Ranking Aggregation, Inference, and Design

Anthony Della Pella
Zijin Zhang

December 09, 2020

1. Introduction

Ranking is a popular topic of recommendation system. One fundamental problem is ranking inference. Given user preference information, ranking inference aims to decide the list of the K most popular items. The essential part of ranking inference problem is how to utilize input information and draw the optimal ranking decision. Papers in various areas including machine learning, approximation algorithms, and operations research study ranking design from different angles (Ailon *et al.* [2005],Radlinski *et al.* [2008],Derakhshan *et al.* [2018]). The ideal preference information is supposed to incorporate comparison results between any two items, but we very often face more complicated situations: only partial preference information is available. For example, users browse videos on YouTube. They only click a couple of attractive videos and provide no data to less interesting ones. Therefore, YouTube can not get access to the full preference information. Ranking agents can also have different objectives. In the e-commerce context, rather than directly inferring ranking sequence by popularity, online retailers focus on objectives like maximizing click through rate, maximizing revenue, etc.

In this report, we analyze some cutting-edge ranking methods which build on different information structures and model assumptions, especially those with approximation algorithm approaches. The presentation is intended for dissemination of new ideas from the theory of ranking aimed at a business audience. Our report is organized as follows. In section 2, we discuss the case where ranking agents have full access to user preference information. In section 3, we introduce the partial preference information case with additional choice mechanisms. In section 4, we extend the discussion to online learning algorithms without historical data. We conclude in section 5.

2. Full Preference Information

In this section, we introduce a stream of literature which assumes that each user submits the full ranking preference in input information. Suppose there are three items A, B, and

C. The full preference information consists of ranking sequence of each individual with form, $A \succeq B \succeq C$.

2.1. Definition of Problems

Several closely related problems are studied in the paper Ailon *et al.* [2005]. An algorithm is given and then analyzed for each problem type whereby approximation results are obtained. We state the problems below with the convention that $V = \{1, \ldots, n\}$:

Definition 2.1 (FAS-TOURNAMENT:)**.**
Input: A tournament represented as a digraph $G = (V, A)$ with the property that either $(i, j) \in A$ or $(j, i) \in A$ for all distinct $i, j \in V$.
Output: A permutation π on V minimizing the number of pairs i, j with $i <_\pi j$ and $(j, i) \in A$. We call such pairs backward edges with respect to the permutation π.

Definition 2.2 (RANK-AGGREGATION)**.**
Input: A list of k permutations (viewed as rankings) $\pi_1, \ldots, \pi_k$ on V.
Output: A permutation π minimizing the sum of distances:

$$\sum_{i=1}^{k} d(\pi, \pi_i) \tag{1}$$

where $d(\pi, \rho)$ is the number of i, j pairs such that $i <_\pi j$ but $j <_\rho i$. This is known as the Kemeny distance.

As mentioned in the introduction, aggregating ranking information is an important theoretical and applied problem. Several "real world" examples have been considered from a more theoretical stand point. These include voting, social choice theory, high-dimensional data mining and combining of micro-array databases. The latter two problems have been tackled by *consensus* or *ensemble clustering* and were studied in detail in Filkov & Skiena [2004] and Strehl & Ghosh [2003]. In Ailon *et al.* [2005] another closely related problem is given which captures the clustering studied in the aforementioned references.

Definition 2.3 (CONSENSUS-CLUSTERING)**.** Input: A list of k different clusterings $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of V. Output: One clustering $\mathcal{C}$ that minimizes

$$\sum_{i=1}^{k} d(\mathcal{C}, \mathcal{C}_i) \tag{2}$$

where the distance between two clusterings is the number of $\{i, j : i \neq j\} \subset V$ that are clustered together by one and separated by the other.

One last closely related notion of clustering presented as a computational problem is the following:

Definition 2.4 (CORRELATION-CLUSTERING)**.** Input: A relation $R : \{\{i, j\} : i, j \in V\} \to \{\oplus, \ominus\}$. Output: m disjoint clusters $C_1, \ldots, C_m$ covering V and minimizing the number of disagreement pairs. A disagreement pair here means a $\oplus$ labeled pair ending up in different clusters, or a $\ominus$ pair ending up in the same cluster.

We do not present analyses of Definition 2.3 or Definition 2.4 here, but do note that these problems are approximated in Ailon *et al.* [2005] utilizing and extension of the primary algorithm presented in the next subsection.

2.2. Pivoting Algorithm

Let $G = (V, A)$ be an instance of FAS-TOURNAMENT, Definition 2.1.

Algorithm 1: FAS-PIVOT($G = (V, A)$)

```
Set V_L → ∅, V_R → ∅.
Pick random pivot i ∈ V.
for j ∈ V\{i} do
    if (j, i) ∈ A then
        Add j to V_L (placing j on the left side);
    end
    else if (i, j) ∈ A then
        Add j to V_R (placing j on the right side);
    end
    Let G_L = (V_L, A_L) be the tournament induced by V_L.
    Let G_R = (V_R, A_R) be the tournament induced by V_R.
    Return FAS-PIVOT(G_L), i, FAS-PIVOT(G_R).
end
```

Theorem 2.5. *The algorithm* FAS-PIVOT *is a randomized expected* 3*-approximation algorithm for* FAS-TOURNAMENT.

The proof of this theorem relies on the observation that an edge $(i, j) \in A$ becomes a "backward" edge precisely when there is a third vertex k such that (i, j, k) form a directed triangle in G and k was chosen as a pivot when all three were input to the same recursive call. In short, the pivot step would then have i to its right and j to its left thus making (i, j) a backward edge.

The analysis then follows by first considering the set T of all directed triangles and then writing the cost of the algorithm in terms of a sum over this set. By writing the following linear programming problem :

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in A} x_e && (3) \\ \text{subject to} \quad & x_{e_1} + x_{e_2} + x_{e_3} \geq 1 && (4) \\ & \{e_1, \epsilon_2, e_3\} \in T && (5) \\ & x_e \geq 0. && (6) \end{aligned}$$

This LP lower bounds C^{OPT} and since a packing of triangles $\{\beta_t\}$ is a feasible solution it is also a lower bound on the optimal.

Lastly, such a packing is demonstrated using a probabilistic argument and the events $A_t :=$ one of the vertices of triangle t is chosen as a pivot when all three vertices are part of the same recursive call.

2.3. Weighted Tournaments And Rank Aggregation

Somewhat remarkably, the following strategy can be proven as a good approximation algorithm in many cases for the weighted version of FAS-TOURNAMENT, Definition 2.1. Given an instance $G_w = (V, A_w)$ with weights defined in the obvious way, construct the unweighted majority tournament $G_w = (V, A_w)$ and return FAS-PIVOT(G_w). Several results are then shown, the most important of which for our purposes are the following:

Lemma 2.6. *For an optimal (w.r.t.* FAS-TOURNAMENT*) permutation π^*, let $c^*(e)$ denote the cost incurred by e so that $C^{OPT} = \sum_{e \in A_w} c^*(e)$.*

If the weights satisfy the "probability constraints" $w_{ij} + w_{ji} = 1$, then $w(t) \leq 5c^*(t)$ for all $t \in T$, and if the weights satisfy the "triangle inequality" constraints $w_{ij} \leq w_{ik} + w_{kj}$ then $w(t) \leq 3c^*(t)$ for all $t \in T$.

This lemma allows us to prove approximation guarantees for FAS-PIVOT on RANK-AGGREGATION summarized in the following theorem:

Theorem 2.7. *The best of* FAS-PIVOT *on G_w and selecting a random permutation uniformly at random is an expected $11/7$ approximation for* RANK-AGGREGATION*, Definition 2.2.*

3. Partial Preference Information

Ranking aggregation assumes the unique optimal ranking which minimizes disagreements under contradictory information. However, there are three limitations about the above setting. First, ranking preferences are stochastic. The assumption that users have uniform and deterministic ranking preferences can be relaxed to a general setting where heterogeneous users make random decisions following some probability distribution. Second, the underlying intent mechanism is known. The mechanism provides additional information to infer the best ranking. In this sense, partial preference information could provide enough support in the ranking design phase. Third, objectives are various. Instead of simply ranking items according to relevance or popularity, different measures occur according to various objectives. For example, there is strong empirical evidence that on the e-commerce page, ranking positions affect clicks and ultimately, demand. Smart ranking design can affect choice behavior, such as click and purchase decisions on online platforms, therefore improve the total revenue. This leads to a new stream of literature, *display optimization*.

3.1. Definition of Problems

Position-based ranking model has been well-studied in literature varying among different assumptions and approaches. A recent paper Ferreira *et al.* [2019] studies how online platforms rank items to maximize the total impressed users, defined as people who click at least one item. Define user (indexed by t) type as $(\mathbf{p}_t, k_t)$, where $\mathbf{p}_t$ is the vector of intrinsic click probabilities, k_t is the attention window. Define the ranking sequence as the function $\pi : [n] \to [n]$ mapping from product to position.

Definition 3.1 (Impressed-User)**.**
Define the event that user t is "impressed" as

$$H_t(\pi) = \begin{cases} 1 & \text{if } \sum_{i \in [n]} C_{it}(\pi) \geq 1 \\ 0 & \text{if } \sum_{i \in [n]} C_{it}(\pi) = 0 \end{cases},$$

where C_{it} is a binary random variable indicating if item i is clicked or not.

It is based on the setting that each user has individual preference and attention window, where they will continue browsing if at least one item interests them within attention window. Therefore, the objective is:

$$\max_{\pi_1, \pi_2, \ldots, \pi_T} E\left[\sum_{t=1}^{T} H_t(\pi_t)\right],$$

The decision variables are ranking sequences π_t for each arriving user. This paper shows that in the offline setting where the user type distribution is known, this problem is NP-Hard. It is proved by constructing a special case and showing that the special case is equivalent to a stochastic version of the maximum coverage problem.

3.2. OffAR Algorithm

To address the above problem, this paper develops a greedy algorithm achieving $\frac{1}{2}$ approximation factor.

Algorithm 2: Greedy Algorithm for OffAR

Initialize null ranking $\pi^g(i) = \emptyset$ for all $1 \leq i \leq n$, and initialize unranked items $U = [n]$.
for $r \leftarrow 0$ **to** n **do**
  for $i \in U$ **do**
    Let $\tilde{\pi}^g \leftarrow \pi^g$;
    $\tilde{\pi}^g(i) \leftarrow r$;
    $\Delta_{ir} = E_{t \sim D}[H_t(\tilde{\pi}^g) - H_t(\pi^g)]$;
  end
  Let $i_r^* = \text{argmax}_{i \in U} \Delta_{ir}$;
  Set $\pi^g(i + r^*) \leftarrow r$;
  $U \leftarrow U \backslash \{i_r^*\}$.
end

Theorem 3.2. *The algorithm* OffAR *is a randomized expected* $\frac{1}{2}$*-approximation algorithm for* Impressed-User Ranking*.*

The intuition of OffAR algorithm is straightforward. OffAR ranks items from top positions to low positions. At each update, it places the product which has the largest attraction increment to the current position. The proof idea is similar to vertex cover problem. It estabishes a relationship between the optimal ranking π^* and the algorithm ranking π^g and prove the bound on the ratio of $H_t(\pi^*)$ to $H_t(\pi^g)$.

Furthermore, under pairwise independent assumptions, the greedy algorithm achieves a $(1 - \frac{1}{e})$-approximation factor. The key part of the proof states that this problem is a special case of stochastic submodular optimization problem. Badanidiyuru & Vondrák [2014] gives a $(1 - \frac{1}{e} - \epsilon)$-approximation for maximizing monotone stochastic submodular optimization functions. Derakhshan *et al.* [2018] considers different choice structure: two-stage sequential search model. This model is derived by Weitzman [1979]. Through characterizing the optimal strategy, it shows that ranking items in decreasing order of their preference weights does not necessarily maximize market share or user utility. Since the original problems are NP-complete, they design a PTSA algorithm, *w-ordered algorithm*, which achieves a multiplicative approximation factor of $\frac{1}{2}$ and an additive factor of 0.1716. The PTSA algorithm includes a DP-based approach.

4. Online Ranking

In practice, preference information is not available at the beginning. User data is collected while conducting the algorithm. Motivated by real-world examples in recommendation systems, web search, etc., some recent work studies learn-to-rank algorithms, i.e. underlying mechanism is known whereas parameters are unknown. A popular approach is the classic reinforcement leaning method, *multi-armed bandit (MAB)*. MAB demonstrates the exploration-exploitation tradeoff, allocating the limited resources under unknown environment.

Ferreira *et al.* [2019] proposes online algorithms to offer the ranking for users, balancing popularity and diversity. The theoretical bounds provide the performance ratio $\frac{1}{2+\alpha}$ with probability $1-\delta$ and the regret is bounded by $\mathcal{O}(nT^{2/3}(\log_{1+\alpha}(T))^{1/3}\log(T))$. The learning-to-rank problem studies how to learn rank web documents while learning Radlinski *et al.* [2008]. Instead of assuming that the relevance of one document is independent of other documents, this paper argues that the usefulness and relevance of a document does depend on other documents ranked higher, therefore clicks are dependent among documents. They consider the measure, regrets minimization, where the goal is to minimize the total number of poor rankings displayed over all time. Based on the regret measure, two algorithms are proposed to directly minimize the abandonment rate, where low abandonment rate implies high relevance and better ranking decision. These two algorithms achieve $(1-\frac{1}{e})$-approximation ratio and rather good practical performance. Kveton *et al.* [2015] modifies the upper confidence bound algorithm and proposes the cascading bandits algorithm, *UCB-cascade*, assuming user behavior follows cascade model.

5. Conclusion

Ranking has proven to be an interesting problem from the perspectives of both theoretical computer science and applied computer science. Progress has been made from both of these perspectives and in both cases optimal results have been proven. One promising aspect of ranking (both on the theory side and applied side of things) is the flexibility in the models. For instance, in Ailon *et al.* [2005] 5 models are discussed all of which follow the same general "full information inference" paradigm. When we consider that each of these models can be adapted to a weighted setting, it becomes clear that ranking is a general problem which can be modified slightly leading to new questions. In Ferreira *et al.* [2019] online ranking is discussed which is another promising area – especially in terms of applications as web based platforms (such as YouTube or Amazon) are constantly utilizing ranking in this setting. Kveton *et al.* [2015] does a good job of "combining" the theory and applied side of things by introducing the cascading bandits algorithm which has both practical applications and is interesting as a potential approach to the theoretical multi-armed bandit problem. Regardless of perspective, ranking as a whole is both a challenging (and fun) theoretical problem to consider as well as an idea that has countless real world applications in many different disciplines.

References

Ailon, Nir, Charikar, Moses, & Newman, Alantha. 2005. Aggregating inconsistent information: ranking and clustering. *Proceedings of the thirty-seventh annual acm symposium on theory of computing*, 684–693.

Badanidiyuru, Ashwinkumar, & Vondrák, Jan. 2014. Fast algorithms for maximizing submodular functions. *Pages 1497–1514 of: Proceedings of the twenty-fifth annual acm-siam symposium on discrete algorithms*. SIAM.

Derakhshan, Mahsa, Golrezaei, Negin, Manshadi, Vahideh, & Mirrokni, Vahab. 2018. Product ranking on online platforms. *Available at ssrn 3130378*.

Ferreira, Kris, Parthasarathy, Sunanda, & Sekar, Shreyas. 2019. Learning to rank an assortment of products. *Available at ssrn 3395992*.

Filkov, Vladimir, & Skiena, Steven. 2004. Integrating microarray data by consensus clustering. *International journal on artificial intelligence tools*, **13**(04), 863–880.

Kveton, Branislav, Szepesvari, Csaba, Wen, Zheng, & Ashkan, Azin. 2015. Cascading bandits: Learning to rank in the cascade model. *Pages 767–776 of: International conference on machine learning*.

Radlinski, Filip, Kleinberg, Robert, & Joachims, Thorsten. 2008. Learning diverse rankings with multi-armed bandits. *Pages 784–791 of: Proceedings of the 25th international conference on machine learning*.

Strehl, Alexander, & Ghosh, Joydeep. 2003. Relationship-based clustering and visualization for high-dimensional data mining. *Informs journal on computing*, **15**(2), 208–230.

Weitzman, Martin L. 1979. Optimal search for the best alternative. *Econometrica: Journal of the econometric society*, 641–654.