Pairwise Ranking Aggregation by Non-interactive Crowdsourcing with Budget Constraints

Changjiang Cai*, Haipei Sun*, Boxiang Dong†, Bo Zhang*, Ting Wang‡, Hui (Wendy) Wang*

* Stevens Institute of Technology, Hoboken, NJ, {ccai1,hsun15,bzhang31,Hui.Wang}@stevens.edu

† Montclair State University, Montclair, NJ, dongb@montclair.edu

‡ Lehigh University, Bethlehem, PA, ting@cse.lehigh.edu

***Abstract*—Crowdsourced ranking algorithms ask the crowd to compare the objects and infer the full ranking based on the crowdsourced pairwise comparison results. In this paper, we consider the setting in which the task requester is equipped with a limited budget that can afford only a small number of pairwise comparisons. To make the problem more complicated, the crowd may return *noisy* comparison answers. We propose an approach to obtain a good-quality full ranking from a small number of pairwise preferences in two steps, namely *task assignment* and *result inference*. In the task assignment step, we generate pairwise comparison tasks that produce a full ranking with high probability. In the result inference step, based on the transitive property of pairwise comparisons and truth discovery, we design an efficient heuristic algorithm to find the best full ranking from the potentially conflictive pairwise preferences. The experiment results demonstrate the effectiveness and efficiency of our approach.**

I. Introduction

Crowdsourcing has emerged over the last few years as an important new tool for a variety of tasks, as it enables faster task completion and cheaper cost than in-house solutions. Among various types of crowdsourcing tasks, ranking objects based on *the pairwise comparison* results from the crowd has witnessed many applications (e.g., image processing and data cleaning). In practice, the requester may only have a limited *budget* that is not able to afford all pairwise comparisons.

Despite the economical highlight, there are several major challenges related to ranking aggregation from crowdsourcing. First, the crowd may return *noisy* and *conflicting* pairwise preferences. Second, in the *non-interactive setting*, the requester releases the tasks to the crowd only once, and accepts the results generated from the collected answers. Therefore, the requester is not aware of the quality of the workers beforehand. Last but not least, even given a set of accurate pairwise preferences, it is computationally expensive to find the full ranking of the highest preference.

In this paper, we leverage the *transitive property* of pairwise comparisons to infer the preferences for the pairs that are not assigned to any worker. Based on the answers collected from the crowd, we estimate the quality of each worker and the true preference, named *truth*, of every pairwise comparison.

In particular, we consider the non-interactive crowdsourced ranking setting that allows transitive pairwise comparisons. The first step (called *task assignment*) generates a small number of pairwise comparisons from whose answers we can obtain a full ranking with high probability. The second step (called *result inference*) infers the ranking based on the transitivity and truth estimation. We design an efficient algorithm to find the full ranking. The experiment results demonstrate the efficiency and effectiveness of our approach.

II. Preliminaries

Ranking Setting. Given a set of n objects $\mathcal{O} = \{O_1, \dots, O_n\}$, a full ranking π is a permutation of $\{1, \dots, n\}$, such that $O_{\pi[i]}$ is preferred to $O_{\pi[i+1]}$ (denoted as $O_{\pi[i]} \prec O_{\pi[i+1]}$). For each pairwise comparison task $T = (O_i, O_j)$, a worker may vote for either $O_i \prec O_j$ or $O_j \prec O_i$. Note that the workers' preferences may be inconsistent, e.g., a worker votes that $O_i \prec O_j$, while another worker votes that $O_j \prec O_i$.

Crowdsourcing Setting. The requester is equipped with a limited budget B that can only afford $\ell < \binom{n}{2}$ comparison tasks. To increase the accuracy, each task is assigned to c workers, where c is a constant value. We assume that the ground truth of ranking is not available to the requester.

III. Approach Design

A. Task Assignment

Given n objects $\mathcal{O}$ and ℓ pairwise comparison tasks $\mathcal{T}$, $\mathcal{T}$ can be modeled as a *task graph*, which is a *un-weighted*, *un-directed* graph G_T that consists of n vertices and ℓ edges: (1) each object $O \in \mathcal{O}$ corresponds to a vertex $v \in G_T$; (2) each pairwise comparison $(O_i, O_j) \in \mathcal{T}$ corresponds to an edge $(v_i, v_j) \in G_T$. Based on the pairwise preferences collected from the workers, we can construct a *weighted*, *directed* graph G_P such that each edge $v_i \to v_j$ is associated with a weight $w_{ij} \in (0, 1]$, indicating the confidence of $O_i \prec O_j$. We say a node $v \in G_P$ is an *in-node* (*out-node*, resp.) if v only has incoming edges (outgoing edges, resp.).

A nice feature of pairwise preference is the *transitive property*, i,e, if $O_i \prec O_j$ and $O_j \prec O_k$, then it can be inferred that $O_i \prec O_k$. Following this, a new edge (v_i, v_j) can be inserted if there exists a path $P(v_i, \dots, v_j)$ in G_P. As we focus on the full ranking of the n objects, we only consider the paths in the preference graph whose length is no more than $n-1$. Let G_P^* be the *transitive closure* (i.e., the reachability relation) inferred, apparently, any Hamiltonian path (HP) (i.e., a path that visits each vertex exactly once) of G_P^* corresponds to a full ranking of $\mathcal{O}$.

We use $Pr(G_P^*)$ to denote the probability that the transitive closure G_P^* has an HP. In the task assignment procedure, our objective is to design the tasks $\mathcal{T}$ (and thus G_T) that is of *high HP-likelihood*, i.e., it maximizes $Pr(G_P^*)$ for any G_P constructed from G_T.

Theorem 3.1: Given a task graph G_T, let G_P and G_P^* be any preference graph and transitive closure constructed from G_T, there is no HP in G_P^* if: (1) G_T does not have an HP; or (2) G_P^* has at least two in-nodes or out-nodes.

According to Theorem 3.1, we infer that for any given task graph G_T, if all the vertices have the same degree, the probability to have at least two in-nodes or out-nodes in G_P^* is minimized, i.e., it is more likely that G_P^* has at least one HP. We design the following procedures to generate the task graph G_T that guarantees high probability of the existence of HP in G_P^*. First, we randomly construct an acyclic path P that connects all vertices in G_T. Next, we randomly insert edges so that the degree of each vertex reaches $\frac{2\ell}{n}$, where ℓ is the number of affordable comparison tasks.

B. Result Inference

Based on the pairwise preferences from the workers, we compute the full ranking in the following four steps.

Step 1. Truth Discovery of Direct Pairwise Comparisons. We observe that the workers' quality and the inferred truth are correlated: the workers who provide true pairwise preferences more often will be assigned higher quality (i.e., reliability degrees), and the pairwise preference that is supported by reliable workers will be regarded as the truth. Based on this intuition, given any task $T(O_i, O_j)$, the true preference $\hat{w_{ij}}$ is the weighted average of the workers' preferences, namely,

$$\hat{w_{ij}} = \frac{\sum_{W_k \in \overline{W_{ij}}} w_{ij}^k \times q_k}{\sum_{W_k \in \overline{W_{ij}}} q_k}, \quad (1)$$

where $\overline{W_{ij}}$ is the set of workers who perform the task, $w_{ij}^k \in \{0, 1\}$ is the k-th worker's preference, and q_k is the quality of the k-th worker.

We make the same assumption as [1] that the workers' error follows the normal distribution $\epsilon_k \sim \mathcal{N}(0, \sigma_k^2)$, where σ_k is the standard error deviation of worker W_k. Therefore, having $\hat{w_{ij}}$, we update the worker's quality, namely,

$$q_k \propto \frac{\mathcal{X}^2(\alpha/2, |\mathcal{T}_k|)}{\sum_{t_j \in \mathcal{T}_k} (w_{ij}^k - \hat{w_{ij}})^2}, \quad (2)$$

where α is a pre-defined value of the confidence interval, and $\mathcal{X}^2(\alpha/2, |\mathcal{T}_k|)$ is the $\frac{\alpha}{2}$-th percentile of a $\mathcal{X}^2$-distribution with degree of $|\mathcal{T}_k|$.

Using Equation (1) and (2), we keep updating the task truth preference $\{\hat{w_{ij}}\}$ and worker quality $\{q_k\}$, until it reaches convergence.

Step 2: Preference Smoothing. Our task assignment scheme cannot guarantee the existence of HP in G_P (i.e., full rankings). Following Theorem 3.1, the existence of in-/out- nodes in G_P is the main reason of HP failure. To ensure the existence of HP in G_P (and its transitive closure), our solution is to *smooth* the preference of each 1-edge (v_i, v_j) by estimating the *unknown* preference on the edge (v_j, v_i), such that $w_{ij} < 1$, $w_{ji} > 0$, while $w_{ij} + w_{ji} = 1$. In particular, given an 1-edge $(O_i, O_j) \in G_P$, we apply *smoothing* on w_{ij} and w_{ji} by the following: $w_{ij} = w_{ij} - \frac{\sum_{\forall W_k \in \overline{W_{ij}}} err_k}{|\overline{W_{ij}}|}$, and $w_{ji} = w_{ji} + \frac{\sum_{\forall W_k \in \overline{W_{ij}}} err_k}{|\overline{W_{ij}}|}$, where err_k is the error of the worker W_k that follows the distribution of $\mathcal{N}(0, \sigma_k^2)$.

Step 3: Computation of Indirect Pairwise Preference. Given a path $P(v_i, \ldots, v_j)$ whose length is larger than 1, the weight of the *indirect* edge (v_i, v_j) inferred from the path P is calculated as $w_{ij}^P = \prod_{\forall (v_x, v_y) \in P} w_{xy}$. Let w_{ij}^* be the indirect weight of (v_i, v_j), which is the sum of the weights of all paths between v_i and v_j, the final preference is computed as

$$\check{w_{ij}} = \beta w_{ij} + (1 - \beta) w_{ij}^*, \quad (3)$$

where β is a user-specified value that defines the weights on the (in)direct preferences. We also normalize the weights so that $\check{w_{ij}} + \check{w_{ji}} = 1$.

We have the following theorem that shows there always exists an HP in G_P^*.

Theorem 3.2: Given a preference graph G_P, let G_P^* be its transitive closure constructed by Step 1 - 3. Then there always exists an HP in G_P^*.

Step 4: Find the Best Ranking. Among the multiple HPs in G_P^*, we aim at picking the HP of the highest preference probability. Due to the potentially large number of HPs, we design an efficient algorithm, namely the simulated annealing based path search (SAPS) algorithm that returns the *heuristic* solution. In particular, we adapt the Simulated Annealing (SA) algorithm to heuristically searching a solution P^*. The key idea is to generate a new HP by rotating, reversing and swapping the current HP, and retain it with a probability based on Boltzmann probability distribution if the preference probability is not improved. After sufficient number of iterations, we can find a global optimal solution with high probability.

IV. Experiments

We compare our method with three baseline approaches in terms of both ranking accuracy and time performance, and report the result in Table I. The result suggests that our method delivers good ranking accuracy, if not the best, at cheap computational cost.

TABLE I: Comparison with Baselines

# of objects	100		200		300	
	Accu.	Time(s)	Accu.	Time(s)	Accu.	Time(s)
This Papaer	89.2%	0.363	92.9%	1.464	94.2%	3.882
RC [2]	2.00%	0.038	1.70%	0.090	1.50%	0.179
QS [3]	19.6%	0.275	23.1%	0.654	19.6%	1.159
CrowdBT [4]	93.8%	3529	92.4%	14938	89.1%	26012

V. Conclusion

In this paper, we study budget-conscious pairwise ranking aggregation problem in the non-interactive crowdsourcing setting. We design efficient algorithms that select a small number of pairwise comparisons by the crowd, and construct the full ranking from the workers' pairwise preferences by taking the workers' quality into consideration. In the future, we plan to investigate the top-k ranking problem in the same setting.

VI. Acknowledgement

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 1350324 and 1464800.

References

[1] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *PVLDB*, 2014.

[2] N. Ailon, "Aggregation of partial rankings, p-ratings and top-m lists," *Algorithmica*, vol. 57, no. 2, pp. 284–300, 2010.

[3] M. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," in *CIKM*, 2002, pp. 538–548.

[4] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, "Pairwise ranking aggregation in a crowdsourced setting," in *WSDM*, 2013, pp. 193–202.