UE Project: Friendly

Maël Alet, Anaël Dufrechou, and Morteza Ezzabady

Supervisor: Prof.Josiane Mothe

Paul Sabatier University, Toulouse, France

April 2021

Abstract

We have been asked to create a global ranking of scientific articles using the grades given by users of an online article database. In this report, we present several approaches to create a total order using partial orders. We also present our approach, which is a variant of OPCA. we then compare them regarding their result, and how well they perform with the input data of our specific problem. We also propose a way to generate data, and compare the approaches properly.

1 Introduction

EasyChair is an online submission and revision article database. During a conference, a two or three reviewers review each article individually, grade it and then decide whether the article is accepted or rejected. One very simple and immediate algorithm to decide on whether a paper is accepted or rejected is ranking the papers by the means of their grades. However, as we will see later, this approach presents a lot of problems. We have been asked to create another algorithm that can rank articles using the grades given by the users, so it can replace the old algorithm and hopefully also replace the trusted team's decision on whether the article is accepted or rejected. Fundamentally, this problem reduces to the problem of creating a total order using multiple partial orders.

A total order is a set plus a relation on the set (called a total order) that satisfies the conditions for a partial order plus an additional condition known as the comparability condition. A relation $\preceq$ is a total order on a set S if the following properties hold[Wei].

- Reflexivity: $a \preceq a$ for all $a \in S$
- Antisymmetry: $a \preceq b$ and $b \preceq a$ implies $a = b$

- Transitivity: $a \preceq b$ and $b \preceq c$ implies $a \preceq c$
- Comparability(trichotomy law): For any $a, b \in S$, either $a \preceq b$ or $b \preceq a$

The first three are the axioms of a partial order, while addition of the trichotomy law defines a total order.

To achieve that goal, we look at different approaches to fuse multiple partial orders into one total order and compare their results with the article database as input. In order to complement the results obtained on the real data, we also create an algorithm that can generate synthetic data on which we run and compare the proposed algorithms.

2 Data set

2.1 Generated Data

Because we lacked real data, we built a program[1] to generate random data with some constraints. These constraints are :

- minimum count of reviews for each item
- minimum count of items to review for each reviewer

In this program we create a random weighted bipartite graph under those constraints. The range of grades is 0 to 20. Here is an example of generated data:

[1] You can access and download our code on this link.

Reviewer	Submission	Grade
0	4	6
0	2	16
1	0	7
1	3	19
2	0	8
2	4	14
2	3	13
3	1	18
3	2	1
4	4	4
4	5	20
5	1	6
5	5	1
6	3	1
6	0	11
6	5	0
7	1	11
7	2	1

Table 1: 8 reviewers with minimum 2 submissions to review, 6 submissions with at least 3 review

2.2 Real Data

In order to test our solutions in real conditions, Prof.Mothe gave us two datasets of confrences held and the real reviews of submissions. We talked in more details about these in section 5.

3 Approaches

In this section we propose some approaches alongside an explanation of an algorithm from [FMM15] to solve our problem and in section 5 we will compare these approaches to find the best one. Before continuing, keep in mind that in the following description, m reviewers have reviewed n submissions and the grades they have given are in an $n \times m$ matrix P.

3.1 Average

For each submission, consider the set S of reviewers which reviewed the submission. The final scores are calculated by the following formula:

$$S_i = \{k | P_{ik} \neq NULL, 1 <= k <= m\}, \; score_i = \frac{\sum_{k \in S_i} P_{ik}}{|S_i|} \quad (1)$$

One of the disadvantages of this approach is the difference in the grading range of each reviewer. For instance, one reviewer might use the entire range of grades available by giving grades between 0 and 20. While some others might give only grades between 8 and 12, we can interpret this by saying that in some way, the second reviewer believes that no submission deserves to be rated less than 8 and that a grade of 12 is an excellent grade.

3.2 Standardized Average

First, let us introduce the operation of standardization. This operation is applied to a set of numbers $X = \{x_1, x_2, \dots, x_n\}$ in order to scale and center them so that the mean and standard deviation of the scaled and centered numbers is 0 and 1, respectively. To do that, we first start by computing the mean and standard deviation of the original set and then obtain the standardized set of numbers X' and final scores by following formulas:

$$\mu = \frac{\sum x_i}{|X|},\ \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{|X|}},\ x'_i = \frac{x_i - \mu}{\sigma} \tag{2}$$

3.3 Ordered Paired-Comparisons Algorithm (OPCA)

[FMM15] introduced a new algorithm called OPCA for aggregating multi-agent preference orderings problem. This algorithm has multiple phases.

- construction of the sets of paired comparisons
- synthesis of the sets of paired comparisons
- construction of the fused ordering

In summary, without considering their thresholds we can formulate the procedure: Set A_{ij} contains reviewers who evaluate both submissions i and j and then C_{ij} is defined as the number of reviewers who prefer submission i over submission j. In the next step, the comparison between every two submissions is determined by a comparison of C_{ij} and C_{ji}, and finally, the score of each submission is equal to the average of its comparison results.

$$\begin{aligned} A_{ij} &= \{k | P_{ik} \neq NULL \wedge P_{ik} \neq NULL,\ 1 <= k <= m\} \\ C_{ij} &= \sum_{\substack{k \in A_{ij} \\ P_{ik} > P_{jk}}} 1 + \sum_{\substack{k \in A_{ij} \\ P_{ik} = P_{jk}}} 0.5 + \sum_{\substack{k \in A_{ij} \\ P_{ik} < P_{jk}}} 0 \end{aligned} \tag{3}$$

$$\begin{aligned} B_i &= \{j | C_{ij} \neq NULL, 1 <= j <= n\} \\ D_i &= \sum_{\substack{j \in B_i \\ C_{ij} > C_{ji}}} 1 + \sum_{\substack{j \in B_i \\ C_{ij} = C_{ji}}} 0.5 + \sum_{\substack{j \in B_i \\ C_{ij} < C_{ji}}} 0 \end{aligned} \tag{4}$$

$$score_i = \frac{D_i}{|B_i|} \tag{5}$$

This method won't work well if we don't have enough data. By experience, we need at least 3 reviews for each submission and also the number of reviewers should not be greater than 1.5 times the number of submissions.

3.4 Modified OPCA

Despite the problems mentioned in the first OPCA method, we still wanted to use the core of the method on the data we have. So, we changed the comparison system of OPCA. Instead of counting whether one item is better or not than another item in a review, we included the difference of items in that review. Also, to be more accurate we can use the supplementary feature, confidence, for each reviewer. We changed the C_{ij} formula, so the given marks difference is involved now:

$$C_{ij} = \sum_{k \in A_{ij}} \max(0, P_{ik} - P_{jk}) * confidence_k \tag{6}$$

As we said above, data completeness affects the result of the OPCA. Therefore, we tried to estimate the comparison for pairs of submissions that didn't compare in our data. We used the transitive nature of partially ordered sets. To estimate the comparison between two submissions, we look for the submissions which are better than one of them and worse than the other one. For each of those target submissions, we count the defined auxiliary submissions and set a weight like other comparisons (Although, we can set a lower weight than the normal comparison). After this phase, we have a full comparison table and we can sum up the weights for each submission and sort them.

$$B_i^c = U - B_i,\ C'_{ij} = |\{k | C_{ik} > C_{ki} \wedge C_{kj} > C_{jk} \wedge j \in B_i^c \wedge k \in B_i \wedge k \in B_j\}| \tag{7}$$

$$D'_i = \sum_{\substack{j \notin B_i \\ C'_{ij} > C'_{ji}}} 1 + \sum_{\substack{j \notin B_i \\ C'_{ij} = C'_{ji}}} 0.5 + \sum_{\substack{j \notin B_i \\ C'_{ij} < C'_{ji}}} 0 \tag{8}$$

$$score_i = D_i + w * D'_i,\ 0 < w <= 1 \tag{9}$$

4 Evaluation Measures

We need an objective function to compare these approaches and choose the best one. We defined three factors to achieve this goal. These measures are defined according to total order properties mentioned in section 1. It's expected from the final ranking that has the fewest conflicts as possible with the initial partial orders. The ideal situation is when all reviewers reviewed all submissions and made the same orders in this case the final ranking has no conflict.

4.1 Duel Conflict

This value shows number of triples (i, j, k), where$i$ and j are submissions and k is reviewer, with this property:

$$P_{ik} < P_{jk} \wedge score_i > score_j \tag{10}$$

4.2 Trans Conflict

We know that if i better than j and j is better than k, then i is also better than k. So, we count the number of situations (i, j, k, u, v) with submissions i, j, k and reviewers u, v that this statement is true:

$$P_{iu} > P_{ju} \wedge P_{jv} > P_{kv} \wedge score_i < score_k \tag{11}$$

5 Experiments

We evaluated our methods on two categories of data mentioned.

5.1 Generated Data

The rankings obtained from each method on the data in Table 1 are given below. The evaluation on these results showed that in this case, method 3.4 has a better ranking with fewer conflicts.

Rank	Average	Standardized Average	Original OPCA	Modified OPCA
1	1	1	1	1
2	3	3	3	2
3	0	5	4	4
4	4	4	2	3
5	5	0	0	0
6	2	2	5	5

Table 2: Ranking given by each algorithms

	Average	Standardized Average	Original OPCA	Modified OPCA
Duel Conflict	5	4	3	2
Trans Conflict	6	6	5	3
Total	11	10	8	5

Table 3: Number of conflicts for the results of each algorithms

5.2 Real Data

In this part we provide results for the real given datasets. To protect users' privacy, we anonymized data by assigning an ID to each submission and reviewer.

- Co18

  24 submissions, 48 reviewers and 71 reviews.

- ECIR

  200 submissions, 222 reviewers and 626 reviews.

	Top 10 submissions										Duel Conflicts	Trans Conflicts
Average	6	31	11	25	15	28	10	4	21	13	3	0
Standardized Average	12	16	10	28	25	15	4	31	6	11	1	0
Original OPCA	6	25	15	4	31	11	16	10	28	14	1	0
Modified OPCA	12	4	11	10	6	13	9	14	16	15	3	1

Table 4: Result of the different algorithms on the Co18 dataset

Classification

By defining an acceptance rate λ, we assigned the label Accepted to the top $\lambda * n$ submissions and assigned the label Rejected to the rest. After this step, we can compare different methods by their classifications. Here we prepared some confusion matrices for the methods after applying them on ECIR dataset.

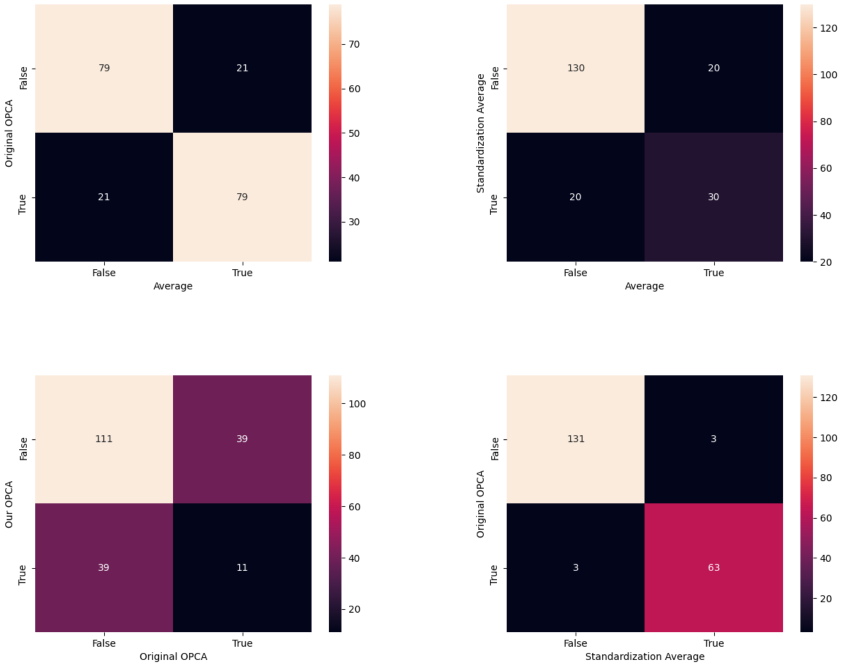

Figure 1: Original OPCA achieved fewest conflicts among the algorithms on the ECIR dataset. The λ values are 0.5, 0.25, 0.25 and 0.33 for top-left, top-right, bottom-left and bottom-right matrices respectively.

6 Conclusion

In this report we introduced multiple algorithms to create a total order out of multiple partial orders given by papers reviews. We also defined two metrics to evaluate these different methods. Using these metrics we showed that, for real data, the original Ordered Paired-Comparisons Algorithm is better than the other proposed algorithms. For the future works, probability models like bradley-terry[BT52] could lead to better performance.

References

[BT52] Ralph Allan Bradley and Milton E. Terry. “Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons”. In: *Biometrika* 39.3/4 (1952), pp. 324–345. ISSN: 00063444. URL: http://www.jstor.org/stable/2334029.

[FMM15] Fiorenzo Franceschini, Domenico Maisano, and Luca Mastrogiacomo. "A paired-comparison approach for fusing preference orderings from rank-ordered agents". In: *Information Fusion* 26 (Feb. 2015). DOI: 10.1016/j.inffus.2015.01.004.

[Wei] Eric W. Weisstein. *Totally Ordered Set*. en. Text. Publisher: Wolfram Research, Inc. URL: https://mathworld.wolfram.com/TotallyOrderedSet.html.