Decision Analytics with Rank Uncertainty

Jiayi Zeng; Songhao Wang; Qiao-Chu He
Southern University of Science and Technology

October 23, 2022

Abstract

In this paper, we propose a general framework to deal with decision-making problems wherein uncertainty occurs in a rank aggregation process. A novel solution approach for the integrated problem is proposed via robust optimization and an efficient algorithm is designed based on constraint generation for tractable computation. By investigating the statistical properties of our problem, we are able to derive theoretical guarantees for the likelihood of the true rank, which in turn helps to quantify the uncertainty set for the robust optimization problem. We illustrate applications of our framework using examples in advertising revenue management and supply chain design problems.
Keywords: data-driven optimization, rank aggregation, Bayesian statistics, prescriptive decision analytics.

1 Introduction

Many well-known theories and models in economics and operations rely crucially on solving decision-making problems (e.g., revenue maximization and cost minimization). Some of these problems can depend on the rank information related to the decision variables, such as the rank of different candidates in a resource allocation problem. Specifically, we consider the following decision-marking problem wherein a payoff function h is to be maximized :

$$\max_{\boldsymbol{x}} \quad h(\pi(\boldsymbol{v}), \boldsymbol{x}). \tag{1}$$

In this problem, $\boldsymbol{x} \in \mathbb{R}^n$ is a n-dimensional decision variable where the i-th component, x_i, represents the decision with respect to the i-th candidate, for instance, the resource allocated to the

i-th candidate. This problem involves n known values v_j, $j = 1, ..., n$, and a rank π. The value v_j can be treated as some problem-specific parameters, such as the utility, for the candidate at the j-th position in rank π. A rank π can be seen as a permutation, mapping i to $\pi(i)$, and thus $v_{\pi(i)}$ represents the parameters for the i-th candidate whose rank is $\pi(i)$. Here, we slightly abuse the notation and denote $(v_{\pi(1)}, \cdots, v_{\pi(n)})$ as $\pi(\boldsymbol{v})$. The rank π will interact with the decision variable $\boldsymbol{x}$ through $\boldsymbol{v}$ in the objective function h, and thus it is essential for the decision makers to have a good understanding of π before solving problem (1).

The true rank π is typically unknown in reality. Instead, it can be estimated from related information, for example, a thorough evaluation of different candidates prior to the decision-making process. Like estimating any other parameters, estimating a rank π from real-world data will bring unavoidable uncertainty. For instance, we may need to aggregate the information from multiple sources, such as multiple human experts or multiple purchase choices of a customer. Each resource can have its rank towards the candidates, which can be treated as a random realization of the true rank. The subjective human elements in the evaluation will further add to the randomness of the data. Moreover, due to a limited budget, the evaluation may be incomplete and thus the quality of the data will be negatively affected. Therefore, it is important to take the uncertainty in the estimation of π into account.

A *rank aggregation* process is considered to estimate the rank π from multiple information sources. Specifically, we assume the data available are in the form of pair-wise comparisons, i.e., each data record is a result of whether candidate i beats candidate j, where i and j are two arbitrary candidates. We note that this setting is fairly general and a rank from one source can be easily decomposed into multiple pair-wise comparisons. Moreover, it also captures the situation where only a partial (rather than a full) rank can be obtained from a source. An example in marketing research is known as the *random utility model* (RUM). In RUM, the choices of consumers are captured by pair-wise random preferences, from which a rank can be learned (McFadden 1980). One widely used RUM, the Luce model (Luce 1959), is equivalent to the Bradley–Terry model (Bradley and Terry 1952), which is a popular model for rank aggregation. In a different light, rank aggregation is a data-driven approach to recovering the candidates' priority and detecting the uncertainty in it, which also connects the literature in ranking and selection (Hong et al. 2021), expert learning (Volkovs et al. 2012), and crowdsourcing (Niu et al. 2015).

A natural approach to connect the pair-wise comparison data and the objective function in (1) is to obtain a point estimator $\hat{\pi}$ of the rank π from data and then plug it in h. The estimator $\hat{\pi}$

is treated as if it were the true rank and the optimization problem under this plug-in estimator is carried out. However, the plug-in method over-optimistically employs the single point estimator and ignores its uncertainty, which is unavoidable and can be quite significant as aforementioned. This type of uncertainty is known as the 'input uncertainty' in the stochastic simulation and optimization literature (Lam 2016). It is well recognized that the optimal decision can be seriously misled if the input uncertainty is not considered properly (Wang et al. 2020).

In this paper, we propose a general framework to deal with decision-making problems wherein uncertainty occurs in a related rank aggregation process. Here, we do not make any parametric assumption on the rank aggregation model, e.g. BTL model, which presumes a specific structure of the probabilities of comparisons. Instead, to avoid the potential risk incurred by a poor estimator, we build an uncertainty set of the rank based on the data and then employ a *robust optimization* (RO) procedure to solve the integrated decision problem. RO considers the worst scenario in the uncertainty set and thus achieves a robust decision. The uncertainty set is decided through a novel *relaxed maximum likelihood estimation* (RMLE) approach. By investigating the statistical properties of the rank aggregation problem, we are able to derive theoretical guarantees that the true rank is recovered in this set under a given probability level. The proposed framework can adapt to different levels of risk attitude by changing this probability level. A more risk-averse decision maker can choose a higher probability level and ensure that true rank is more likely to be included and further considered in the following decision-making process. To solve the integrated problem in a tractable manner, a constraint generation algorithm is proposed to reduce the time complexity of robust optimization. With a numerical case, we show that our integrated framework outperforms the standard plug-in approach (i.e., estimate-then-optimize) while its numerical performance is guaranteed by state-of-art solvers. We also provide two applications in online advertising and supply chain management which illustrate the practical implications of our decision framework.

We summarize our main contributions as follows.

1. We propose a new analytical paradigm for decision analytics with rank uncertainty, which integrates a rank aggregation process and a related decision problem.

2. We propose a novel RMLE approach to build the uncertainty set for the rank and further derive theoretical guarantees for the likelihood of the true rank.

3. A novel solution approach for the integrated problem is proposed via robust optimization and an efficient algorithm is designed based on constraint generation for tractable computation.

The rest of this paper is organized as follows. Section 2 further reviews related literature. Section 3 formally defines the integrated decision problem and provides an overview of the robust optimization approach. Section 4 introduces the RMLE approach to construct the uncertainty set and the statistical analysis to quantify the confidence level of this set. Section 5 presents the proposed constraints generation algorithm for the integrated problem. Section 6 provides two applications and Section 7 concludes this paper.

2 Related Works

2.1 Rank Aggregation

The goal of rank aggregation is to aggregate many partial ranks and get a consensus rank. This process may appear in the case of aggregating different preferences or evaluations. Jagabathula et al. (2022) learn the preferences of customers from historical purchase data by rank aggregation and then predict purchases of customers, according to which products are promoted. Benjamin Armbruster (2015) consider a decision-making problem with the utility function being unknown. They assume the decision maker's preference for lotteries is known and incorporate this pairwise comparison information into the decision-making process. Rank aggregation is also closely related to the *learning-to-rank* algorithms in the information retrieval area (Liu 2009). One example is the famous PageRank (Page et al. 1999), which is used for ranking web pages in Google's search engine results. There are also applications in online advertising (Chaudhuri et al. 2017; Wang et al. 2022) and e-commerce search (Karmaker Santu et al. 2017; Hu et al. 2018).

Most research in rank aggregation/learning-to-rank focuses on finding an optimal rank with respect to some accuracy measures. However, far too little attention has been paid to decision-making problems with rank information. We list some related works here. Christoph and Michael (2005) investigate a linear optimization problem of ranks. Lu and Boutilier (2010) study the voting problem in social choice theory when candidates may be unavailable. Agrawal et al. (2020) consider an optimal stopping problem where the player has to decide the order of observing. Different from dealing with a specific problem, in this paper, we will discuss a general type of decision problem where rank information exists with uncertainty.

The rank information we are concerned with in this article is a rank recovered from pairwise comparisons. There are a variety of implementations of this rank aggregation problem. A simple way is to consider comparisons as votes, from which one can easily get a rank, e.g. Ammar and Shah (2012). Some other studies use parametric models such as the Mallows model (Mallows 1957) or the Bradley-Terry-Luce model (Bradley and Terry 1952; Luce 1959), which enable them to exploit the structure of the comparison matrix. Levya et al. (2021) propose a rank aggregation algorithm based on low-rank matrix completion. Negahban et al. (2017) treat the comparison process as a random walk and then assign each object a Rack Centrality score. From a graph theory perspective, the rank aggregation problem is to find an acyclic subgraph in a directed graph, which is exactly the *minimum feedback arc set in tournaments* problem. Mitliagkas et al. (2011) use this method to learn users' preferences for a set of items. The FAST problem can be solved directly (Baharev et al. 2021) or by approximation algorithms such as in Kenyon-Mathieu and Schudy (2007), Alon et al. (2009). In this paper, we use integer programming to get an exact solution.

2.2 Data-Driven Optimization

Data-driven optimization provides a different perspective to examine similar decision problems wherein parameters are simultaneously estimated from a data-collection pool or process. Feng and Shanthikumar (2017) discuss some new trends and concepts in operation management in the era of big data. Predictive machine learning and statistic techniques are gradually embraced and incorporated into the decision-making paradigm. Ban and Rudin (2019) investigate a feature-based newsvendor problem where some historical feature data are available, and a machine learning algorithm based on kernel regression is presented. Bertsimas and Kallus (2020) propose an analysis framework for prescribing decisions from auxiliary data. Bertsimas and Koduri (2021) present two methods for decision-making problems based on regression in reproducing kernel Hilbert spaces, which also use historical data with auxiliary data. Chen et al. (2021) use a spline-based approximation algorithm to solve a joint pricing and inventory management problem. Behrendt et al. (2021) study the courier scheduling problem and produce online solutions with a neural network that learns from simulation results.

2.3 Robust Optimization

Robust optimization (RO) and distributionally robust optimization (DRO) are also popular methods for dealing with uncertainty in decision-making problems. Optimizing the performance under the worst case in a given ambiguity set, one can derive a robust solution by robust optimization, which achieves good out-of-sample performance and is especially effective for risk-averse problems. Hall et al. (2015) propose a measure for the risk of not achieving the target return in the project portfolio selection problem, and the risk is minimized in a DRO approach. Qi et al. (2021) discuss the quantile prediction problem under a DRO framework. He et al. (2019) investigate the fleet repositioning problem for vehicle sharing, and a multi-stage DRO model is used. Wang et al. (2020) adopt a RO framework to study the process flexibility problem and introduce a new index for evaluating the worst-case performance.

To model the uncertainty in a decision problem, an ambiguity set is needed, which can be constructed in various ways. A direct approach is to specify the moment information, which may come from the empirical distribution. For example, Zhang et al. (2018) choose the ambiguity set under the given mean and variance, and the chance constraints are calculated from the ambiguity set. Mak et al. (2014) study the appointment scheduling problem with moment information. The method we use in this paper is more related to another line of work using a data-driven approach, which constructs the ambiguity set as a region under a distance function where the empirical distribution is contained. Bertsimas et al. (2018a) discuss uncertainty sets design in data-driven robust optimization. Wang et al. (2016) and Bertsimas et al. (2018b) study DRO on a confidence region of distributions, which motivates us to construct the uncertainty set by likelihood. Some other widely used distance functions are the Wasserstein metric (Mohajerin Esfahani and Kuhn 2018; Zhi Chen 2022), Prokhorov metric (Erdoğan and Iyengar 2006), etc. We refer the readers to Lu and Shen (2021) for a recent survey. Due to the high computational complexity of robust optimization, constraint generation algorithms are widely used. This type of algorithm can be traced back to the 1960s (Benders 1962). Bienstock and Özbay (2008) use a constraint generation algorithm to solve a robust inventory problem. For two-stage robust optimization, Zeng and Zhao (2013) proposed a column and constraint generation algorithm. Chan et al. (2018) present a data-driven model for deploying AEDs whose solution is also based on column and constraint generation.

3 Problem Formulation

In this section, we formally define the integrated decision problem with rank uncertainty. A simple example is also given to illustrate the robust solution to the problem. We represent the general decision-making problem in (1) as follows:

$$\max_{\mathbf{x}\in\mathcal{G}} \quad h(\pi(\boldsymbol{v}), \boldsymbol{x}). \tag{2}$$

Recall that the rank of candidate i is $\pi(i)$ and $\pi(\boldsymbol{v})$ stands for $(v_{\pi(1)}, \cdots, v_{\pi(n)})$. More specifically, we assume that the feasible set is $\mathcal{G}$, which is a polytope $\mathcal{G} = \{G\boldsymbol{x} \leq g\}$, and $h : \mathbb{R}^{n+n} \longrightarrow \mathbb{R}$ is a function of $\boldsymbol{v}$ and $\boldsymbol{x}$. Assuming that the values, $v_1, ..., v_n$, are known is a key assumption in our model. In contrast to alternative models with valuation uncertainty, rank uncertainty reflects scenarios where a fixed utility is assigned to agents with a given rank. For instance, the best team who wins a tournament receives a fixed prize. Furthermore, the payoff can be a function of the rank. Consider the following example: for n independent identically distributed (i.i.d) uniform random variables in $[0, 1]$, the ith order statistic converges to $i/(n+1)$ as $n \to \infty$. A real world example is the power law, which is a relation in the form of exponential functions, existing in many economic phenomena (Gabaix 2016), such as city sizes, firm sizes, distribution of income, etc.

As π is unknown, we cannot determine the optimal x directly. We take a robust optimization approach and make decisions with the help of related data. The problem can be formulated as:

$$\max_{\mathbf{x} \in \mathcal{G}} \min_{\pi \in U} \quad h(\pi(\boldsymbol{v}), \boldsymbol{x}). \tag{3}$$

Here U is the uncertainty set, which is determined from the data under some probabilistic restriction (See details in Section 4). (3) finds a robust solution by considering the worst case in U. Our robust framework generalizes the "plug-in" method in the sense that the latter is a special case of the former wherein the uncertainty set (confidence region) contains a singleton of estimated rank, i.e., the uncertainty set shrinks to an atom. By leveraging on the hyperparameter of this uncertain set, we are able to fine-tune the bias-variance trade-off between a conservative solution that allocates resources more evenly, and an aggressive solution that pursues more accurate rank information. The integrated framework we propose can deal with decision problems with rank uncertainty, wherein the true rank information is recovered without any parametric assumption. Under this framework, we are able to quantify the uncertainty set (confidence region) for the decision problem as well as derive theoretical guarantees for the likelihood of the true rank.

To make Problem (3) tractable, we reformulate it into

$$\begin{aligned} \max_{\boldsymbol{x} \in \mathcal{G}, z \in \mathbb{R}} \quad & z, \\ \text{s.t.} \quad & z \leq h(\pi(\boldsymbol{v}), \boldsymbol{x}), \ \forall \pi \in U. \end{aligned} \tag{4}$$

If h is a concave function w.r.t $\boldsymbol{x}$ for any $\boldsymbol{v}$, this robust optimization problem is still a convex problem and thus the optimal solution exists. We next give an example to illustrate how does the solution of (4) look like.

Example 1. *Consider the following simple problem where h is a linear function between $\pi(\boldsymbol{v})$ and $\boldsymbol{x}$:*

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \pi(\boldsymbol{v})^T \boldsymbol{x}, \\ \text{s.t.} \quad & \|\boldsymbol{x}\| \leq 1. \end{aligned} \tag{5}$$

When the true value of π is estimated by a point estimator $\hat{\pi}$, the plug-in estimation leads to the following solution:

$$\boldsymbol{x} = \frac{\hat{\pi}(\boldsymbol{v})}{\|\hat{\pi}(\boldsymbol{v})\|}. \tag{6}$$

The solution is essentially the unit vector along the direction of $\hat{\pi}(\boldsymbol{v})$. Suppose we derive an uncertainty set for π consisting of not only $\hat{\pi}$ but also another rank $\tilde{\pi}$, i.e., the uncertainty set $U = \{\hat{\pi}, \tilde{\pi}\}$. The problem becomes

$$\begin{aligned} \max_{\boldsymbol{x}, z} \quad & z, \\ \text{s.t.} \quad & z \leq \hat{\pi}(\boldsymbol{v})^T \boldsymbol{x}, \\ & z \leq \tilde{\pi}(\boldsymbol{v})^T \boldsymbol{x}, \\ & \|\boldsymbol{x}\| \leq 1. \end{aligned} \tag{7}$$

One can verify that the solution is in the form of

$$\boldsymbol{x} = \frac{\lambda \hat{\pi}(\boldsymbol{v}) + (1-\lambda)\tilde{\pi}(\boldsymbol{v})}{\|\lambda \hat{\pi}(\boldsymbol{v}) + (1-\lambda)\tilde{\pi}(\boldsymbol{v})\|}, \ 0 \leq \lambda \leq 1. \tag{8}$$

The solution becomes an unit vector pointing in between the direction of $\hat{\pi}(\boldsymbol{v})$ and $\tilde{\pi}(\boldsymbol{v})$. Intuitively, the robust solution carefully utilizes the rank information collected from the uncertainty set U in a systematic way.

To solve (4), we need to first determine the uncertainty set U and then develop an efficient algorithm to solve the following decision problem. These two issues are discussed in Sections 4 and 5, respectively.

4 Constructing Uncertainty Sets

We next discuss how to build the uncertainty set U. In Section 4.1, we review the ranking mechanism for rank aggregation. In Section 4.2, we propose a new RMLE estimator as well as an integer programming to maximize the RMLE, based on which we obtain the uncertainty set. Finally, in Section 4.3, we derive the statistical properties of the uncertainty set and quantify its confidence level.

4.1 Ranking mechanism

Suppose we have access to an amount of pair-wise comparison data, which may come from direct comparison or breaking partial ranks (Ford 1957). We model the ranking mechanism by a weighted graph $G = (V, E, W)$, where the vertex set V consists of the items to be ranked, denoted as $\{V_1, V_2, \ldots, V_n\}$. E is a set of edges such that the edge (V_i, V_j) connecting vertices V_i and V_j belongs to set E if and only if at least one comparison between items V_i and V_j are conducted. The weight W_{ij} of edge $(V_i, V_j) \in E$ is defined as the number of comparisons made between items V_i and V_j. Then the edge set E can be represented as the set $\{(V_i, V_j) \in V^2 : W_{ij} > 0\}$.

Assume that there exists a true rank (order) $\pi^* \in S_n$ of the items $\{V_1, \ldots, V_n\}$ to be learned, where S_n denotes the set of all permutations over $\{1, \ldots, n\}$, such that V_i is of higher rank to V_j if and only if $\pi^*(i) < \pi^*(j)$. Associated with each edge $(V_i, V_j) \in E$ is an observation Y_{ij} as the number of times among the W_{ij} comparisons that item V_i is of higher rank to item V_j. The probability that the result of a comparison between item V_i and V_j is consistent with the true rank π^* is denoted by p_{ij}, and we assume that $p_{ij} > \frac{1}{2} + \eta$ for some $\eta > 0$ for all i and j. Then the distribution of Y_{ij} can be described as

$$Y_{ij} = \sum_{k=1}^{W_{ij}} Z_{ijk}, \tag{9}$$

where Z_{ijk} are i.i.d. Bernoulli results of the kth comparison between V_i and V_j:

$$\begin{aligned} \mathbb{P}(Z_{ijk} = 1) &= p_{ij}^{I[\pi^*(i) \leq \pi^*(j)]} (1 - p_{ij})^{I[\pi^*(i) > \pi^*(j)]}, \\ \mathbb{P}(Z_{ijk} = 0) &= (1 - p_{ij})^{I[\pi^*(i) \leq \pi^*(j)]} p_{ij}^{I[\pi^*(i) > \pi^*(j)]}. \end{aligned} \tag{10}$$

Here $I[\cdot]$ represents the indicator function. For any rank $\pi \in S_n$, we can build a pairwise comparison matrix $A = \{A_{ij}\}$ defined as

$$A(\pi) : S^n \mapsto \mathbb{R}^{n \times n}, A_{ij} = I[\pi(i) < \pi(j)], \tag{11}$$

which gives an one-to-one mapping from a rank to a matrix.

In reality, the problem is that the true rank π^* is typically unknown. In this work, though, we assume that there are several pairwise comparisons from which we can estimate the true rank. Based on our rank aggregation mechanism, we use the standard maximum likelihood estimation to recover the true rank. The likelihood function for $A(\pi)$ takes the form

$$\begin{aligned}\mathcal{L}(A|Y) &= \prod_{i<j} p_{ij}^{A_{ij}Y_{ij}}(1-p_{ij})^{A_{ij}(W_{ij}-Y_{ij})} p_{ij}^{(1-A_{ij})(W_{ij}-Y_{ij})}(1-p_{ij})^{(1-A_{ij})Y_{ij}} \\ &= \prod_{i<j} p_{ij}^{A_{ij}(2Y_{ij}-W_{ij})+(W_{ij}-Y_{ij})}(1-p_{ij})^{-A_{ij}(2Y_{ij}-W_{ij})+Y_{ij}}.\end{aligned} \tag{12}$$

Therefore, the corresponding conditional log-likelihood function for $A(\pi)$ is

$$l(A|Y) \propto \sum_{i<j} A_{ij}(2Y_{ij}-W_{ij}) \log\Big(\frac{p_{ij}}{1-p_{ij}}\Big). \tag{13}$$

4.2 A Relaxed Maximum Likelihood Estimator

A problem is that the probabilities of correct comparisons $P = \{p_{ij}\}$ are generally unknown. One possible way to deal with this problem is to use the standard assumption in classic rank aggregation literature, e.g., the BTL model, which gives a specific form of $\{p_{ij}\}$. However, these assumptions can be restrictive. We choose to drop the p_{ij} term in the likelihood function, as in the following. We propose to use the following relaxed maximum likelihood estimator (RMLE) $\hat{\pi}$ as the solution to the optimization problem

$$\max_{\pi \in S_n} \quad L(A) = \sum_{i<j} A_{ij}(\pi)(2Y_{ij}-W_{ij}). \tag{14}$$

Here S_n is the permutation class over $\{1,\ldots,n\}$, and $A(\pi)$ is the comparison matrix of π, $A_{ij} = I[\pi(i) < \pi(j)], \forall i < j$. When p_{ij} are all identical, i.e. $p_{ij} = p, \forall i,\ j$, (12) coincides with our relaxation (14). Our relaxation can be seen as an approximation when p_{ij} are close to each other, which implies that the expert make comparisons with similar accuracies.

From a graph-theoretic point of view, A is an adjacency matrix of a weighted directed graph: there are n nodes and there is an edge from i to j if $A_{ij} > 0$, whose weight is $2Y_{ij} - W_{ij}$. A rank π can be seen as a directed graph, and it should be acyclic since a ring will lead to inconsistency, i.e. $\exists i\ ,j\ ,k,\ s.t.\ \pi(i) > \pi(j),\ \pi(j) > \pi(k),\ \pi(k) > \pi(i)$. Therefore, recovering a rank from A is equivalent to finding an acyclic subgraph. This problem is known as the minimum feedback arc set in tournaments problem, and here we use integer programming to solve it (Baharev et al. 2021).

Proposition 1. *Problem* (14) *can be solved by the following integer programming.*

$$\max_{A} \quad \sum_{j=2}^{n} \sum_{i=1}^{j-1} (2Y_{ij} - W_{ij}) A_{ij}, \tag{15a}$$

$$\text{s.t.} \quad A_{ij} + A_{jk} - A_{ik} \leq 1, \forall 1 \leq i < j < k \leq n, \tag{15b}$$

$$- A_{ij} - A_{jk} + A_{ik} \leq 0, \forall 1 \leq i < j < k \leq n, \tag{15c}$$

$$A_{ij} \in \{0, 1\}, \forall 1 \leq i < j \leq n, \tag{15d}$$

$$A_{ij} = 0, \forall 1 \leq j \leq i \leq n. \tag{15e}$$

Intuitively, constraint (15b) and (15c) avoid rings in the corresponding graph of A and the last two constraints ensure that A is a comparison matrix. Therefore, for any matrix A in the feasible space induced by the above constraints, there exists a rank π whose pairwise comparison matrix is A. This integer programming problem can find the optimal matrix that maximize the likelihood function and thus find the optimal rank. For a rigorous proof, we refer the interested readers to Grötschel et al. (1984).

4.3 Statistical Properties

We define the uncertainty set U as:

$$U := \{\Delta(A(\hat{\pi}), A(\pi)) \leq D\}. \tag{16}$$

Recall that $A(\pi)$ is the comparison matrix of π and $\hat{\pi}$ is the RMLE. Similar to Wang et al. (2016), we define Δ to be the difference of likelihood:

$$\Delta(A(\hat{\pi}), A(\pi)) = L(A(\hat{\pi})) - L(A(\pi)). \tag{17}$$

Intuitively, Δ can be thought of as a "distance function" in terms of the likelihood function (though it is not a strict metric as it does not satisfies symmetry). The uncertainty set U is thus a ball centered at $\hat{\pi}$ with radius D, including all the ranks that are less than D away from $\hat{\pi}$. The parameter D can be used to control how conservative the decision is. With a higher value of D, increasing choices of ranks are included in U and thus the probability that the true rank is recovered is higher. Compared with other metrics for ranks, such as Kendall tau distance and Spearman's footrule, the difference in likelihood has some advantages:

- Likelihood has an explicit statistical meaning. So the uncertainty set can be interpreted as all ranks with a probability higher than a given threshold.

- Likelihood leverages the data and thus has a practical meaning. For example, exchanging any two neighbors of a rank will cause the same increase in Kendall tau distance. However, the comparison results between each pair of neighbors may vary.

As mentioned above, changing the value of D can fine-tune the confidence level of U. In this subsection, we discuss how to choose D to satisfy certain probabilistic restrictions. The proofs of all lemmas and theorems in this section are deferred to the Electronic Companion. We treat D as the α *value-at-risk* of $\Delta(A(\hat{\pi}), A(\pi^*))$. That is to say the ground truth π^* has at least a probability of $1-\alpha$ to be in U, $\mathbb{P}\big(\Delta(A(\hat{\pi}), A(\pi^*)) \geq D\big) \leq \alpha$. First we will get an upper bound of $\Delta\big(A(\hat{\pi}), A(\pi^*)\big)$ by relaxing problem (15). Removing constraints (15b) and (15c), we get a new optimization problem. Recall that these two constraints ensure that the graph is corresponding to a rank, so now A can be an arbitrary upper triangular matrix consisting of 0 and 1. The new problem is

$$\begin{aligned} \max_{A} \quad & L(A) = \sum_{i<j} A_{ij}(2Y_{ij} - W_{ij}), \\ \text{s.t.} \quad & A_{ij} = 0, \ \forall n \geq i \geq j \geq 1, \\ & A_{ij} \in \{0,1\}. \end{aligned} \tag{18}$$

The solution $\widetilde{A}$ satisfies

$$\widetilde{A}_{ij} = \begin{cases} 1, \ if \ 2Y_{ij} - W_{ij} > 0, \\ 0, \ if \ 2Y_{ij} - W_{ij} \leq 0. \end{cases} \tag{19}$$

Then we search for a permutation $\hat{\pi}$ which is the solution of

$$\min_{\pi \in S_n} \quad \Delta(\widetilde{A}, A(\pi)) = L(\widetilde{A}) - L(A(\pi)). \tag{20}$$

In fact, $\hat{\pi}$ is the RMLE. Since $\hat{\pi}$ is the minimizer of this problem, $L(A(\pi^*)) \leq L(A(\hat{\pi}))$. Now $\widetilde{A}$ is the solution of the relaxed problem (18) while $A(\hat{\pi})$ solves the original problem (15), so we have $L(A(\hat{\pi})) \leq L(\widetilde{A})$. We conclude the above discussion in the following lemma.

Lemma 1. *The difference of likelihood between the RMLE $\hat{\pi}$ and the ground truth π^* satisfies*

$$0 \leq \Delta(A(\hat{\pi}), A(\pi^*)) \leq \Delta(\widetilde{A}, A(\pi^*)). \tag{21}$$

Then we uses some simple random variables to estimate the expectation of $\Delta(\widetilde{A}, A(\pi^*))$.

Lemma 2. *Supposing that $p_{ij} \geq p$, let $\widetilde{U}_{ij} \sim Binomial(W_{ij}, p)$ and $U_{ij} = \max\{W_{ij} - 2\widetilde{U}_{ij}, 0\}$. Then*

$$\mathbb{E}[\Delta(\widetilde{A}, A(\pi^*))] \leq \mathbb{E} \sum_{i<j} U_{ij}. \tag{22}$$

We combine the two lemmas above and derive an upper bound for the expectation of the difference in likelihood.

Theorem 1. *Under the assumption that $p_{ij} \geq p$, the expectation of the difference of likelihood between RMLE and ground truth $\Delta(A(\hat{\pi}), A(\pi^*))$ has the following property,*

$$\mathbb{E}\Delta(A(\hat{\pi}), A(\pi^*)) \leq \sum_{i<j} f(W_{ij}, p). \tag{23}$$

Here

$$f(W,p) = \begin{cases} \dfrac{W+1}{(2p-1)W+3-2p}(4p(1-p)\dfrac{W^2}{W^2-1})^{\frac{W}{2}}(\dfrac{(W-1)(1-p)}{(W+1)p})^{\frac{1}{2}}, \text{ if } W \text{ is odd,} \\ \dfrac{2W(1-p)}{(2p-1)W+2-2p}(4p(1-p))^{\frac{W}{2}}, \text{ if } W \text{ is even.} \end{cases} \tag{24}$$

Especially, when $W_{ij} = m, \forall 1 \leq i < j \leq n$,

$$\mathbb{E}\Delta(A(\hat{\pi}), A(\pi^*)) \leq \frac{n(n-1)}{2} f(m,p). \tag{25}$$

Applying the Hoeffding inequality gives an upper bound of the value-at-risk.

Corollary 1. *Under the assumption that $p_{ij} \geq p$, $\Delta(A(\hat{\pi}), A(\pi^*))$ satisfies*

$$\mathbb{P}\left(\Delta\left(\hat{A}, A(\pi^*)\right) - \sum_{i<j} f(W_{ij}, p) \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i<j} W_{ij}^2}\right). \tag{26}$$

Remark $\sum_{i<j} f(m,p) = \mathcal{O}(n^2[4p(1-p)]^{\frac{m}{2}})$ and $t = \mathcal{O}(mn)$, so the magnitude of D mainly depends on the choice of t when n is small. For example, we set $\alpha = 0.05$, which means the ground truth has a probability of at least 95% to appear in the uncertainty set U. Then

$$\begin{aligned} \log(0.05) &\geq -\frac{4t^2}{n(n-1)m^2}, \\ t &\geq (\frac{\log 20n(n-1)m^2}{4})^{\frac{1}{2}} \approx 0.87mn. \end{aligned} \tag{27}$$

and D can be set as

$$D = \sum_{i<j} f(m,p) + 0.87mn. \tag{28}$$

The proposed framework provides a flexible way to consider the risk tolerance of the decision makers when the uncertainty of the rank is not negligible. In such situations, the plug-in method over-optimistically employs the single point estimator from the data and ignores its uncertainty due to potential data quality issues such as limited observability of data and measurement errors. In contrast, the proposed approach considers the risk of the estimation error, and more specifically, the framework can adapt to different levels of risk attitude by changing the probability level α and the resulting uncertainty set. For a more risk-averse decision maker, a smaller α ensures that with a high probability $1-\alpha$. the ground truth rank is recovered in D and then considered in the following decision-making process.

5 A Constraint Generation Algorithm for the Integrated Decision Problem

Given the uncertainty set U, we further develop a constraint generation algorithm to solve the decision-making problem. Recall that the robust optimization problem is

$$\begin{aligned} \max_{\boldsymbol{x}\in\mathcal{G},z\in\mathbb{R}} \quad & z, \\ \text{s.t.} \quad & z \le h(\pi(\boldsymbol{v}),\boldsymbol{x}),\ \forall \pi \in U. \end{aligned} \tag{29}$$

Since $|U|$ grows rapidly with the number of decision variables n, it is costly to solve this problem directly. Though there are many *constraint generation* (or *cutting plane*) algorithms for large-scale linear/convex programs, such as Kelley (1960), Veinott (1967), Ben-Ameur and Neto (2006), they require these constraints to be exactly known. However, just enumerating all elements of U is already expensive both in time and memory. Thus, we propose a constraint generation algorithm for (4) which is inspired by *Bender's Decomposition* (Benders 1962). The basic idea is to generate constraints on the fly instead of enumerating all the constraints. In each iteration, the algorithm solves a master problem firstly which is a relaxation of (4).

Master Problem at the k_{th} Iteration

$$\begin{aligned} \max_{\boldsymbol{x}\in\mathcal{G},z\in\mathbb{R}} \quad & z, \\ \text{s.t.} \quad & z \le h(\pi(\boldsymbol{v}),\boldsymbol{x}),\ \forall \pi \in U_k. \end{aligned} \tag{30}$$

Then the algorithm solves a subproblem known as the *separation problem* to get the most violated constraint and adds it to the master problem.

Separation Problem at the k_{th} Iteration

$$\min_{\pi \in U} \quad h(\pi(\boldsymbol{v}), \boldsymbol{x_{k-1}}). \tag{31}$$

The separation problem is solved by integer programming, which enables us to take advantage of state-of-the-art solvers. We specify that $h(\boldsymbol{v}, \boldsymbol{x})$ needs to be a linear or quadratic function of $\boldsymbol{v}$ in order to be solved by a solver. This subproblem is in fact optimization over part of a permutation group. To reformulate the separation problem into an integer programming problem, we introduce a standard matrix representation of permutations (Christoph and Michael 2005),

$$B(\pi)_{ij} = \begin{cases} 1, & \text{if } \pi(i) = j, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

However, so far we have modeled the rank π as a directed acyclic graph and used the adjacency matrix to build the uncertainty set. The following lemma connects these two matrix representations.

Lemma 3. *Let A be the comparison matrix* (11) *of a rank π_A and B be the matrix representation* (32) *of a rank π_B. Then $\pi_A = \pi_B$ if*

$$\sum_{j=1}^{i} A_{ji} + \sum_{j=i+1}^{n} (1 - A_{ij}) = \sum_{j=1}^{n} jB_{ij}, \ \forall 1 \leq i \leq n. \tag{33}$$

Proof. Since $\pi(i) < \pi(j)$ if $A(\pi)_{ij} = 1$, then the left-hand side calculates how many elements precede i in π_A, which equals to $\pi_A(i)$. The right-hand side is $\pi_B(i)$, so $\pi_A = \pi_B$. $\square$

Proposition 2. *The separation problem* (31) *can be solved by the following integer programming.*

$$
\begin{aligned}
\min_{A,B} \quad & h(B\boldsymbol{v}, \boldsymbol{x}), \\
\text{s.t.} \quad & A_{ij} + A_{jk} - A_{ik} \leq 1, \forall 1 \leq i < j < k \leq n, \\
& -A_{ij} - A_{jk} + A_{ik} \leq 0, \forall 1 \leq i < j < k \leq n, \\
& A_{ij} \in \{0,1\}, \forall 1 \leq i < j \leq n, \\
& A_{ij} = 0, \forall 1 \leq j \leq i \leq n. \\
& \sum_{i=1}^{n} B_{ij} = 1, \ \forall 1 \leq j \leq n, \\
& \sum_{j=1}^{n} B_{ij} = 1, \ \forall 1 \leq i \leq n, \\
& B_{ij} \in \{0,1\}, \forall 1 \leq i, j \leq n, \\
& \sum_{j=2}^{n} \sum_{i=1}^{j-1} (2Y_{ij} - W_{ij}) \left(A(\hat{\pi})_{ij} - A_{ij} \right) \leq D, \\
& \sum_{j=1}^{i} A_{ji} + \sum_{j=i+1}^{n} (1 - A_{ij}) = \sum_{j=1}^{n} jB_{ij}, \ \forall 1 \leq i \leq n.
\end{aligned}
\tag{34}
$$

Proof. The first four constraints ensure that $\{A_{ij}\}$ corresponds to a rank. Also, the next three constraints ensure that $\{B_{ij}\}$ corresponds to a rank, as they require each column and row has only one non-zero element. The second last constraint is just $\Delta(A(\hat{\pi}), A(\pi)) \leq D$. From Lemma 3, the last constraint ensures that A and B represent the same rank. Noting that $\sum_{j=1}^{n} B_{ij} v_j = v_{\pi(i)}$, we have

$$B(\pi)\boldsymbol{v} = (v_{\pi(1)}, v_{\pi(2)}, ..., v_{\pi(n)})^T = \pi(\boldsymbol{v}). \tag{35}$$

and thus the objective function equals to $h(\pi(\boldsymbol{v}), \boldsymbol{x})$. □

The convergence of Algorithm 1 is shown in the following proposition.

Proposition 3. *If (4) has a feasible optimal solution $(\boldsymbol{x}^*, z^*)$, Algorithm 1 will reach z^* in finite steps.*

Proof. Since the algorithm adds an element of U to U_k in each iteration, it will terminate in at most $|U|$ steps. Note that if a repeated π is added to U_k, the algorithm will stop. Let $(\boldsymbol{x}_t, z_t)$ be the output of Algorithm 1. Since the master problem is a relaxation of (4), we have $z^* \leq z_t$. Suppose that $z^* < z_t$, then there must be a $\sigma \in U \backslash U_k$ s.t. $h(\sigma(\boldsymbol{v}), \boldsymbol{x_t}) < z_t$. In the last iteration, the solution

Algorithm 1 Constraint Generation Algorithm for (4)

$U_0 \leftarrow \{\hat{\pi}\}$ $\triangleright$ $\hat{\pi}$ is the RMLE rank

$(\boldsymbol{x}_0, z_0) \leftarrow$ a solution of the Master Problem (30)

$k \leftarrow 0$

while $\boldsymbol{x}_{k-1} \neq \boldsymbol{x}_k$ **do**

  $k \leftarrow k+1$

  $\pi_{k+1} \leftarrow$ a solution of the Separation Problem (31)

  $U_{k+1} \leftarrow U_k \cup \{\pi_{k+1}\}$

  $(\boldsymbol{x}_{k+1}, z_{k+1}) \leftarrow$ a solution of the Master Problem (30)

end while

of the separation problem π_t satisfies $h(\pi_t(\boldsymbol{v}), \boldsymbol{x_{t-1}}) \leq h(\sigma(\boldsymbol{v}), \boldsymbol{x_{t-1}})$. From the stopping criterion we know $\boldsymbol{x_{t-1}} = \boldsymbol{x_t}$, so $z_t \leq h(\pi_t(\boldsymbol{v}), \boldsymbol{x_t}) \leq h(\sigma(\boldsymbol{v}), \boldsymbol{x_t})$, which leads to a contradiction. $\square$

Other Stopping Criteria Since a new constraint is added in each iteration, the time cost of each iteration increases as the number of iterations increases. However, the solution we get in the early rounds is usually good enough. To save computational resources, we may terminate the algorithm before it converges. For instance, a stopping criterion can be $|z_t - z_{t-1}| < \epsilon$.

6 Applications

In this section, we provide a real-world application of our framework. Additional applications and examples are illustrated in the Electronic Companion.

Revenue management for advertising In online display advertising, the profit of the publisher not only depends on the bid but also depends on the number of ad clicks. Consider a revenue management problem similar to Shen et al. (2020), where advertisers are bidding on different slots and different groups of viewers. A slot only displays an ad to its targeted viewers, and we call a slot with a corresponding viewer group an audience unit. The publisher decides how to assign ads to audience units.

Let I and J be the set of audience units and ads respectively, and b_{ij} be the bid price of ad j targeting audience unit i. Noting that an audience unit needs not to be assigned to a single ad, we define x_{ij} to be the proportion of audience unit i's impressions assigned to ad j. $h_{ij}(x_{ij})$ is a prediction function of the number of clicks from audience unit i for ad j. Because viewers are less likely to click the ad when seeing repeated ads, h should be a non-decreasing and concave function, which is studied by Chatterjee et al. (2003) and Braun and Moe (2013). For simplicity, we set $h_{ij}(x_{ij}) = \alpha_{ij}\log(10x_{ij}+1)$, where $\alpha_{ij}$is a coefficient that describes how attracting ad j is for audience unit i. Values of α_{ij} are evaluated by sub-models in the form of ranks (Wang et al. 2022), and then the output of these sub-models are aggregated. The problem can be formulated as follows.

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \sum_{i\in I}\sum_{j\in J} b_{ij}\alpha_{\pi^*(i,j)}\log(10x_{ij}+1), \\ \text{s.t.} \quad & \sum_{j\in J} x_{ij} \le 1, \quad \forall i \in I, \\ & x_{ij} \ge 0, \forall i \in I, j \in J. \end{aligned} \tag{36}$$

Maximizing the total profit is equivalent to maximizing the profit of each audience unit i, and thus we only consider one audience unit for convenience.

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \sum_{j\in J} b_j\alpha_{\pi^*j}\log(10x_j+1), \\ \text{s.t.} \quad & \sum_{j\in J} x_j \le 1, \\ & x_j \ge 0, \forall j \in J. \end{aligned} \tag{37}$$

We reformulate the robust version of problem (37) as a conic exponential programming problem and solve it with MOSEK 9.3.8. Let $y_j = \log(10x_j+1)$, then

$$\begin{aligned} \max_{\boldsymbol{y},z} \quad & z, \\ \text{s.t.} \quad & z \le \sum_{j\in J} b_j\alpha_{\pi j}y_j, \forall \pi \in U, \\ & \sum_{j\in J} q_j \le n+10, \\ & e^{y_j} \le q_j, \forall j \in J, \\ & y_j \ge 0, \forall j \in J. \end{aligned} \tag{38}$$

The parameters are set as follows. There are 10 ads to be displayed in one audience unit. b_j are drawn from $Uniform(1,2)$ and $a_j = 0.05 * (j+9)$. We repeat the experiment for 100 times and the stopping criterion is $|z_t - z_{t-1}| < \epsilon$ with $\epsilon = 0.001$. The experiments are conducted on a laptop with i7-6560U and the solvers are CPLEX 12.10.0.0 and MOSEK 9.3.10.

n	m	p	D	meanCG	meanMLE	stdCG	stdMLE	time	nIter
10	3	0.6	4	7.106	7.093	0.084	0.110	0.20	3.67
10	3	0.7	2	7.204	7.211	0.061	0.069	0.12	2.75
10	3	0.8	1	7.266	7.272	0.014	0.013	0.07	2.09
10	5	0.6	5	7.129	7.111	0.076*	0.125	0.24	3.66
10	5	0.7	2	7.240	7.247	0.036	0.036	0.09	2.45
10	5	0.8	1	7.273	7.275	0.010	0.009	0.06	1.72

$^{*}p < 0.01.$

Table 1: Revenue management for advertising. Mean values and variances of these two methods are compared by Welch's t-test and Levene's test respectively.

7 Conclusion

In this paper, we propose a general framework to deal with decision problems wherein uncertainty occurs in a rank aggregation process. We employ a robust optimization procedure to solve the integrated decision problems with rank uncertainty, which is data-driven by nature and utilizes comparison information. Our approach involves searching a relaxed maximal likelihood estimator (RMLE) of the true rank and constructing the uncertainty set according to the likelihood function. The proposed framework provides a flexible way to consider the risk tolerance of the decision makers when the uncertainty of the rank is not negligible. In such situations, the plug-in method over-optimistically employs the single point estimator from the data and ignores its uncertainty due to potential data quality issues such as limited observability of data and measurement errors. In contrast, the proposed approach considers the risk of the estimation error, and more specifically, the framework can adapt to different levels of risk attitude by changing the probability level α and the resulting uncertainty set. We model the optimization problem as an integer programming problem and design an efficient algorithm based on constraint generation for tractable computation. By investigating the statistical properties of our problem, we derive the upper bound of the value-at-risk of the likelihood difference between the true rank and the RMLE, which in turn helps to quantify the uncertainty set for the robust optimization problem. We apply our decision framework to a revenue management problem to show the practical usage of this approach, and numerical results are provided, which suggests our decision framework significantly reduces variance as well as achieves better mean performance.

References

Agrawal, S., J. Sethuraman, and X. Zhang (2020). On optimal ordering in the optimal stopping problem. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, New York, NY, USA, pp. 187–188. Association for Computing Machinery.

Alon, N., D. Lokshtanov, and S. Saurabh (2009). Fast fast. In *ICALP*.

Ammar, A. and D. Shah (2012). Efficient rank aggregation using partial data. *SIGMETRICS Perform. Eval. Rev. 40*(1), 355–366.

Arratia, R. and L. Gordon (1989). Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology 51*(1), 125–131.

Baharev, A., H. Schichl, A. Neumaier, and T. Achterberg (2021, April). An exact method for the minimum feedback arc set problem. *ACM J. Exp. Algorithmics 26*, 1–28.

Ban, G.-Y. and C. Rudin (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research 67*(1), 90–108.

Behrendt, A., M. Savelsbergh, and H. Wang (2021). A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Optimization Online*.

Ben-Ameur, W. and J. Neto (2006). A constraint generation algorithm for large scale linear programs using multiple-points separation. *Mathematical Programming 107*(3), 517–537.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik 4*(1), 238–252.

Benjamin Armbruster, E. D. (2015). Decision making under uncertainty when preference information is incomplete. *Management Science 61*(1), 111–128.

Bertsimas, D., V. Gupta, and N. Kallus (2018a). Data-driven robust optimization. *Mathematical Programming 167*(2), 235–292.

Bertsimas, D., V. Gupta, and N. Kallus (2018b). Robust sample average approximation. *Mathematical Programming 171*(1), 217–282.

Bertsimas, D. and N. Kallus (2020). From predictive to prescriptive analytics. *Management Science 66*(3), 1025–1044.

Bertsimas, D. and N. Koduri (2021). Data-driven optimization: A reproducing kernel hilbert space approach. *Operations Research 70*(1), 454–471.

Bienstock, D. and N. Özbay (2008). Computing robust basestock levels. *Discrete Optimization 5*(2), 389–414.

Bradley, R. A. and M. E. Terry (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika 39*, 324.

Braun, M. and W. Moe (2013, 09). Online display advertising: Modeling the effects of multiple creatives and individual impression histories. *Marketing Science 32*, 753–767.

Chan, T. C. Y., Z.-J. M. Shen, and A. Siddiq (2018). Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation. *Operations Research 66*(2), 358–379.

Chatterjee, P., D. L. Hoffman, and T. P. Novak (2003). Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science 22*, 520–541.

Chaudhuri, S., A. Bagherjeiran, and J. Liu (2017). Ranking and calibrating click-attributed purchases in performance display advertising. In *Proceedings of the ADKDD'17*. Association for Computing Machinery.

Chen, B., X. Chao, and C. Shi (2021). Nonparametric learning algorithms for joint pricing and inventory control with lost sales and censored demand. *Mathematics of Operations Research 46*.

Christoph, B. and J. Michael (2005). Linear optimization over permutation groups. *Discrete Optimization 2*(4), 308–319.

Erdoğan, E. and G. Iyengar (2006). Ambiguous chance constrained problems and robust optimization. *Mathematical Programming 107*, 37–61.

Feng, Q. and J. G. Shanthikumar (2017). How research in production and operations management may evolve in the era of big data. *Production and Operations Management 27*.

Ford, L. R. (1957). Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly 64*(8P2), 28–33.

Gabaix, X. (2016). Power laws in economics: An introduction. *Journal of Economic Perspectives 30*(1), 185–206.

Gotoh, J.-Y., M. J. Kim, and A. E. Lim (2018, jul). Robust empirical optimization is almost the same as mean–variance optimization. *Oper. Res. Lett. 46*(4), 448–452.

Grötschel, M., M. Jünger, and G. Reinelt (1984, 12). A cutting plane algorithm for the linear ordering problem. *Operations Research 32*, 1195–1220.

Hall, N. G., D. Z. Long, J. Qi, and M. Sim (2015). Managing underperformance risk in project portfolio selection. *Operations Research 63*(3).

He, L., Z. Hu, and M. Zhang (2019). Robust repositioning for vehicle sharing. *Manufacturing & Service Operations Management 22*(2), 241–256.

Hong, L. J., W. Fan, and J. Luo (2021). Review on ranking and selection: A new perspective. *Frontiers of Engineering Management 8*, 321–343.

Hu, Y., Q. Da, A. Zeng, Y. Yu, and Y. Xu (2018). Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, New York, NY, USA, pp. 368–377. Association for Computing Machinery.

Jagabathula, S., D. Mitrofanov, and G. Vulcano (2022). Personalized retail promotions through a directed acyclic graph–based representation of customer preferences. *Operations Research 70*(2), 641–665.

Karmaker Santu, S. K., P. Sondhi, and C. Zhai (2017). On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, New York, NY, USA, pp. 475–484. Association for Computing Machinery.

Kelley, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics 8*(4), 703–712.

Kenyon-Mathieu, C. and W. Schudy (2007). How to rank with few errors. *Proceedings of the Annual ACM Symposium on Theory of Computing*, 95–103.

Lam, H. (2016). Advanced tutorial: Input uncertainty and robust analysis in stochastic simulation. In *2016 Winter Simulation Conference (WSC)*, pp. 178–192. IEEE.

Levya, T., A. Vahidb, and R. Giryes (2021). Ranking recovery from limited pairwise comparisons using low-rank matrix completion. *Applied and Computational Harmonic Analysis 54*, 227–249.

Liu, T.-Y. (2009, mar). Learning to rank for information retrieval. *Found. Trends Inf. Retr. 3*(3), 225–331.

Lu, M. and Z. M. Shen (2021). A review of robust operations management under model uncertainty. *Production and Operations Management 30*(6), 1927–1943.

Lu, T. and C. E. Boutilier (2010). The unavailable candidate model: A decision-theoretic view of social choice. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, New York, NY, USA, pp. 263–274. Association for Computing Machinery.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. New York: Wiley.

Mak, H.-Y., Y. Rong, and J. Zhang (2014). Appointment scheduling with limited distributional information. *Management Science 61*(2), 316–334.

Mallows, C. L. (1957, 06). Non-null ranking models. i. *Biometrika 44*(1-2), 114–130.

McFadden, D. (1980). Econometric models for probabilistic choice among products. *The Journal of Business 53*(3), s13–s29.

Mitliagkas, I., A. Gopalan, C. Caramanis, and S. Vishwanath (2011). User rankings from comparisons: Learning permutations in high dimensions. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1143–1150.

Mohajerin Esfahani, P. and D. Kuhn (2018). Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming 171*(1), 115–166.

Negahban, S., S. Oh, and D. Shah (2017). Rank centrality: Ranking from pair-wise comparisons. *Operations Research 65*(1), 266–287.

Niu, S., Y. Lan, J. Guo, X. Cheng, L. Yu, and G. Long (2015). Listwise approach for rank aggregation in crowdsourcing. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, New York, NY, USA, pp. 253–262. Association for Computing Machinery.

Page, L., S. Brin, R. Motwani, and T. Winograd (1999, November). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Pelekis, C. (2017). Lower bounds on binomial and poisson tails: an approach via tail conditional expectations.

Qi, M., Y. Cao, and M. Shen (2021). Distributionally robust conditional quantile prediction with fixed design. *Management Science*.

Shen, H., Y. Li, J. Guan, and G. Tso (2020, 10). A planning approach to revenue management for non-guaranteed targeted display advertising. *Production and Operations Management 30*.

Veinott, A. F. (1967). The supporting hyperplane method for unimodal programming. *Operations Research 15*(1), 147–152.

Volkovs, M. N., H. Larochelle, and R. S. Zemel (2012). Learning to rank by aggregating expert preferences. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, New York, NY, USA, pp. 843–851. Association for Computing Machinery.

Wang, H., J. Yuan, and S. H. Ng (2020). Gaussian process based optimization algorithms with input uncertainty. *IISE Transactions 52*(4), 377–393.

Wang, S., X. Wang, and J. Zhang (2020). Robust optimization approach to process flexibility designs with contribution margin differentials. *Manufacturing & Service Operations Management 24*(1).

Wang, X., G. Huzhang, Q. Lin, and Q. Da (2022). Learning-to-ensemble by contextual rank aggregation in e-commerce. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, New York, NY, USA, pp. 1036–1044. Association for Computing Machinery.

Wang, Z., P. W. Glynn, and Y. Ye (2016). Likelihood robust optimization for data-driven problems. *Computational Management Science 13*(2), 241–261.

Zeng, B. and L. Zhao (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters 41*(5), 457–461.

Zhang, Y., R. Jiang, and S. Shen (2018). Ambiguous chance-constrained binary programs under mean-covariance information. *SIAM Journal on Optimization 28*(4), 2922–2944.

Zhi Chen, Daneil Kuhn, W. W. (2022). Data-driven chance constrained programs over wasserstein balls. *arXiv*.

A Electronic Companion

A.1 Proofs

Lemma 2

Proof. According to the definitions of $\Delta(\widetilde{A}, A(\pi^*))$ we have

$$\begin{aligned}\Delta(\widetilde{A}, A(\pi^*)) &= \sum_{i<j} (\widetilde{A}_{ij} - 1)(2Y_{ij} - W_{ij}) \\ &= \sum_{i<j} (1 - \widetilde{A}_{ij})(W_{ij} - 2\sum_{k=1}^{W_{ij}} Z_{ijk}) \\ &= \sum_{i<j} \max\{W_{ij} - 2\sum_{k=1}^{W_{ij}} Z_{ijk}, 0\}.\end{aligned}$$

The last step comes from the definition of $\widetilde{A}_{ij}$ (19). Now we have

$$\mathbb{E}[\Delta(\widetilde{A}, A(\pi^*))] = \mathbb{E}\sum_{i<j} \max\{W_{ij} - 2\sum_{k=1}^{W_{ij}} Z_{ijk}, 0\}.$$

Let $r_{ij} = \lfloor W_{ij}/2 \rfloor$. With the assumption $p_{ij} \geq p, \forall i, j$, we get

$$\begin{aligned}\mathbb{E}\sum_{i<j} \max\{W_{ij} - 2\sum_{k=1}^{W_{ij}} Z_{ijk}, 0\} &= \sum_{i<j}\sum_{k=0}^{r_{ij}} \binom{W_{ij}}{k} p_{ij}^k (1 - p_{ij})^{W_{ij}-k}(W_{ij} - 2k) \\ &\leq \sum_{i<j}\sum_{k=0}^{r_{ij}} \binom{W_{ij}}{k} p^k (1 - p)^{W_{ij}-k}(W_{ij} - 2k).\end{aligned}$$

The last step comes from the fact that $p^k(1-p)^{W-k}$ decreases with p when $1/2 < p < 1, k \leq W_{ij}/2$. Let $\widetilde{U}_{ij} \sim \mathit{Binomial}(W_{ij}, p)$ and $U_{ij} = \max\{W_{ij} - 2\widetilde{U}_{ij}, 0\}$. Then

$$\mathbb{E}[\Delta(\widetilde{A}, A(\pi^*))] \leq \mathbb{E}\sum_{i<j} U_{ij}. \tag{39}$$

□

Theorem 1

Proof. We proceed from equation (39) with the same notations. Recall that $r_{ij} = \lfloor W_{ij}/2 \rfloor$ and $\widetilde{U}_{ij} \sim \mathit{Binomial}(W_{ij}, p)$. First, to get an upper bound of $\mathbb{E}U_{ij}$, we decompose it into two parts,

$$\mathbb{E}U_{ij} = \mathbb{P}(\widetilde{U}_{ij} \leq r_{ij})\mathbb{E}\big[W_{ij} - 2\widetilde{U}_{ij} | \widetilde{U}_{ij} \leq r_{ij}\big].$$

The probability part can be bounded by the Chernoff bound of binomial distribution (Theorem 1 in Arratia and Gordon (1989)),

$$\begin{aligned}\mathbb{P}(\widetilde{U}_{ij} \leq r_{ij}) &= \mathbb{P}(W_{ij} - \widetilde{U}_{ij} \geq W_{ij} - r_{ij}) \\ &\leq \exp\Big(- W_{ij} D\big(1 - \frac{r_{ij}}{W_{ij}}||1-p\big)\Big) \\ &= \exp\Big(- W_{ij} D\big(\frac{r_{ij}}{W_{ij}}||p\big)\Big),\end{aligned}$$

where $D(a||p) = a\log(\frac{a}{p}) + (1-a)\log(\frac{1-a}{1-p})$ is the relative entropy between an a-coin and a p-coin. Hence

$$\mathbb{P}(\widetilde{U}_{ij} \leq r_{ij}) \leq \begin{cases} (4p(1-p)\dfrac{W_{ij}^2}{W_{ij}^2 - 1})^{\frac{W_{ij}}{2}}(\dfrac{(W_{ij}-1)(1-p)}{(W_{ij}+1)p})^{\frac{1}{2}}, \text{ if } W_{ij} \text{ is odd,} \\ (4p(1-p))^{\frac{W_{ij}}{2}}, \text{ if } W_{ij} \text{ is even.} \end{cases} \tag{40}$$

Next, we introduce a lemma (Theorem 1.1 in Pelekis (2017)) to bound the tail conditional expectation part.

Lemma 4. *Let $X \sim B(n,p)$ and k be an integer such that $np < k \leq n$. Then*

$$\mathbb{E}[X|X \geq k] \leq k + \frac{(n-k)p}{k - np + p}.$$

Noting that $W_{ij} - \widetilde{U}_{ij}$ is again a binomial variable, from Lemma 4 we have

$$\mathbb{E}\big[W_{ij} - \widetilde{U}_{ij}|\widetilde{U}_{ij} \leq r_{ij}\big] \leq \begin{cases} \dfrac{W_{ij}+1}{2} + \dfrac{(W_{ij}-1)(1-p_{ij})}{(2p_{ij}-1)W_{ij} + 3 - 2p_{ij}}, \text{ if } W_{ij} \text{ is odd,} \\ \dfrac{W_{ij}}{2} + \dfrac{W_{ij}(1-p_{ij})}{(2p_{ij}-1)W_{ij} + 2 - 2p_{ij}}, \text{ if } W_{ij} \text{ is even.} \end{cases}$$

Since

$$\mathbb{E}\big[W_{ij} - 2\widetilde{U}_{ij}|\widetilde{U}_{ij} \leq r_{ij}\big] = 2\mathbb{E}\big[W_{ij} - \widetilde{U}_{ij}|\widetilde{U}_{ij} \leq r_{ij}\big] - W_{ij},$$

we have

$$\mathbb{E}\big[W_{ij} - 2\widetilde{U}_{ij}|\widetilde{U}_{ij} \leq r_{ij}\big] \leq \begin{cases} \dfrac{W_{ij}+1}{(2p_{ij}-1)W_{ij} + 3 - 2p_{ij}}, \text{ if } W_{ij} \text{ is odd,} \\ \dfrac{2W_{ij}(1-p_{ij})}{(2p_{ij}-1)W_{ij} + 2 - 2p_{ij}}, \text{ if } W_{ij} \text{ is even.} \end{cases} \tag{41}$$

Combining (40) and (41), we get

$$\mathbb{E}\Delta(\widetilde{A}, A(\pi^*)) = \sum_{i<j} \mathbb{E}U_{ij} \leq \sum_{i<j} f(W_{ij}, p_{ij}),$$

$$
f(W,p)=\begin{cases}\frac{W+1}{(2p-1)W+3-2p}(4p(1-p)\frac{W^2}{W^2-1})^{\frac{W}{2}}(\frac{(W-1)(1-p)}{(W+1)p})^{\frac{1}{2}}, \text{ if } W \text{ is odd,}\\ \frac{2W(1-p)}{(2p-1)W+2-2p}(4p(1-p))^{\frac{W}{2}}, \text{ if } W \text{ is even.}\end{cases}
$$

□

Corollary 1

Proof. Recall that $\mathbb{E}\Delta(\widetilde{A}, A(\pi^*)) = \sum_{i<j}\mathbb{E}U_{ij}$, and $U_{ij} \in [0, W_{ij}]$ are bounded variables. Applying the Hoeffding inequality we have

$$
\mathbb{P}\left(\Delta\left(\widetilde{A}, A(\pi^*)\right) - \mathbb{E}\Delta\left(\widetilde{A}, A(\pi^*)\right) \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i<j} W_{ij}^2}\right).
$$

Thus

$$
\begin{aligned}
\mathbb{P}\left(\Delta\left(\hat{A}, A(\pi^*)\right) - \sum_{i<j} f(W_{ij}, p) \geq t\right) &= \mathbb{P}\left(\Delta\left(\hat{A}, A(\pi^*)\right) \geq t + \sum_{i<j} f(W_{ij}, p)\right) \\
&\leq \mathbb{P}\left(\Delta\left(\widetilde{A}, A(\pi^*)\right) \geq t + \sum_{i<j} f(W_{ij}, p)\right) \\
&\leq \mathbb{P}\left(\Delta\left(\widetilde{A}, A(\pi^*)\right) \geq t + \mathbb{E}\Delta\left(\widetilde{A}, A(\pi^*)\right)\right) \\
&\leq \exp\left(-\frac{2t^2}{\sum_{i<j} W_{ij}^2}\right).
\end{aligned}
$$

□

A.2 Numerical Results

In this section, we give one simple case study to examine the influence of each parameter. An extra real application is also provided. All the experiments are conducted on a laptop with i7-6560U. The solver is CPLEX 12.10.0.0. Consider a general decision problem. There are n objectives, and we have pairwise comparison information of these objectives. We know the value of a ranking i objective is $v_i = i/n$. $\boldsymbol{x} = (x_i)$ is the decision variable with a constraint $\sum x_i \leq 1$. Here we set $h_i(x) = \sqrt{x_i}$.

This problem can be formulated as

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \sum_{i=1}^{n} v_{\pi^* i}\sqrt{x_i}, \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \leq 1, \\ & x_i \geq 0, \quad 1 \leq i \leq n. \end{aligned} \tag{42}$$

It is equivalent to the following problem and can be solved by a solver.

$$\begin{aligned} \max_{\boldsymbol{y}} \quad & \sum_{i=1}^{n} v_{\pi^* i} y_i, \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i^2 \leq 1, \\ & y_i \geq 0, \quad 1 \leq i \leq n. \end{aligned} \tag{43}$$

For these experiments, we use a simple setting $W_{ij} = m$, $P_{ij} = p$, i.e. each couple of objectives are compared for m times and the success probability of comparison is p. The stopping criterion is $|z_t - z_{t-1}| < \epsilon$ with $\epsilon = 0.001$. Each experiment is repeated 100 times, and Table 2 shows our results. Suffix 'CG' stands for our constraint generation algorithm, and 'MLE' stands for a plug-in method that treats RMLE as π^*. 'time' is the mean time cost of our constraint generation algorithm, and 'nIter' is the average number of iterations.

Size of the Uncertainty Set Next, we discuss how different Ds affect the performance of our algorithm. The left figure shows a bell-shaped curve, where the optimal mean value is attained at $D = 6$. In the right figure, the variance decreases with D and is always less than the variance of the plug-in method.

n	m	p	D	meanCG	meanMLE	stdCG	stdMLE	time	nIter
10	3	0.6	3	1.813*	1.773	0.067*	0.107	0.45	5.15
10	3	0.6	6	1.813*	1.773	0.049*	0.107	0.68	6.38
10	3	0.7	3	1.903	1.889	0.038*	0.070	0.45	4.78
10	3	0.7	6	1.889	1.889	0.033*	0.070	0.68	5.76
10	3	0.8	3	1.945	1.954	0.011	0.012	0.27	4.01
10	3	0.8	6	1.938	1.954	0.014	0.012	0.61	4.88
10	5	0.6	3	1.819	1.790	0.076*	0.124	0.42	4.94
10	5	0.6	6	1.827*	1.790	0.056*	0.124	0.68	5.83
10	5	0.7	3	1.930	1.929	0.023	0.036	0.24	4.05
10	5	0.7	6	1.920	1.929	0.024	0.036	0.51	4.87
10	5	0.8	3	1.955	1.957	0.006	0.008	0.11	3.45
10	5	0.8	6	1.952	1.957	0.006	0.008	0.21	4.06
15	5	0.6	3	2.222*	2.191	0.061*	0.085	3.17	4.76
15	5	0.6	6	2.227*	2.191	0.048*	0.085	9.65	6.25
15	5	0.6	9	2.222*	2.191	0.039*	0.085	13.77	6.75
15	5	0.7	6	2.319	2.319	0.016*	0.026	3.98	4.64
15	5	0.8	6	2.341	2.345	0.004	0.004	1.41	3.75

$^{*}p < 0.01$.

Table 2: Mean values and variances of these two methods are compared by Welch's t-test and Levene's test respectively.

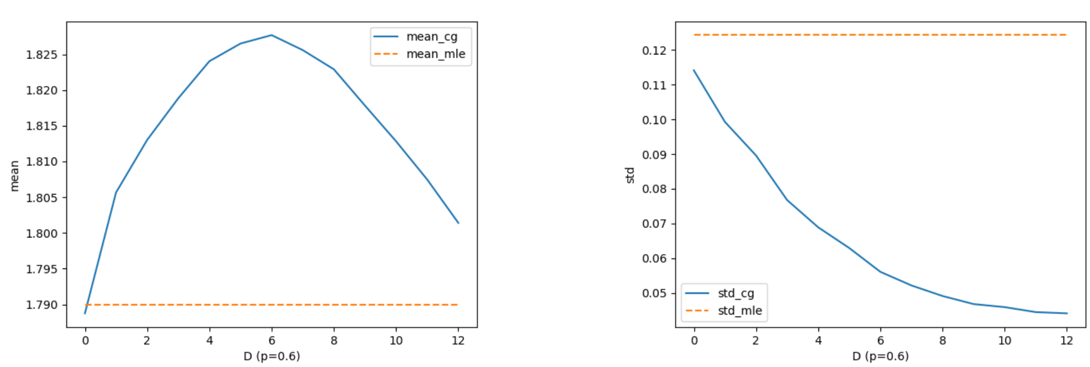

Figure 1: Changing D with fixed p

Intuitively, a large uncertainty set leads to a conservative solution, and thus a low variance. If D is large enough that all $n!$ rankings are put into the uncertainty set, we will get a trivial solution. In contrast, the plug-in method tries to predict the true ranking and is easily affected by the noise. Roughly speaking, the bell-shaped curve implies a bias-variance trade-off. This phenomenon has been investigated for a general type of distributionally robust optimization problem (Gotoh et al. 2018).

Stopping Criteria We compare three stopping criteria in Table 3. The tolerance ϵ is set to be 0.01 and 0.001. A hyphen stands for complete convergence,

$$\boldsymbol{x_{t-1}} = \boldsymbol{x_t}.$$

n	m	p	D	ϵ	meanCG	meanMLE	stdCG	stdMLE	time	nIter
10	3	0.6	3	-	1.813*	1.773	0.067*	0.107	1.10	8.83
10	3	0.6	3	0.001	1.813*	1.773	0.067*	0.107	0.45	5.15
10	3	0.6	3	0.01	1.812*	1.773	0.067*	0.107	0.26	3.65
10	5	0.6	6	-	1.828*	1.790	0.056*	0.124	2.20	10.28
10	5	0.6	6	0.001	1.827*	1.790	0.056*	0.124	0.68	5.83
10	5	0.6	6	0.01	1.824	1.790	0.059*	0.124	0.30	3.72
15	5	0.6	6	-	2.227*	2.191	0.049*	0.085	34.12	11.67
15	5	0.6	6	0.001	2.227*	2.191	0.048*	0.085	9.65	6.25
15	5	0.6	6	0.01	2.224*	2.191	0.051*	0.085	2.71	3.71

$^{*}p < 0.01$.

Table 3: Results with different stopping criteria. Mean values and variances of these two methods are compared by Welch's t-test and Levene's test respectively.

Reliable facility location design Consider a facility location problem, where each site has a probability of unexpected failure. Denote the set of customers by I and the set of candidate locations by J. A facility located at j has an annualized fixed location cost f_j and a failure probability q_j. Customer i's annual demand is λ_i and the unit cost of shipment from location j to customer i is d_{ij}. If the alternate for customer i also fails, there will be a penalty l_i. x_j, y_{ij}, z_{ik} are binary decision variables. $x_j = 1$ means there is a facility in location j, $y_{ij} = 1$ means facility j is assigned to customer i, and z_{ik} means facility k is an alternate for customer i. Probabilities of failure q_j are evaluated by experts in the form of rankings, which are modeled by the rank aggregation process. The problem can be formulated as follows.

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad & \sum_{j\in J} f_j x_j + \sum_{i\in I, j\in J} \lambda_i d_{ij}(1-q_{\pi^* j}) y_{ij} + \sum_{i\in I, j,k\in J} \lambda_i q_{\pi^* j} y_{ij} (1-q_{\pi^* k}) d_{ik} z_{ik} + \sum_{i\in I, j,k\in J} \lambda_i q_{\pi^* j} y_{ij} q_{\pi^* k} z_{ik} l_i, \\
\text{s.t.} \quad & \sum_{j\in J} y_{ij} = 1, \\
& \sum_{k\in J} z_{ik} = 1, \\
& y_{ij} \leq x_j, \\
& z_{ik} \leq x_k, \\
& y_{ij} + z_{ij} \leq 1, \\
& x_i, y_{ij}, z_{ij} \in \{0,1\}.
\end{aligned} \tag{44}$$

The first term of the objective function is the location cost. The second/third term is the cost of shipment when the assigned facility works/fails, and the last one is the penalty of not serving the customer. The robust version is

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},t} \quad & t, \\
\text{s.t.} \quad & t \geq \sum_{j\in J} f_j x_j + \sum_{i\in I, j\in J} \lambda_i d_{ij}(1-q_{\pi j}) y_{ij} + \sum_{i\in I, j,k\in J} \lambda_i q_{\pi j} y_{ij} (1-q_{\pi k}) d_{ik} z_{ik} \\
& \quad + \sum_{i\in I, j,k\in J} \lambda_i q_{\pi j} y_{ij} q_{\pi k} z_{ik} l_i, \forall \pi \in U \\
& \sum_{j\in J} y_{ij} = 1, \\
& \sum_{k\in J} z_{ik} = 1, \\
& y_{ij} \leq x_j, \\
& z_{ik} \leq x_k, \\
& y_{ij} + z_{ij} \leq 1, \\
& x_i, y_{ij}, z_{ij} \in \{0,1\}.
\end{aligned} \tag{45}$$

The parameters are set as follows. The fixed costs of building a distribution center $f_j = 1$, the probabilities of failure $q_j = 0.22j$, the demand of each retailer $\lambda_i = 5$, and the penalty for no supply is $l_j = 33$. 9 locations $(i/2, j/2), i, j = 0, 1, 2$ in $[0, 1] \times [0, 1]$ are available.

n	m	p	D	meanCG	meanMLE	stdCG	stdMLE	time	nIter
9	3	0.6	3	13.259	13.305	0.000	0.088	34.40	5.20

$^{*}p < 0.01$.

Table 4: Reliable facility location design. Mean values and variances of these two methods are compared by Welch's t-test and Levene's test respectively.

Due to the structure of this problem, our robust approach always chooses to build a distribution center in every location, and thus avoid the risk of failure.