Pairwise Ranking Aggregation in a Crowdsourced Setting

Xi Chen[*]
Carnegie Mellon University
Pittsburgh, PA U.S.A.
xichen@cs.cmu.edu

Paul N. Bennett,
Kevyn Collins-Thompson, Eric Horvitz
Microsoft Research
Redmond, WA U.S.A. 98052
{pauben,kevynct,horvitz}@microsoft.com

ABSTRACT

Inferring rankings over elements of a set of objects, such as documents or images, is a key learning problem for such important applications as Web search and recommender systems. Crowdsourcing services provide an inexpensive and efficient means to acquire preferences over objects via labeling by sets of annotators. We propose a new model to predict a gold-standard ranking that hinges on combining pairwise comparisons via crowdsourcing. In contrast to traditional ranking aggregation methods, the approach learns about and folds into consideration the quality of contributions of each annotator. In addition, we minimize the cost of assessment by introducing a generalization of the traditional active learning scenario to jointly select the annotator and pair to assess while taking into account the annotator quality, the uncertainty over ordering of the pair, and the current model uncertainty. We formalize this as an active learning strategy that incorporates an exploration-exploitation tradeoff and implement it using an efficient online Bayesian updating scheme. Using simulated and real-world data, we demonstrate that the active learning strategy achieves significant reductions in labeling cost while maintaining accuracy.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.1.2 [**Information Systems**]: User/Machine Systems—*Human factors*

General Terms

Algorithms, Human Factors

Keywords

Ranking, Crowdsourcing, Pairwise Preference

1. INTRODUCTION

Obtaining a set of gold-standard labels for a set of objects is a critical step in learning to rank. For example, when determining how to rank the results returned in response to a Web search, the results are often passed through a ranking model that has been learned using a machine learning procedure [12]. In order to learn this model, learning methods typically require a training set of queries and documents where gold-standard labels on relevance with respect to a query have been provided. The learning method optimizes some objective with respect to the labels.

A variety of approaches can be employed to acquire labels. We may obtain binary relevance judgments, graded relevance judgments, or preferences [1]. Among these there may be tradeoffs in the amount of information the label contains and the noise associated with obtaining the label. For example, while a graded relevance judgment on a five-point scale may contain more information than a binary judgment, annotators may also make more errors due to the complexity of assigning finer-grained judgments. One approach to acquiring inputs on rank is to obtain relative preference judgments for pairs of items. This method promises assessments that are easier and faster to obtain, is less prone to assessor error, and enables fine-grained comparisons. Such pairwise assessment may be especially valuable for ranking in tasks with higher numbers of gradations, *e.g.* assessing reading difficulty into one of the standard 12 American grade levels [3]. Pairwise assessment may also be valuable for inferring global rankings in such settings as developing recommender systems where we desire to order a set of products based on a small number of observed preferences from individual users.

We focus on the task of inferring a gold-standard ranking from a set of preferences over objects given in the form of pairwise comparisons (*i.e.*, i is preferred to j denoted as $i \succ j$). As in any collection of gold-standard data, we seek to obtain the most accurate labeling with minimal labeling cost. To this end, we seek to take advantage of crowdsourcing services, such as Amazon Mechanical Turk, which enables one to programmatically obtain large collections of pairwise comparisons from sets of annotators at low cost. However, the reliability of annotators available via crowdsourcing can vary significantly. In addition, seeking pairwise assessments from the crowd can lead to inconsistent pairs (*e.g.*, $i \succ j$ by one annotator and $j \succ i$ by another annotator; or $i \succ j$, $j \succ k$ and $k \succ i$). Many existing ranking aggregation methods [20, 15, 18, 13, 2, 19] are either incapable of modeling the quality of work by annotators or are inadequate for dealing with inconsistent pairs.

[*]This work was performed during an internship at Microsoft Research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WSDM '13 Rome, Italy
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

To address the challenge of learning a global ranking in a crowdsourced setting, we introduce the *Crowd-BT* algorithm, which extends the widely used Bradley-Terry model [20] by explicitly incorporating the quality of contributions provided by different annotators. The Crowd-BT algorithm can both appropriately weight annotators' contributions by their annotation quality as well as distinguish between *spammers* and *malicious annotators*: spammers assign random labels,[1] while malicious annotators (or poorly informed annotators) assign the wrong label most of the time.[2] Many existing crowdsourcing algorithms treat assessments provided by spammers or malicious annotators the same as they do assessments of low quality. In contrast, Crowd-BT can exclude pairs labeled by spammers from the modeling while automatically correcting the pairs provided by malicious annotators.

Beyond appropriately handling error, spam, and malicious inputs, we seek to be budget-conscious; we typically prefer to harness fewer labeled samples while achieving reasonably good accuracy. Thus, we seek a formal model of active learning to guide the allocation of effort in crowdsourcing (*e.g.*, see [9]). Most active learning methods [22] assume the availability of an *oracle* that can provide the correct label. In such settings, we only need to decide how to select the next pairwise assessment. We typically do not have access to such expertise in a crowdsourced setting. Thus, we face the challenge of simultaneously selecting the best next pair to be labeled and the best annotator to label the next pair. We shall formulate and study an *exploration-exploitation tradeoff* in crowdsourcing, previously explored in bandit and reinforcement learning. More precisely, *exploration* refers to using pairs with high-confidence labels to test the quality of annotators, while *exploitation* refers to asking for labels for the most uncertain pairs. We need to balance the tradeoff between exploration and exploitation carefully: too much exploration could lead to samples being repeatedly labeled, so that we do not have a sufficient number of unique samples within the assessment budget; however, too much emphasis on exploitation may result in a large number of noisy labels provided by low-quality annotators.

In the remainder of the paper, we first provide background on the Bradley-Terry model and demonstrate how to incorporate annotator quality into the model. We then demonstrate how to address situations that arise in practice via a regularization term before discussing how to optimize the objective function to infer the model parameters. Next, we formalize the active learning problem as an exploration-exploitation tradeoff and derive an approach that enables the efficient updates needed in an active learning setting. Finally, we present a series of experiments with synthetic data to better understand model properties. The experiments with real-world data demonstrate that modeling annotator quality improves inferred ranking quality, and furthermore, that our active learning approach achieves 90% of the best gold-standard accuracy with only 3% of the total labeling cost.

2. RELATED WORK

Early work for modeling annotator quality is presented in [5], where the true category of an object is inferred from the crowd. With the availability of programmatic access to human effort via crowdsourcing platforms, a range of studies have applied machine learning to data collected from the crowd. Raykar *et al.* [21] extended Dawid & Skene's work [5] by introducing a logistic classification model to incorporate features of the input data. Wang *et al.* [25] proposed to separate malicious annotators from spammers in a binary classification setting. Our Crowd-BT algorithm extends the latter work to pairwise ranking aggregation problems.

Karger *et al.* [10] proposed an iterative algorithm to infer consensus class labels with asymptotic consistency guarantees. Welinder *et al.* [26] extended Dawid & Skene's work [5] to Bayesian updating procedures. While most of the work in learning from the crowd has focused on classification problems, several studies examine ordinal regression and ranking problems with assessments [21, 24]. For example, the formulation presented by Volkovs & Zemel [24] models annotator quality as the variance term in a logistic formulation and could be used to address our challenge. However, the methodology suffers from the weakness that it cannot distinguish spammers and malicious (or poorly informed) annotators. In addition, Bayesian modeling and active learning with the variance term in their logistic formulation provide more difficult computational challenges. In contrast, we will show computationally efficient solutions for our formulations.

Cost and efficiency within a budget are important for learning about and harnessing a crowd for problem solving. Costs can be throttled with selective assessments guided by active-learning procedures. Unlike many applications of active learning, in a crowd setting, we cannot assume we have access to an oracle with answers. Yan *et al.* [29] proposed an active learning strategy for binary classification with a crowd. This work mainly focused on selection of an annotator who can provide the most confident label for an actively selected sample. Kamar *et al.* [9] describe methods for guiding the acquisition of votes in a crowdsourcing system for citizen science with a decision-theoretic computation of value of information within a POMDP representation, using a voting rule on a training set to define ground truth.

A great deal of prior work has been devoted to challenges with aggregation of rankings. Methods studied include permutation-based methods (*e.g.*, Mallows [2] and CPS [19] models), matrix factorization methods (*e.g.*, [6]) and score-based probabilistic methods (*e.g.*, Bradley-Terry [20], Plackett-Luce [15, 18] and Thurstone [13] models).

Permutation-based methods are generally computationally expensive while matrix factorization methods lack probabilistic interpretation. Thus, we build our work on score-based methods which are both more suitable for modeling pairwise comparisons and computationally efficient.

In summary, in contrast to previous work in pairwise ranking aggregation, our method can learn annotator quality with a unified model and distinguishes malicious annotators from spammers. More importantly, the active learning strategy proposed in this paper explicitly models the tradeoff between the learning of annotator quality versus the learning of pairwise preference. Our work formalizes this as the important concept of an *exploration-exploitation tradeoff* in active learning with the crowd.

[1] Annotators who either do not actually look at instances, or robots pretending to be human annotators, presumably to quickly receive pay for work.

[2] That is, they label $i \succ j$ whenever $j \succ i$ and vice versa, perhaps because the annotators are malicious or misunderstand the labeling criteria.

3. CROWD-BT: EXTENDING BRADLEY -TERRY MODEL TO CROWDSOURCING

As mentioned above, we choose to extend the Bradley-Terry model because it has a well-understood probabilistic interpretation, is well-suited to preferences, and can be optimized for computational efficiency. In particular, we extend the Bradley-Terry model [20] to incorporate parameters for individual annotator quality. We first review the basic Bradley-Terry model before demonstrating how annotator quality can be incorporated.

For any two objects X and Y, Bradley-Terry models the probability that X is preferred over Y as $\Pr(X \succ Y) = \frac{\pi_X}{\pi_X + \pi_Y}$, where $\pi_X, \pi_Y > 0$ can be viewed as relevance scores for X and Y respectively (alternative interpretations in other settings are as skill scores or difficulty scores). By defining $\pi_X = \exp\{s_X\}$, we obtain:

$$\Pr(X \succ Y) = \frac{e^{s_X}}{e^{s_Y} + e^{s_Y}} = \frac{e^{(s_X - s_Y)}}{1 + e^{(s_X - s_Y)}}. \tag{1}$$

The Bradley-Terry model can be easily extended to model preferences among a small set of objects:

$$\Pr(X \succ \{Y, Z\}) = \frac{e^{s_X}}{e^{s_X} + e^{s_Y} + e^{s_Z}}. \tag{2}$$

It can also model a chain-complete partial order by decomposing it into pairwise preferences: $\Pr(X \succ Y \succ Z) \doteq \Pr(X \succ Y)\Pr(Y \succ Z)$.

We assume there are N objects $\{o_1, \ldots, o_N\}$ and a pool of K annotators $\{a_1, \ldots, a_K\}$. We denote the set of labeled pairs by the k-th annotator as $S_k = \{(i,j) : o_i \succ_k o_j\}$, where $o_i \succ_k o_j$ represents that the k-th annotator prefers o_i over o_j. Here, we make an implicit assumption that an annotator never simultaneously claims $o_i \succ o_j$ and $o_i \prec o_j$, so that each pair (i,j) in S_k can be ordered by $o_i \succ_k o_j$. Directly applying the Bradley-Terry model without distinguishing each annotator's quality, we have $\Pr(o_i \succ_k o_j) = \frac{e^{s_i}}{e^{s_i} + e^{s_j}}$. Then, pairwise ranking aggregation can be directly formulated into a *log-likelihood maximization problem* as follows:

$$\max_{\mathbf{s}} \log \left(\prod_{k=1}^{K} \prod_{(i,j) \in S_k} \Pr(o_i \succ_k o_j) \right) = \sum_{k=1}^{K} \sum_{(i,j) \in S_k} \log \left(\frac{e^{s_i}}{e^{s_i} + e^{s_j}} \right)$$

$$\text{s.t.} \sum_{i=1}^{N} s_i = 0. \tag{3}$$

Because the objective function on the left of Eq. (3) is *not scale-invariant*: if we increase all s_i by any given constant c, the log-likelihood will remain the same. Therefore, to make the objective identifiable, we use a standard trick (e.g., [8]), which adds one additional constraint, $\sum_{i=1}^{N} s_i = 0$. By maximizing Eq. (3), we can obtain a global ranking over N objects by sorting the obtained $\mathbf{s}$.

When directly applying the Bradley-Terry model in crowdsourcing, as in Eq. (3), each annotator is treated equally and, hence, the model is incapable of capturing the variability in quality of contribution across individual annotators. We now introduce a parameter η_k for the k-th annotator which is defined as the probability that the k-th annotator agrees with the true pairwise preference. In particular, for any pair with the true preference $X \succ Y$:

$$\eta_k \equiv \Pr(X \succ_k Y | X \succ Y). \tag{4}$$

If the k-th annotator is perfect, we have $\eta_k \approx 1$; if he/she is a spammer, we have $\eta_k \approx 0.5$; while if he/she is a malicious or poorly informed annotator, we have $\eta_k \approx 0$. Applying the law of total probability, we have

$$\begin{aligned} \Pr(o_i \succ_k o_j) &= \Pr(o_i \succ_k o_j | o_i \succ o_j) \Pr(o_i \succ o_j) \\ &\quad + \Pr(o_i \succ_k o_j | o_i \prec o_j) \Pr(o_i \prec o_j) \\ &= \eta_k \frac{e^{s_i}}{e^{s_i} + e^{s_j}} + (1 - \eta_k) \frac{e^{s_j}}{e^{s_i} + e^{s_j}}. \end{aligned} \tag{5}$$

The log-likelihood $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s})$ thus takes the form

$$\begin{aligned} \mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) &= \sum_{k=1}^{K} \sum_{(i,j) \in S_k} \log \Pr(o_i \succ_k o_j) \\ &= \sum_{k=1}^{K} \sum_{(i,j) \in S_k} \log \left[\eta_k \frac{e^{s_i}}{e^{s_i} + e^{s_j}} + (1 - \eta_k) \frac{e^{s_j}}{e^{s_i} + e^{s_j}} \right]. \end{aligned} \tag{6}$$

3.1 Thurstone model

A closely related model to the Bradley-Terry model is the Thurstone model [13], which assumes that the score for each object X has a Gaussian distribution $N(S_X, \sigma_X)$. For simplicity, here we only consider the Case V Thurstone model where $\sigma_X = 1$ for all objects. Then, the difference between the score of X and that of Y follows a Gaussian distribution $N(S_X - S_Y, \sqrt{2})$ and thus

$$\Pr(X \succ Y) = \Phi \left(\frac{S_X - S_Y}{\sqrt{2}} \right), \tag{7}$$

where $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function. The likelihood $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s})$ under the Thurstone model thus takes the following form:

$$\mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) = \sum_{k=1}^{K} \sum_{(i,j) \in S_k} \log \left[\eta_k \Phi \left(\frac{s_i - s_j}{\sqrt{2}} \right) + (1 - \eta_k) \Phi \left(\frac{s_j - s_i}{\sqrt{2}} \right) \right].$$

To solve this corresponding maximum likelihood problem, we need to evaluate $\Phi(\cdot)$ many times, which involves an integration and hence is computationally more expensive than maximizing Eq. (6). Thus, we adopt the Bradley-Terry model in the paper. However, we note that the performance of Bradley-Terry and Thurstone models have been shown to be very similar [23]; and all the developed methods in this paper can be used in a straightforward way to extend the Thurstone model for use in crowdsourcing.

3.2 Regularization

For better visualization and interpretation, pairwise comparisons are often presented as a comparison graph: if an annotator prefers o_i over o_j, we draw a *directed* edge from o_j to o_i. We first point out that application of the Bradley-Terry model in Eq. (3) can face numerical challenges if the underlying comparison graph is not strongly-connected.[3] More specifically, one cannot have a maximizer of the log-likelihood in Eq. (3) when the comparison graph is not strongly-connected. As an example, we consider the comparison graph in Fig. 1 with *two* strongly-connected components in red circles. For *any* given solution s_A, s_B, s_C, s_D, if one adds an arbitrary positive constant c to s_A, s_B and subtracts the constant c from s_C, s_D, then the likelihood terms corresponding to the edges $C \to A$ and $D \to B$ (*i.e.*, $\frac{e^{s_A}}{e^{s_A} + e^{s_C}}$ and $\frac{e^{s_B}}{e^{s_B} + e^{s_D}}$) will always increase, while the likelihood terms corresponding to the edges within each strongly-connected component

[3] A directed graph is called strongly-connected if there is a path from each node in the graph to every other node.

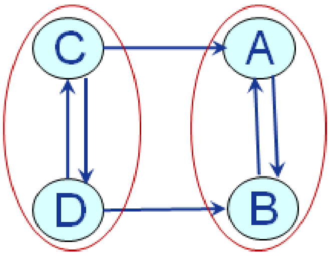


Figure 1: Example of comparison graph.

(*i.e.*, $A \to B$, $B \to A$, $C \to D$ and $D \to C$) will remain the same. Therefore, we could not have a maximizer of the log-likelihood.

We address this numerical problem by introducing a virtual object (node) o_0 with the score e^{s_0}. We assume that each object o_i is compared to o_0 by a perfect annotator with one virtual win and one virtual loss. We can make any graph strongly-connected with this procedure. According to [14], the log-likelihood will then have a unique maximizer. The virtual node regularization technique has been used for ranking problems in different settings (e.g., [4]). In the setting of our problem, this amounts to using a regularized form $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) + \lambda\mathcal{R}(\mathbf{s})$, where the regularization term $\mathcal{R}(\mathbf{s})$ is defined as

$$\mathcal{R}(\mathbf{s}) = \sum_{i=1}^{N}\left(\log\left(\frac{e^{s_0}}{e^{s_0}+e^{s_i}}\right) + \log\left(\frac{e^{s_i}}{e^{s_0}+e^{s_i}}\right)\right)$$

and $\lambda > 0$ is the predefined regularization parameter.

3.3 Crowd-BT

The final Crowd-BT formulation for pairwise ranking aggregation in crowdsourcing is essentially a *regularized maximum likelihood problem*:

$$\begin{aligned}\max_{\boldsymbol{\eta},\mathbf{s}} \quad & \mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) + \lambda\mathcal{R}(\mathbf{s}) \qquad (8)\\ & \doteq \sum_{k=1}^{K}\sum_{(i,j)\in S_k} \log\left[\eta_k \frac{e^{s_i}}{e^{s_i}+e^{s_j}} + (1-\eta_k)\frac{e^{s_j}}{e^{s_i}+e^{s_j}}\right]\\ & \quad + \lambda\sum_{i=1}^{N}\left(\log\left(\frac{e^{s_0}}{e^{s_0}+e^{s_i}}\right) + \log\left(\frac{e^{s_i}}{e^{s_0}+e^{s_i}}\right)\right)\\ \text{s.t.} \quad & 0 \le \eta_k \le 1, \qquad \forall k \in \{1,\ldots,K\}.\end{aligned}$$

As we can see from Eq. (8), another benefit of this extra regularization is that the constraint $\sum_{i=1}^{N} s_i = 0$ in Eq. (3) is no longer needed if we fix s_0. Recall that this constraint is used to address the scale-invariant problem in the objective function in Eq. (3). Now, in Eq. (8), if we fix s_0, the objective is no longer scale-invariant and hence the constraint $\sum_{i=1}^{N} s_i = 0$ could be dropped.

Let us provide more detailed explanations for the objective in Eq. (8). The first term $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s})$ is the log-likelihood: that is, $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) = \sum_{k=1}^{K}\sum_{(i,j)\in S_k} \log \Pr(o_i \succ_k o_j)$ with $\Pr(o_i \succ_k o_j) = \eta_k \frac{e^{s_i}}{e^{s_i}+e^{s_j}} + (1-\eta_k)\frac{e^{s_j}}{e^{s_i}+e^{s_j}}$ as defined in Eq. (5). Due to the principle of maximum likelihood and desirable properties of maximum-likelihood estimates (*e.g.*, consistency, normality, invariance and efficiency), it is natural to maximize $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s})$.

Now we examine each term in $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s})$. For a perfect annotator with $\eta_k = 1$, $\Pr(o_i \succ_k o_j)$ will reduce to the Bradley-Terry model. For a spammer with $\eta_k = 0.5$, $\Pr(o_i \succ_k o_j) \equiv 0.5$ for any s_i, s_j and hence all the pairs provided by a spammer will not affect our objective in Eq. (8). In other words, once we detect a spammer, we automatically discard all the pairs labeled by him/her. For a malicious annotator with $\eta_k = 0$, we have $\Pr(o_i \succ_k o_j) = \frac{e^{s_j}}{e^{s_i}+e^{s_j}}$, which is equivalent to having $o_j \succ o_i$ provided by a perfect annotator. This means that our model can automatically recover the errors made by a malicious annotator. On the other hand, if $s_i \gg s_j$ (*i.e.*, there is a significant difference between these two objects), we have $\Pr(o_i \succ_k o_j) \approx \eta_k$, which indicates that the probability depends largely on the annotator's quality. If $s_i \approx s_j$, we have $\Pr(o_i \succ_k o_j) \approx 0.5$ which indicates that for two very similar objects, they are indistinguishable regardless of the annotator's quality.

The second part of Eq. (8) is the regularization term. First, as we show in Section 3.2, without the regularization, the corresponding optimization is not well-defined when the graph is not strongly connected, and we will not have a finite solution. Therefore, the regularization is indispensable for comparison graphs that are not strongly connected. Secondly, when the graph is indeed strongly connected, the regularization might change the solution. However, if we set λ to be sufficiently small, the ranking inferred from $\mathbf{s}$ will be the same as that obtained from the un-regularized problem. Interestingly, in many problems, the regularized problem could lead to an even better solution than the un-regularized one as shown in our experiments. In practice, if one only wants to recover the ranking from the un-regularized problem (assuming it is well-defined), one could just use a sufficiently small λ. On the other hand, if gold pairs are available (*i.e.*, samples with the true label provided by experts),[4] one can carefully use them to tune the parameter λ to achieve the best performance on gold samples. Regardless of the optimal choice, we show empirically later that for a broad range of $\lambda \in [0.1, 10]$ our algorithm outperforms the baseline (see Tables 2 and 3).

To maximize the objective $\mathcal{L}(\boldsymbol{\eta}, \mathbf{s}) + \lambda\mathcal{R}(\mathbf{s})$, a natural optimization strategy is the alternating approach [16]: fix $\boldsymbol{\eta}$ and optimize over $\mathbf{s}$; then fix $\mathbf{s}$ and optimize over $\boldsymbol{\eta}$; and iterate over these two steps. In particular, we adopt limited-memory BFGS [16] to optimize $\mathbf{s}$ and the projected Newton method to optimize $\boldsymbol{\eta}$ [11]. We note that as Eq. (8) is a non-concave maximization problem, a good initialization is important to avoid being trapped in local minima. As long as the average quality of the annotators is better than that of spammers, we suggest starting with $\eta_k = 1$ for each annotator and first optimizing over $\mathbf{s}$. In fact, this strategy is better than the traditional multiple random initialization strategy for solving non-convex optimization since we utilize the prior side information of the problem (*i.e.*, most annotators are good). If there are more spammers and malicious annotators (although it may happen rarely in practice), we could initialize η_k by measuring the performance of each annotator on a handful of gold samples. In particular, η_k could be initialized as the ratio of the correct answers on the gold samples.

[4] In fact, even if there is no expert, one can construct gold pairs from the data. For example, we could treat a pair as a gold sample if more than 10 annotators rank the pair and at least 90% of them agree on the same preference.

4. ACTIVE LEARNING

Active learning in crowdsourcing is fundamentally different from traditional active learning in two different aspects: (1) we do not have access to an oracle for labels and (2) beyond selecting pairs to be labeled (exploitation) we also need to probe each annotator's quality (exploration), and carefully balance this exploration-exploitation tradeoff. For pairwise comparison, at each round, we need to choose a triplet *(object i, object j, annotator k)* and ask for the preference between *object i* and *object j* from the annotator k. Let T be the total budget, i.e., the total number of triplets that we can query. The high-level picture of the active learning in crowdsourcing for our problem is as follows. For each round $t = 1, \dots, T$ run the following steps:

1. Choose the triplet (o_i, o_j, k) that maximizes the impact on the model uncertainty given the expectation over the annotator k's response.

2. Query the preference between o_i and o_j from the annotator k.

3. Update the model with the elicited preference.

However, there are three major challenges for implementing the above approach. First, the sheer number of possible triplets is $KN(N-1)/2$ so that the maximization in Step 1 is taken over a large space. The second is quantifying the impact on the model uncertainty in a way that also incorporates the notion of an exploration-exploitation tradeoff. The third is how to update the model in a time-efficient manner without re-training the whole model. For the first issue, since the maximization can be solved in a straightforward parallel manner, this challenge can be addressed given enough computational power. For the second issue, we establish a Bayesian framework for our problem and introduce a novel definition of the *expected information gain* by extending the traditional Kullback-Leibler (KL) divergence to incorporate the exploration-exploitation tradeoff. Finally, in the next subsection, we introduce an efficient online update of the model parameters that will enable a constant time update for a new incoming triplet.

We first extend Crowd-BT into a Bayesian framework to enable the definition of the information gain/model uncertainty and facilitate the development of an online updating method. We assume $\{s_i\}_{i=1}^N$ and $\{\eta_k\}_{k=1}^K$ are independent random variables and introduce a Gaussian prior for each s_i (*i.e.*, $s_i \sim N(\mu_i, \sigma_i)$) and a Beta prior for each η_k (*i.e.*, $\eta_k \sim \mathrm{Beta}(\alpha_k, \beta_k)$). Given a pair labeled by the k-th annotator, $(o_i \succ_k o_j)$, we have the prior

$$p(s_i, s_j, \eta_k) = N(s_i; \mu_i, \sigma_i) N(s_j; \mu_j, \sigma_j) B(\eta_k; \alpha_k, \beta_k)$$

with the likelihood $l(s_i, s_j, \eta_k)$ given in Eq. (5). Then the posterior can be calculated from Bayes' rule. However, since the marginal posterior will be again used as the prior for the coming pairs, it is difficult to directly use the exact inferred posterior. Therefore, we approximate the posterior $p(s_i, s_j, \eta_k | o_i \succ_k o_j)$ using the variational approximation:

$$\begin{aligned} & p(s_i, s_j, \eta_k | o_i \succ_k o_j) = l(s_i, s_j, \eta_k) p(s_i, s_j, \eta_k)/C \qquad (9) \\ & \approx N(s_i; \mu_i^{i \succ_k j}, \sigma_i^{i \succ_k j}) N(s_j; \mu_j^{i \succ_k j}, \sigma_j^{i \succ_k j}) B(\eta_k; \alpha_k^{i \succ_k j}, \beta_k^{i \succ_k j}). \end{aligned}$$

where

$$l(s_i, s_j, \eta_k) = \Pr(o_i \succ_k o_j) = \eta_k \frac{e^{s_i}}{e^{s_i} + e^{s_j}} + (1 - \eta_k) \frac{e^{s_j}}{e^{s_i} + e^{s_j}},$$

is the likelihood function and $C = \Pr(o_i \succ_k o_j)$ is the normalization constant. In particular, we assume s_i, s_j and η_k are (conditionally) independent in posterior, the posterior distributions for s_i and s_j are still Gaussian, and η_k is Beta.

Let us defer the discussion of how to efficiently update the posterior parameters to the next subsection and first present the proposed active learning strategy. For each potential triplet in the pool (o_i, o_j, a_k) (*i.e.*, represents asking the k-th annotator to compare o_i and o_j), we compute the expected information gain:

$$\begin{aligned} \Pr(o_i \succ_k o_j) \quad & \Big(\mathrm{KL}\left(N(\mu_i^{i \succ_k j}, \sigma_i^{i \succ_k j}) || N(\mu_i, \sigma_i)\right) \qquad (10) \\ & + \mathrm{KL}\left(N(\mu_j^{i \succ_k j}, \sigma_j^{i \succ_k j}) || N(\mu_j, \sigma_j)\right) \\ & + \gamma \, \mathrm{KL}\left(\mathrm{Beta}(\alpha_k^{i \succ_k j}, \beta_k^{i \succ_k j}) || \mathrm{Beta}(\alpha_k, \beta_k)\right)\Big) \\ + \Pr(o_i \prec_k o_j) \quad & \Big(\mathrm{KL}\left(N(\mu_i^{i \prec_k j}, \sigma_i^{i \prec_k j}) || N(\mu_i, \sigma_i)\right) \\ & + \mathrm{KL}\left(N(\mu_j^{i \prec_k j}, \sigma_j^{i \prec_k j}) || N(\mu_j, \sigma_j)\right) \\ & + \gamma \, \mathrm{KL}\left(\mathrm{Beta}(\alpha_k^{i \prec_k j}, \beta_k^{i \prec_k j}) || \mathrm{Beta}(\alpha_k, \beta_k)\right)\Big) \end{aligned}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback-Leibler (KL) divergence. Since we do not know whether $o_i \succ_k o_j$ or not before the pair is labeled, we take the *expected* information gain over the Bernoulli outcome; the computation of $\Pr(o_i \succ_k o_j)$ is shown in the next section (Eq. (16) and Eq. (19)). At each iteration, we choose the triplet (o_i, o_j, a_k) that maximizes Eq. (10). In other words, we use a pure greedy strategy to select the most informative triplet. We also realize that other mixed methods may work better in practice. For example, one can use an ϵ-greedy approach, *i.e.*, with probability $1-\epsilon$ select the triplet that maximizes the expected information gain, else with probability ϵ, select a random triplet.

The expected information gain defined via KL divergence has been a popular utility function in traditional active learning [22] and used for ranking problems [17]. To extend active learning to crowdsourcing, our formulation in Eq. (10) generalizes the traditional expected information gain by introducing an extra parameter γ. In particular, recall that the traditional information gain is simply defined by the KL divergence between the posterior and the prior:

$$\begin{aligned} & \mathrm{KL}\big(p(s_i, s_j, \eta_k | o_i \succ_k o_j) \, || \, p(s_i, s_j, \eta_k)\big) \qquad (11) \\ & = \mathrm{KL}\left(N(\mu_i^{i \succ_k j}, \sigma_i^{i \succ_k j}) \, || \, N(\mu_i, \sigma_i)\right) \\ & + \mathrm{KL}\left(N(\mu_j^{i \succ_k j}, \sigma_j^{i \succ_k j}) \, || \, N(\mu_j, \sigma_j)\right) \\ & + \mathrm{KL}\left(\mathrm{Beta}(\alpha_k^{i \succ_k j}, \beta_k^{i \succ_k j}) \, || \, \mathrm{Beta}(\alpha_k, \beta_k)\right). \end{aligned}$$

As compared to Eq. (11), our formulation in Eq. (10) introduces the parameter γ which represents the tradeoff between exploration and exploitation. A larger γ will give more weight to the KL divergence terms related to annotator quality in the objective in Eq. (10), which means that we are willing to spend more to explore the quality of annotators. On the other hand, a smaller γ will result in relatively more emphasis on exploiting the information in the observed pairwise comparisons. When there are gold samples, one could choose γ by testing the performance on the gold samples. However, when gold samples are not available, according to our experience, any $\gamma \in [5, 10]$ could lead to much better performance than setting $\gamma = 0$ (*i.e.*, traditional active learning without exploring annotator quality) or $\gamma = 1$ (*i.e.*, traditional information gain defined by KL divergence). Meanwhile, setting γ larger than 10 could lead to too much exploration at the beginning–especially when the budget is limited. Therefore, as a simple rule of thumb, one could set $\gamma = 5$ when the budget is limited while $\gamma = 10$ when the budget is sufficient. Although such a simple rule is by no means an optimal choice of γ, it often leads to superior empirical performance. We can adopt a more sophisticated guideline for the selection of γ. Specifically, we can start from a large γ; and gradually reduce the parameter γ by half for every $\tau\%$ of the budget (*e.g.*, $\tau = 25$). The reason behind such a dynamic strategy of setting γ is as follows: at the beginning, we may typically have very little knowledge about the quality of annotators, so more exploration should be carried out with a larger γ. As we gradually gather more information about annotator quality, we should do more exploitation instead of exploration using a smaller γ.

4.1 Online Learning

To update the posterior parameters efficiently in Eq. (9), we use a moment-matching strategy. We first approximate the first- and second-order moments for s_i, s_j, η_k under the true posterior distribution and then update the posterior parameters accordingly. To compute $\mathbb{E}(s_i)$, $\mathbb{E}(s_j)$, $\mathrm{Var}(s_i)$, $\mathrm{Var}(s_j)$ under the true posterior, we first integrate out η_k and the marginal posterior for (s_i, s_j) takes the form:

$$f_s(s_i, s_j)N(s_i; \mu_i, \sigma_i)N(s_j; \mu_j, \sigma_j),$$

where

$$f_s(s_i, s_j) = \frac{\alpha_k}{\alpha_k + \beta_k}\frac{e^{s_i}}{e^{s_i} + e^{s_j}} + \frac{\beta_k}{\alpha_k + \beta_k}\frac{e^{s_j}}{e^{s_i} + e^{s_j}}.$$

Let $z_i = \frac{s_i - \mu_i}{\sigma_i} \sim N(0, 1)$ and $z_j = \frac{s_j - \mu_j}{\sigma_j} \sim N(0, 1)$. We can view $f_s(s_i, s_j)$ as a function of z_i, z_j and rewrite it as:

$$\begin{aligned} f_z(z_i, z_j) = \quad & \frac{\alpha_k}{\alpha_k + \beta_k}\frac{e^{\sigma_i z_i + \mu_i}}{e^{\sigma_i z_i + \mu_i} + e^{\sigma_j z_j + \mu_j}} \\ & + \frac{\beta_k}{\alpha_k + \beta_k}\frac{e^{\sigma_j z_j + \mu_j}}{e^{\sigma_i z_i + \mu_i} + e^{\sigma_j z_j + \mu_j}}. \end{aligned}$$

Using the extension of Stein's Lemma [28], the expectation $\mathbb{E}(z_i)$ can be approximated as in [27]:

$$\mathbb{E}(z_i) \quad = \quad \mathbb{E}\Big(\frac{\partial f_z(z_i, z_j)/\partial z_i}{f_z(z_i, z_j)}\Big) \approx \frac{\partial \log f_z(z_i, z_j)}{\partial z_i}\bigg|_{z_i = z_j = 0}.$$

Therefore, we have:

$$\begin{aligned} \mu_i^{i \succ_k j} \quad &= \quad \mathbb{E}(s_i) = \mu_i + \sigma_i \mathbb{E}(z_i) \qquad (12) \\ &\approx \quad \mu_i + \sigma_i^2 \left(\frac{\alpha_k e^{\mu_i}}{\alpha_k e^{\mu_i} + \beta_k e^{\mu_j}} - \frac{e^{\mu_i}}{e^{\mu_i} + e^{\mu_j}}\right). \end{aligned}$$

We can interpret this updating rule as follows. For a perfect annotator with $\alpha_k \gg \beta_k$, we have $\frac{\alpha_k e^{\mu_i}}{\alpha_k e^{\mu_i} + \beta_k e^{\mu_j}} \approx 1$ and hence $\mu_i^{i \succ_k j} \approx \mu_i + \sigma_i^2 \frac{e^{\mu_j}}{e^{\mu_i} + e^{\mu_j}}$. This formulation captures the intuition that μ_i should increase when an observation $o_i \succ_k o_j$ is made by a good annotator. Secondly, if $\mu_i \gg \mu_j$, the extra information of observing $o_i \succ_k o_j$ is limited and hence the amount of increase of μ_i, $\sigma_i^2 \frac{e^{\mu_j}}{e^{\mu_i} + e^{\mu_j}}$, is very small. On the other hand, for a random annotator with $\alpha_k \approx \beta_k$, we have $\mu_i^{i \succ_k j} \approx \mu_i$ while for a malicious annotator, $\mu_i^{i \succ_k j} \approx \mu_i - \sigma_i^2 \frac{e^{\mu_i}}{e^{\mu_i} + e^{\mu_j}}$. Similarly, we have:

$$\begin{aligned} \mu_j^{i \succ_k j} \quad &= \quad \mathbb{E}(s_j) = \mu_j + \sigma_j \mathbb{E}(Z_j) \qquad (13) \\ &\approx \quad \mu_j + \sigma_j^2 \left(\frac{\beta_k e^{\mu_j}}{\alpha_k e^{\mu_i} + \beta_k e^{\mu_j}} - \frac{e^{\mu_j}}{e^{\mu_i} + e^{\mu_j}}\right) \\ &\approx \quad \mu_j - \sigma_j^2 \left(\frac{\alpha_k e^{\mu_i}}{\alpha_k e^{\mu_i} + \beta_k e^{\mu_j}} - \frac{e^{\mu_i}}{e^{\mu_i} + e^{\mu_j}}\right). \end{aligned}$$

We can derive $\sigma_i^{i \succ_k j}$, $\sigma_j^{i \succ_k j}$ in the same way. Using the extension of Stein's Lemma as in [28, 27]:

$$\begin{aligned} & \mathrm{Var}(z_i) = \mathbb{E}(z_i^2) - (\mathbb{E}(z_i))^2 \\ & = \left(1 + \mathbb{E}\left(\frac{\partial^2 f_z(z_i, z_j)/\partial z_i^2}{f_z(z_i, z_j)}\right)\right) - \mathbb{E}\Big(\frac{\partial f_z(z_i, z_j)/\partial z_i}{f_z(z_i, z_j)}\Big)^2 \\ & = 1 + \mathbb{E}\left(\frac{\partial^2 \log f_z(z_i, z_j)}{\partial z_i^2}\right) \approx 1 + \frac{\partial^2 \log f_z(z_i, z_j)}{\partial z_i^2}\bigg|_{z_i = z_j = 0}. \end{aligned}$$

Then we have:

$$\begin{aligned} & \left(\sigma_i^{i \succ_k j}\right)^2 = \mathrm{Var}(s_i) = \sigma_i^2 \mathrm{Var}(z_i) \qquad (14) \\ & = \sigma_i^2 \max\left(1 + \sigma_i^2 \left(\frac{\alpha_k e^{\mu_i} \beta_k e^{\mu_j}}{(\alpha_k e^{\mu_i} + \beta_k e^{\mu_j})^2} - \frac{e^{\mu_i} e^{\mu_j}}{(e^{\mu_i} + e^{\mu_j})^2}\right), \kappa\right), \\ & (\sigma_j^{i \succ_k j})^2 = \mathrm{Var}(s_j) = \sigma_j^2 \mathrm{Var}(z_j) \qquad (15) \\ & = \sigma_j^2 \max\left(1 + \sigma_j^2 \left(\frac{\alpha_k e^{\mu_i} \beta_k e^{\mu_j}}{(\alpha_k e^{\mu_i} + \beta_k e^{\mu_j})^2} - \frac{e^{\mu_i} e^{\mu_j}}{(e^{\mu_i} + e^{\mu_j})^2}\right), \kappa\right). \end{aligned}$$

where the parameter κ is a small constant (*e.g.*, 10^{-4}) to ensure the positivity of variance.

To update α_k and β_k, let $C_1 = \mathbb{E}_N\left(\frac{e^{s_i}}{e^{s_i} + e^{s_j}}\right)$, where $\mathbb{E}_N$ denotes the expectation over the prior Gaussian distribution of (s_i, s_j) and let $C_2 = \mathbb{E}_N\left(\frac{e^{s_j}}{e^{s_i} + e^{s_j}}\right) = 1 - C_1$. The normalization constant $C = \Pr(o_i \succ_k o_j)$ in Eq. (9) can be computed as:

$$\begin{aligned} C \quad &= \quad \int_{[0,1]} (C_1 \eta_k + C_2(1 - \eta_k))\,\mathrm{Beta}(\eta_k; \alpha_k, \beta_k) d\eta_k \\ &= \quad \frac{C_1 \alpha_k + C_2 \beta_k}{\alpha_k + \beta_k}. \qquad (16) \end{aligned}$$

Then we can compute the first and second order moment of η_k as follows:

$$\begin{aligned} \mathbb{E}(\eta_k) \quad &= \quad \frac{1}{C}\int_{[0,1]} \eta_k\,(C_1 \eta_k + C_2(1 - \eta_k))\,\mathrm{Beta}(\eta_k; \alpha_k, \beta_k) d\eta_k \\ &= \quad \frac{C_1(\alpha_k + 1)\alpha_k + C_2 \alpha_k \beta_k}{C(\alpha_k + \beta_k + 1)(\alpha_k + \beta_k)}. \\ \mathbb{E}(\eta_k^2) \quad &= \quad \frac{C_1(\alpha_k + 2)(\alpha_k + 1)\alpha_k + C_2(\alpha_k + 1)\alpha_k \beta_k}{C(\alpha_k + \beta_k + 2)(\alpha_k + \beta_k + 1)(\alpha_k + \beta_k)}. \end{aligned}$$

and update the α_k and β_k as follows:

$$\alpha_k^{i \succ_k j} \quad = \quad \frac{(\mathbb{E}(\eta_k) - \mathbb{E}(\eta_k^2))\mathbb{E}(\eta_k)}{\mathbb{E}(\eta_k^2) - (\mathbb{E}(\eta_k))^2} \qquad (17)$$

$$\beta_k^{i \succ_k j} \quad = \quad \frac{(\mathbb{E}(\eta_k) - \mathbb{E}(\eta_k^2))(1 - \mathbb{E}(\eta_k))}{\mathbb{E}(\eta_k^2) - (\mathbb{E}(\eta_k))^2} \qquad (18)$$

Now the challenge is to compute C_1 efficiently. Let $g(s_i, s_j) = \frac{e^{s_i}}{e^{s_i} + e^{s_j}}$ and we take the second-order Taylor expansion of $g(s_i, s_j)$ at (μ_i, μ_j):

$$\begin{aligned} & g(s_i, s_j) \approx g(\mu_i, \mu_j) + (s_i - \mu_i)\nabla_{s_i} g(\mu_i, \mu_j) + \\ & (s_j - \mu_j)\nabla_{s_j} g(\mu_i, \mu_j) + \frac{1}{2}(s_i - \mu_i)^2 \nabla_{s_i, s_i} g(\mu_i, \mu_j) + \\ & (s_i - \mu_i)(s_j - \mu_j)\nabla_{s_i, s_j} g(\mu_i, \mu_j) + \frac{1}{2}(s_j - \mu_j)^2 \nabla_{s_j, s_j} g(\mu_i, \mu_j). \end{aligned}$$

We take the expectation of $g(s_i, s_j)$ under the prior distribution and we obtain C_1. In particular, by the fact that $\mathbb{E}_N(s_i - \mu_i) = \mathbb{E}_N(s_j - \mu_j) = 0$, we have:

$$C_1 \approx \frac{e^{\mu_i}}{e^{\mu_i} + e^{\mu_j}} + \frac{1}{2}(\sigma_i^2 + \sigma_j^2)\frac{s^{\mu_i} s^{\mu_j}(s^{\mu_j} - s^{\mu_i})}{(s^{\mu_i} + s^{\mu_j})^3}. \qquad (19)$$

Algorithm 1 Active Ranking Aggregation in Crowd

```
Input: Prior distribution parameters $\{\mu_i\}$, $\{\sigma_i\}$, $\{\alpha_k\}$, $\{\beta_k\}$, the tradeoff parameter $\gamma$ and the total budget $T$.
for $t = 1, \ldots T$ do
    Select a pair $(o_a, o_b)$ and an annotator $k$ which maximize the expected information gain in Eq. (10).
    Query the annotator $k$ on the preference between $o_a$ and $o_b$.
    if $o_a \succ_k o_b$ then
        Set $i = a$ and $j = b$
    else
        Set $i = b$ and $j = a$.
    end if
    Update $\mu_i$, $\mu_j$, $\sigma_i$, $\sigma_j$, $\alpha_k$ and $\beta_k$ according to Eq. (12), (13), (14), (15), (17) and (18).
end for
Output: Rank objects by sorting the obtained $\{\mu_i\}$.
```

We can use the above closed-form updating rules to infer approximate posterior distributions in constant time. Combining the posterior update rules with our selection criteria based on expected information gain in Eq. (10), the entire active algorithm is presented in Algorithm 1. We note that, after labeling the pairs, we can simply sort $\{\mu_i\}$ to obtain the ranking (which is used in our experiments), or we could rank the objects by using Crowd-BT. In general, solving the optimization of Crowd-BT leads to a slightly better performance than sorting $\{\mu_i\}$ but is also computationally more expensive.

5. EXPERIMENTS

5.1 Simulated Study

5.1.1 *Accuracy for Different Distributions of Annotator Quality*

We first conduct experiments with simulated data to test the performance where the average quality of annotators is varied. We assume that there are 100 objects, each with an underlying true score in the range of 1 to 100. We randomly sample 400 pairs of objects and assume that each pair is labeled by 10 different annotators. In this way, we gather 4000 labeled pairs. We assume that the ground truth quality of 100 annotators $\{\eta_k^*\}_{k=1}^{100}$ follow a Beta distribution Beta(α, β). For any pair (o_i, o_j) labeled by the k-th annotator, he/she will claim $o_i \succ_k o_j$ with the probability η_k and vice versa with the probability $1 - \eta_k$. We test our Crowd-BT method Eq. (8) with two different initialization schemes: (1) initialize each η_k by 1, *i.e.*, starting by assuming that all annotators are perfect (Crowd-BT-One) and (2) initialize each η_k by the accuracy on 5 gold pairs with known true relationship (Crowd-Gold); and compare them with the vanilla Bradley-Terry model (BT) in Eq. (3). We evaluate algorithms using the accuracy based on Wilcoxon-Mann-Whitney statistics:

$$\mathrm{ACC} := \frac{\sum_{i,j} \mathbf{I}(y_i > y_j \wedge s_i > s_j)}{\sum_{i,j} \mathbf{I}(y_i > y_j)}, \tag{20}$$

where $\mathbf{y}$ is the true relevance score and $\mathbf{s}$ is the estimated score.

We first vary the distribution of annotator quality and report the accuracy for each distribution in Table 1 with virtual node regularization parameter $\lambda = 0.5$. The sensitivity of λ will be further investigated in another simulated experiment. As we can see from Table 1, when the average quality is above 0.5, the two initialization strategies for Crowd-BT achieve very similar performance and are both better than the Bradley-Terry model. For a difficult scenario with many more malicious annotators, the performance of Crowd-BT with "all ones" initialization is indeed quite bad. However, initialization by a very rough estimate of quality using only five gold pairs will lead to a significant boost in performance. This is because our method has the ability to automatically recover from the errors made by malicious annotators with a reasonable initialization. In summary, when there are more good annotators, which is often the case in practice, we could directly apply Crowd-BT with "all ones" initialization; otherwise, it is necessary to obtain a rough estimate of quality via several gold samples.

5.1.2 *Exploration-Exploitation Tradeoff*

We now investigate the effect of considering the control of an exploration-exploitation tradeoff in active learning and compare our active learning method with different configurations of the tradeoff to the random selection strategy. In particular, we assume that the true quality for each annotator is drawn from the Beta distribution Beta$(2, 1)$. This is similar to one of the most common settings in practice where we have many good annotators but also some spammers and malicious contributors. We initialize the prior of the quality with Beta$(10, 1)$ to reflect our starting assumption that all annotators are very good. We plot the number of sampled pairs against accuracy by varying the parameter γ. As displayed in Figure 2, the active learning strategy with an appropriate γ (*e.g.*, blue line with $\gamma = 5$) significantly outperforms the random selection strategy. If one uses too small a value for γ (*e.g.*, red line with $\gamma = 0$), the accuracy has a sharp increase at the beginning, but becomes worse as we sample more pairs. This outcome arises because in the absence of enough exploration of annotator quality, we may assign many pairs to bad annotators and thus harm the performance in the long run. On the other hand, if we adopt too large γ (*e.g.*, black line with $\gamma = 30$), the increase in accuracy is slow at the beginning. The main reason for this is that during the first few hundred iterations, we perform too much exploration and hence obtain limited information about pairwise preferences.

We also study the exploration-exploitation tradeoff under different settings of averaged annotator quality. For each distribution of annotator quality and each setting of γ, we calculate the normalized area under the active learning curve. A typical active learning curve is displayed in Figure 2. As presented in Figure 3, in both (a) and (b) with different priors, when the average quality is relatively low (*e.g.*, 0.667), a larger γ (*e.g.*, $\gamma = 5, 10$) performs best, which means that we need more exploration in that scenario. On the other hand, when average quality approaches one, exploration becomes unnecessary and $\gamma = 0$ or $\gamma = 1$ leads to the best performance.

5.1.3 *Virtual Node Regularization*

Finally, we conduct a simulated experiment to investigate the effect of the virtual node regularization parameter λ.

(α, β)	(10, 1)	(5,1)	(2,1)	(2,2)	(1,2)	(1,5)
Average Quality	0.909	0.833	0.667	0.500	0.333	0.166
BT	0.882	0.890	0.800	0.542	0.171	0.144
Crowd-BT-One	**0.899**	**0.918**	**0.869**	0.849	0.109	0.122
Crowd-BT-Gold	**0.899**	0.917	**0.869**	**0.850**	**0.897**	**0.878**

Table 1: Accuracies of different approaches on the simulated datasets. Average Quality is the mean $\alpha/(\alpha+\beta)$ of the Beta distribution. Best performance in each column is in bold.

$n \times m$	4000×1			400×10			200×20		
λ	0.1	0.5	1	0.1	0.5	1	0.1	0.5	1
BT (ACC)	0.849	0.849	0.849	0.803	0.804	0.804	0.745	0.748	0.749
Crowd-BT (ACC)	**0.955**	0.946	0.936	0.893	**0.894**	0.883	0.793	0.803	**0.810**
Crowd-BT Estimate vs. Truth									
Quality Correlation	0.956	0.950	0.945	0.957	0.950	0.947	0.972	0.970	0.967

Table 2: Simulated studies for the virtual node regularization. Best performance in each block is in bold.

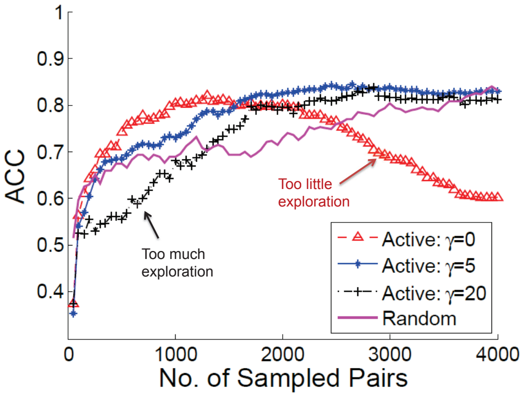
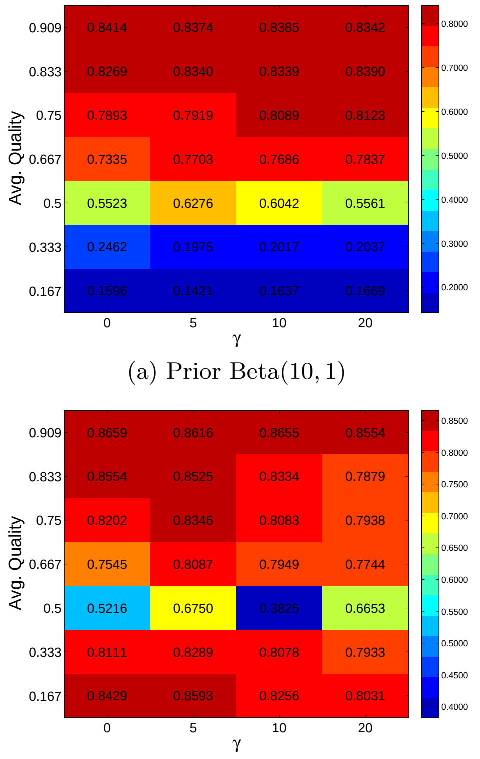


Figure 2: Active Learning with different γ.

We generate the synthetic data in a similar manner as in the previous section. Again, we assume that there are 100 objects, each with an underlying true score in the range 1 to 100, and 100 annotators with the ground truth quality $\{\eta_k^*\}_{k=1}^{100}$ following a Beta distribution Beta$(2,1)$. We randomly sample n pairs of objects and assume that each pair is labeled by $m = (4000/n)$ distinct annotators. We test our Crowd-BT algorithm in Eq. (8) with different virtual node regularization λ under different settings of n and compare the accuracies with that from the Bradley-Terry (BT) model. We report the accuracy and the correlation between the estimated quality $\widehat{\eta}$ and true quality η^* in Table 2. For all settings, Crowd-BT is superior to the baseline Bradley-Terry model. In addition, the correlations between the estimated quality and true quality are very close to one. More interestingly, when n is smaller (*i.e.*, the number of unique pairs is small), we need a larger regularization weight λ to achieve better performance. In fact, when $n = 4000$, it is very likely that the underlying comparison graph is strongly connected and hence we do not need a strong regularization. On the other hand, when $n = 200$, then we have at most 400 directed edges in the graph. In such a sparse graph, strong regularization will help to improve performance.

5.2 Real-World Challenge: Reading Level

We now apply Crowd-BT to the task of ranking documents by their reading difficulty. Our dataset is composed of 491 documents, each assigned a gold-standard reading difficulty level from 1 to 12, as described in [3] in more detail. Using the CrowdFlower crowdsourcing platform,[5] a total of 624 distinct annotators in the United States and Canada were shown representative passages from randomly selected pairs of these documents, and asked to decide which of the two texts was more challenging to read and understand. To help avoid an imbalanced judgment pool that is biased to-



(a) Prior Beta$(10,1)$

(b) Prior estimated on 5 gold pairs

Figure 3: Normalized area under the active learning curve for different averaged quality and γ.

[5] http://crowdflower.com/

	ACC				
	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 10$	$\lambda = 50$
BT	0.6760	0.6796	*0.6815*	0.6802	0.6629
Majority-Vote-BT	0.6686	*0.6700*	0.6688	0.6483	0.6409
Variance-BT	0.6790	0.6835	*0.6862*	0.6828	0.6658
Crowd-BT	**0.6924**	**0.6961**	***0.6978***	**0.6874**	**0.6690**

Table 3: ACC for different methods on reading level dataset (with all 12,728 pairs). Best performance in each column is in bold. Best performance in each row is in italics.

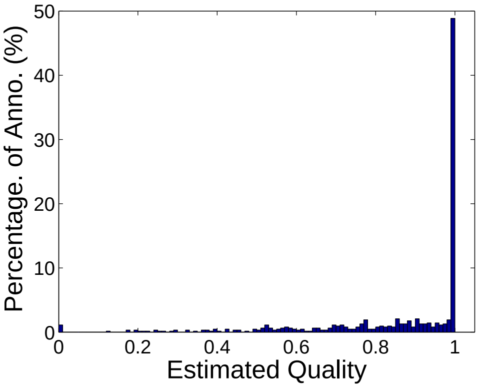


Figure 4: Histogram for the estimated η for different annotators.

	ACC
TrueSkill	0.6722 (0.002)
Online-Crowd-BT	0.6822 (0.002)

Table 4: Comparisons of online learning methods on reading level dataset (with all 12,728 pairs).

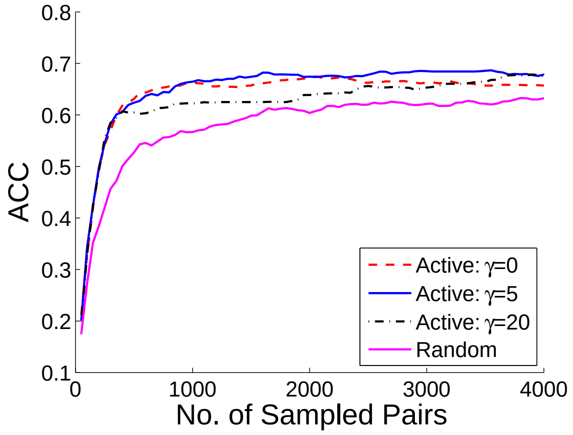


Figure 5: Active learning *vs.* random strategy on reading level dataset.

ward a few prolific annotators, each annotator was allowed to contribute a maximum of 40 judgments. We obtained a total of 12,728 pairwise comparisons. The overall quality of annotators on this task is known to be relatively high.

We compare Crowd-BT with $\eta_k = 1$ as the initialization to several competitors: (1) Bradley-Terry (BT) model; (2) for each pair of objects (o_i, o_j), we first use majority vote to obtain the preference between them and apply the BT model (*e.g.*, if 3 annotators claim $o_i \succ o_j$ and 2 claim $o_i \prec o_j$, then we generate a pair $o_i \succ o_j$ as labeled by a perfect annotator); (3) a model proposed in [24] where the difference for annotators is captured by a variance term in the logistic form in Bradley-Terry model. We call this method *Variance-BT*. As the evaluation metric, we again use the accuracy in Eq. (20) as in the simulated experiments which measures the overall accuracy across all pairs in the gold-standard ranking. The results are presented in Table 3. As we can see, Crowd-BT performs the best for any λ, followed by Variance-BT and Majority-Vote-BT, which has the worst performance. We also plot the histogram for the estimated η in Figure 4 and we observe that about half of the annotators are estimated to be perfect annotators on this dataset.

We also compare our Bayesian online Crowd-BT with another well-known online ranking aggregation algorithm: Trueskill [7]. Since the sample ordering in an online algorithm is random, the results could be slightly different for each run, so we report the mean and standard deviation over 50 runs in Table 4. As we can see, our method improves over the TrueSkill method's accuracy. We note that the performance of the online methods is worse than that of the deterministic methods. The main advantages of online methods are their computational efficiency and the ability to handle streaming data.

Finally, we compare the active-learning strategy with different exploration-exploitation tradeoffs, against the random strategy. For better visualization, we only present the accuracy for the first 4,500 labeled pairs in Figure 5. As captured in the figure, the active learning strategy significantly outperforms the random strategy. The exploration-exploitation tradeoff can also be observed from Figure 5. In particular, the accuracy for $\gamma = 0$ (red line) increases sharply at the beginning; on the other hand, the accuracy for $\gamma = 50$ (black line) increases slowly at the beginning but outperforms the $\gamma = 0$ case after about 3,000 samples. This indicates that different from traditional active learning, the exploration-exploitation tradeoff leveraged by γ is very important for active learning in crowdsourcing. In practice, according to our experience, we suggest choosing $\gamma \in [1, 10]$ to achieve better performance. To further quantify the improvement, we report the number of sampled pairs to first achieve a certain accuracy. In particular, the best accuracy over different methods after sampling all 12,728 pairs is 0.6843. We report the number of sampled pairs to achieve a certain ratio of best accuracy in Table 5. As we can see, the number of pairs needed for the active learning strategy is much smaller than that for the random strategy. With active learning, we can achieve 90% of the best accuracy with only about $400/12,728 \approx 3.14\%$ of the total pairs.

Ratio to best accuracy	$\gamma = 0$	$\gamma = 5$	$\gamma = 20$	Random
98 %	1850	*1400*	3650	7250
95 %	*700*	850	2450	5350
90 %	*400*	450	850	2150

Table 5: Number of pairs required to achieve a specified level of accuracy. Best performance in each row is in italics.

6. CONCLUSIONS AND FUTURE WORK

We have explored the challenge of learning a global ranking from pairwise comparisons via inputs from the crowd. We generalized the widely applied Bradley-Terry model by incorporating annotator quality. We further proposed an active learning strategy that can adaptively sample the next assessment pair and annotator. We introduced and studied an exploration-exploitation tradeoff in active learning with crowdsourcing pairwise comparisons, and demonstrated the importance of the configuration of this tradeoff via empirical studies. Although we developed methods on the foundation provided by the Bradley-Terry model, the proposed methods can be applied to other models for pairwise ranking, such as the Thurstone model [13].

We see several interesting future directions for extending this work. In one direction of research, we see opportunities for reducing the computational cost via narrowing the sampling space for active learning. Heuristics for narrowing the space promise to be valuable. For example, if we are certain about $o_i \succ o_j$ and $o_j \succ o_k$, we may exploit the nearly certain preference between o_i and o_k which can be inferred by the transitivity rule. A second direction of research centers on optimizing in an automated manner both the exploration-exploitation tradeoff parameter γ and virtual node regularization parameter λ. We are also interested in better ways of harnessing limited sets of gold samples in validation sets.

7. ACKNOWLEDGMENTS

We would like to thank TK Huang, Denny Zhou and Susan Dumais for helpful discussions and Vaughn Hester for support obtaining information from Crowdflower runs needed for this research.

8. REFERENCES

[1] B. Carterette, P. Bennett, D. Chickering, and S. Dumai. Here or there: Preference judgments for relevance. In *ECIR*, 2008.

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

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

[4] R. Coulom. Computing elo ratings of move patterns in the game of go. Technical report, Université Charles de Gaulle, 2007.

[5] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28:20–28, 1979.

[6] D. Gleich and L. Lim. Rank aggregation via nuclear norm minimization. In *KDD*, 2011.

[7] R. Herbrich, T. Minka, and T. Graepel. Trueskill(tm): A bayesian skill rating system. In *NIPS*, 2007.

[8] T.-K. Huang, R. C. Weng, and C.-J. Lin. Generalized bradley-terry models and multiclass probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.

[9] E. Kamar, S. Hacker, and E. Horvitz. Combing human and machine intelligence in large-scale crowdsourcing. In *AAMAS*, 2012.

[10] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, 2011.

[11] C. Lin and J. J. More. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9:1100–1127, 1999.

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

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

[14] L.R.Ford. Solution of a ranking problem from binary comparisons. *American Math Monthly*, 64:28–33, 1957.

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

[16] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2000.

[17] T. Pfeiffer, X. A. Gao, A. Mao, Y. Chen, and D. G. Rand. Adaptive polling for information aggregation. In *AAAI*, 2012.

[18] R. Plackett. The analysis of permutations. *Applied Statistics*, 24:193–302, 1975.

[19] T. Qin, X. Geng, and T.Y.Liu. A new probabilistic model for rank aggregation. In *NIPS*, 2010.

[20] R.A.Bradley and M. Terry. The rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324—345, 1952.

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

[22] B. Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

[23] K. Tsukida and M. R. Gupta. How to analyze paired comparison data. Technical report, University of Washington, 2011.

[24] M. N. Volkovs and R. S. Zemel. A flexible generative model for preference aggregation. In *WWW*, 2012.

[25] J. Wang, P. G. Ipeirotis, and F. Provost. Managing crowdsourcing workers. In *The 2011 Winter Conference on Business Intelligence*, 2011.

[26] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010.

[27] R. C. Weng and C.-J. Lin. A bayesian approximation method for online ranking. *Journal of Machine Learning Research*, 12:267–300, 2011.

[28] M. Woodroofe. Very weak expansions for sequentially designed experiments: linear models. *The Annals of Statistics*, 17:1087–1102, 1989.

[29] Y. Yan, R. Rosales, G. Fung, and J. G. Dy. Active learning from crowds. In *ICML*, 2011.