Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

co-rank: An Online Tool for Collectively Deciding Efficient Rankings Among Peers

Ioannis Caragiannis
University of Patras

George A. Krimpas
University of Patras

Marianna Panteli
University of Patras

Alexandros A. Voudouris
University of Patras

The rationale behind co-rank

Our aim with co-rank is to facilitate the grading of exams or assignments in massive open online courses (MOOCs). MOOCs have emerged as a new education trend; they are online platforms that offer, to a huge number of students from all around the world, open access to top-class courses and related educational material. As the number of students attending a single online course usually exceeds 50,000, the issue of effective grading and assessment of students has been recognized as an important bottleneck for the success of such platforms. Since professional graders are costly, inexpensive grading is absolutely necessary in order to make the new educational experience beneficial for the students and, at the same time, viable for the MOOC platforms.

Not surprisingly, the problem has attracted a lot of attention in several fields of computer science, including AI. The general approach, which has been proposed to address it, is known as *peer grading*. In peer grading, the students themselves act as graders of the exam or the assignment in which they participate. That is, each student has to grade a small number of exam papers submitted by other students and report to the course instructor. Peer grading seems to be the only applicable method, especially when one aims to assess the abilities of the students in proving a mathematical statement or in demonstrating their creative writing skills. In such cases, the much simpler alternative of organizing the exam through a multiple-choice questionnaire (which could be graded automatically) is infeasible and student assessment and grading is an inherently *human computation* task.

In particular, co-rank provides functionalities for a variant of peer grading, known as *ordinal peer grading* (Shah et al. 2013). Ordinal peer grading comes to address the problem that is usually faced by the students when they are asked to act as graders and assign cardinal scores to exam papers: they are not experienced in this task and need extensive guidance in order to grade accurately. Even worse, they may have strong incentives to assign low scores to their fellow students in order to boost their own relative performance. Hence, peer grading with cardinal scores cannot be effective unless extensive calibration of the individual grades is used; actually, this is the main subject of recent work on peer grading — e.g., see (Piech et al. 2013). Ordinal peer grading is much simpler. It requires each student to grade a small number of exam papers submitted by other students and report a ranking representing the relative performance of these students in the exam. Then, an *aggregation* step will merge all the partial rankings reported into a single one.

co-rank functionalities

We describe the co-rank functionalities through a step-by-step scenario that will explain what an instructor and a student can do using the tool. The whole process is represented graphically in Figure 1.

First, the instructor creates a new exam. To do so, she uploads a file that contains the exam questions, and defines the submission and grading deadlines. She also specifies grading-related information, such as the number of exam papers that each student will have to grade and the aggregation method. She has also the option to allow the students to submit feedback or raise flags in order – for example – to report plagiarism. The instructor can define the rules for communication between herself and the students during grading; this can be either unidirectional (from her to the students) or bidirectional. Typically, the instructor broadcasts grading guidelines. In the case of bidirectional communication, the students can ask the instructor for clarifications but they cannot communicate with each other through co-rank.

Once the exam has been created, it is stored in the co-rank database and is allocated a unique id, which the students can use to track the exam and participate in it. Then, the students have access to the questions of the exam and can upload a file containing their answers by the submission deadline. Of course, students can update their files as many times as they like before the submission deadline.

After this deadline, the instructor can initiate the grading process. By doing so, a *bundle computation algorithm* computes the bundles of exam papers that students will receive for grading. Grading is performed by the students through the user-friendly interface of co-rank. Initially, each student can see the exam papers in her bundle in a random order. Using the tool, she can rank them or distinguish between the exam papers she approves and the ones she disapproves, depending on the grading method defined by the instructor. During this process, the students can assign mnemonic names to exam papers in order to facilitate grading.

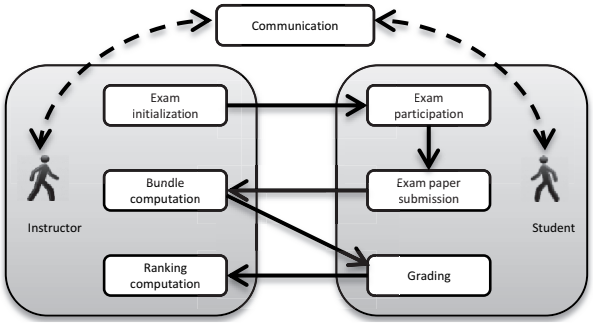

Figure 1: A system workflow diagram for co-rank.

After the grading deadline, the instructor can initiate the aggregation step. In this step, the selected aggregation method is invoked and produces a global ranking of the students. Finally, after possible adjustments by the instructor (e.g., a projection of the global ranking on cardinal scores according to a predefined distribution), the students become aware of their performance and the whole process is over.

Algorithms

Two are the main algorithmic components in co-rank. The first one is the bundle computation algorithm. Its objective is to compute a bundle of k exam papers per student so that each paper is assigned to exactly k students for grading and no student gets a bundle that contains her own exam paper. Typically, the parameter k is small (e.g., between 6 and 10, but higher or lower numbers are supported as well). This is the main strength of co-rank, namely that by assigning moderate grading tasks to the students, it produces meaningful global rankings. The bundle computation algorithm creates the bundles among n students, by picking $k+1$ disjoint perfect matchings from the complete bipartite graph $K_{n,n}$ uniformly at random. Assuming that nodes in each side of the bipartition of $K_{n,n}$ are denoted by the positive integers up to n, the adjacent nodes to a left node i in the first k matchings denote the k exam papers in a bundle and the adjacent node in the $(k+1)$-th matching indicates the student that will grade this particular bundle. The (expected) running time of the algorithm is only linear in n. In our implementation, this allows to compute bundles of size 10 among 1,000,000 students in less than 4 minutes using a very modest laptop; the same computation takes 20 seconds for 100,000 and runs instantly for significantly smaller instances.

Currently, co-rank supports three simple aggregation methods that are motivated by social choice theory. The first method is a variation of the *Borda* count, according to which the exam paper ranked first in a partial ranking gets k points, the one ranked second gets $k-1$ points, and so on. The Borda score of an exam paper is the total points it gets. The final global ranking follows by simply sorting the exam papers in decreasing order in terms of their Borda scores; ties are broken uniformly at random. This aggregation method has been studied in our recent paper (Caragiannis, Krimpas, and Voudouris 2015), where it is shown to be particularly robust both in theory and in simulations. The other two supported aggregation methods are variations of the *approval* voting rule. According to the first one, which we call *partition*, every student approves half of the exam papers in her bundle. In the second one, which we call *randomized approval*, the number of exam papers each student approves is chosen randomly between 1 to $k-1$. In both cases, the final score of an exam paper is the total number of approvals; again, ties are broken uniformly at random. These two rules have not been considered before in the peer grading literature. Their main characteristic is that the grading task required by the students is even simpler than ranking: all a student has to do is to approve a predefined number of exam papers (expressing a preference for all these papers in comparison to her disapproved ones). Still, preliminary experiments indicate that these rules can compute reasonable global rankings. In particular, the theoretical guarantees proved for Borda count in (Caragiannis, Krimpas, and Voudouris 2015) hold for randomized approval as well; in a (technical) sense, randomized approval is a variation of Borda count.

Implementation details and demo

The implementation of co-rank is based on web development technologies. On the front end, it uses Bootstrap, a unified framework for HTML, CSS, and JavaScript, which allows co-rank to have a responsive user-friendly interface. On the back end, the bundle computation algorithm and the aggregation methods are implemented in the web-programming language PHP. The database, where all the information about instructors, students, and exams is stored, has been implemented using MySQL. Our current implementation is available at `co-rank.ceid.upatras.gr`.

In the AAAI demo, we will present the co-rank functionalities both from the instructor and the student side and will show how it can handle grading in an exam with 10,000 students. In our future work, we plan to continue the development of co-rank by adding functionalities (i.e., more aggregation methods), and refining its architecture so that it is as scalable as possible. We also aim to incorporate more findings from our theoretical work on ordinal peer grading and from real-life experiments in it.

References

Caragiannis, I.; Krimpas, G. A.; and Voudouris, A. A. 2015. Aggregating partial rankings with applications to peer grading in massive online open courses. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 675–683.

Piech, C.; Huang, J.; Chen, Z.; Do, C.; Ng, A.; and Koller, D. 2013. Tuned models of peer assessment in MOOCs. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM)*, 153–160.

Shah, N. B.; Bradley, J. K.; Parekh, A.; Wainwright, M.; and Ramchandran, K. 2013. A case for ordinal peer-evaluation in MOOCs. In *Neural Information Processing Systems (NIPS): Workshop on Data Driven Education*.