Supplementary Material for
Active Learning for Top-K Rank Aggregation from Noisy Comparisons

Soheil Mohajer [1] **Changho Suh** [2] **Adel Elmahdy** [1]

1. Supplementary Algorithms for Heap Structure

For the sake of completeness, we also present a modified version of `Heapify` and `BuildHeap` algorithms that work based on noisy observations in Algorithm 1 and Algorithm 2, respectively. It is worth mentioning the sample complexity of building a `HEAP` over N items using noiseless measurements is $O(N)$, and insertion of a new item to an existing `HEAP` with N items requires $O(\log N)$ comparisons. Finally, `HEAP` maintains the maximum item at the beginning of the list, so that max-extraction can be done for free. Here, in the modified version of the algorithms, we repeat each comparison used for `HEAP` for m times, and decide based on the majority of the observation. The variable m is a design parameter, which will be determined later.

Algorithm 1 `Heapify`$(Z, X, i; m)$

Input: Integers i and m
Data: Array Z of indices and Data X
Output: Array Z with sub-tree at node i being a max-heap
left $\leftarrow 2i$
right $\leftarrow 2i + 1$
$T \leftarrow 0$
for $t \leftarrow 1$ **to** m **do**
  $e_t \leftarrow (Z[\text{left}], Z[i])$
  $T \leftarrow T + Y_t$
end for
if left $\leq |Z|$ ***and*** $T \geq \frac{m}{2}$ **then**
  max $\leftarrow$ left
else
  max $\leftarrow i$
end if
$T \leftarrow 0$
for $t \leftarrow 1$ **to** m **do**
  $e_t \leftarrow (Z[\text{right}], Z[\text{max})]$
  $T \leftarrow T + Y_t$
end for
if right $\leq |Z|$ ***and*** $T \geq \frac{m}{2}$ **then**
  max $\leftarrow$ right
end if
if max $\neq i$ **then**
  `swap`$(Z[i], Z[\text{max}])$
  `Heapify`$(Z, X, \text{max}; m)$
end if

Algorithm 2 `BuildHeap`$(Z, X; m)$

Input: Integer m
Data: Array Z of indices and Data X
Output: max-heap Z
for $i \leftarrow \lfloor |Z|/2 \rfloor$ **down to** 1 **do**
  `Heapify`$(Z, X, i; m)$
end for

2. Supplementary Simulation Results

2.1. Performance Evaluation of Top-1 and Top-K Ranking Algorithms

Figs. 1 and 2 depict the trade-off between the average number of pairwise comparisons and the empirical success rate of identifying top-1 and top-8 ranked items under two pairwise comparison probability models, namely uniform noise model and BTL model, respectively. The total number of items, from which we are interested in Top-K, is $n = 2^{12} = 4096$. We set two different values for the preference separation measure as $\Delta_K = 0.0025$ and $\Delta_K = 0.01$. The BTL model parameters are $w_i \in [1, 50]$, $1 \leq i \leq n$. Furthermore, for all simulation results presented in this section, success rate and total number of pairwise comparison are averaged over 10^4 Monte Carlo trials.

It is obvious that the success rate monotonically increases with the number of pairwise comparisons under all probability models. Moreover, we can see that as K increases or Δ_K decreases, the number of pairwise comparisons required for a target success rate increases. All these observations are consistent with our theoretical analysis.

[1]ECE, University of Minnesota, Twin Cities, MN, USA. [2]EE, KAIST, Daejeon, South Korea. Correspondence to: Soheil Mohajer <soheil@umn.edu>, Changho Suh <chsuh@kaist.ac.kr>, Adel Elmahdy <adel@umn.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

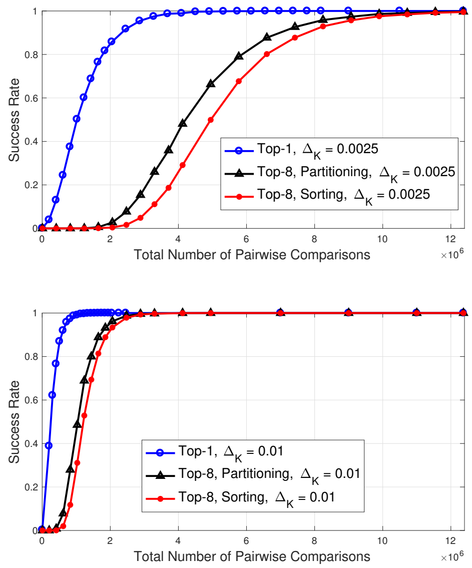

Figure 1. Evolution of success rate of the proposed algorithm with respect to the total number of pairwise comparisons under uniform noise model for different values of K and $\Delta_K (= \gamma^2)$ when $n = 4096$.

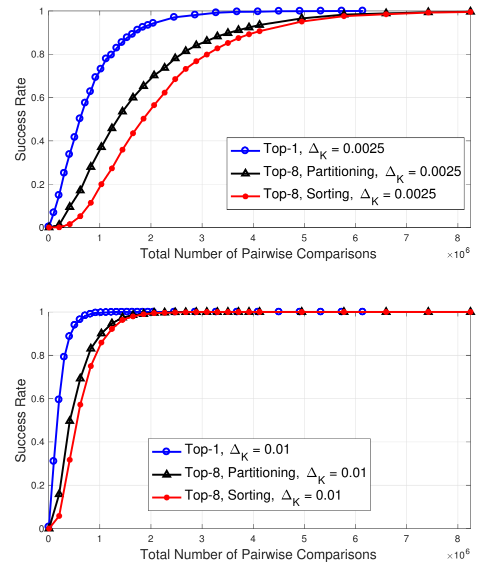

Figure 2. Evolution of success rate of the proposed algorithm with respect to the total number of pairwise comparisons under BTL model for different values of K and Δ_K when $n = 4096$ and $w_i \in [1, 50], 1 \leq i \leq n$.

For Top-K ranking with $K > 1$, we study the performance of the proposed heap-based ranking algorithm under two different settings: (1) Top-K Partitioning, and (2) Top-K Sorting. It should be noted that the second setting is more restrictive than the first one, and hence, the algorithm performance in Top-K Sorting is always inferior to the one in Top-K Partitioning. It is illustrated, through Figs. 1 and 2, that the performance gap between the two scenarios is small, for all probability models. This shows that the proposed algorithm is robust in that it provides the ordered top-K items without resulting in severe performance degradation, compared to the performance of only identifying the top-K items.

2.2. Performance Comparison of Active Ranking Algorithms

An interesting fact about our algorithm is revealed by Fig. 3. It is obvious that as the number of pairwise comparisons increases, the corresponding probability of error falls down. However, in Braverman algorithm, there is a threshold after which any further increase in the number of pairwise comparisons does not yield any improvement in the probability of error. These findings are consistent with our analysis of the sample complexity of both algorithms, and the upper bound imposed on c, a parameter that the sample complexity of Braverman algorithm scales with, such that $10 \leq c \leq \log(n)$ (Braverman et al., 2016).

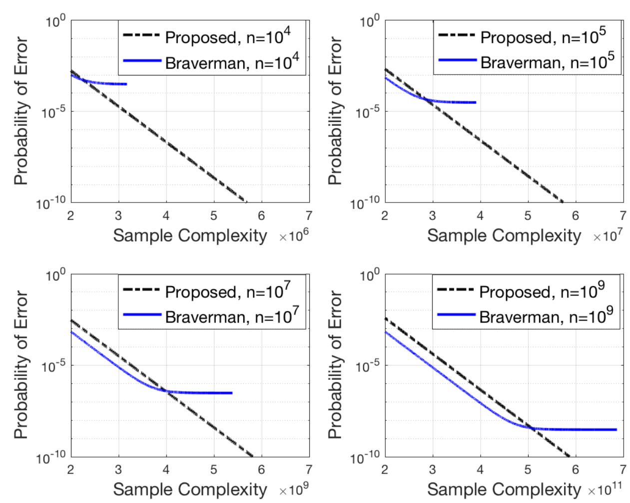

Figure 3. The analytical tradeoff between probability of error and sample complexity for Top-1 algorithm and Braverman algorithm under the uniform noise model for different values of n when $\Delta_K = \gamma^2 = 0.0225$.

References

Braverman, M., Mao, J., and Weinberg, S. M. Parallel algorithms for select and partition with noisy comparisons. *STOC*, 2016.