliable than CrowdBT for noisy preferences; (2) The standard deviations of the online models were lower because of the improved quality of the cleaned datasets; (3) The accuracy for CrowdBT, OnlinePL, and OnlineBT with the PO (COUPLE) dataset was higher than that with the PO (CrowdBT) dataset, respectively. This demonstrates that COUPLE is more reliable than CrowdBT for noisy worker detection; (4) Similar to the observations for the original PO dataset, PeerGrader achieved higher or comparable accuracy to CrowdBT on all cleaned PO datasets. However, PeerGrader's accuracy was inferior to OnlinePL and OnlineBT on all cleaned FR datasets. This observation further verifies our analysis in Sect. 3.3 that the series of methods introduced by Raman and Joachims (2014) cannot model the nature of human annotation noise well, and, therefore, those methods are likely to fail in real-world applications.

5.4 Empirical results in online image-rating

We further explored the reliability of COUPLE on the *BabyFace* dataset. Following the experiment on the *PeerGrading* datasets, we first duplicated the *BabyFace* dataset five times to reduce the adverse effects of random unknown factors. Then, we repeated the experiment 10^3 times and measured the accuracy of all models in terms of mean and standard deviation, as shown in Table 6.

From the accuracy of the five models on the *BabyFace* dataset (the second line in Table 6), we observe that: (1) Although COUPLE delivered comparable accuracy to PeerGrader on the *BabyFace* dataset, COUPLE is more efficient because it relies on Bayesian analytical updating rules, while PeerGrader relies on gradient information for updates. (2) COUPLE, CrowdBT, and PeerGrader outperformed OnlinePL and OnlineBT, which demonstrates the superiority

Table 7 Six workers with lowest work quality identified by COUPLE, CrowdBT, or PeerGrader on *BabyFace* datasets

Dataset	COUPLE	CrowdBT	PeerGrader
BabyFace	1, 2, 5, 12, 13, 17	1, 2, 5, 12, 13, 17	1, 2, 5, 12, 13, 17

of the three models in term of modelling worker quality. Since OnlinePL and OnlineBT do not model the quality of crowd workers, their accuracies were easily affected by noisy preferences; and (3) it is interesting to note that the standard derivation of OnlineBT was smaller. Since BT-based methods resort to full rank-breaking to split each k-ary preference into C_k^2 pairwise preferences, OnlineBT generates more preferences and is more stable on small datasets. However, CrowdBT does not enjoy the benefit of rank-breaking; therefore, we conjecture that the noisy preferences would magnify the effect of statistical inconsistency, even with a consistent rank-breaking method.

Noisy worker detection Following the procedure in Sect. 5.3, we leveraged the uncertainty vector $\boldsymbol{\eta}_w$ (COUPLE), worker quality η_w (CrowdBT) and worker reliability η_w (PeerGrader as indicators to identify the noisy workers in *BabyFace* dataset. The six lowest-quality workers, detected by COUPLE, CrowdBT, or PeerGrader, are presented in Table 7.

It is quite interesting that COUPLE, CrowdBT and PeerGrader detected the same six noisy workers for the *BabyFace* dataset. Similar to the setup on the *PeerGrading* datasets, we first removed the preferences annotated by these six noisy workers. Then, we ran the five models on the cleaned *BabyFace* dataset again. We repeated the experiment 10^3 times and collected the results (in the bottom line of Table 6).

In comparing the results with the *BabyFace* dataset in Table 6, we make the following observations: (1) The accuracy of COUPLE stabilized at around 92.30%, which means that COUPLE is reliable for preferences provided by noisy workers; (2) The accuracy of OnlinePL, OnlineBT improved significantly on the cleaned *BabyFace* dataset. It means that the three models indeed detected the actual noisy (low-quality) workers in the dataset. They can be used to clean the noisy dataset by removing the preferences from the detected noisy workers; (3) The standard deviations of all online methods decreased as the quality of the dataset improved; and (4) it is notable that the accuracy of CrowdBT also improved on the cleaned *BabyFace* dataset. It is because CrowdBT tries to reduce the impact of low-quality preferences from crowd workers, while COUPLE aims to recover the ground the truth from noisy preferences. Therefore, COUPLE distils more useful information from the noisy preferences, resulting in higher accuracy with noisy rank aggregation.

5.5 Computation cost

The *BabyFace* dataset is a satisfactory candidate for verifying the complexity analysis, as shown in Table 2. Since the length of the preferences in the *BabyFace* dataset is fixed to three, according to Table 2, the computation cost of processing each preference remains constant for COUPLE and other baselines. Hence, the time cost of all online models increase linearly with the number of samples, which means that the proportional relations between the time cost of four models should be consistent with that in Table 2. We duplicated the *BabyFace* dataset five times and repeated the experiment 10^2 times to measure the time cost in terms of mean and standard deviation for all models, as shown in Fig. 8.

The theoretical analysis in Table 2 is consistent with our observations in Fig. 8: (1) The computation cost of CrowdBT was much higher than that of other models, because CrowdBT needs to break each k-ary preferences into C_k^2 pairwise comparisons before aggregation; (2) The proportion of the time cost between COUPLE and OnlinePL was about 2.8, which is close to the theoretical result $17 : 6$ in Table 2 when $k = 3$. Similar observations can be made of the other pairs; (3) The computation cost of COUPLE was higher than those of OnlinePL and OnlineBT because we introduce an uncertainly vector to model the worker quality. However, COUPLE was more reliable than all other baselines at the acceptable expense of greater computational cost; and (4) The time cost for PeerGrader was much higher than the online updating methods, because online methods update with simple Bayesian analytical updating rules, while PeerGrader uses SGD to solve the batch objective.

6 Conclusion

In this paper, we outline a method to reliably aggregate large-scale noisy preferences annotated by crowd workers into one global preference using a reliable crowdsourced Plackett–Luce model, called (COUPLE) combined with an efficient Bayesian learning technique. To ensure reliability, an uncertainty vector in COUPLE recovers the ground truth from each worker's noisy preferences with a certain level of probability. An online Bayesian moment matching technique ensures that COUPLE scales naturally to large-scale preferences. Empirical results show that COUPLE combined with the OnlineGBMM algorithm delivers substantially more reliable results than current approaches. In the future, we intend to extend this research in several ways. With active learning, different policies for COUPLE could be designed to select samples more wisely, so as to maximize the gain against some criteria. Additionally, a theoretical analysis of OnlineGBMM's convergence rate and the approximation accuracy of GBMM would allow COUPLE to be applied to more complex situations.

Acknowledgements This paper was supported by the ARC Future Fellowship FT130100746, ARC grant LP150100671 and DP180100106.

Appendix

Detailed derivation for Section 4

Renormalization in the Bayesian framework

Suppose $\boldsymbol{\eta}_w = (\eta_w^1, \eta_w^2, \ldots, \eta_w^K)^T$ and $P_0(\boldsymbol{\eta}_w) = Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w) = \frac{\Gamma(\sum_{i=1}^K \alpha_w^i)}{\prod_{i=1}^K \Gamma(\alpha_w^i)} \prod_{i=1}^K (\eta_w^i)^{(\alpha_w^i-1)}$, if $\bar{\eta}_w^i$ is renormalized based on the first k elements of η_w, i.e. $\bar{\eta}_w^i = \frac{\eta_w^i}{\sum_{v=1}^k \eta_w^v}$, $i = 1, 2, \ldots, k$, then we have

$$\bar{\eta}_w^i \sim Dir(\bar{\boldsymbol{\eta}}_w|\bar{\boldsymbol{\alpha}}_w) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_w^i\right)}{\prod_{i=1}^k \Gamma\left(\alpha_w^i\right)} \prod_{i=1}^k \left(\bar{\eta}_w^i\right)^{(\alpha_w^i-1)}$$

where $\bar{\boldsymbol{\eta}}_w = (\bar{\eta}^1, \bar{\eta}^2, \ldots, \bar{\eta}^k)^T$ and $\bar{\boldsymbol{\alpha}}_w = (\alpha^1, \alpha^2, \ldots, \alpha^k)^T$.

Second-order Taylor approximation

Suppose $R_t = E_{N(\boldsymbol{\lambda}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)}\left[\frac{e^{\lambda_t}}{\sum_{i=1}^{L} e^{\lambda_i}}\right]$, where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_L)^T$, then the second-order Taylor approximation of R_t at $\boldsymbol{\mu}$ can be represented as follows:

$$\begin{aligned} R_t &= E_{N(\boldsymbol{\lambda}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)}\left[\frac{e^{\lambda_t}}{\sum_{i=1}^{L} e^{\lambda_i}}\right] \\ &\approx E_{N(\boldsymbol{\lambda}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)}\left[\frac{e^{\mu_t}}{\sum_{i=1}^{L} e^{\mu_i}} + \sum_{i=1}^{L}(\lambda_i - \mu_i)\frac{\partial}{\partial\lambda_i}\frac{e^{\lambda_t}}{\sum_{m=1}^{L} e^{\lambda_m}}\Big|_{\boldsymbol{\lambda}=\boldsymbol{\mu}}\right. \\ &\quad \left. + \frac{1}{2}\sum_{i=1}^{L}(\lambda_i - \mu_i)^2\frac{\partial^2}{\partial\lambda_i^2}\frac{e^{\lambda_t}}{\sum_{m=1}^{L} e^{\lambda_m}}\Big|_{\boldsymbol{\lambda}=\boldsymbol{\mu}}\right] \\ &= \frac{e^{\mu_t}}{\sum_{i=1}^{L} e^{\mu_i}} - \frac{1}{2}\sum_{i=1}^{L}\sigma_i^2\frac{e^{\mu_i}e^{\mu_t}}{\left(\sum_{m=1}^{L} e^{\mu_m}\right)^2} + \frac{1}{2}\sigma_t^2\frac{e^{\mu_t}\left(\sum_{m=1}^{L} e^{\mu_m}\right)}{\left(\sum_{m=1}^{L} e^{\mu_m}\right)^2} \end{aligned},$$

Then we do the following normalized nonnegative transformation to R_t,

$$R_t = \frac{\max(R_t, \kappa_2)}{\sum_{i=1}^{L}\max(R_i, \kappa_2)} \qquad t = 1, 2, \ldots, L,$$

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

Data distribution evidence

$$R = P\left(\tilde{X} = \rho^{(i)}\right) = E_{Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}\left[P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\eta}_w\right)\right] = \sum_{t=1}^{(k-i+1)}\left(\frac{\alpha_w^t}{\sum_{m=1}^{(k-i+1)}\alpha_w^m} \times R_t\right)$$

Moments of Dirichlet distribution

$$\begin{aligned} E[\eta_w^t] &= \int \eta_w^t P\left(\boldsymbol{\eta}_w|\tilde{X} = \rho^{(i)}\right)d\boldsymbol{\eta}_w = \int \eta_w^t\frac{P\left(\tilde{X} = \rho^{(i)}|\boldsymbol{\eta}_w\right)Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)}{R}d\boldsymbol{\eta}_w \\ &= \frac{1}{R}\int \eta_w^t\sum_{v=1}^{(k-i+1)}\left(\eta_w^v \times R_v\right)Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)d\boldsymbol{\eta}_w \\ &= \frac{1}{R}\sum_{v=1}^{(k-i+1)}R_v\int \eta_w^t\eta_w^v Dir(\boldsymbol{\eta}_w|\boldsymbol{\alpha}_w)d\boldsymbol{\eta}_w \\ &= \frac{1}{R}\sum_{v=1}^{(k-i+1)}R_v\int \eta_w^t\eta_w^v\frac{\Gamma\left(\sum_{m=1}^{(k-i+1)}\alpha_w^m\right)}{\prod_{m=1}^{(k-i+1)}\Gamma(\alpha_w^m)}\prod_{m=1}^{(k-i+1)}(\eta_w^m)^{(\alpha_w^m-1)}d\boldsymbol{\eta}_w \\ &= \frac{1}{R}\left(\frac{\sum_{v=1}^{(k-i+1)}R_v\alpha_w^t\alpha_w^v + R_v\alpha_w^t}{\left(\sum_{m=1}^{(k-i+1)}\alpha_w^m + 1\right)\left(\sum_{m=1}^{(k-i+1)}\alpha_w^m\right)}\right) \\ &= \frac{\alpha_w^t\left(\sum_{v=1}^{(k-i+1)}(R_v \times \alpha_w^v) + R_t\right)}{R\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v + 1\right)\left(\sum_{v=1}^{(k-i+1)}\alpha_w^v\right)}. \end{aligned}$$

Table 8 Statistics for the poster (PO) and final report (FR) datasets from the classroom experiment

Dataset	Objects (L)	Workers (W)	TAs
PO	42	148	7
FR	44	153	9

$$
\begin{aligned}
E[(\eta_w^t)^2] &= \int (\eta_w^t)^2 P\left(\boldsymbol{\eta}_w | \tilde{X} = \rho^{(i)}\right) d\boldsymbol{\eta}_w = \int (\eta_w^t)^2 \frac{P\left(\tilde{X} = \rho^{(i)} | \boldsymbol{\eta}_w\right) Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w)}{R} d\boldsymbol{\eta}_w \\
&= \frac{1}{R} \int (\eta_w^t)^2 \sum_{v=1}^{(k-i+1)} \left(\eta_w^v \times R_v\right) Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w \\
&= \frac{1}{R} \sum_{v=1}^{(k-i+1)} R_v \int (\eta_w^t)^2 \eta_w^v Dir(\boldsymbol{\eta}_w | \boldsymbol{\alpha}_w) d\boldsymbol{\eta}_w \\
&= \frac{1}{R} \sum_{v=1}^{(k-i+1)} R_v \int (\eta_w^t)^2 \eta_w^v \frac{\Gamma\left(\sum_{m=1}^{(k-i+1)} \alpha_w^m\right)}{\prod_{m=1}^{(k-i+1)} \Gamma(\alpha_w^m)} \prod_{m=1}^{(k-i+1)} (\eta_w^m)^{(\alpha_w^m - 1)} d\boldsymbol{\eta}_w \\
&= \frac{1}{R} \left(\frac{\sum_{v=1}^{(k-i+1)} R_v (\alpha_w^t + 1) \alpha_w^t \alpha_w^v + 2 R_v (\alpha_w^t + 1) \alpha_w^t}{\left(\sum_{m=1}^{(k-i+1)} \alpha_w^m + 2\right) \left(\sum_{m=1}^{(k-i+1)} \alpha_w^m + 1\right) \left(\sum_{m=1}^{(k-i+1)} \alpha_w^m\right)} \right) \\
&= \frac{\alpha_w^t (\alpha_w^t + 1) \left(\sum_{v=1}^{(k-i+1)} \left(R_v \times \alpha_w^v\right) + 2 R_t\right)}{R \left(\sum_{v=1}^{(k-i+1)} \alpha_w^v + 2\right) \left(\sum_{v=1}^{(k-i+1)} \alpha_w^v + 1\right) \left(\sum_{v=1}^{(k-i+1)} \alpha_w^v\right)}.
\end{aligned}
$$

Statistics for the PeerGrading datasets

Table 8 summarizes some of the key statistics of the Poster (PO) and Final Report (FR) datasets. More information can be found in Raman and Joachims (2014).

Data preprocessing

All gradings are done on 10-point (cardinal) Likert scale, where 10 denotes “perfect”, 8 “good”, 5 “borderline”, 3 “deficient”and 1 “unsatisfactory”. For instance, preference ρ_w : $O_5(9.0) \gtrsim O_3(7.0) \gtrsim O_2(1.0)$ denotes preference ρ_w was annotated by crowd worker w. However, students are not trained graders so they may be grading on different scales. In particular, different students may have different preconceptions of what constitutes a score 8 if they come from different universities. Therefore, we only use the ordinal information provided by each grader. In addition, as ties are not considered in COUPLE and CrowdBT, the preferences containing ties (equal scores) are splitted into independent preferences. To

Table 9 Number of k-ary preferences in the PO and FR datasets

k-ary preferences	2-ary	3-ary	4-ary	5-ary	6-ary	7-ary	Total number
Poster (PO)	165	348	309	85	4	1	912
Final report (FR)	153	146	17	0	0	0	316

be specific, preference $\rho_w : O_5 \tilde{>} O_3 \tilde{\doteq} O_4 \tilde{>} O_2$ breaks into two preferences $\rho_{1,w} : O_5 \tilde{>} O_3 \tilde{>} O_2$ and $\rho_{2,w} : O_5 \tilde{>} O_4 \tilde{>} O_2$. Detailed information about the two post-processed datasets appears in Table 9.

Poster (PO) dataset contains 912 distinct preferences, , and most have a length of 3 or 4. The Final Report (FR) dataset is much smaller (316), and almost 95% of data are 2-ary and 3-ary preferences. Overall, PO and FR are consistent with the definition of crowdsourced k-ary preferences.

References

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

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

Chen, X., Bennett, P., Collins-Thompson, K., & Horvitz, E. (2013). Pairwise ranking aggregation in a crowdsourced setting. In *WWW*.

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

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR* (pp. 248–255). IEEE.

Desarkar, M. S., Sarkar, S., & Mitra, P. (2016). Preference relations based unsupervised rank aggregation for metasearch. *Expert Systems with Applications*, *49*, 86–98.

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

Elo, A. E. (1978). *The rating of chessplayers, past and present*. Nagoya: Arco Pub.

Fligner, M. A., & Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society Series B (Methodological)*, *48*, 359–369.

Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *48*(3), 377–394.

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

Herbrich, R., Minka, T., & Graepel, T. (2007). Trueskill$^{\text{TM}}$: A Bayesian skill rating system. In *NIPS* (pp. 569–576).

Jaini, P., Chen, Z., Carbajal, P., Law, E., Middleton, L., Regan, K., Schaekermann, M., Trimponias, G., Tung, J., & Poupart, P. (2016). Online Bayesian transfer learning for sequential data modeling. In *ICLR*

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*.

Khare, R., Good, B. M., Leaman, R., Su, A. I., & Lu, Z. (2015). Crowdsourcing in biomedicine: Challenges and opportunities. *Briefings in Bioinformatics*, *17*(1), 23–32.

Khetan, A., & Oh, S. (2016). Data-driven rank breaking for efficient rank aggregation. *Journal of Machine Learning Research*, *17*(193), 1–54.

Knight, H., & Keith, O. (2005). Ranking facial attractiveness. *The European Journal of Orthodontics*, *27*(4), 340–348.

Kulkarni, C., Wei, K. P., Le, H., Chia, D., Papadopoulos, K., Cheng, J., et al. (2013). Peer and self assessment in massive online classes. *ACM Transactions on Computer-Human Interaction*, *20*(6), 1–31.

Lijphart, A. (1994). *Electoral systems and party systems: A study of twenty-seven democracies, 1945–1990*. Oxford: Oxford University Press.

Liu, Q., Peng, J., & Ihler, A. T. (2012). Variational inference for crowdsourcing. In *NIPS* (pp. 692–700).

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

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.

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

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

Mollica, C., & Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, *82*, 442–458.

Negahban, S., Oh, S., & Shah, D. (2016). Rank centrality: Ranking from pairwise comparisons. *Operations Research*, *65*(1), 266–287.

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

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

Raman, K., & Joachims, T. (2014). Methods for ordinal peer grading. In *KDD*.

Rashwan, A., Zhao, H., & Poupart, P. (2016). Online and distributed Bayesian moment matching for parameter learning in sum-product networks. In *AISTATS* (pp. 1469–1477).

Richard, B. (2013). Cheap solutions: Managing a co-producing crowd of strangers to solve your problems. *Contemporary perspectives on technical innovation, management and policy* (pp. 261–287).

Saari, D. G. (1999). Explaining all three-alternative voting outcomes. *Journal of Economic Theory*, *87*(2), 313–355.

Shah, N., Balakrishnan, S., Bradley, J., Parekh, A., Ramchandran, K., & Wainwright, M. (2015). Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *AISTATS*.

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*.

Soufiani, H. A., Chen, W., Parkes, D. C., & Xia, L. (2013). Generalized method-of-moments for rank aggregation. In *NIPS* (pp. 2706–2714).

Soufiani, H. A., Parkes, D. C., & Xia, L. (2014). Computing parametric ranking models via rank-breaking. In *ICML* (pp. 360–368).

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

Tsiporkova, E., & Boeva, V. (2006). Multi-step ranking of alternatives in a multi-criteria and multi-expert decision making environment. *Information Sciences*, *176*(18), 2673–2697.

Turner, T. L., & Miller, P. M. (2012). Investigating natural variation in drosophila courtship song by the evolve and resequence approach. *Genetics*, *191*(2), 633–642.

Vitelli, V., Sørensen, Ø., Frigessi, A., & Arjas, E. (2014). Probabilistic preference learning with the mallows rank model. arXiv:1405.7945.

Volkovs, M., & Zemel, R. (2012). A flexible generative model for preference aggregation. In *WWW*.

Vuurens, J., de Vries, A. P., & Eickhoff, C. (2011). How much spam can you take? An analysis of crowdsourcing results to increase accuracy. In *SIGIR Workshop on CIR*.

Wainwright, M. J., & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, *1*, 1–305.

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

Woodroofe, M., et al. (1989). Very weak expansions for sequentially designed experiments: Linear models. *The Annals of Statistics*, *17*(3), 1087–1102.

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

Zagel, C., Piazza, A., Petrov, Y., & Bodendorf, F. (2018). Sciencomat: A gamified research platform for evaluating visual attractiveness. In L. E. Freund & W. Cellary (Eds.), *Advances in the human side of service engineering* (pp. 50–60). Berlin: Springer.