

Initial Plan: Implementing a system for tournament peer grading

Author: Ricky Chu

Supervisor: Richard Booth

Project description

Peer grading is an increasingly popular way to grade students' work, especially in MOOCs (Massive Open Online Courses). If there are many students a course instructor may not be able to mark all of the work. In order to solve this problem we can introduce peer grading. The idea is that each student grades the work of a subset of the total population excluding their own.

The main way to do this is to employ cardinal peer grading. This is where students mark the work of other students and assign them a score which is then used to estimate the student's actual score. The issue with this is that students may mark inaccurately or mark maliciously to inflate their grades.

To avoid the latter, we can use ordinal peer grading. Students are given a set of papers and are asked to give an ordered ranking of all the papers from best to worst. From this information we can obtain a ranking of students or estimate the student's score. One problem with this may be that ranking can be difficult when presented with many choices.

In this project we'll be exploring pairwise grading. Student as given a set of papers. The set is split off into pairs and the student compares the papers in these pairs. For example, given the set of papers $P_1, P_2, P_3, P_4 \dots P_n$ the student may be asked to compare P_1 vs P_2 P_3 vs $P_4 \dots P_{n-1}$ vs P_n and from this information we'll obtain a ranking of students or estimate the student's score.

With all three of these methods we want to give more weight to students that do better as they're more likely to recognise good work. However, we do not know who these students are until we give them a ranking making this a challenging problem.

Another challenge is the distribution of the papers, we don't want to overwhelm the students with many papers to rank, so the subsets cannot be too large. However, that means that for a large number of students we may be left with incomplete rankings. Plus, the actual distribution of the subsets may be difficult as we must not give a subset of papers to a student who's paper is in that subset.

Previous work has been done on cardinal, ordinal and pairwise grading.

For ordinal ranking there is peerrank[1] which uses fixed point iteration over a grade matrix to estimate grades. This is resitant to malicious and poor graders.

For pairwise ranking I've found some previous work that examines noisy rankings. One common method is to model the tournament with a Bradley-Terry Model and uses Maximum Likelihood Estimation to calculate skill levels for students.[2] [3]

Another approach for pairwise ranking is graph editing. This is where we flip, remove or add edges to our directed graph to obtain some sort of way to rank the students. For example we could turn it into a DAG and get a total order.[4]. Or we could model our tournament as a biparte graph and apply chain editing to obtain an ordering over the students.[5].

Aims and Objectives

The overall aim is to formulate an efficient, accurate method of ranking students from pairwise comparisons made by fellow students.

1. Formulate a method for peer ranking.

We need to first set out our requirements for a good method and adapt/create a method that satisfies these requirements.

2. Find an efficient method for assigning papers to students.

In order to run our ranking algorithm we need to distribute the papers to the students first. Finding an efficient method for this is the first step. We want to avoid students ranking their own papers, have a fair distribution of papers. Also we want to select pair in a way that we can maximise the information we can obtain from them.

3. Test my method for peer ranking and find optimal parameters.

This will involve implementing previous methods and testing and comparing it to my method. I will need a dataset to test them on. If there isn't one I will create synthetic data.

4. Create a basic website allowing users to test my method. This w

As an extension, if I have time I will compare my method to existing methods.

Risk	Likelihood	Mitigation
Lack of test data	Medium	Create synthetic data to test on
Unable to obtain licence for constrain satisfaction solver	Low	Use a free/open source solver
Equipment/Backup failure	Low	Version control and cloud backups

Work Plan

This work plan contains tasks and milestones over 12 weeks. The plan may change depending on progress and meetings with my supervisor. Each week I will meet up with my supervisor and discuss progress.

Week 1

- Background reading on social choice
- Background reading on current methods for peer ranking
- Working on the initial report
- Find existing list of algorithms for peer ranking
- Select tools required. E.g. Programming languages, Solvers, IDEs

Deliverables

- Initial report
- List of tools
- List of algorithms

Milestones

- Finish initial report
- List of ideas for a method for peer ranking

Week 2 & 3

- Selection or modification of an existing algorithm for ranking students
- Selection or modification of an existing algorithm for distribution of papers

Deliverables

- Mathematical formulation of the problems
- Fleshed out idea for paper distribution algorithm
- Fleshed out idea for ranking algorithm

Week 3 & 4

- Implementing papers distribution algorithm
- Implementing ranking algorithm

Deliverables

- Working code for the paper distribution
- Working code for the student ranking

Week 4 & 5

- Looking for a dataset/creating synthetic data
- Testing my method

Deliverables

- A dataset to test my problem on
- A report with graphs and figures explaining the efficacy of my method
- A confirmation that my method works or doesn't work

Week 6 & 7

- Working on website
- Halfway point meeting with supervisor

Deliverable

- Working website that allows you to test our algorithm

Week 8

- Website Evaluation

Deliverable

- A report on the usability of the website

Milestone

- All of the implementation is finished

Week 9 - 12

- Continuing on any tasks that may be incomplete
- Writing Report

Deliverable

- Submit the final report, code and any supporting material

Milestone

- The project is over

References

- [1] T. Walsh, The PeerRank Method for Peer Assessment, (2014). <http://arxiv.org/abs/1405.7192> (accessed February 1, 2023).
- [2] N.B. Shah, J.K. Bradley, A. Parekh, M. Wainwright, K. Ramchandran, A Case for Ordinal Peer-evaluation in MOOCs, (n.d.).
- [3] X. Chen, P.N. Bennett, K. Collins-Thompson, E. Horvitz, Pairwise ranking aggregation in a crowd-sourced setting, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, ACM, Rome Italy, 2013: pp. 193–202. <https://doi.org/10.1145/2433396.2433420>.
- [4] C. Kenyon-Mathieu, W. Schudy, How to rank with few errors, in: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, ACM, San Diego California USA, 2007: pp. 95–103. <https://doi.org/10.1145/1250790.1250806>.
- [5] J. Singleton, R. Booth, Rankings for Bipartite Tournaments via Chain Editing, (2021). <http://arxiv.org/abs/2101.02476> (accessed February 1, 2023).