### Problem

One of the most important applications of rank aggregation [3], that is merging multiple ranked lists of objects, is in the field of information retrieval (IR). Typically, rank aggregation in IR is conducted for two different purposes:

1. Aggregating ranked lists of a document set ranked by different ranking methods.
2. Aggregating ranked lists of different document sets ranked by the same ranking method.

Our goal is to provide a solution for the second problem.

### Optimal Rank Aggregation: a rank aggregation problem in which, with the presence of relevancy information, items from different ranked lists are aggregated into a single final list so that:

- The final ranking has the best possible achievable performance;
- Local ordering of the initial lists are preserved in the final list.

#### Chen’s Algorithm: Bucketing Strategy

A bucket is denoted in the form of \((n_1, n_2, . . . , n_k)\) with each \(n_i\) representing the number of irrelevant and relevant documents in the bucket. In this example, based on Chen’s algorithm, the initial active set will be: \(\{0.1\}[A_1], (2.1)[B_1, B_2, B_3]\). Continuing the algorithm, it will produce the following rankings:

- The final ranking has the best possible achievable performance.
- The greedy approach does not necessarily lead to an optimal solution.
- In this example, the average precision of Chen’s algorithm output is 0.5280, while that of the optimal aggregated list is 0.6198.

#### Proposed DP Algorithm

We introduce an m-th order tensor \(M\) of size \((|b_0| + 1) \times \cdots \times (|b_m| + 1)\), in which value of an entry \(M[i_1, i_2, \ldots, i_m]\) shows the maximum AP obtainable by aggregating top \(i_1, i_2, \ldots, i_m\) buckets of the ranked lists 1, 2, . . . , respectively:

\[
M[i_1, i_2, \ldots, i_m] = \max_{k} \left( M[i_1, i_2, \ldots, i_{m−1}, k] + \text{AP-GAIN}(k, i_1, i_2, \ldots, i_m) \right)
\]

AP-GAIN is defined as follows:

\[
p = \sum_{i \in S} \sum_{j \in S} \text{RelevancyScore}(i, j)
\]

#### Acknowledgment

The authors would like to thank Google and SIGIR for providing a travel grant to attend this conference.

### References

