Blind Men and The Elephant: Thurstonian Pairwise Preference for Ranking in Crowdsourcing

Xiaolong Wang*, Jingjing Wang*, Luo Jie†, Chengxiang Zhai*, and Yi Chang†

University of Illinois*
Urbana, IL 61801
{xwang95, jwang112, czhai}@illinois.edu

Yahoo Research†
701 First Avenue, Sunnyvale, CA 94089
{luojie, yichang}@yahoo-inc.com

***Abstract*—Crowdsourcing services make it possible to collect huge amount of annotations from less trained crowd workers in an inexpensive and efficient manner. However, unlike making binary or pairwise judgements, labeling complex structures such as ranked lists by crowd workers is subject to large variance and low efficiency, mainly due to the huge labeling space and the annotators' non-expert nature. Yet ranked lists offer the most informative knowledge for training and testing in various data mining and information retrieval tasks such as *learning to rank*. In this paper, we propose a novel generative model called "Thurstonian Pairwise Preference" (TPP) to infer the true ranked list out of a collection of crowdsourced pairwise annotations. The key challenges that TPP addresses are to resolve the inevitable incompleteness and inconsistency of judgements, as well as to model variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness. Experimental results on both synthetic and real-world datasets demonstrate that TPP can effectively bind pairwise preferences of the crowd into rankings and substantially outperforms previously published methods.**

I. Introduction

Collecting reliable annotation at scale has been a critical issue in the development of machine learning techniques. Crowdsourcing services make it possible to collect huge amount of annotations from less trained crowd workers in an inexpensive and efficient manner. The general philosophy of crowdsourcing is that instead of collecting one single expert-annotated label for each instance, multiple labels per example are collected from non-expert crowd workers at low cost to infer the ground truth [1], [2].

A. Motivation

In different tasks of learning, the form of labels can be as simple as binary/pairwise judgements, but can also be structured and complex. An example of the latter case is a ranked list of documents with respect to a query. *Ranked lists* offer the most informative knowledge for training and testing in various data mining and information retrieval tasks such as *learning to rank* [3], [4]. Nevertheless, unlike making binary or pairwise judgements, labeling complex structures such as ranked lists by crowd workers is subject to large variance and low efficiency. In order to generate a ranked list of N items, a worker needs to consider a number of $N!$ possibilities. Annotation in such a huge labeling space is time consuming and uneconomic. Furthermore, the non-expert nature of crowdsourcing workers makes it even more difficult to reach consensus on the ground-truth ranked lists than binary/pairwise judgements.

The fact that ranked lists are highly useful but hard to be directly annotated motivates us to seek for alternative strategies. Our idea is based upon a metaphor in which we can only *learn* what *an elephant* is like through *a group of blind men*. Each one holds onto a different part, but only one part, such as the side or the tusk. In the original story, they then discuss their observations which leads to argument and complete disagreement. However, a smarter treatment is to analysis all the observations and to find an probable explanation that most fits. In this paper, we implement such idea by decomposing the task of labeling ranked lists into a series of smaller and easier tasks: *annotating pairwise preferences*, each of which requires a worker to compare only a pair of items out of the entire set. In addition, the pairwise judgements by crowd workers are more reliable and can be easily scaled up. Pairs of items can be randomly generated out of the set and will be labeled by multiple workers. The goal is to infer the true *ranked list* out of the *crowdsourced pairwise* annotations.

B. Challenges

Leveraging pairwise preferences to infer the full ranked list is promising but also challenging. The key challenge comes from *incomplete* and *inconsistent* annotations.

Pairwise preferences can be *incomplete* due to time and budget constraints. Not every two items are compared either directly or indirectly (For items A, B and C, an *indirect* annotation of $A \succ B$ may be obtained if *direct* annotations of $A \succ C$ and $C \succ B$ have been given). The available annotations can also be *inconsistent*, resulting from either the disagreements between multiple workers, or the intrinsic uncertainty within one single worker. A common mistake of the latter case is that one labels $A \succ B, B \succ C,$ and $C \succ A$ at the same time.

The discussion below reveals a number of factors that lead to inconsistent annotations:

- *Query difficulty:* More difficult queries, such as ambiguous and vague queries, demand more effort to interpret and to judge, making them intrinsically more prone to errors.
- *Worker expertise across domains:* Different workers have different domain expertise; the same worker can also have varying domain knowledge across different task, making the quality of their labels vary accordingly. In practice, neither the task domain nor the worker's expertise is known apriori.
- *Truthfulness of Workers:* Truthfulness of workers is a prevailing issue in crowdsourcing tasks. Two typical adversarial groups are spammer workers and malicious workers: Spammers give random judgments and offer little information about the ranked lists; Malicious workers, on the other hand, sabotage the utility of annotations by giving false preferences.

Identifying the sources of such incompleteness and inconsistency, and properly modeling them, are critical to infer the true ranked list from the crowdsourced pairwise annotations.

C. Our Proposal

We propose a novel generative model called "Thurstonian Pairwise Preference" (TPP) to bind pairwise preferences of the crowd into rankings. The key modeling challenges that TPP addresses are to resolve the inevitable incompleteness and inconsistency of judgements, as well as to model variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness.

TPP is built on top of the Thurstonian Ranking Model (TRM) [5], which takes noisy ranked lists of items as observations and estimates the true rankings. When applied to crowdsourcing, TRM models the generation of the noisy ranked lists annotated by crowd workers, taking variable query difficulty into account. It infers the relevance score of each item to form the ranked list. In contrast to TRM, the observations of TPP are pairwise preferences. Specifically, TPP naturally simulates the generative process of incomplete pairwise annotations, and seamlessly integrates a worker-aware layer with the original query-aware layer to model the inconsistency of the labeling process. The advantage of TPP is that it does not require full rankings as observations, and pairwise preferences can be efficiently labeled at scale.

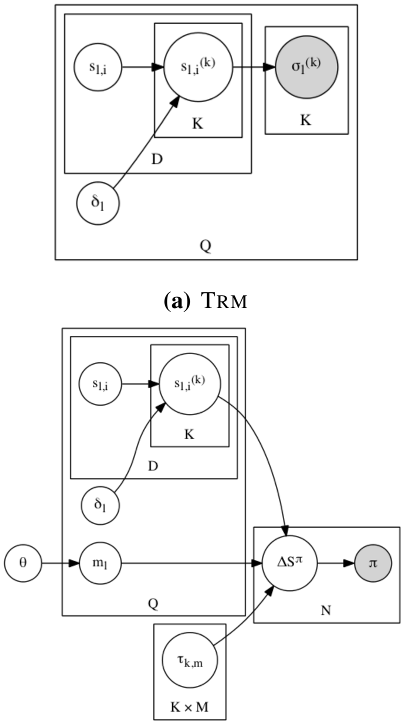
While there have been earlier research efforts on (pairwise) ranking aggregation with similar goals, most of them investigated a "non-crowd" setting, or only a subset of the above factors are taken into account (See Section VI for details). In sharp contrast, TPP provides a unified and principled strategy to handle various influential factors, which effectively binds pairwise preferences of the crowd into rankings.

Organization. We briefly introduce the original Thurstonian Ranking Model in Section II, and present our proposed Thurstonian Pairwise Preference model (TPP) in Section III. The inference of TPP is given in Section IV. We provide the experimental study in Section V, review related work in Section VI and conclude our study in Section VII.

II. Thurstonian Ranking Model

The original Thurstonian ranking model (TRM) [5] is devised for analyzing ordinal data. Suppose in a ranking annotation task, K workers $\{t_k\}_{k=1}^{K}$ are given Q queries $\{q_l\}_{l=1}^{Q}$ and D documents $\{d_i\}_{i=1}^{D}$. It is postulated that the optimal ranked list[1] for query q_l is determined by the *ground truth relevance score* $s_{l,i}$ of each document d_i. Precisely, the larger the value of $s_{l,i}$, the higher rank is assigned to d_i. Each worker t_k produces a ranked list $\sigma_l^{(k)}$ by ordering documents according to his *perceived relevance scores* $s_{l,i}^{(k)}$, which are assumed to be Gaussian distributed: $s_{l,i}^{(k)} \sim \mathrm{N}(s_{l,i}, \delta_l^2)$. The variance δ_l^2 quantifies the *query difficulty* of q_l: δ_l^2 is larger for more difficult query, and the perceived score can deviate more from the ground truth score.

The plate notation of the above generative process is given in Figure 1a. With the workers' annotated rankings $\{\sigma_l^{(k)}\}$ given as observations, the goal of TRM is to infer $\{s_{l,i}\}$ as well as $\{\delta_l^2\}$. Algorithmic development for inference previously investigated includes maximum likelihood estimation [6] and Bayesian inference [7]. A derivation of the maximum likelihood estimation is given in Appendix B.



(a) TRM

(b) TPP

Fig. 1: Plate notation for TRM and TPP

[1] a permutation of documents

TABLE I: Summary of Notations

Notation	Explanation
t_k, q_l, d_i	worker t_k, query q_l and document d_i
$s_{l,i}$	ground truth relevance score of d_i *w.r.t.* q_l
δ_l^2	the difficulty of query q_l
m_l	the domain of query q_l
$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)^T$	the distribution of query domains, $m_l \sim \mathrm{Mult}(\boldsymbol{\theta})$
$\tau_{k,m}$	worker t_k's expertise & truthfulness on domain m
$s_{l,i}^{(k)}$	worker t_k's perceived score of d_i *w.r.t.* q_l
$\pi = \langle k, l, i_1, i_2 \rangle$	pairwise preference π: t_k prefers document d_{i_1} to document d_{i_2} *w.r.t.* q_l
$\tilde{s}_{i_1}^{\pi}, \tilde{s}_{i_2}^{\pi}$	noisy scores of d_{i_1} and d_{i_2} to determine pairwise preference π
Δs^{π}	noisy score difference $\Delta s^{\pi} = \tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi}$
$\mathbf{\Theta} = \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\}$	model parameters
$\mathbf{Z} = \{m_l, s_{l,i}^{(k)}\}$	latent variables of interest
$\mathbf{V} = \{\Delta s^{\pi}\}$	auxiliary latent variables
$\mathbf{D} = \{\pi\}$	observations

III. Thurstonian Pairwise Preference

TRM specifies the generation of ranked lists in a crowd-sourced setting, with variable query difficulty taken into account. However, the difficulty in obtaining annotated *ranked lists* makes it hardly applicable in practice. We propose a novel generative model called "Thurstonian Pairwise Preference" (TPP), which extends TRM to accommodate *pairwise preferences* as observations. Meanwhile, TPP seamlessly integrates a *worker-aware* layer with the original query-aware layer to incorporate workers' variable expertise across different domains and their truthfulness, which explains the generation of the inconsistent pairwise preferences at modeling time.

The plate notation of TPP is given in Figure 1b. The notations used throughout this paper are summarized in Table I. Suppose worker t_k compares documents d_{i_1} and d_{i_2} *w.r.t.* query q_l. The pairwise preference π is either t_k prefers d_{i_1} to d_{i_2}, denoted by $\langle k, l, i_1, i_2 \rangle$, or $\pi = \langle k, l, i_2, i_1 \rangle$ if t_k prefers d_{i_2}[2]. The preference depends on query difficulty, as well as the domain expertise and truthfulness of the worker.

TPP first generates the workers' perceived scores in the same way as TRM does. Then it introduces a worker-aware layer to simulate the generation of pairwise annotations, which involves a delicate modeling of query domains. We assume there are M domains. For query q_l, its domain m_l is drawn from a multinomial distribution: $m_l \sim \mathrm{Mult}(\boldsymbol{\theta})$. In order to generate the pairwise preference π, worker t_k generates two noisy scores $\tilde{s}_{i_1}^{\pi}$ and $\tilde{s}_{i_2}^{\pi}$, which are Guassian distributed: $\tilde{s}_{i_1}^{\pi} \sim \mathrm{N}(\mathrm{sgn}(\tau_{k,m_l}) s_{l,i_1}^{(k)}, \tau_{k,m_l}^{-2})$ and $\tilde{s}_{i_2}^{\pi} \sim \mathrm{N}(\mathrm{sgn}(\tau_{k,m_l}) s_{l,i_2}^{(k)}, \tau_{k,m_l}^{-2})$.[3] The parameter $\tau_{k,m}$ encodes worker t_k's expertise and truthfulness on domain m. Specifically, the sign of $\tau_{k,m}$ indicates whether worker t_k is truthful or malicious on domain m. A malicious worker would have a negative $\tau_{k,m}$, giving false preferences by "flipping" his perceived scores. The absolute value of $\tau_{k,m}$ measures the expertise of t_k on m: a larger $|\tau_{k,m}|$ means a smaller variance of the noisy score, i.e., t_k is more knowledgeable on m; for a very small $|\tau_{k,m}|$, the noisy score is nearly uniformly distributed, implying t_k likely to be a spammer. Given the noisy scores $\tilde{s}_{i_1}^{\pi}$ and $\tilde{s}_{i_2}^{\pi}$, the pairwise preference is uniquely determined: $\pi = \langle k, l, i_1, i_2 \rangle$ if $\tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi} \geq 0$ and vice versa. We define the *noisy score difference* in this case as:

$$\Delta s^{\pi} = \tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi} \tag{1}$$

and thus $\mathrm{P}(\pi = \langle k, l, i_1, i_2 \rangle) = \mathrm{P}(\Delta s^{\pi} \geq 0)$.

The generative process of TPP is summarized as follows:

- **Generate Perceived Scores:** Generate worker t_k's perceived score of document d_i *w.r.t.* query q_l: $s_{l,i}^{(k)} \sim \mathrm{N}(s_{l,i}, \delta_l^2)$
- **Generate Query Domains:** For query q_l, draw its domain: $m_l \sim \mathrm{Mult}(\boldsymbol{\theta})$.
- **Generate Noisy Scores:** To compare two documents d_{i_1} and d_{i_2}, worker t_k generate noisy scores $\tilde{s}_{i_1}^{\pi}$ and $\tilde{s}_{i_2}^{\pi}$.

$$\tilde{s}_{i_j}^{\pi} \sim \mathrm{N}(\mathrm{sgn}(\tau_{k,m_l}) s_{l,i_j}^{(k)}, \tau_{k,m_l}^{-2}) \; (j = 1, 2) \tag{2}$$

- **Generate Pairwise Preferences:** The pairwise preference π is determined by the noisy score difference: $\pi = \langle k, l, i_1, i_2 \rangle$ if $\Delta s^{\pi} = \tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi} \geq 0$, and $\pi = \langle k, l, i_2, i_1 \rangle$ if $\Delta s^{\pi} < 0$.

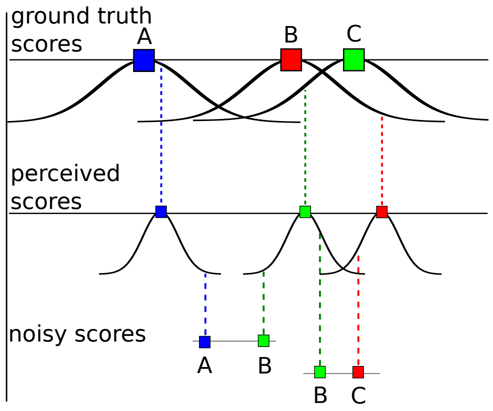


Fig. 2: An Illustration Example of TPP: The generation of two pairwise preferences by a crowd worker for a given query

The true ranking is determined by the ground truth scores of each document. The perceived score of each document is Gaussian distributed based on the true score and the query difficulty. Each time a worker is asked to compare a pair of documents, The perceived scores, together with the domain expertise and truthfulness of the worker, specify another two Gaussian distributions from which the noisy scores are drawn. The pairwise preference is given accordingly. The worker is truthful in this example.

Figure 2 illustrates the generation of two pairwise preferences by a crowd worker for a given query. The ground truth scores for three documents A, B, C imply the true ranking to be $A \prec B \prec C$. The worker's perceived scores deviate from the

[2] We adopt the assumption made in TRM that no ties exist in rankings. However, if two documents are indeed equally relevant, the workers shall randomly prefers either one, and the ground truth relevance scores of the two documents would be close.

[3] $\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}$

ground truth scores due to query difficulty. In fact, the perceived scores imply $A \prec C \prec B$, which contradicts with the true ranking. We further assume that the worker is truthful and has reasonable domain knowledge (This example does not include the generation of query domains for the sake of clarity). The worker generates noisy scores which are close to his perceived scores, and gives pairwise preferences ($A \prec C$, $C \prec B$) accordingly. It is worth noting that a pair of noisy scores are drawn each time a worker judges a pair of documents. Thus TPP respects intra-worker inconsistency as well as inter-worker inconsistency.

IV. Inference

The model parameters $\mathbf{\Theta} = \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\}$ are learned by Maximum Likelihood Estimation (MLE) with the Expectation-Maximization (E-M) [8] algorithm. The posterior distribution of the *latent variables of interest* $\mathbf{Z} = \{m_l, s_{l,i}^{(k)}\}$ given the observations $\mathbf{D} = \{\pi\}$ is approximated via alternate sampling of $\mathbf{Z}$ and the *auxiliary latent variables* $\mathbf{V} = \{\Delta s^\pi\}$. The inference algorithm of TPP is summarized in Algorithm 1.

Algorithm 1: Inference of TPP

```
Input: Pairwise preferences $\mathbf{D}$
Output: Model parameters $\mathbf{\Theta}$
Initialize $\mathbf{V}, \mathbf{Z}, \mathbf{\Theta}$;
while convergence criteria not met do
    (E-step) Sample the posterior distribution of $\mathbf{V}$ and $\mathbf{Z}$;
    (M-step) Update $\mathbf{\Theta}$;
    Model rescaling;
```

A. Model Parametrization

The pairwise preference $\pi = \langle k, l, i_1, i_2 \rangle$ between two documents d_{i_1} and d_{i_2} hinges on $\Delta s^\pi = \tilde{s}_{i_1}^\pi - \tilde{s}_{i_2}^\pi$. We introduce *auxiliary latent variables* $\mathbf{V} = \{\Delta s^\pi\}$ to parameterize TPP.

Our results rely on the following lemma of (truncated) Gaussian distribution, the proof of which can be found in [9]):

Lemma 4.1: If x_1 and x_2 are independently sampled from $x_i \sim \mathrm{N}(\mu_i, \sigma^2)$, $i = \{1, 2\}$, then we have

(a) $x_1 - x_2 \sim \mathrm{N}(\mu_1 - \mu_2, 2\sigma^2)$ and $\mathrm{P}(x_1 - x_2 \geq 0) = Q(-\frac{\mu_1 - \mu_2}{\sqrt{2}\sigma})$, where $Q(\cdot)$ denotes the tail probability of the standard normal distribution: $Q(s) := \Pr(x \geq s)$, $x \sim \mathrm{N}(0, 1)$.

(b) $x_1 - x_2 | x_1 - x_2 \geq 0 \sim \mathrm{TN}_0^\infty(\mu_1 - \mu_2, 2\sigma^2)$, where $\mathrm{TN}_a^b(m, s^2)$ $(a < b, a, b \in \mathbb{R} \cup \{\pm\infty\})$ is the truncated Gaussian distribution bounded by interval (a, b) with the embedded Gaussian distribution being $\mathrm{N}(m, s^2)$. □

Given Eq. 1 and 2, it follows from Lemma 4.1(a) that the auxiliary latent variable Δs^π follows the truncated Gaussian distribution:

$$\begin{aligned} &\tilde{s}_{i_1}^\pi - \tilde{s}_{i_2}^\pi | \tau_{k,m_l}, s_{l,i_1}^{(k)}, s_{l,i_2}^{(k)} \\ &\sim \mathrm{N}\left(\mathrm{sgn}(\tau_{k,m_l})(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)}), 2\tau_{k,m_l}^{-2}\right) \end{aligned} \tag{3}$$

and we have

$$\mathrm{P}(\Delta s^\pi \geq 0 | \tau_{k,m_l}, s_{l,i_1}^{(k)}, s_{l,i_2}^{(k)}) = \mathrm{Q}\left(-\frac{\tau_{k,m_l}}{\sqrt{2}}(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)})\right) \tag{4}$$

In view of the above results, the joint probability of $\mathbf{D}, \mathbf{Z}, \mathbf{V}$ can be factorized as:

$$\begin{aligned} &\mathrm{P}(\mathbf{D}, \mathbf{Z}, \mathbf{V} | \mathbf{\Theta}) = \mathrm{P}(\mathbf{Z}|\mathbf{\Theta})\mathrm{P}(\mathbf{V}|\mathbf{Z})\mathrm{P}(\mathbf{D}|\mathbf{V}) \\ &= \prod_l \mathrm{P}_{\mathtt{Mult}}(m_l | \boldsymbol{\theta}) \cdot \prod_{k,l,i} \mathrm{P}_{\mathtt{N}}(s_{l,i}^{(k)} | s_{l,i}, \delta_l^2) \\ &\prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \Big(\mathrm{P}_{\mathtt{N}}\Big(\Delta s^\pi | \mathrm{sgn}(\tau_{k,m_l})(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)}), 2\tau_{k,m_l}^{-2}\Big) \\ &\prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \mathbf{1}(\Delta s^\pi \geq 0) \end{aligned} \tag{5}$$

Integrating out $\mathbf{V}$, we get the joint probability of the observations $\mathbf{D} = \{\pi\}$ and the latent variables of interest $\mathbf{Z} = \{m_l, s_{l,i}^{(k)}\}$:

$$\begin{aligned} &\mathrm{P}(\mathbf{D}, \mathbf{Z} | \mathbf{\Theta}) = \int_{\mathbf{V}} \mathrm{P}(\mathbf{D}, \mathbf{Z}, \mathbf{V} | \mathbf{\Theta}) \ \mathrm{d}\mathbf{V} \\ &= \prod_l \mathrm{P}_{\mathtt{Mult}}(m_l | \boldsymbol{\theta}) \cdot \prod_{k,l,i} \mathrm{P}_{\mathtt{N}}(s_{l,i}^{(k)} | s_{l,i}, \delta_l^2) \\ &\prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \mathrm{Q}\left(-\frac{\tau_{k,m_l}}{\sqrt{2}}(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)})\right) \end{aligned} \tag{6}$$

The model parameters $\mathbf{\Theta} = \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\}$ are learned by optimizing the log likelihood $\mathbf{\Theta} = \arg\max_{\mathbf{\Theta}} \ln \mathrm{P}(\mathbf{D}|\mathbf{\Theta})$ with the E-M algorithm. At the t-th iteration, the posterior distribution of $\mathbf{Z}|\mathbf{D}, \mathbf{\Theta}^{(t)}$ is computed (E-step), followed by the model update, i.e. maximizing the expected joint log likelihood: $\mathbf{\Theta}^{(t+1)} = \arg\max_{\mathbf{\Theta}} \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)})$ (M-step), where

$$\mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) = \mathbf{E}_{\mathbf{Z}|\mathbf{D}, \mathbf{\Theta}^{(t)}}[\ln \mathrm{P}(\mathbf{D}, \mathbf{Z} | \mathbf{\Theta})] \tag{7}$$

B. Posterior Sampling

The analytic calculation of $\mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)})$ is impossible due to the intractability of $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$. Instead, we approximate the posterior distribution by sampling. Nevertheless, sampling $\mathbf{Z}$ from $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$ is still difficult because we cannot effectively integrate over Eq. 6 to obtain the distribution of $s_{l,i}^{(k)} | \mathbf{Z} \setminus \{s_{l,i}^{(k)}\}, \mathbf{D}, \mathbf{\Theta}$. Therefore, we reintroduce the auxiliary latent variables $\mathbf{V}$. A blocked Gibbs sampler [10] is applied to sample $\mathbf{V}$ and $\mathbf{Z}$. Each block of variables, i.e., query domains $\{m_l\}$, perceived scores $\{s_{l,i}^{(k)}\}$, and noisy score differences $\{\Delta s^\pi\}$, are sampled in sequence.

1) Sample Query Domain m_l: It follows from Eq. 5 that the posterior distribution of the domain m_{l^*} for a query l^* is given by the following multinomial distribution:

$$\begin{aligned} &\mathrm{P}(m_{l^*} = m^* | \mathbf{D}, \mathbf{Z} \setminus \{m_{l^*}\}, \mathbf{V}, \mathbf{\Theta}) \qquad (8) \\ &\propto \ \theta_{m^*} \prod_{\substack{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D} \\ l = l^*}} \mathrm{P}_{\mathtt{N}}\left(\Delta s^\pi | \mathrm{sgn}(\tau_{k,m^*})(s_{l^*,i_1}^{(k)} - s_{l^*,i_2}^{(k)}), 2\tau_{k,m^*}^{-2}\right) \end{aligned}$$

Note that there is no coupling (inter-dependency) among $\{m_l\}$, and the multinomial sampling can be accelerated with parallel implementation.

2) Sample Perceived Score $s_{l,i}^{(k)}$: It follows from Eq. 5 that the posterior distribution for the perceived score is given by

$$\begin{aligned} & \mathrm{P}(s_{l^*,i^*}^{(k^*)} = s^*|\mathbf{D}, \mathbf{Z} \setminus \{s_{l^*,i^*}^{(k^*)}\}, \mathbf{V}, \mathbf{\Theta}) \\ \propto\ & \mathrm{P}_{\mathtt{N}}(s^*|s_{l^*,i^*}, \delta_{l^*}^2) \qquad (9) \\ & \prod_{\pi=\langle k^*,l^*,i^*,i\rangle\in\mathbf{D}} \mathrm{P}_{\mathtt{N}}(\Delta s^\pi|\mathrm{sgn}(\tau_{k^*,m_{l^*}})(s^* - s_{l^*,i}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}) \\ & \prod_{\pi=\langle k^*,l^*,i,i^*\rangle\in\mathbf{D}} \mathrm{P}_{\mathtt{N}}(\Delta s^\pi|\mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i}^{(k^*)} - s^*), 2\tau_{k^*,m_{l^*}}^{-2}) \end{aligned}$$

To derive the sampling rule for perceived score $s_{l,i}^{(k)}$, we employ the following lemma that an exponential-family distribution is uniquely determined by its sufficient statistics and natural parameters [11]:

Lemma 4.2: If $\mathrm{P}(x)$ is a valid distribution and $\mathrm{P}(x) \propto \exp(c_1 x + c_2 x^2)$, then $x \sim \mathrm{N}(-\frac{c_1}{2c_2}, -\frac{1}{2c_2})$ □

And it follows immediately that:

$$s_{l^*,i^*}^{(k^*)} \sim \mathrm{N}(\frac{a_1}{a_2}, \frac{1}{a_2}) \qquad (10)$$

where

$$\begin{aligned} a_1 =\ & \frac{1}{\delta_{l^*}^2} s_{l^*,i^*} \qquad (11) \\ & + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \pi=\langle k^*,l^*,i^*,i\rangle\in\mathbf{D}}} s_{l^*,i}^{(k^*)} + \mathrm{sgn}(\tau_{k^*,m_{l^*}})\Delta s^\pi \right) \\ & + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \pi=\langle k^*,l^*,i,i^*\rangle\in\mathbf{D}}} s_{l^*,i}^{(k^*)} - \mathrm{sgn}(\tau_{k^*,m_{l^*}})\Delta s^\pi \right) \\ a_2 =\ & \frac{1}{\delta_{l^*}^2} + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \langle k^*,l^*,i^*,i\rangle\in\mathbf{D}}} 1 + \sum_{\substack{i \\ \langle k^*,l^*,i,i^*\rangle\in\mathbf{D}}} 1 \right) \qquad (12) \end{aligned}$$

Intuitive Interpretation. Here is an intuitive interpretation of the above calculation which provides more insights into the behaviors of TPP:

First, the mean value $\frac{a_1}{a_2}$ is a weighted average of three sources of estimation:

- s_{l^*,i^*}, the ground truth relevance score (1st term in Eq. 11). It is discounted by the query difficulty $\delta_{l^*}^2$. The easier the query, the more it contributes to the perceived score $s_{l^*,i^*}^{(k^*)}$.
- $\left(s_{l^*,i}^{(k^*)} + \mathrm{sgn}(\tau_{k^*,m_{l^*}})\Delta s^\pi\right)$ where $\pi = \langle k^*, l^*, i^*, i\rangle \in \mathbf{D}$, (2nd term in Eq. 11). It corresponds to a pairwise preference π when t_{k^*} *prefers* d_{i^*} to the other document d_i. It estimates $s_{l^*,i^*}^{(k^*)}$ by combining the perceived score $s_{l^*,i}^{(k^*)}$ of the less preferred document d_i and the noisy score difference $\Delta s^\pi = \tilde{s}_{i^*}^\pi - \tilde{s}_i^\pi$ multiplied by the worker's truthfulness $(\mathrm{sgn}\,(\tau_{k^*,m_{l^*}}))$. This estimation is then weighted by the worker's domain expertise $(\frac{1}{2}\tau_{k^*,m_{l^*}}^2)$.
- The third source of estimation (3rd term in Eq. 11) corresponds to the case when d_{i^*} is less preferred by t_{k^*}. The analysis is analogous to that of the 2nd term.

In addition, the variance $\frac{1}{a_2}$ in Eq. 10 is the harmonic average of the query difficulty $\delta_{l^*}^2$ and the worker's domain expertise $2\tau_{k^*,m_{l^*}}^{-2}$, which determines the uncertainty of the perceived score $s_{l^*,i^*}^{(k^*)}$. The sampled perceived scores are more localized to the mean value $\frac{a_1}{a_2}$ with easier queries and more knowledgeable workers.

3) Sample Noisy Score Difference Δs^π: Denote the pairwise preference by $\pi^* = \langle k^*, l^*, i_1^*, i_2^*\rangle$. It follows from Eq. (5) that

$$\begin{aligned} & \mathrm{P}(\Delta s^{\pi^*} = \Delta s^*|\mathbf{D}, \mathbf{Z}, \mathbf{V} \setminus \{\Delta s^{\pi^*}\}, \mathbf{\Theta}) \qquad (13) \\ \propto\ & \mathrm{P}_{\mathtt{N}}\left(\Delta s^*|\mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i_1^*}^{(k^*)} - s_{l^*,i_2^*}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}\right) \mathbf{1}(\Delta s^* \geq 0) \end{aligned}$$

By Lemma 4.1(b), the posterior distribution of Δs^{π^*} is a truncated Gaussian distribution:

$$\mathrm{TN}_0^\infty\left(\mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i_1^*}^{(k^*)} - s_{l^*,i_2^*}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}\right)$$

Efficient sampling from a truncated Gaussian distribution can be found in [9].

With the above sampling rules, $\{m_l\}$, $\{s_{l,i}^{(k)}\}$, and $\{\Delta s^\pi\}$ are sampled in blocks. After the burn-in period, samples of $\mathbf{Z}$ are collected to approximate the posterior distribution $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$ (samples of $\mathbf{V}$ are discarded).

C. Model Updating

The model parameters are updated by

$$\begin{aligned} \mathbf{\Theta}^{(t+1)} &= \arg\max_{\Theta} \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) \\ \text{where } \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) &= \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[\ln \mathrm{P}(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta})] \qquad (14) \end{aligned}$$

with the posterior distribution $\mathbf{Z}|\mathbf{D}, \mathbf{\Theta}^{(t)}$ approximated by blocked Gibbs sampling.

Optimization details are given in Appendix A. Closed forms are obtained for the update of ground truth scores $\{s_{l,i}\}$, query difficulties $\{\delta_l^2\}$, and the domain distribution $\{\theta_m\}$. (Inexact) Newton's method is applied to update the domain expertise and truthfulness of workers $\{\tau_{k,m}\}$.

D. Identifiability

Identifiability is a property which a model must satisfy in order for precise inference to be possible. In plain words, it requires that different values of the parameters must generate different probability distributions of the observable variables.

For modeling rankings of documents, the extra degree of freedom of the model can potentially lead to an arbitrary scaling of the ground truth scores (or parameters), and thus must be carefully avoided.

One may observe that for the same collection of observations $\mathbf{D}$, the following two models have the same likelihood $\mathrm{P}(\mathbf{D}|\mathbf{\Theta_1}) = \mathrm{P}(\mathbf{D}|\mathbf{\Theta_2})$ for any global factor $\sigma > 0$ and query-level biases $\{b_l\}$.[4]

$$\begin{aligned} \mathbf{\Theta_1} &= \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\} \\ \mathbf{\Theta_2} &= \{(s_{l,i} - b_l)/\sigma, \delta_l^2/\sigma^2, \theta_m, \tau_{k,m}\sigma\} \qquad (15) \end{aligned}$$

Therefore, these two sets of parameters are not identifiable.

To cancel such extra freedom, we regularize the model by adding the following two constraints:

[4] This can be verified by comparing $\int_{\mathbf{Z}} \mathrm{P}(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta})$ using Eq.(6) for $\mathbf{\Theta} = \mathbf{\Theta_1}$ and $\mathbf{\Theta} = \mathbf{\Theta_2}$.

Identification Conditions

$$\begin{cases} \sum_l \delta_l^2 = 1 & (16) \\ \min_i s_{l,i} = 0, \forall l & (17) \end{cases}$$

The constraints are imposed after the model update in each iteration. Rescaling in this way keeps the model from undesired drifting and scaling.

V. Experiments

In this section, we systematically evaluate the techniques presented in this paper on both synthetic and real-world datasets. Code and datasets are available at the following repository: https://github.com/dragonxlwang/crowd_thurstonian

A. Simulated Study

1) Datasets: In order to test the effectiveness of TPP under various scenarios, we generate synthetic datasets with the following parameter settings.

The ground truth relevance scores of a list of documents $\{s_{l,i}\}_{i=1,2,...}$ for query q_l are generated from a uniform distribution $\mathcal{U}[0,1]$. Two different lengths are investigated: 5 (DOC5) and 30 (DOC30). Query difficulty δ_l^2 is generated from a uniform distribution $\mathcal{U}[0,0.1]$. To characterize the variable quality of answers given by crowd workers, we assume that worker t_k's expertise and truthfulness $\tau_{k,m}$ on domain m falls into one of the following categories:

- *Expert*: $\tau_{k,m} = 10$
- *Average*: $\tau_{k,m} = 5$
- *Spammer*: $\tau_{k,m} = 1$
- *Malicious*: $\tau_{k,m} = -10$

Three demographic groups are formed by changing the distributions over these four categories. Let p denote the categorical distribution over [*expert, average, spammer, malicious*]:

- DEMO1: $p = [0.2, 0.6, 0.1, 0.1]$. This group represents the most common case where average workers are dominant.
- DEMO2: $p = [0.2, 0.4, 0.3, 0.1]$. This group has a large proportion of spammers that can hurt the annotation quality.
- DEMO3: $p = [0.2, 0.4, 0.1, 0.3]$. The pairwise preferences given by this group can be overwhelmingly misleading due to the presence of too many malicious workers.

In order to simulate the incompleteness of annotations, which in real world often depends on factors such as time and budget constraints, we introduce a variable, *sparsity ratio* (SR), to control the probability that a pair of documents is judged by a worker. For example, if there are a list of 30 documents, and SR $= 0.05$, each worker will judge $\frac{30\times(30-1)}{2} \times 0.05 = 21.75$ randomly selected pairs.

Finally, the following 8 datasets are generated. Each of them contains 10 workers, 10 query domains and 100 queries: DOC5SR1.0DEMO1, DOC5SR0.5DEMO1, DOC5SR0.5DEMO2, DOC5SR0.5DEMO3, DOC30SR0.1DEMO1, DOC30SR0.05DEMO1, DOC30SR0.05DEMO2 and DOC30SR0.05DEMO3.

2) Baselines: We compare the performance of TPP against the following four baselines:

- TPPUNIDOM: TPP without modeling query domains, i.e., all queries are treated as from one single domain.
- TPPUNIEXP: TPP without modeling the domain expertise/truthfulness of workers, i.e., all workers have the same expertise and truthfulness for a given query domain: $\tau_{k_1,m} = \tau_{k_2,m} = \tau_m$, $\forall k_1, k_2$.
- TPPUNIDIFF: TPP with identical query difficulty, i.e., all queries are equally difficult: $\delta_l^2 = 1/Q$, $\forall l$ with some constant Q.
- CROWDBT: CROWDBT [12] is proposed to infer the ground truth scores out of pairwise preferences, which extends the Bradley-Terry model by taking worker accuracy into consideration. Specifically, a "worker-independent" pairwise preference between d_{i_1} and d_{i_2} for q_l is drawn from a Bernoulli distribution. The probability of $d_{i_1} \succ d_{i_2}$ is computed by the Sigmoid function:

$$\sigma(s_{l,i_1} - s_{l,i_2}) = \Big(1 + \exp\big(-(s_{l,i_1} - s_{l,i_2})\big)\Big)^{-1}$$

  Once the pairwise preference is drawn, each worker has a certain probability (accuracy) to report it truthfully or "flip" it. Compared with TPP, CROWDBT lacks the mechanism to model multiple query domains, thus incapable to characterize workers' domain-dependent expertise and truthfulness. Furthermore, it simplifies the generation of inconsistent annotations as solely a result from worker accuracy.

3) Performance Studies: We test all the methods on synthetic datasets under various parameter settings, and report Kendall's tau distance [13] between the inferred optimal ranking and the ground truth ranking. Kendall's tau distance is often used to measure the dissimilarity between two ranked lists [14], which is computed as the number of discordant pairs of the two ranked lists. A pair of documents is discordant if their relative order is reversed in the two rankings. For example, suppose two ranked lists of length 5 are $d_1 \succ d_2 \succ d_3 \succ d_4 \succ d_5$ and $d_3 \succ d_4 \succ d_1 \succ d_2 \succ d_5$. There are in total $\frac{5(5-1)}{2} = 10$ pairs and 4 of them are discordant: $\{d_1, d_3\}$, $\{d_1, d_4\}$, $\{d_2, d_3\}$, $\{d_2, d_4\}$, thus the Kendall's tau distance is 4. A small Kendall's tau distance indicates good performance. We run each method on every dataset 5 times and report the mean and standard deviation in Table II.

Overall Performance. TPP outperforms all other methods in general (the only exception is on DOC5SR0.5DEMO3, where TPPUNIDIFF gives the best result with a small margin). Among the three variants of TPP, TPPUNIEXP has the worst performance in recovering the ground truth rankings. This justifies the importance of modeling workers' domain expertise and truthfulness. Compared with CROWDBT, TPP consistently behaves significantly better, implying that the assumed generative process provides more flexibility in modeling and better explains the generation of inconsistent annotations.

Performance on Different Demographic Groups. Spammers and malicious workers have negative effects on all the methods. The decrease in performance due to malicious workers is more

TABLE II: Crowd Pairwise Preferences Binding Performance (Kendall's tau Distance)

Dataset	Tpp		TppUniDom		TppUniExp		TppUniDiff		CrowdBt	
Doc5Sr1.0Demo1	**0.386**	±0.031	0.414	±0.037	0.466	±0.023	0.402	±0.046	0.468	±0.047
Doc5Sr0.5Demo1	**0.574**	±0.067	0.728	±0.066	0.846	±0.080	0.628	±0.069	0.856	±0.028
Doc5Sc0.5Demo2	**0.734**	±0.021	0.852	±0.033	0.940	±0.037	0.754	±0.047	0.960	±0.041
Doc5Sr0.5Demo3	1.592	±0.237	1.760	±0.077	2.550	±0.029	**1.540**	±0.288	2.990	±0.060
Doc30Sr0.1Demo1	**22.442**	±1.238	25.636	±0.302	29.204	±0.291	26.866	±0.456	24.420	±0.906
Doc30Sr0.05Demo1	**40.640**	±0.926	45.498	±0.408	45.636	±0.178	47.258	±0.959	48.820	±2.161
Doc30Sr0.05Demo2	**61.818**	±2.713	70.548	±0.821	81.782	±0.145	66.488	±2.026	104.500	±2.469
Doc30Sr0.05Demo3	**129.156**	±1.892	139.154	±0.243	142.496	±0.587	135.04	±1.864	153.390	±1.031

striking than that due to spammers. Nevertheless, the proposed Tpp is more robust in resisting the attack from malicious workers than the baselines. Specifically, we observe that the Kendall's tau increases by 88.516 for Tpp when changing the dataset from Doc30Sr0.5Demo1 to Doc30R0.5Demo3 [5], while this number is 93.656 for TppUniDom, 96.860 for TppUniExp, and 104.57 for CrowdBt. This demonstrates that Tpp does a better job in recognizing adversarial workers.

Performance w.r.t. *Sparsity Ratio.* Sparser annotations provide less evidence to infer the ground truth rankings. It is observed that the best performance on Doc5Sr0.5Demo1 (0.574) is still much higher (and thus worse) than the worst performance on Doc5Sr1.0Demo1 (0.468). Similar observations are obtained on the Doc30 datasets.

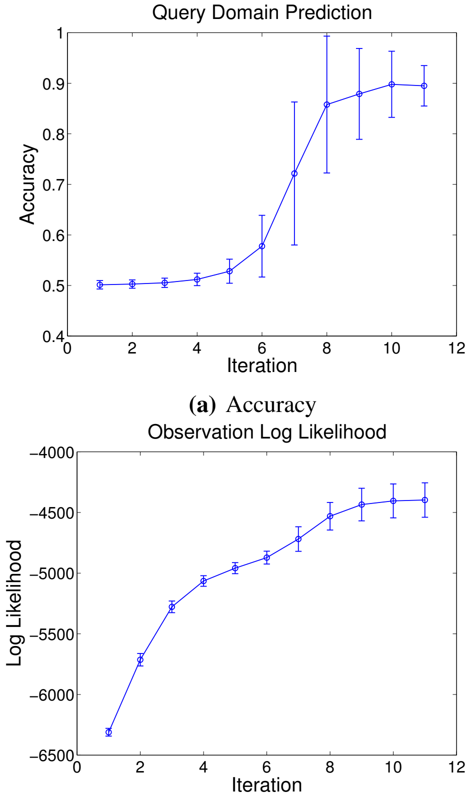
4) Query Domain Prediction: We investigate the capability of Tpp in distinguishing between queries from different domains.

We use the setting of Doc5Sr0.5Demo1 and generate the pairwise preferences with only two domains evenly distributed among 100 queries, for the ease of illustration. We run Tpp for 10 times and plot the prediction accuracy and the log likelihood. As shown in Figure 3, the algorithm starts from random guess with accuracy around 0.5, and converges to an accuracy around 0.895 in less than 10 iterations, implying that Tpp is able to learn query domains effectively and efficiently.

5) More Workers but Sparser Annotation: In practice, when time is the constraining factor, it is plausible to employ a large number of crowd workers and each worker labels only a few pairs. However, the situation of *"More Workers but Sparser Annotation"* can potentially lead to a critical limitation for Tpp. On one hand, the number of parameters $\{\tau_{k,m}\}$ grows with the number of workers. On the other hand, the amount of data to estimate each $\tau_{k,m}$ decreases.

TABLE III: Tpp Performance with More Workers but Sparser Annotation (Kendall's tau Distance)

Dataset	Kendall's tau	
Anno100Sr0.01	36.208	±0.292
Anno100Sr0.02	24.328	±0.451
Anno200Sr0.01	25.734	±0.394
Anno200Sr0.02	16.290	±0.435



(a) Accuracy

(b) Log Likelihood

Fig. 3: Domain Prediction Accuracy and Model Log Likelihood with Standard Deviations

To evaluate the performance in such scenarios, we create another four datasets under the setting of Doc30Demo1 with more annotators (Anno100 of 100 annotators and Anno200 of 200 annotators) and lower sparsity ratios (Sr0.01 and Sr0.02).

As shown Table III, the performance of Tpp becomes worse with "More Workers but Sparser Annotation" as Kendall's tau increases from 22.442 (Doc30Sr0.1Demo1) to 36.208 (Anno100Sr0.01). This is anticipated because the two datasets have the same amount of pairwise judgements but Anno100Sr0.01 involves more workers and has sparser annotations. However, Anno100Sr0.01 drastically reduces the time cost and may take only a tenth of the time that

[5] The maximal Kendall's tau distance for Doc30 is $\frac{30(30-1)}{2} = 435$.

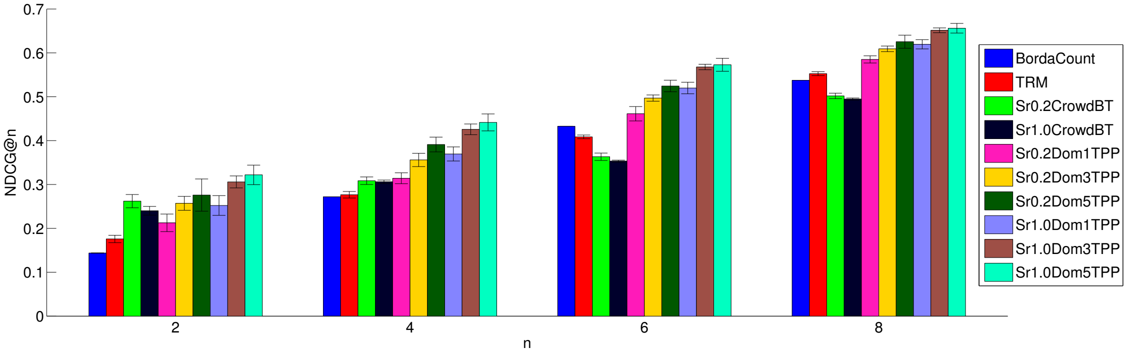


Fig. 4: NDCG@n evaluated on MQ2008-agg Dataset

Doc30Sr0.1Demo1 takes. In fact, by doubling the number of workers to 200 or doubling the sparsity ratio to 0.02, comparable performance can be achieved with Doc30Sr0.1Demo1. With an even more aggressive setting Anno200Sr0.02 (20 times the number of workers and five times sparser annotations), the performance further improves. Therefore we conclude that the performance of Tpp is reasonably robust even at the situation of "more workers and sparser annotation."

6) Malicious Worker Detection: Identifying malicious workers is a difficult task since the number of malicious workers is usually small so that the classification is highly imbalanced. We assess the performance of malicious worker detection by plotting the averaged Receiver Operating Characteristic (R.O.C.) curves in Figure 5. In the experiment, with 100 workers from Demo1 and Sr $= 0.01$, Tpp performs well with AUC $= 0.837$ (Area Under the Curve). When the annotation is denser (Anno100Sr0.02), AUC improves remarkably (0.924). However, with 200 workers (Demo1), the difference of AUC between Sr = 0.01 and Sr = 0.02 is not significant. This can be explained by the fact that malicious workers are easier to identify in a larger group, even with sparser annotations.

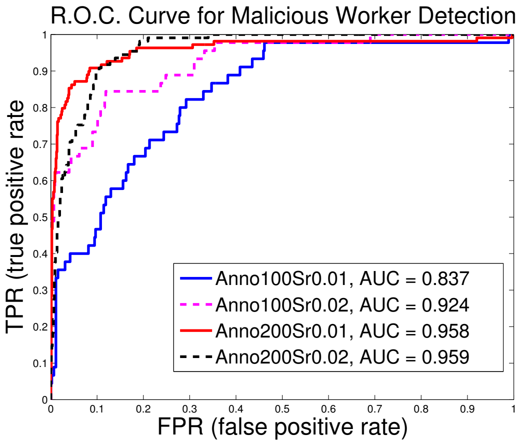


Fig. 5: R.O.C. Curve for Malicious Worker Detection

B. Experiments on Real-World Data

To validate our proposed strategy of binding pairwise preferences into rankings, we utilize a real-world benchmark MQ2008-agg (part of LETOR 4.0[6]) which is originally devised for the rank aggregation (meta-ranking) task. The MQ2008-agg dataset consists of ranked lists from 25 retrieval systems (workers). Each document is labeled as *highly relevant* (2), *relevant* (1) or *irrelevant* (0). For rank aggregation algorithms (Trm and BordaCount), ranked lists generated from each retrieval system are taken as input to infer the true ranked list for each query. The pairwise preference binding algorithms (Tpp and CrowdBt), on the other hand, estimate the true ranking out of the pairwise judgements from each retrieval system ("worker"). The pairwise judgements are randomly sampled with a sparsity ratio Sr. In the experiment, we use sparsity ratios Sr = 1.0 (all pairwise judgements are observed) and Sr = 0.2. We evaluate Tpp with 1, 3 and 5 domains. The performance is compared against both the pairwise preference binding algorithm CrowdBt, and the rank aggregation algorithms BordaCount [15] and Trm (see Section II and Appendix B). In particular, Bordacount is a simple yet robust algorithm which is essentially a ranking version of *majority voting*. It infers the true ranking by averaging the rank positions from each worker. The performance is measured by NDCG (Normalized Discounted Cumulative Gain) [16]. We use NDCG@n where $n = 2, 4, 6, 8$.

The results are presented in Figure 4. In general, similar performances are observed for the two rank aggregation algorithms with Trm slightly outperforming Bordacount. With Sr1.0, Tpp and CrowdBt have the same amount of information from observations as the rank aggregation counterparts. However, Sr1.0CrowdBt performs better than Trm and Bordacount only at NDCG@2 and NDCG@4, while it gets worse at NDCG@6 and NDCG@8. In contrast, Tpp consistently outperforms all the baselines, with better performance achieved if more domains are incorporated.

When the available annotations become sparser (Sr0.2), the performance of both Tpp and CrowdBt become worse: NDCGs decrease across different settings. However, Tpp still

[6] http://research.microsoft.com/en-us/um/beijing/projects/letor

significantly outperforms CrowdBt even with a single domain. In addition, it also outperforms Trm and Bordacount although the annotation is incomplete. This is because that the flexible generative process of Tpp properly resolves the inconsistency from multiple sources.

VI. Related Work

Early research of crowdsourcing can be dated back to the study of *integration of labels from multiple annotators* for image classification [17]. Later on, studies including [2], [18] began focusing on explicitly modeling annotator quality such as expertise, truthfulness in crowdsourcing settings. The dual tasks of inferring ground truth labels as well as worker quality have been investigated in some recent studies [1], [2], [18], including this paper.

Previous research mainly focused on simple tasks (classification, regression, etc.) while we tackle complex labeling problem such as ranking. In this direction, [19] reconstructs the order of facts from individual worker annotated *whole ranked lists* with the Thurstonian Ranking Model (Trm) [5] and the Mallows model [20], which features a distance-based distribution of rankings (permutations) using Kendall's tau. Other studies on "Rank Aggregation" are also related to this work, including [14], [21]. They adapt the Mallows model for inferring ground truth rankings as well as the quality of ranking algorithms. However, the above approaches do not fit well for information retrieval and web search tasks as it is not practical for annotators to label the whole ranked lists. This motivates us to investigate binding pairwise preferences from crowd workers into rankings.

There is one recent study [12] that adopts a similar philosophy, which extends the Bradley-Terry model, a pairwise special case of the Plackett-Luce model [22], [23]. Nevertheless, their model (CrowdBt) lacks the mechanism to model multiple query domains, thus incapable to characterize workers' domain-dependent expertise and truthfulness. CrowdBt does not take query difficulty into account either. Furthermore, unlike Tpp, CrowdBt does not model the generation of rankings. Therefore, it is not capable of modeling the annotation inconsistency from multiple sources, which makes it less favorable as demonstrated by the experimental study.

VII. Conclusions and Future Work

In this paper, we present a novel generative model called "Thurstonian Pairwise Preference" (Tpp) to infer the true ranked list out of a collection of crowdsourced pairwise annotations, which is highly useful in various data mining and information retrieval tasks such as *learning to rank*. Tpp resolves the inevitable incompleteness and inconsistency of pairwise judgements, by carefully modeling variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness. Experimental results on both synthetic and real-world datasets demonstrate that Tpp can effectively bind pairwise preferences of the crowd into rankings and substantially outperforms previously published methods. To further explore the benefit from the inferred ranked lists, it is promising to extend Tpp to jointly learn the ranking model of the end application, which we leave for future work.

Acknowledgment

This work is supported in part by the National Science Foundation under Grant Numbers CNS-1513939 and CNS-1408944. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *NIPS*, 2010.

[2] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, 2009, pp. 2035–2043.

[3] H. Valizadegan, R. Jin, R. Zhang, and J. Mao, "Learning to rank by optimizing ndcg measure," in *NIPS*, 2009.

[4] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *SIGIR*, 2007.

[5] L. L. Thurstone, "A law of comparative judgment." *Psychological review*, vol. 34, no. 4, p. 273, 1927.

[6] U. Böckenholt, "Applications of thurstonian models to ranking data," in *Probability models and statistical analyses for ranking data*. Springer, 1993.

[7] G. Yao and U. Böckenholt, "Bayesian estimation of thurstonian ranking models based on the gibbs sampler," *British Journal of Mathematical and Statistical Psychology*, 1999.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[9] N. Chopin, "Fast simulation of truncated gaussian distributions," *Statistics and Computing*, 2011.

[10] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1984.

[11] M. G. Kendall *et al.*, "The advanced theory of statistics." *The advanced theory of statistics.*, no. 2nd Ed, 1946.

[12] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, "Pairwise ranking aggregation in a crowdsourced setting," in *WSDM*. ACM, 2013.

[13] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[14] A. Klementiev, D. Roth, and K. Small, "Unsupervised rank aggregation with distance-based models," in *ICML*, 2008.

[15] J. A. Aslam and M. Montague, "Models for metasearch," in *SIGIR*. ACM, 2001.

[16] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd annual international ACM SIGIR*. ACM, 2000, pp. 41–48.

[17] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of venus images," *NIPS*, 1995.

[18] Y. Yan, R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," in *AISTATS*, 2010.

[19] M. Steyvers, B. Miller, P. Hemmer, and M. D. Lee, "The wisdom of crowds in the recollection of order information," in *NIPS*, 2009.

[20] C. L. Mallows, "Non-null ranking models. i," *Biometrika*, vol. 44, no. 1/2, pp. 114–130, 1957.

[21] A. Klementiev, D. Roth, K. Small, and I. Titov, "Unsupervised rank aggregation with domain-specific expertise," in *IJCAI*, 2009.

[22] R. D. Luce, *Individual choice behavior: A theoretical analysis*. Dover-Publications. com, 2012.

[23] R. L. Plackett, "The analysis of permutations," *Applied Statistics*, pp. 193–202, 1975.

[24] M. Chiani, D. Dardari, and M. K. Simon, "New exponential bounds and approximations for the computation of error probability in fading channels," *Wireless Communications, IEEE Transactions on*, vol. 2, no. 4, 2003.

Appendix A
Model Updating

By zeroing the derivatives of $\mathcal{Q}(\boldsymbol{\Theta}^{(t+1)};\boldsymbol{\Theta}^{(t)})$ with respect to $\boldsymbol{\Theta}^{(t+1)}$, the following closed forms are obtained for the update of $\{s_{l,i}^{(t+1)}\}$, $\{\delta_l^{2(t+1)}\}$ and $\{\theta_m^{(t+1)}\}$:

$$\begin{cases} s_{l^*,i^*}^{(t+1)} = \dfrac{\sum\limits_{k\in\mathcal{W}_{l^*,i^*}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\boldsymbol{\Theta}^{(t)}}[s_{l^*,i^*}^{(k)}]}{\sum\limits_{k\in\mathcal{W}_{l^*,i^*}} 1} & (18)\\ \delta_{l^*}^{2(t+1)} = \dfrac{\sum\limits_{(k,i)\in\mathcal{P}_{l^*}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\boldsymbol{\Theta}^{(t)}}[(s_{l^*,i}^{(k)} - s_{l^*,i}^{(t+1)})^2]}{\sum\limits_{(k,i)\in\mathcal{P}_{l^*}} 1} & (19)\\ \theta_{m^*}^{(t+1)} \propto \sum\limits_{l} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\boldsymbol{\Theta}^{(t)}}[\mathbf{1}(m_l = m^*)] & (20) \end{cases}$$

where $\mathcal{W}_{l^*,i^*}$ denotes the set of workers who have judged d_{i^*} for q_{l^*}, and $\mathcal{P}_{l^*}$ denotes the set of $\langle$worker, document$\rangle$ pairs involved in the annotation for q_{l^*}, i.e.,

$$\begin{aligned} \mathcal{W}_{l^*,i^*} &= \{k \mid \exists i,\ \langle k,l^*,i^*,i\rangle \in \mathbf{D} \text{ or } \langle k,l^*,i,i^*\rangle \in \mathbf{D}\} \\ \mathcal{P}_{l^*} &= \{(k,i) \mid \exists\tilde{i},\ \langle k,l^*,i,\tilde{i}\rangle \in \mathbf{D} \text{ or } \langle k,l^*,\tilde{i},i\rangle \in \mathbf{D}\} \end{aligned}$$

Unfortunately, $\{\tau_{k,m}^{(t+1)}\}$ do not have a closed-form analytic solution, where we employ Newton's method. The partial derivatives *w.r.t.* $\tau_{k^*,m^*}^{(t+1)}$ are given by:

$$\begin{aligned} \frac{\partial\mathcal{Q}}{\partial\tau_{k^*,m^*}^{(t+1)}} = &\sum_{\substack{l,i_1,i_2\\ \langle k^*,l,i_1,i_2\rangle\in\mathbf{D}}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\boldsymbol{\Theta}^{(t)}} \Bigg[\mathbf{1}(m_l = m^*)\frac{s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)}}{\sqrt{2}} \\ &f\Big(-\frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}}(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big)\Bigg] \qquad (21) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2\mathcal{Q}}{\partial\tau_{k^*,m^*}^{(t+1)\ 2}} = -&\sum_{\substack{l,i_1,i_2\\ \langle k^*,l,i_1,i_2\rangle\in\mathbf{D}}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\boldsymbol{\Theta}^{(t)}}\Bigg[\mathbf{1}(m_l = m^*)\frac{(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})^2}{2} \\ &\Big\{f\Big(-\frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}}(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big)\frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}}(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)}) \\ &+ f^2\Big(-\frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}}(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big)\Big\}\Bigg] \qquad (22) \end{aligned}$$

where we used the fact that

$$\partial\mathrm{Q}(x)/\partial x = -\mathrm{P}_{\mathtt{N}}(x|0,1)$$

And $f(\cdot)$ is defined as

$$f(x) = \frac{\mathrm{P}_{\mathtt{N}}(x|0,1)}{\mathrm{Q}(x)} \qquad (23)$$

whose derivative is calculated as:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = -xf(x) + f^2(x) \qquad (24)$$

An important implementation issue of Newton's method is numeric stability. For large $x > 0$, computing $f(x)$ using Eq. 23 is not advised as both $\mathrm{P}_{\mathtt{N}}(x|0,1)$ and $\mathrm{Q}(x)$ approach zero fast. To address this issue, we use the following approximation [24]:

$$\mathrm{Q}(x) \approx \frac{1}{12}e^{-\frac{x^2}{2}} + \frac{1}{4}e^{-\frac{2}{3}x^2} \qquad (25)$$

Using this result, $f(x) \approx \frac{12}{\sqrt{2\pi}}$ can be found to be a good approximation for $x > 8$.

Appendix B
Inference of Trm

Trm is the building block of the proposed Tpp and is investigated as a baseline in the experiment. We present the maximum likelihood estimation (MLE) using the Expectation-Maximization (E-M) algorithm.

The joint likelihood is given by

$$\begin{aligned} &\mathrm{P}\big(\{\sigma_l^{(k)}\},\{s_{l,i}^{(k)}\}|\{s_{l,i}\},\{\delta_l^2\}\big) \\ =&\prod_{l,i,k}\mathrm{P}_N(s_{l,i}^{(k)}|s_{l,i},\delta_l^2)\cdot\mathbb{1}(\sigma_l^{(k)},\{s_{l,i}^{(k)}\}) \qquad (26) \end{aligned}$$

where $\mathbb{1}(\sigma_l^{(k)},\{s_{l,i}^{(k)}\}) = 1$ if the ranking derived from the order of $\{s_{l,i}^{(k)}\}$ is consistent with $\sigma_l^{(k)}$ and 0 otherwise.

In addition, like Tpp, the posterior distribution is approximated by Gibbs sampling,

$$\begin{aligned} &\mathrm{P}\big(s_{l^*,i^*}^{(k^*)}|\{s_{l,i}\},\{\delta_l^2\},\{\sigma_l^{(k)}\},\{s_{l,i}^{(k)}\}\ \{s_{l^*,i^*}^{(k^*)}\}\big) \\ =&\mathrm{P}_N(s_{l^*,i^*}^{(k^*)}|s_{l^*,i^*},\delta_{l^*}^2)\cdot\mathbb{1}(s_- \le s_{l^*,i^*}^{(k^*)} \le s_+\}) \qquad (27) \end{aligned}$$

where s_+ (or s_-) denotes the worker's perceived score $s_{l^*,i}^{(k^*)}$ of the document d_i which immediately precedes (or follows) d_{i^*} as ranked by $\sigma_{l^*}^{(k^*)}$ if such d_i exists or otherwise evaluated as $+\infty$ (or $-\infty$). Consequently, we samples $s_{l^*,i^*}^{(k^*)}$ by

$$s_{l^*,i^*}^{(k^*)} \sim \mathrm{TN}_{s_-}^{s_+}(s_{l^*,i^*},\delta_{l^*}^2)$$

Lastly, we update the parameters by optimizing the expected joint log likelihood, which yields the same updating rules as in Eq. 18, 19 with the only difference being that k is ranged over all workers that rank for q_{l^*} in Eq. 18 and (k,i) over all workers that judge q_{l^*} and documents in the ranking list of q_{l^*} in Eq. 19.

A final note of Trm is about its identifiability: It requires rescaling in the same manner as in Eq. 16 and Eq. 17 to cancel extra freedom in order to prevent the model from undesired drifting and scaling.