The Rankability of Weighted Data

DOUBLEBLIND*
DOUBLEBLIND
DOUBLEBLIND, DOUBLEBLIND

DOUBLEBLIND
DOUBLEBLIND
DOUBLEBLIND, DOUBLEBLIND

DOUBLEBLIND
DOUBLEBLIND
DOUBLEBLIND, DOUBLEBLIND

DOUBLEBLIND
DOUBLEBLIND
DOUBLEBLIND, DOUBLEBLIND

DOUBLEBLIND
DOUBLEBLIND
DOUBLEBLIND, DOUBLEBLIND

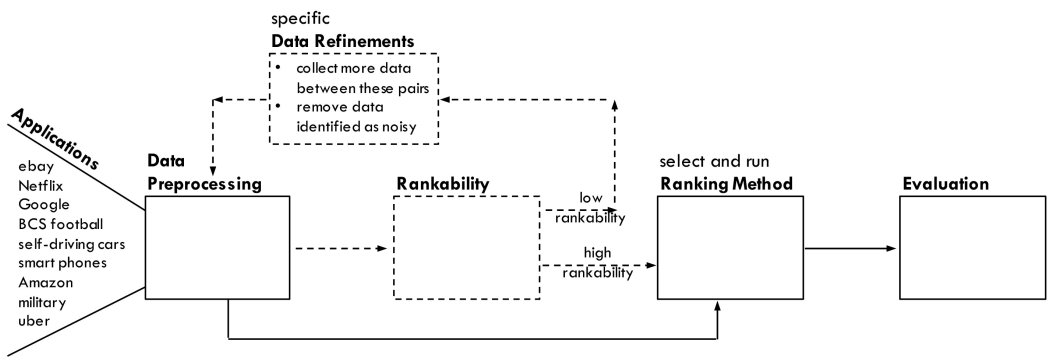


Figure 1: Common Pipeline for Ranking with Rankability. Ranking problems follow the pipeline shown in solid lines. Herein we advance the algorithms of rankability shown in dashed lines. Source: https://doubleblindremoved

ABSTRACT

The *rankability problem* attempts to measure to a dataset's inherent ability to produce a meaningful ranking of its items. While ranking algorithms impact numerous applications including web search, data mining, cybersecurity, and machine learning, little attention has been paid to the question of whether a dataset is suitable for ranking. Further, when a ranking method is applied to a dataset, the diversity of alternative optimal rankings is often not considered, resulting in a single finalized ranking.

Current rankability methods are limited to unweighted data for which the dominance relations are binary, i.e., an item either dominates or is dominated by another item. In this paper, we introduce novel rankability methods on *weighted data* for which an item may dominate another by any finite amount. We present combinatorial approaches to a weighted rankability measure and present methods for studying the diversity in alternative rankings. We compare several algorithms for computing this new measure and apply them to several weighted datasets from sports and movie recommendation.

*corresponding author

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*; Recommender systems.

KEYWORDS

ranking, rankability, linear program, integer program, combinatorial optimization, relaxation

ACM Reference Format:
DOUBLEBLIND, DOUBLEBLIND, DOUBLEBLIND, DOUBLEBLIND, and DOUBLEBLIND. 2018. The Rankability of Weighted Data. In *Proceedings of KDD (KDD 2020).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Ranking touches almost every aspect of translating computational and algorithmic results into a form that a human can use. Ranking probes the question "what's best?" Finding the "Best" involves quantification which then involves ranking. There are many applications of ranking such as movie recommendation, resource allocation, web search, optimization, cybersecurity, and machine learning. Generalized ranking models have been developed for many of these applications using learning to rank approaches [8, 12, 15, 20]. With the exceptions of Arrow's paradox and some theories from social choice (Section 2), there are few examples of significant research investigating particular foundational issues associated with ranking algorithms. Investigators often choose a ranking method or learning to rank approach arbitrarily and then accept the resulting ranking or model without asking questions of such as: Are certain parts of the ranking too similar to be disambiguated and, therefore, possibly meaningless? The field of rankability aims to measure and present information that practitioners may use to answer these and other questions. Figure 1 shows how rankability may be incorporated into a common ranking pipeline.

Ranking can be formulated as a graph problem, finding the order or rank of vertices in a (weighted) directed graph. In this paper, we use data matrices and graphs interchangeably [16]. Anderson et al. presented a rankability measure for unweighted (or uniformly weighted) graphs [3]. Ranking and rankability problems for *unweighted* data use binary dominance relations in a matrix $\underset{\sim}{D}$ where d_{ij} is 1 if a link exists in the graph from item i to item j, meaning $i > j$ (i dominates j) and 0, otherwise. A 1 in the (i, j) position of the dominance matrix $\underset{\sim}{D}$ means that i dominated j by winning either a single event or the majority of its multiple events. Applications that create wins, losses, or draws yet no differential data create unweighted data. Binary survey data (product A is preferred over product B) is an example of unweighted data.

The purpose of this paper is to present a novel extension of rankability of a set of items with a known or derived dominance relationship between pairs of items (i.e., rankability of ***weighted*** *graphs*.) This result is significant as many applications provide more information than a strict binary dominance relationship and contain a margin of belief that item *A* is better than item *B*. One obvious example is the final score in sports that provides a margin of victory or a point differential when two teams play. In these examples, the teams are the items and the scores provide the dominance relationships between pairs of items. For another example, consider surveys that use star ratings (e.g., hotel A has 5 stars while hotel B received only 2 stars). For the purpose of this paper we will often resort to sports terminology (i.e., teams and scores), yet the work is applicable to any dataset that provides the extent to which item *A* is better than item *B*.

This paper is organized as follows: Section 2 presents related work and discusses relevant literature, Section 3 presents methods in detail with examples, Section 4 presents our results and discussion for three datasets, Section 5 presents our conclusions, and Section 6 provides information on finding full implementations and reproducing our results.

2 RELATED WORK

The minimum feedback arc set problem, the linear ordering problem (LOP), and the rank aggregation problem are three classic and equivalent problems in ranking; one problem can be transformed to the another. Our rankability problem is connected to yet distinct from these problems.

One class of LOP methods are based on measuring minimum violations [2, 4, 5, 9, 10, 14, 16, 19, 21]. These methods create an ordering that minimizes the number of upsets (or feedback arcs as they are called in the minimum feedback arc set problem [11]). Uparcs represent violations between the data and the ranking and manifest as nonzero elements in the lower triangular part of the matrix of dominance relations that has been symmetrically reordered according to the ranking. In the context of rankability, they are the y_{ij} variables of our rankability formulation presented in [3] and Section 2. The y_{ij} variables are links that must be removed. Minimum violation work, however, does not consider link additions, the x_{ij} variables in our rankability formulation. Thus, our rankability work is more general than minimum violations work. Another way of summarizing the difference between our rankability work and minimum violations work: rankability considers both existing links and potential links, while minimum violations work considers only existing links.

The rank aggregation problem is a special case of the linear ordering problem [1]. In rank aggregation, k rankings are aggregated into one unified ranking. Like the rankability problem, the minimum feedback arc set problem, the linear ordering problem, and the rank aggregation problem use optimization models. Thus, the value of the objective function of the three classic problems may be considered a measure of rankability. Yet this approach to rankability is incomplete and gives unsatisfactory results because the minimum feedback arc set, the linear ordering, and the rank aggregation models were designed to produce a ranking, not a rankability measure. In contrast, our rankability model was designed, from the start, to produce a rankability measure by using a much stricter and more appropriate definition of perfection against which to measure a distance. Further, to define rankability, the rankability problem also incorporates another important measure, distance from uniqueness (i.e., the number of rankings at the optimal distance from perfection) that the minimum feedback arc set, the linear ordering, and the rank aggregation problems neglect to consider. With these two distances and our stricter definition of perfect rankability, we are able to make suggestions on how to improve the rankability of a given graph, a feature that, to our knowledge, we are the first to provide.

Another way of understanding rankability measures is through finding ties. Very few ranking methods consider tied events as part of the input data and even fewer allow for the possibility of ties in the output ranking [9]. On the other hand, rankability work allows for input ties and makes use of output ties.

A final approach to rankability is sensitivity. Perhaps a graph is unrankable if it is highly sensitive to small changes in the graph. The sensitivity of linear systems and eigensystems, which are at the heart of most ranking methods, is well-studied [6, 7, 18]. Yet such sensitivity studies compare the original ranking to the perturbed ranking and thus fail the independence property of a rankability measure. Rankability measure does incorporate ideas from sensitivity analysis. Traditional sensitivity analysis asks how small changes in the input affect the output. Instead, with rankability, we ask how *many* changes in the input are needed to create a *particular* output.

3 METHODS

3.1 Summary of Rankability for Unweighted Data

This section summarizes the key ideas for the rankability of *unweighted* graphs that, in Section 3.3, we will adapt to weighted graphs. Consider four items with the following binary matrix $\underset{\sim}{D}_1$ of pairwise dominance relations. Suppose the items are teams and each team played every other team exactly once and there were no ties in these matchups. Team 1 is in the first rank position because it beat every other team, followed by team 2 which beat all teams except the superior ranked team 1. Team 3 beat only team 4 and gets the third position and winless team 4 fills in last place. It is clear that there is one unquestionable ranking of these teams. Anderson et al. call such a matrix *perfectly rankable*. The matrix $\underset{\sim}{D}_2$ is also perfectly rankable, which becomes apparent after symmetrically reordering the rows and columns according to the ranking of $[2 \quad 4 \quad 3 \quad 1]$.

$$\underset{\sim}{D}_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

$$\underset{\sim}{D}_2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \text{ and reordered } \underset{\sim}{D}_2 = \begin{array}{c} \\ 2 \\ 4 \\ 3 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 4 & 3 & 1 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

In real applications, perfectly rankable data is rare. For example, in the seventeen seasons from 1995-2012 and 24 conferences of NCAA Division 1 college football, there was only one perfect season (the 2009 Mountain West conference). In terms of rankability, all the other seasons and conferences in college football had imperfect data. A goal of this paper is to determine a more fine-grained status of rankability beyond just the two classes of perfect and imperfect.

Anderson et al. define rankability as the degree of imperfection of the dominance matrix, i.e., its distance from the perfectly rankable upper triangular matrix. In particular, the count k, the number of link changes (additions and removals) required to make a matrix perfect. For example, the matrix $\underset{\sim}{D}_3$ requires just $k = 1$ change to make it into a 4×4 strictly upper triangular matrix. Either add a link from 3 to 4 resulting in the ranking of $[1 \quad 2 \quad 3 \quad 4]$ or add a link from 4 to 3 resulting in the ranking of $[1 \quad 2 \quad 4 \quad 3]$. Then Anderson et al. denote p as the number of rankings that are this distance k from perfection. Thus, for $\underset{\sim}{D}_3$, $p = 2$. The matrix $\underset{\sim}{D}_4$ below is less rankable since it is much farther ($k = 5$) from perfect and there are many (precisely $p = 6$) rankings that with five changes could be transformed into a perfect dominance graph.

$$\underset{\sim}{D}_3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \quad \underset{\sim}{D}_4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

In summary, the rankability measure of Anderson et al. for unweighted data involves two ideas: [3].

- *Distance from perfection.* The scalar k is the distance that the input data of pairwise dominance relations is from perfectly rankable data. In particular, k is the minimum number of edges that must be added or removed from the graph to transform it into a perfectly rankable graph.
- *Distance from uniqueness.* The scalar p is the number of rankings that are a distance k from the given graph. And the set of these rankings is denoted P.

The *rankability integer program* of [3], shown below as Model (1), takes as input the matrix of binary dominance relations $\underset{\sim}{D}$. The integer program has two sets of decision variables, x_{ij} and y_{ij}, that give information about which links should be added or deleted to transform $\underset{\sim}{D}$ into a perfect dominance graph. The decision variable x_{ij} is 1 if a link is added from i to j, and 0, otherwise. The decision variable y_{ij} is defined similarly for the removal of a link from i to j.

$$\begin{aligned} & \min \sum_{i \neq j} (x_{ij} + y_{ij}) & (1) \\ & (d_{ij} + x_{ij} - y_{ij}) + (d_{ji} + x_{ji} - y_{ji}) = 1 \quad \forall\, i < j \\ & \text{(anti-symmetry)} \\ & (d_{ij} + x_{ij} - y_{ij}) + (d_{jk} + x_{jk} - y_{jk}) + (d_{ki} + x_{ki} - y_{ki}) \leq 2 \\ & \forall\, j \neq i, k \neq j, k \neq i \\ & \text{(transitivity)} \\ & 0 \leq x_{ij} \leq 1 - d_{ij} \quad \forall\, i, j \\ & \text{(only add potential links)} \\ & 0 \leq y_{ij} \leq d_{ij} \quad \forall\, i, j \\ & \text{(only remove existing links)} \\ & x_{ij}, y_{ij} \in \{0, 1\} \quad \forall\, i \neq j \\ & \text{(binary)} \end{aligned}$$

The anti-symmetry and transitivity constraints force the perturbed matrix $\mathbf{D} + \mathbf{X} - \mathbf{Y}$ to be a dominance matrix that can be symmetrically reordered to strictly upper triangular form. The ordering of nodes that achieves this upper triangular form is the ranking. The optimal objective function value gives k, which is the minimum number of perturbations to $\underset{\sim}{D}$ (link additions in $\underset{\sim}{X}$ and link deletions in $\underset{\sim}{Y}$) required to achieve a dominance graph. The number of optimal extreme point solutions to this rankability integer program is p and the set of optimal extreme point solutions is P.

An alternative form first presented in [3] is shown below in Model (2).

$$\begin{aligned} & \max \sum_{i \neq j} d_{ij} z_{ij} & (2) \\ & z_{ij} + z_{ji} = 1 \quad \forall\, i < j \text{ (anti-symmetry)} \\ & z_{ij} + z_{jk} + z_{ki} \leq 2 \quad \forall\, j \neq i, k \neq j, k \neq i \quad \text{(transitivity)} \\ & z_{ij} \in \{0, 1\} \quad \forall\, i \text{ (binary)} \end{aligned}$$

The constraints of this alternative formulation, which is now a maximization, encompass those of the original Anderson et al.'s Model (1) and are arrived at with the simple substitution $z_{ij} = d_{ij} + x_{ij} - y_{ij}$. The following rules are used to translate the solution from this alternative formulation into the solution for the original formulation. If $z_{ij} = 0$ and $d_{ij} = 1$, then set $y_{ij} = 1$. If $z_{ij} = 1$ and $d_{ij} = 0$, then set $x_{ij} = 1$. Then k is the number of nonzeros in $\underset{\sim}{X}$ plus the number of nonzeros in $\underset{\sim}{Y}$, i.e., $k = nnz(\underset{\sim}{X}) + nnz(\underset{\sim}{Y})$.

3.2 Defining D for Weighted Data

Consider an example where the items are hotels and each review scores a hotel from 1 to 5. There are many ways to create a dominance matrix from such weighted data. A few follow.

- point differential. If item i beat item j by 5 points, then $d_{ij} = 5$ and $d_{ji} = 0$.

- point score. If item i scores 50 and item j 45, then $d_{ij} = 50$ and $d_{ji} = 45$.
- point ratio. If item i beat item j by a score of 50 to 45, then $d_{ij} = 50/45$ and $d_{ji} = 45/50$.

If there are multiple matchups between i and j, then average or cumulative values may be used. Once a D matrix is constructed we ask what is the standard of perfection for such a matrix?

3.3 Hillside Form: The Standard of Perfection for Weighted Data

This paper brings two rankability concepts, distance from perfection and distance from uniqueness, to weighted data. A distance from perfection for weighted data first requires a *definition* of perfection for weighted data. As shown in the previous section, for unweighted data, perfection is defined as a dominance matrix in strictly upper triangular form (or a matrix that can be symmetrically reordered to such form). Is there an analogous standard of perfection for weighted data? Prior work by Pedings et al. [14] provides an answer. Pedings et al. defined a so-called **hillside form**.

Definition 3.1. A matrix $\underline{D}$ is in *hillside form* if

$$\begin{aligned} d_{ij} &\le d_{ik}, \quad \forall\, i \text{ and } \forall\, j \le k && (3)\\ &\text{(ascending order across the rows)} && (4)\\ d_{ij} &\ge d_{kj}, \quad \forall\, j \text{ and } \forall\, i \le k. && (5)\\ &\text{(descending order down the columns)} && (6) \end{aligned}$$

The name is suggestive as a 3D cityplot of a matrix in hillside form looks like a sloping hillside as seen in image on the right of Figure 2. The matrix $\underline{D}_5$ of weighted data below is in hillside form, while $\underline{D}_6$ is not.

$$\underline{D}_5 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1\\2\\3\\4\\5 \end{array} & \begin{pmatrix} 0 & 3 & 5 & 8 & 15\\ 0 & 0 & 2 & 4 & 9\\ 0 & 0 & 0 & 3 & 6\\ 0 & 0 & 0 & 0 & 5\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \text{and} \quad \underline{D}_6 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1\\2\\3\\4\\5 \end{array} & \begin{pmatrix} 0 & 3 & 5 & 8 & 15\\ 0 & 0 & 2 & 4 & 9\\ 7 & 0 & 0 & 3 & 4\\ 0 & 0 & 0 & 0 & 5\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

A matrix in hillside form (or one that can be symmetrically reordered to such form) has one unquestionable ranking of its items. For example, matrix $\underline{D}_5$ says that not only is team 1 ranked above teams 2, 3, 4, and 5, but we expect team 1 to beat team 2 by some margin of victory, then team 3 by an even greater margin, and so on. For $n \times n$ matrices in hillside form, the ranking of the items is clear: $[1 \quad 2 \quad \cdots \quad n]$.

As with unweighted data, it is rare for real applications with weighted data to have (or be able to be reordered to have) hillside form. For example, recall the 2009 Mountain West season, which was perfectly rankable when win-loss binary unweighted data were used. When, instead, point differential and thus, weighted data, is used, this season is no longer perfectly rankable, i.e., there is no reordering that transforms the original data into a hillside matrix. Thus, the next question becomes how to define distance from perfection, i.e., distance from hillside form.

3.4 Hillside Count

The Hillside Count method counts the number of violations of the hillside conditions of ascending rows and descending columns and denotes this as k, the distance from perfection. A matrix with more violations is farther from hillside form and thus less rankable than one with fewer violations. For example, the matrix $\underline{D}_5$ above has 0 violations while $\underline{D}_6$ has 7 violations. Often a matrix that appears to be non-hillside can be symmetrically reordered so that it is in hillside or near hillside form. In fact, the non-hillside matrix $\underline{D}_7$ shown below is the perfect hillside matrix $\underline{D}_5$ when $\underline{D}_7$ is reordered according to the vector $[4 \quad 2 \quad 5 \quad 3 \quad 1]$.

$$\underline{D}_7 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1\\2\\3\\4\\5 \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0\\ 9 & 0 & 4 & 0 & 2\\ 5 & 0 & 0 & 0 & 0\\ 15 & 3 & 8 & 0 & 5\\ 6 & 0 & 3 & 0 & 0 \end{pmatrix} \end{array} \quad \underline{D}_7 = \underline{D}_5 = \begin{array}{c c} & \begin{array}{ccccc} 4 & 2 & 5 & 3 & 1 \end{array} \\ \begin{array}{c} 4\\2\\5\\3\\1 \end{array} & \begin{pmatrix} 0 & 3 & 5 & 8 & 15\\ 0 & 0 & 2 & 4 & 9\\ 0 & 0 & 0 & 3 & 6\\ 0 & 0 & 0 & 0 & 5\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Typically after a data matrix has been reordered to be as close to hillside form as possible, violations remain. These violations are of two types: *type 1 transitivity violations* and *type 2 transitivity violations*. Type 1 violations violate transitivity in the ranking and manifest as nonzero entries in the lower triangular part of the reordered matrix. In the context of sports, type 1 violations correspond to upsets, i.e., when a lower ranked team beat a higher ranked team. On the other hand, type 2 violations violate the differentials required by hillside form. These violations occur in the upper triangular part of the matrix. In the context of sports, type 2 violations are weak wins, which occur when a high ranked team beats a low ranked team but by a smaller margin of victory than expected. In the hillside method, an upset (i.e., type 1 violation) typically naturally accounts for more violations than a weak win (i.e., type 2 violation) as the example matrix $\underline{D}_6$ above demonstrates. The 7 in the lower triangular part of the $\underline{D}_6$ matrix accounts for 6 of the 7 violations whereas the weak win in the last column accounts for just one violation. It is possible to weight these two types of violations in other non-uniform ways if the modeler has a greater aversion to one type of violation over the other.

Finding the hidden hillside structure of a weighted dominance matrix was exactly the aim of [14]. The method of Pedings et al. finds a reordering of the items that when applied to the item-item matrix of weighted dominance data forms a matrix that is as close to *hillside form* as possible [14]. Figure 2 summarizes the method pictorially. The left is a cityplot of an 8×8 matrix in its original ordering of items. The right is a cityplot of the same data displayed with the new optimal hillside ordering.

Pedings et al. use hillside form to find a minimum violations ranking of the items, the ranking with the minimum k value. *In contrast, our goal is to produce a rankability score, rather than a ranking.* Like Pedings et al. we use k, but we also find another scalar p and we combine these to create a rankability measure for weighted data. In particular, we define p, the distance from uniqueness, as the number of rankings that, starting from $\underline{D}$, are a distance of k violations from hillside form.

Pedings et al. use the integer program of Model (7) to get k. Our contribution is a method for finding the exact value or an upper

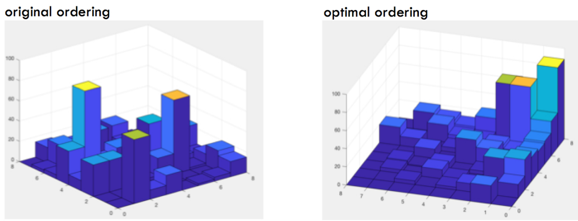


Figure 2: Cityplot of 8×8 data matrix with original ordering and hillside reordering

bound of p(see Section 3.5), which is the number of optimal extreme point solutions of this integer program.

$$
\begin{aligned}
\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (7)\\
x_{ij} + x_{ji} = 1 \quad \forall \;\; i < j \text{ (antisymmetry)}\\
x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall \;\; j \neq i, k \neq j, k \neq i \text{ (transitivity)}\\
x_{ij} \in \{0, 1\} \text{ (binary)}
\end{aligned}
$$

The objective coefficients c_{ij} are built from the weighted input matrix $\underset{\sim}{D}$ of dominance relations and are defined as $c_{ij} := \#\{\, k \mid d_{ik} < d_{jk} \,\} + \#\{\, k \mid d_{ki} > d_{kj} \,\}$, where # denotes the cardinality of the corresponding set. Thus, for example, $\#\{\, k \mid d_{ik} < d_{jk} \,\}$ is the number of teams receiving a lower point differential against team i than team j. Similarly, $\#\{\, k \mid d_{ki} > d_{kj} \,\}$ is the number of teams receiving a greater point differential against team i than team j. For this weighted rankability integer program, the scalar k is the optimal objective value and p is the number of optimal solutions. In general for linear and integer programs, finding all optimal solutions is a difficult problem. Fortunately for our particular problem, we are able to use properties of the weighted rankability problem to devise an efficient method in Section 3.5 for finding the set of all optimal solutions, which we denote by P, and thus, $p = |P|$.

Figure 3 is a pictorial representation of the difference between a more rankable (bottom half) and a less rankable (top half) weighted matrix. The top half of Figure 3 corresponds to the 2008 Patriot league men's college basketball season, which has rankability values of $k = 155$ and $p = 6$. The bottom half corresponds to the 2005 season, a much more rankable year with lower rankability values of $k = 92$ and $p = 4$. In each year, the left side shows the weighted dominance matrix $\underset{\sim}{D}$ with the original ordering and the right side shows an optimal hillside ordering output by the weighted rankability integer program of Model (7) above. In the top half, the less rankable year does not improve much from its original ordering to its optimal ordering. For that less rankable 2008 year, the right side, though optimal, is not great. Try as the integer program does, the data are just not very close to hillside form. Compare this with the more rankable 2005 data in the bottom half of Figure 3, a matrix that is much closer to hillside form. In other words, some data are just more rankable than others. This paper quantifies exactly how rankable a given weighted dataset is.

3.5 Finding p and P for Hillside Count

With default settings, solvers applied to the rankability integer program conclude with the optimal objective value k and one solution matrix $\underset{\sim}{X}$ from which an optimal ranking can be built. However,

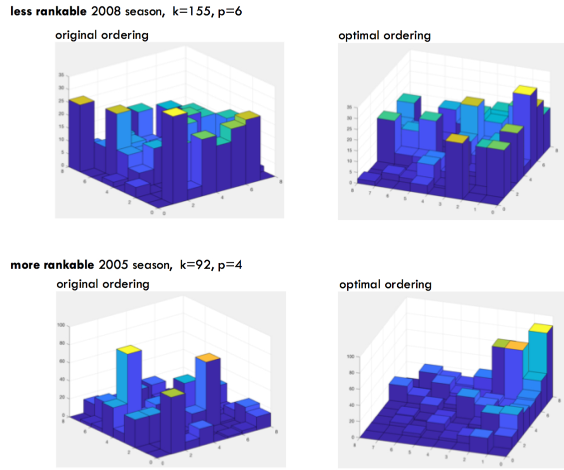


Figure 3: Cityplots of $n = 8$ college football data matrices with the original ordering (left) and the optimal hillside reordering (right). The top row is the 2008 season, a less rankable season with $k = 155$ and $p = 6$. The bottom row is the 2005 season, a more rankable season with $k = 92$ and $p = 4$.

most commercial solvers (e.g., Gurobi) have an option to output any other optimal solutions found along the way. When this option (e.g., in Gurobi, use the PoolSearch option) is set, upon termination, the rankability integer program outputs k and several $\underset{\sim}{X}$ matrices, each of which corresponds to an optimal ranking, and hence, a member of P. We call this set of rankings partial P since we cannot be sure if it is the full set P, the set of *all* optimal rankings, that we desire. We propose the following procedure in order to determine (1) if this partial P is indeed complete and hence the full set P and (2) if this partial P is incomplete, find the remaining members of P to complete the set P.

Our contribution is a novel approach to finding optimal solutions of a weighted rankability problem. This method is much more efficient than the eliminative procedure that Anderson et al. develop for unweighted rankability problems [3]. Rather than eliminating the many branches of an $n!$ tree of rankings, this procedure instead *accumulates* optimal solutions by examining a tiny subset of full rankings from the $n!$ tree of rankings. In particular, this accumulative procedure examines locations of fractional elements in the $\underset{\sim}{X}$ matrix of the linear programming *(LP) relaxation* of the weighted rankability model that is solved by an *interior point*, not an exterior point simplex, method. This last sentence generates two questions; Why an interior point solver? And why the LP relaxation?

First, we explain the interior point solver. For general linear programs, when multiple optimal solutions exist, i.e., when the feasible region has an optimal *face* rather than one optimal point, interior and exterior point solvers both end with **an** optimal solution. However, the difference lies in the location of this optimal solution. The exterior point solution is an extreme point on the optimal face whereas the interior point solution lies in the interior of the optimal face (and on or near the centroid if Mehrotra and Ye's [17] interior point method is used). For our work, we prefer the optimal solution that is in the interior of the optimal face because it is a convex combination of all optimal extreme point solutions. Theorem A.1 below shows that these optimal extreme points on the optimal face are the optimal rankings of the weighted rankability problem.

In other words, the interior point solution can be considered a *summary* of all optimal rankings. This is important as it enables us to work backwards, in 3.5 described later, from this summary solution to deduce all optimal rankings on the optimal face, and, hence, form the full set P.

Input: fractional X^*, k^*

(1) Find $\underset{\sim}{r}$, the indices after sorting the row sums of $\underset{\sim}{X}^*$ in descending order.[1]
(2) Create $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$ by symmetrically reordering $\underset{\sim}{X}^*$ by $\underset{\sim}{r}$.
(3) Identify **fixed positions** in the ranking by locating any so-called *starting arrows*, *ending arrows*, and *binary crosses* in $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$.
(4) The remaining positions are non-fixed, **varying positions**, that correspond to fractional submatrices in $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$.
(5) For each fractional submatrix, create a list of alternative subrankings for these rank positions by letting each fractional element (i, j) take its two extreme values of 0 and 1, meaning $i < j$ and $i > j$.
(6) Assemble the fixed subrankings and alternative fractional subrankings into full rankings in all possible ways.
(7) Evaluate each full ranking from Step 6 for optimality. All optimal rankings create the set P.

Output: P

When $\underset{\sim}{X}^*$, the interior point solution of LP relaxation of Model (7), is binary, $\underset{\sim}{r}$ is an optimal ranking, i.e., a member of P. Thus, in Step 1 of alg:getP when $\underset{\sim}{X}^*$ is fractional, $\underset{\sim}{r}$ may or may not be in P. Nevertheless, this reordering is helpful. For Step 2, if $\underset{\sim}{X}^*$ is binary, then $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$ is a strictly upper triangular matrix. Since we are in Case 2 and $\underset{\sim}{X}^*$ is fractional, $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$ is a nearly strictly upper triangular matrix with deviations from the upper triangular structure that are noticeable and helpful as shown in Step 3. Examples 1-3 on the subsequent pages contain each of the three "fixed position" structures (*starting arrows*, *ending arrows*, and *binary crosses*) of $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$. A binary cross is a band of rows and columns that contain entirely binary elements. For Step 4, a submatrix is called fractional if there exist any fractional elements. Thus, a fractional submatrix can contain both binary and fractional elements. Suppose Step 4 locates a 8×8 fractional submatrix. Then in Step 5, there are a maximum of 8! optimal subrankings of these 8 items in the corresponding 8 rank positions. Yet for Step 5, often many fewer than 8! subrankings need to be created since the 8×8 fractional submatrix typically also has many binary dominance relations that also must be satisfied and this, fortunately, greatly reduces the list of alternative subrankings that are possible. For Step 5, it is also helpful to identify *fractional crosses* in the fractional submatrix. A fractional cross is a **roving item** that can range over all rank positions in the subranking.

In many cases, the fractional submatrix may still contain more fractional components than may be searched in an exhaustive manner. A modification of Algorithm 4.1 enables the determination of upper and lower bounds. The lower bound on the number of optimal solutions may be determined by searching for unique solutions within a given search time or maximum number of iterations. For the lower bounds presented later in this paper, an evolutionary algorithm evolves bit vectors which are then verified to be optimal. The total number of optimal solutions found with this approach provides a lower bound on p for each submatrix. These are then aggregated to produce the global lower bound. An upper bound may be found for a submatrix of size m by a survey of fractional subsets of size $t < m$. For the results in this paper, t was chosen such that the maximum number of fractional subsets was less than 10,000. Each subset of size t is exhaustively searched for all binary permutations of the t fractional elements. Specifically, each binary permutation is analyzed to produce its specific X^* matrix. The fractional elements remaining after fixing the t elements to each binary permutation matrix are recorded, and these individual counts are summed to produce the potential upper bound for the t fractional elements. Finally, the minimum potential upper bound over the set of all t binary elements is returned as the upper bound.

[1]There may be ties in the sorted row sums of $\underset{\sim}{X}^*$, in which case, just break them arbitrarily.

4 RESULTS AND DISCUSSION

4.1 Big 12 College Football Conference

The two examples on the subsequent pages demonstrate the accumulative procedure for finding all optimal solutions for a weighted rankability problem. Both examples are from the Big 12 conference of college football. For each example, we display the optimal solution matrix $\underset{\sim}{X}^*$ output by the Interior Point solver of the linear programming relaxation of the weighted rankability problem. In all examples, the $\underset{\sim}{X}^*$ matrix is fractional, so we can apply ideas from Theorem A.3 and 3.5 to build the set P of all optimal solutions.

Example 1. The 2005 season has the optimal fractional $\underset{\sim}{X}^*$ matrix shown in Figure 4. The first row and column are binary, creating a *starting arrow*. This means that the first item, item 10, belongs in the first rank position. There are no other candidates for this position. Similarly, there is an *ending arrow* in the last rank position so item 9 belongs in the final position. In addition, there is another binary structure in the matrix; notice the *binary cross* near the center of the matrix, covering the bands corresponding to the rows and columns for items 6, 7, 11, and 4. This means that these items must appear in the sixth through ninth rank positions in that order. The remaining rank positions in $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$ contain fractional values, which, from Theorem A.3, we know represent alternatives for the corresponding rank positions. For example, in the second and third rank positions, items can be ordered either 8 then 12 or 12 then 8. In the fourth and fifth rank positions items 3 and 2 can be ordered in any of the 2! ways. Finally, the same thing happens in the tenth and eleventh rank positions with items 1 and 5. This creates a set of $2 \times 2 \times 2 = 8$ rankings that must be evaluated for their optimality. In this case, all 8 rankings shown below built from $\underset{\sim}{X}^*(\underset{\sim}{r}, \underset{\sim}{r})$ are indeed starting arrow creates one fixed position in first place

binary cross creates four fixed positions of items 6, 7, 11, and 4 in sixth through ninth places

$$\mathbf{X}^*(\mathbf{r},\mathbf{r}) = \begin{array}{c} \\ 10 \\ 12 \\ 8 \\ 3 \\ 2 \\ 6 \\ 7 \\ 11 \\ 4 \\ 1 \\ 5 \\ 9 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 10 & 12 & 8 & 3 & 2 & 6 & 7 & 11 & 4 & 1 & 5 & 9 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & .50 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & .50 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & .51 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & .49 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .53 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .47 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

ending arrow creates one fixed position in last place

Figure 4: The interior point solution of Example 1 is a fractional matrix $\underline{\mathrm{X}}^*(\underline{\mathrm{r}}, \underline{\mathrm{r}})$ with a starting arrow, ending arrow, and binary cross.

optimal with a objective value of $k^* = 255$. Thus,

$$P = \left\{ \begin{bmatrix} 10\\12\\8\\3\\2\\6\\7\\11\\4\\1\\5\\9 \end{bmatrix}, \begin{bmatrix} 10\\8\\12\\3\\2\\6\\7\\11\\4\\1\\5\\9 \end{bmatrix}, \begin{bmatrix} 10\\12\\8\\2\\3\\6\\7\\11\\4\\1\\5\\9 \end{bmatrix}, \begin{bmatrix} 10\\8\\12\\2\\3\\6\\7\\11\\4\\1\\5\\9 \end{bmatrix}, \begin{bmatrix} 10\\12\\8\\3\\2\\6\\7\\11\\4\\5\\1\\9 \end{bmatrix}, \begin{bmatrix} 10\\8\\12\\3\\2\\6\\7\\11\\4\\5\\1\\9 \end{bmatrix}, \begin{bmatrix} 10\\12\\8\\2\\3\\6\\7\\11\\4\\5\\1\\9 \end{bmatrix}, \begin{bmatrix} 10\\8\\12\\2\\3\\6\\7\\11\\4\\5\\1\\9 \end{bmatrix} \right\}.$$

Example 2. The 2004 season has the optimal fractional $\underline{\mathrm{X}}^*$ matrix shown in Figure 5. Example 2 has a starting arrow that covers

starting arrow creates three fixed positions in 1st, 2nd, and 3rd places

no binary cross

$$\mathbf{X}^*(\mathbf{r},\mathbf{r}) = \begin{array}{c} \\ 8 \\ 10 \\ 11 \\ 12 \\ 9 \\ 3 \\ 2 \\ 6 \\ 7 \\ 4 \\ 5 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 8 & 10 & 11 & 12 & 9 & 3 & 2 & 6 & 7 & 4 & 5 & 1 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & .50 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & .50 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & .66 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .54 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & .34 & .46 & 0 & .56 & .67 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .44 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .33 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

ending arrow creates two fixed positions in last two places

Figure 5: The interior point solution of Example 2 is a fractional matrix $\underline{\mathrm{X}}^*(\underline{\mathrm{r}}, \underline{\mathrm{r}})$ with a starting arrow, an ending arrow, and two isolated, though neighboring, fractional submatrices. The 5×5 fractional submatrix has a roving item, item 6, that can range over all rank positions in this subranking.

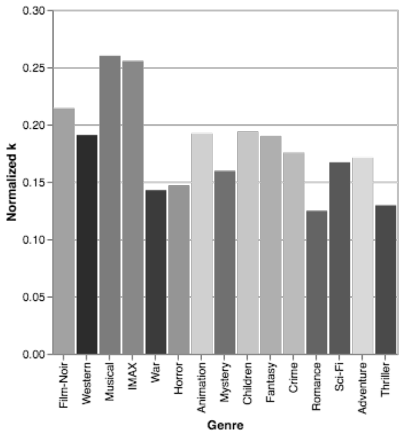
three rank positions and an ending arrow that covers two rank positions. So, in total, 5 of the 12 rank positions are fixed. The remaining seven positions have fractional values that can be used to create the alternative rankings that will be evaluated to see if they belong in P. The fourth and fifth rank positions can be filled as either 12 then 9 or 9 then 12. Then the sixth through tenth rank positions corresponding to the 5×5 fractional submatrix creates a *fractional cross* that can be used to reduce the number of $5! = 120$ subrankings that need to be considered. This fractional cross means that the corresponding item, item 6, is a *roving item* and can appear in all five rank positions in this subranking. Otherwise, the remaining elements in this 5×5 submatrix are binary, meaning that these items must appear in the given order of 3, 2, 7, 4 with 6 inserted in the five slots between these four items. Thus, there are only 5 subrankings ([6 3 2 7 4], [3 6 2 7 4], [3 2 6 7 4], [3 2 7 6 4], [3 2 7 4 6]) that need to be paired with the 2 other subrankings to create 10 full rankings that must be evaluated for optimality. After evaluation, all 10 of these 10 rankings are indeed optimal with an objective value of $k^* = 254$ and $p = 10$.

4.2 MovieLens

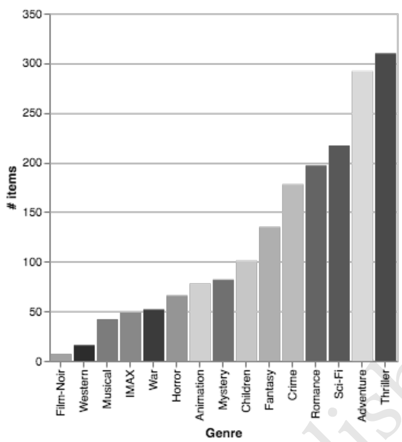
In this section, we apply our methods on a public dataset of movie recommendations to demonstrate the utility of applying weighted rankability analysis to real world datasets of various sizes. MovieLens provides non-commercial movie recommendations and is maintained by the GroupLens Research Group at the University of Minesota [13]. For the purposes of this example, we selected a MovieLens dataset with 100,000 ratings (scored 1-5) and 3,600 tag applications applied to 9,000 movies by 600 users. There are many modeling decisions made when analyzing a real dataset. Below we provide one method for generating the D matrices for the 15 movie genres shown in Figure 6(b). This method aggregates point differentials across many users to produce the final dominance matrices.

- The difference in rating for a given pair of movies (i,j) was ignored unless the number of users that rated both i and j was greater than 20. The number of movies meeting this criterion was 1,132.
- The contribution to the total weight from movie i to j for an individual user is equal to the difference in rating if their rating for movie i is greater than movie j. If the user rated both movie i and j equally, then a weight of 0.5 was added to both the weight of dominance of i over j and j over i.
- Finally, this global dominance matrix was divided into 15 movie genres. The genres and the number of movies in each genre can be seen in Figure 6(b).

To compare the overall rankability across genres, it is important to normalize k according to the size of the matrix. Specifically, k was divided by the maximum number of violations for a D matrix of size n: $n^3 - n^2$. These normalized k values are shown across genres in Figure 6(a). From this data, comparisons can be drawn from the overall rankability such as the Romance genre is more rankable than the Musical genre for these 600 users. The number of constraints increases as the number of movies in a genre increases, and therefore, the overall runtime to solve the integer program increases. The runtime in seconds as a function of the number of movies (# items) is shown in Figure 8. Timing results are reported as the average runtime of 20 runs on a 8 core workstation with 60 GB of RAM.



((a)) Normalized k visualization.



((b)) Number of movies for each genre.

Figure 6: MovieLens rankability results and data summary

Examining the overall rankability of a genre is of value; however, it may be more informative to examine what is driving the rankability within movie subsets. For example, if we select two genres to compare (Musical and Fantasy), then $X^*(r,r)$ provides further insights into what movies are driving the rankability score. This can be seen by visualizing $X^*(\mathbf{r},\mathbf{r})$ for both genres in Figure 7. As previously described, $X^*(\mathbf{r},\mathbf{r})$ is a reordering of the fractional X matrix such that items (movies) of higher rank are at the beginning. From this data, we can reason that there is more agreement on the highest recommended musicals than fantasy movies by examining the upper left corner of Figure 7. When considering the bottom right for both the genres, the opposite pattern emerges with arguably a clearer (more rankable) picture for the Fantasy genre. Using the implementation of Algorithm 4.1 described previously, the lower and upper bound for p of the Musical genre are 4,788 to 565,248, respectively. The lower bound for the Fantasy genre is 100,146,240. Generating the lower bound is equivalent to enumerating equivalent solutions, and the set of those solutions, $\mathbf{P}$, may be examined for additional insights.

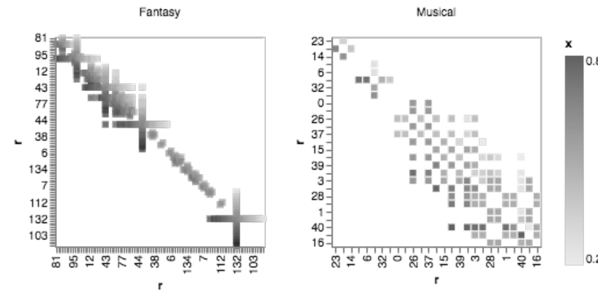


Figure 7: $X^*(r,r)$ visualization.

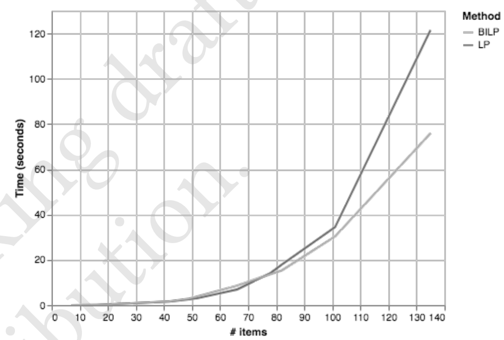


Figure 8: Timing versus size of input visualization.

4.3 ACC Conference

This section presents one final example. This example argues for a potential revised definition of rankability, one that uses k, p, and diversity of P. This example comes from the unweighted data from the 1999 season of the ACC conference of college football. We run the original rankability method of Anderson et al., using the LP relaxation of the alternative formulation of Model (2) so that Theorem A.6 and 3.5 apply.

Example 3. The 1999 season has an integer $k^* = 12$ and the following interesting optimal fractional $\underline{Z}^*$ matrix.

$$\underline{Z}^*(\underline{r},\underline{r}) = \begin{array}{c} \\ 3 \\ 1 \\ 4 \\ 8 \\ 2 \\ 6 \\ 9 \\ 5 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 3 & 1 & 4 & 8 & 2 & 6 & 9 & 5 & 7 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & .36 & .73 & 1 & .62 & 1 & 1 & 1 \\ 0 & .64 & 0 & .36 & 1 & 1 & .64 & 1 & 1 \\ 0 & .28 & .64 & 0 & .64 & .40 & 1 & 1 & 1 \\ 0 & 0 & 0 & .36 & 0 & .26 & .64 & 1 & .64 \\ 0 & .38 & 0 & .10 & .74 & 0 & .38 & .74 & .38 \\ 0 & 0 & .36 & 0 & .36 & .62 & 0 & .36 & 1 \\ 0 & 0 & 0 & 0 & 0 & .26 & .64 & 0 & .64 \\ 0 & 0 & 0 & 0 & .36 & .62 & 0 & .36 & 0 \end{pmatrix} \end{array}$$

The interior point solution of unweighted Example 3 is a highly fractional matrix $\underline{Z}^*(\underline{r},\underline{r})$, which usually portends a large p value, yet p is small, namely $p = 4$. Even though the set P contains just 4 optimal rankings, it is very diverse. Items vary greatly in their rank positions. For instance, item 6 ranges from third place to last place.

$$P = \left\{ \begin{bmatrix} 3\\8\\4\\6\\1\\2\\5\\9\\7 \end{bmatrix}, \begin{bmatrix} 3\\4\\6\\1\\2\\8\\5\\9\\7 \end{bmatrix}, \begin{bmatrix} 3\\4\\1\\2\\8\\5\\9\\7\\6 \end{bmatrix}, \begin{bmatrix} 3\\1\\8\\9\\4\\7\\6\\2\\5 \end{bmatrix} \right\}.$$

For Example 1 discussed at the beginning of the results, these differences are less dramatic and just between neighboring items in the rankings, e.g., items 8 and 12 swap as do items 1 and 5, and 2 and 3. The relative positions of items in the rankings appears rather definite. Further, a study of the average Kendall rank correlation between the two examples shows that Example 1's rankings have a high rank correlation whereas Example 3's rankings do not. This numerical indicator of the diversity of the two P sets corroborates a visual inspection of P. Example 3 also has a much higher percentage of fractional entries than Example 1. A high percentage of fractional entries in the optimal solution matrix can indicate either a large p or a very diverse P. In either case, the rankability is low.

Example 3 makes the case for a revised definition of rankability. For the current definitions, for both weighted and unweighted data, rankability r is a function of two values, k and p. Yet perhaps rankability should be a function of three values, k, p, and the diversity of the set P.

5 CONCLUSIONS

Establishing a linear ordering is inherent in many problems from ranking colleges or sports teams to the recommendation of products. This paper introduces and builds the first novel approaches to study the rankability of weighted data. Binary relationships exist in many forms from thumbs up or down ratings to simply looking at wins and losses. The inclusion of weighted data allows for important nuances to enter our understanding of rankability. This paper looks at various applications that are varied in both in size and context. In particular, the MovieLens example demonstrates that datasets containing 300 or more items can now be explored. While the ACC Conference dataset motivates a revised definition of rankability that incorporates the diversity of P. This paper also introduces novel methods to accumulate optimal solutions, enabling us to not only know the rankability of data but see where differences occur and what portions of a ranking are less certain. Rankability can serve as a critical tool in determining the quality of a dataset relative to ranking and give a sense of the underlying variability one could expect in a linear ordering of the items.

6 IMPLEMENTATIONS AND NOTEBOOKS

Full implementations of all the algorithms described in this paper are available at http://github.com/double_blind_removed. The Python package contained therein depends on freely available open source software and the Gurobi optimizer which is free to use for academic purposes. A separate repository at http://github.com/double_blind_removed contains Jupyter notebooks that provide step by step code and discussion to replicate our experiments.

REFERENCES

[1] Nir Ailon, Moses Charikar, and Alantha Newman. 2005. Aggregating Inconsistent Information: Ranking and Clustering. *Journal of The ACM - JACM* 55. https://doi.org/10.1145/1411509.1411513

[2] Iqbal Ali, Wade D. Cook, and Moshe Kress. 1986. On the minimum violations ranking of a tournament. *Management Science* 32, 6 (1986), 660–672.

[3] Paul Anderson, Timothy Chartier, and Amy Langville. 2019. The Rankability of Data. *SIAM Journal on the Mathematics of Data Science* 1 (2019), 121–143.

[4] Ronald D. Armstrong, Wade D. Cook, and Lawrence M. Seiford. 1982. Priority Ranking and Consensus Formation: The Case of Ties. *Management Science* 28, 6 (1982), 638–645.

[5] C. Richard Cassady, Lisa M. Maillart, and Sinan Salman. 2005. Ranking Sports Teams: A Customizable Quadratic Assignment Approach. *INFORMS: Interfaces* 35, 6 (2005), 497–510.

[6] Timothy P. Chartier, Erich Kreutzer, Amy N. Langville, and Kathryn E. Pedings. 2011. Sensitivity of Ranking Vectors. *SIAM Journal on Scientific Computing* 33, 3 (2011), 1077–1102.

[7] Timothy P. Chartier, Amy N. Langville, Kevin Hutson, and Michael W. Berry. 2013. Identifying influential edges in a directed network: big events, upsets, and non-transitivity. *Complex Networks* (2013).

[8] Daniel Cohen, John Foley, Hamed Zamani, James Allan, and W Bruce Croft. 2018. Universal approximation functions for fast learning to rank: Replacing expensive regression forests with simple feed-forward networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1017–1020.

[9] B. Jay Coleman. 2005. Minimizing Game Score Violations in College Football Rankings. *INFORMS: Interfaces* 35, 6 (2005), 483–496.

[10] Wade D. Cook and Lawrence M. Seiford. 1978. Priority Ranking and Consensus Formation. *Management Science* 24, 16 (1978), 1721–1732.

[11] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman.

[12] Andrey Gulin, Igor Kuralenok, and Dimitry Pavlov. 2011. Winning the transfer learning track of yahoo!'s learning to rank challenge with yetirank. In *Proceedings of the Learning to Rank Challenge*. 63–76.

[13] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2016), 19.

[14] Amy N. Langville, Kathryn Pedings, and Yoshitsugu Yamamoto. 2011. A Minimum Violations Ranking Method. *Optimization and Engineering* (2011), 1–22.

[15] Pan Li, Zhen Qin, Xuanhui Wang, and Donald Metzler. 2019. Combining Decision Trees and Neural Networks for Learning-to-Rank in Personal Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2032–2040.

[16] Rafael Marti and Gerhard Reinelt. 2011. *The Linear Ordering Problem: exact and heuristic methods in combinatorial optimization.* AMS.

[17] Sanjay Mehrotra and Yinyu Ye. 1993. Finding an interior point in the optimal face of linear programs. 62 (02 1993), 497–515.

[18] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra.* SIAM, Philadelphia.

[19] Juyong Park. 2005. On Minimum Violations Ranking in Paired Comparisons.

[20] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2019. Tf-ranking: Scalable tensorflow library for learning-to-rank. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2970–2978.

[21] Gerhard Reinelt. 1985. *The Linear Ordering Problem: Algorithms and Applications.* Heldermann Verlag.

A SUPPLEMENTAL

A.1 Why use LP relaxation?

Interior point methods are designed for linear programs, not integer programs, so we solve the LP relaxation of the rankability problem. The *LP weighted rankability polytope* for the weighted rankability problem is defined as the anti-symmetry constraints $x_{ij} + x_{ji} = 1$), the transitivity constraints ($x_{ij} + x_{jk} + x_{ki} \leq 2$), and the bound constraints ($0 \leq x_{ij} \leq 1$). Notice that the bound constraints are simply a relaxation of the binary constraints of the original integer program, and hence the name, LP relaxation. We compare the LP rankability polytope with the *IP rankability polytope*, which we define as the convex hull of all feasible solutions of the integer program of Model (7). Even though these two polytopes do not always define the same region useful results regarding the IP rankability polytope can be gathered, as Theorem A.1 shows, from the LP rankability polytope, i.e., the relaxed version of the problem.

Theorem A.1. *Every ranking of a weighted rankability problem corresponds to a binary extreme point of the LP weighted rankability polytope.*

Proof. Every ranking $\underset{\sim}{r}$ has a corresponding binary strictly upper triangular matrix $\underset{\sim}{X}(\underset{\sim}{r}, \underset{\sim}{r})$ which denotes $\underset{\sim}{X}$ after it has been symmetrically reordered according to $\underset{\sim}{r}$. The matrix $\underset{\sim}{X}$ is binary and clearly feasible since anti-symmetry and transitivity are easy to verify from the upper triangular form of $\underset{\sim}{X}(\underset{\sim}{r}, \underset{\sim}{r})$. It remains to show that $\underset{\sim}{X}$ is an extreme point, i.e., that $\underset{\sim}{X}$ cannot be written as a convex combination of other extreme points. We do this by contradiction. Suppose that there exists a scalar $0 < \alpha < 1$ and, without loss of generality, exactly two binary feasible matrices $\underset{\sim}{Y} \neq \underset{\sim}{Z}$ such that $\underset{\sim}{X} = \alpha \underset{\sim}{Y} + (1-\alpha)\underset{\sim}{Z}$. Since $\underset{\sim}{Y} \neq \underset{\sim}{Z}$, there exists at least one element, say (i, j) such that $y_{ij} \neq z_{ij}$. Suppose, without loss of generality, that $y_{ij} = 1$ and $z_{ij} = 0$. Then $x_{ij} = \alpha y_{ij} + (1-\alpha) z_{ij} = \alpha$, which means that $\underset{\sim}{X}$ is fractional, which contradicts the statement that $\underset{\sim}{X}$ is binary. Therefore, the assumption that $\underset{\sim}{X}$ is a convex combination of $\underset{\sim}{Y}$ and $\underset{\sim}{Z}$ is false and rather it is that $\underset{\sim}{X}$ is an extreme point. □

The corollary below follows from Theorem A.1.

Corollary A.2. *When the LP relaxation solves the IP, every* optimal *ranking of a weighted rankability problem of Model* (7) *corresponds to a binary extreme point on the* optimal face *of the LP weighted rankability polytope.*

When the LP relaxation of the interior point solver terminates, there are two options for the optimal objective value k^* (integer and non-integer) and two options for the optimal solution matrix $\underset{\sim}{X}^*$ (binary and fractional[2]) creating the following four outcomes.

0. k^* is non-integer and $\underset{\sim}{X}^*$ is binary.
1. k^* is integer and $\underset{\sim}{X}^*$ is binary.
2. k^* is integer and $\underset{\sim}{X}^*$ is fractional.
3. k^* is non-integer and $\underset{\sim}{X}^*$ is fractional.

Case 0 is actually not possible and therefore not an outcome because since $\underset{\sim}{C}$ being a sum of counts is integer and $\underset{\sim}{X}^*$ is binary, then the objective value $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, x_{ij}^*$ must be integer. Case 1 means that $p = 1$, there is a unique optimal solution, and the LP solution is optimal for the IP. Case 2 is the most interesting to us and we will return to it with Theorem A.3 below to build the set P of all optimal solutions. Case 3 means that the LP solution is not optimal for the IP. Our experiments show that Case 3, though possible, is very unlikely. This is also supported by Anderson et al. [3] and Reinelt et al. [16, 21].

Theorem A.3 pertains to Case 2 and gives clues for how to construct all optimal solutions from the Interior Point solver's $\underset{\sim}{X}^*$ matrix.

Theorem A.3. *If the Interior Point solver of the LP relaxed weighted rankability problem of Model* (7) *ends in Case 2 (k^* is integer and $\underset{\sim}{X}^*$ is fractional), then*

(1) *k^* is the optimal objective value for the integer program,*
(2) *$\underset{\sim}{X}^*$ is on the interior of the optimal face (i.e., the convex hull of all optimal solutions) of the integer program, and*
(3) *fractional entry (i, j) in $\underset{\sim}{X}^*$ means that there exists at least one optimal ranking in P with $x_{ij}^* = 1$ (thus, $i > j$) and at least one with $x_{ij}^* = 0$ (thus, $i < j$).*

Proof. (1) (By Contradiction.) Assume otherwise. That is, assume k^*, the optimal objective value of the linear program, is not the optimal objective value of the integer program. Then k^* is suboptimal for the integer program and the integer program's optimal objective value must be an integer superior to k^* such as $k^* - 1$, $k^* - 2, \ldots$. However, this is impossible because the linear program, being a relaxation to the integer program, must have an objective value equal to or superior to the objective value of the integer program. In other words, the only possible superior objective value for the linear program is a non-integer value yet this contradicts the fact that we are in Case 2 with an integer objective value.

(2) We show (2) by proving that the extreme points of the convex hull of the optimal face of the integer program are the extreme points of the optimal face of the linear program. Because the linear program is a relaxation, its optimal face is either: (a) equal to or (b) larger than the optimal face of the integer program. We will show that option (b) is not possible and thus the optimal face of the linear program is the optimal face of the integer program. Suppose the linear program's optimal face is larger than the integer program's optimal face, then the linear program's optimal face must contain at least one fractional extreme point. (Any additional extreme point's on the linear program's optimal face but not on the integer program's optimal face cannot be binary, otherwise they would already be on the integer program's optimal face.) Yet a fractional extreme point on the linear program's optimal face would have a non-integer objective value since the weighted sum of integer c_{ij} with fractional x_{ij} must be non-integer. This contradicts the fact that for Case 2, the optimal objective value k^* is integer. Thus, option (b) is not possible. The only possibility then is option (a): the linear program's optimal face is the integer program's optimal face. Hence, the $\underset{\sim}{X}^*$ in the interior of the linear program's optimal face is in the interior of the integer program's optimal face.

(3) By (2) above, we know that $\underset{\sim}{X}^*$ is in the interior of the optimal face of the integer program, which means that $\underset{\sim}{X}^*$ is a convex combination of the p binary optimal extreme points of the integer program, each of which, by Theorem A.1, corresponds to a ranking $\underset{\sim}{h}$ denoted by the binary matrix $\underset{\sim}{X}^h$. Thus,

$$\underset{\sim}{X}^* = \alpha_1 \underset{\sim}{X}^1 + \alpha_2 \underset{\sim}{X}^2 + \ldots + \alpha_p \underset{\sim}{X}^p,$$

[2] If $\underset{\sim}{X}^*$ contains at least one fractional value, we say it is fractional.

where $0 < \alpha_i < 1$, $\sum_{i=1}^{p} \alpha_i = 1$, and $\underset{\sim}{X}^h$ is the binary matrix corresponding to optimal ranking $\underset{\sim}{h}$. If the (i, j) entry of $\underset{\sim}{X}^*$, x^*_{ij}, is 1, then all rankings in P agree that $i > j$ because x^*_{ij} can only be 1 if all $x^h_{ij} = 1$.

$$\begin{aligned} x^*_{ij} &= \alpha_1 x^1_{ij} + \alpha_2 x^2_{ij} + \ldots + \alpha_p x^p_{ij} \\ &= \alpha_1(1) + \alpha_2(1) + \ldots + \alpha_p(1) \\ &= \alpha_1 + \alpha_2 + \ldots + \alpha_p \\ &= 1. \end{aligned}$$

Similarly, at the other extreme, the only way that $x^*_{ij} = 0$ is if all rankings in P agree that $i < j$, i.e., $x^h_{ij} = 0$ for all h. The remaining option for x^*_{ij} is a fractional value, which can happen only if some $x^h_{ij} = 1$ (meaning $i > j$) and some $x^h_{ij} = 0$ (meaning $i < j$). Thus, a fractional value in the (i, j) entry of $\underset{\sim}{X}^*$ represents disagreement among the members of P about the pairwise ranking of items i and j. □

Theorem A.3 also means that while the values in fractional entries may not be exact (since the interior point method is not guaranteed to converge to the exact centroid), the location of fractional entries is exact. Thus, Theorem A.3 inspires alg:getP, a way to construct all optimal rankings in P.

A.2 Lowerbound on p

In this section, we provide a lowerbound and thus, estimate, on p, the number of rankings in the set P of all optimal rankings. This bound may be helpful for a large example that has a complicated highly fractional $\underset{\sim}{X}^*$ matrix, which, in turn, makes it difficult to assemble rankings to evaluate in accumulative alg:getP.

Theorem A.4. *If $\underset{\sim}{X}^*$ is the exact centroid of all optimal rankings for a weighted rankability problem, then*

$$p \geq \left\lceil \frac{1}{m} \right\rceil,$$

where m is the smallest fractional element in $\underset{\sim}{X}^*$.

Proof. Assume it is the (i, j) entry of $\underset{\sim}{X}^*$ that holds the smallest fractional value m. The only way this entry can have a nonzero value is if at least one of the p binary optimal rankings $\underset{\sim}{X}^h$ for $h = 1, 2, \ldots, n$ has $i > j$, which means there exists at least one $x^h_{ij} = 1$ for $h = 1, 2, \ldots, n$. Suppose that *exactly one* of the optimal rankings, say $\underset{\sim}{X}^1$, has $i > j$ so that $x^1_{ij} = 1$. $\underset{\sim}{X}^*$ is the centroid of all binary optimal rankings $\underset{\sim}{X}^1, \underset{\sim}{X}^2, \ldots, \underset{\sim}{X}^p$ and can be written as the following convex combination

$$\underset{\sim}{X}^* = \frac{1}{p}\underset{\sim}{X}^1 + \frac{1}{p}\underset{\sim}{X}^2 + \cdots + \frac{1}{p}\underset{\sim}{X}^p.$$

Thus, $m = x^*_{ij} = \frac{1}{p}(1) = \frac{1}{p}$ and $p = \frac{1}{m}$. Now suppose exactly two of the p binary optimal rankings have $i > j$, then $m = x^*_{ij} = \frac{1}{p}(1) + \frac{1}{p}(1) = \frac{2}{p}$ and $p = \frac{2}{m} > \frac{1}{m}$. Continuing in this fashion, it follows that $p \geq \frac{1}{m}$, regardless of the number of binary optimal rankings that contribute to the fractional m. Since p is an integer, $\frac{1}{m}$ can be rounded up to the nearest integer. □

The supplemental Theorem A.3 recommended solving the weighted rankability integer program with an LP relaxation solved by an Interior Point method. When the solver concludes in Case 2 (k^* integer, $\underset{\sim}{X}^*$ fractional), then Theorem A.3 showed that $\underset{\sim}{X}^*$ is a convex combination of all optimal rankings. And when an Interior Point solver such as Mehrotra and Ye [17] is used, $\underset{\sim}{X}^*$ is likely near the centroid. While this is not the exact centroid required by the hypothesis of Theorem A.4, it is close enough to give an estimate of a lowerbound. In Table 1, we apply lowerbounding Theorem A.4 to the three examples of the previous section.

Table 1: Applying the lowerbound on p.

	m	$\lceil \frac{1}{m} \rceil$	p
Example 1 (Big 12 season 2005)	.47	3	8
Example 2 (Big 12 season 2004)	.33	4	10

Corollary A.5. *If $\underset{\sim}{X}^*$ is the exact centroid of all optimal rankings for a weighted rankability problem, then fractional entry (i, j) is the percentage of rankings in P that have $i > j$.*

For Case 2, interior point methods conclude near the exact centroid and thus a fractional entry in the optimal solution is an approximation to the percentage of rankings in P that have $i > j$.

A.3 Help constructing all optimal solutions

Theorem A.6 pertains to Case 2 and gives clues for how to construct all optimal solutions from the Interior Point solver's $\underset{\sim}{Z}^*$ matrix.

Theorem A.6. *If the Interior Point solver of the LP relaxed unweighted rankability problem of Model* (2) *ends in Case 2 (k^* is integer and $\underset{\sim}{Z}^*$ is fractional), then*

(1) *k^* is the optimal objective value for the integer program,*
(2) *$\underset{\sim}{Z}^*$ is on the interior of the optimal face (i.e., the convex hull of all optimal solutions) of the integer program, and*
(3) *fractional entry (i, j) in $\underset{\sim}{Z}^*$ means that there exists at least one optimal ranking in P with $z^*_{ij} = 1$ (thus, $i > j$) and at least one with $z^*_{ij} = 0$ (thus, $i < j$).*

Proof. The proof of Theorem A.3 for weighted data revolved around the integrality of the weighted Model (7)'s objective coefficients c_{ij}. The proof for this theorem follows that of Theorem A.3. □

As a result, this means that 3.5 can also be used for the unweighted case. That is, when an interior point solver applied to an unweighted rankability problem, Model (2), concludes with an integer k^* and a fractional optimal solution $\underset{\sim}{Z}^*$, the reordered $\underset{\sim}{Z}^*(\underset{\sim}{r}, \underset{\sim}{r})$ can be analyzed to efficiently build P, the set of all optimal rankings.