Empirical Study of Top-List Rank-Aggregation Methods

Ammar Eltigani and Tai Henrichs

November 23, 2020

Project Proposal

0.1 Focus

A top-k-list for $k \leq n$, where n is the number of candidates, ranks the k most-preferred candidates. Given some top-k-lists, the problem we will focus upon is to output a single ranking of all the candidates. The output ranking should minimize the Kendall-Tau distance across all of the top-k-lists.

0.2 Goals

We will empirically evaluate the performance and accuracy of the approximation algorithms for top-list aggregation provided by [5]. We will consider a range of consensus-levels, similar to [2]. We may evaluate algorithms on both synthetic and real-data sets or only synthetic data. Time permitting, we may apply additional criteria for evaluating top-k-list rank-aggregation methods such as resistance to manipulation, fair treatment of candidates, or ranker-privacy.

0.3 Resources

- [1] This paper introduces the top-list aggregation problem, defining it in relation to the broader partial-list aggregation problem. The author proves the hardness of aggregating top-lists and provides two constant factor approximation algorithms, which are improved by [5].
- [5] This paper contains the top-list aggregation methods we will be empirically evaluating. It provides a handful of approximation algorithms for aggregating Top-lists. They also consider a special case where top-lists have varying lengths.
- [2] The authors empirically evaluate a number of algorithms for outputting an approximate or exact Kemeny-ranking when provided with full rankings as inputs. They find that the best algorithm varies with

the consensus-level of rankers, with tractability growing with consensus. Their methods can guide our own approach to empirically analyzing top-list aggregation.

- [6] This paper details solutions to full-list rank-aggregation, providing useful background for this literature area. It also contains empirical analysis of such approaches.
- [3] The authors consider the full-list rank-aggregation problem in privacy sensitive context. They provide rank-aggregation algorithms that ensure no individual ranking can be re-constructed from the output ranking. They theoretically analyze the accuracy costs incurred by such protections.
- [4] The authors consider the full-list rank-aggregation problem in contexts where people may be ranked, potentially prejudicially, by rankers. They propose full-list rank-aggregation algorithms that satisfy certain fairness criteria for such settings.
- [7] This reviews the literature on strategic manipulation, primarily in voting-settings but also in some others, e.g. matching. Emphasis is placed upon the utility of NP-hardness results for indicating the manipulability of mechanisms in practice. Also presented are alternatives to computational-complexity for assessing the practical manipulability of mechanisms.

0.4 Action Plan

1. Read [1] and [6] by November 25th.
2. Read [5] and [2] by November 27th.
3. Complete implementation of algorithms by December 5th.
4. Run combination of experiments on algorithms and visualize performance using Python matplotlib and numpy by December 8th.
5. Create presentation for class by the end of December 9th.
6. Write final paper by the end of December 17th. This includes integrating data-interpretation and data visualizations.

If we have extra time, we can consider additional criteria mentioned in the goals section (privacy, strategic manipulation, etc.).

References

[1] Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*, 57(2):284–300, 2010.

[2] Alnur Ali and Marina Meilă. Experiments with kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012.

[3] Michael Hay, Liudmila Elagina, and Gerome Miklau. Differentially private rank aggregation. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 669–677. SIAM, 2017.

[4] Caitlin Kuhlman and Elke Rundensteiner. Rank aggregation algorithms for fair consensus. *Proceedings of the VLDB Endowment*, 13(12):2706–2719, 2020.

[5] Claire Mathieu and Simon Mauras. How to aggregate top-lists: Approximation algorithms via scores and average ranks. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2810–2822. SIAM, 2020.

[6] Frans Schalekamp and Anke van Zuylen. Rank aggregation: Together we're strong. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 38–51. SIAM, 2009.

[7] Toby Walsh. Is computational complexity a barrier to manipulation? *Annals of Mathematics and Artificial Intelligence*, 62(1-2):7–26, 2011.