Trends in Computational Social Choice

17

Cite as: Ioannis Caragiannis. Recent Advances in Large-Scale Peer Grading. In Ulle Endriss (editor), *Trends in Computational Social Choice*, chapter 17, pages 327–344. AI Access, 2017.

http://www.illc.uva.nl/COST-IC1205/Book/

CHAPTER 17

Recent Advances in Large-Scale Peer Grading

Ioannis Caragiannis

17.1 Introduction

Peer grading is the standard practice for evaluating research work and has recently become a necessity in online education. Examples include a program committee that has to decide on the papers that will be accepted for presentation at a scientific conference, the members of a professional society that wish to single out a member that should be given an award, or an instructor of an online course who outsources the evaluation of an exam to the students themselves. In all these cases, the individual inputs provided by each program committee member, society member, or student have to be aggregated in order to get the final result.

The challenge that needs to be addressed is to guarantee an as high as possible level of effectiveness in the evaluation outcome, given that the individual inputs will be, in general, *partial* and *inaccurate*. Typically, the number of submissions in a big scientific conference is a few thousand (e.g., more than 2,000 papers were submitted in the last AAAI and IJCAI conferences). Of course, there is no single program committee member that has a complete view of all submitted papers. Instead, each PC member is given very few papers for review. The source of inaccuracy should be clear in the case of students who grade their peers in an online course but it can be a severe problem even among experts. In a recent experiment,[1] the organizers of the NIPS 2014 conference formed two independent program committees. Among the approximately 900 submitted papers, most were assigned to a single PC, but 166 submissions were reviewed by both committees. This let them observe how consistent the two committees were on which papers to accept. The results have revealed a surprisingly high degree of randomness in the decision process: more than half of the papers accepted by one committee were rejected by the other!

So, it should be clear that there is more than one reason why peer grading can be challenging. In this chapter, we will focus on the extremely challenging scenario that manifests itself when grading an exam in a massive open online

[1] See `blog.mrtz.org/2014/12/15/the-nips-experiment.html` for a detailed discussion on the NIPS experiment.

course (or, simply, MOOC or online course). We will follow the vision of the MOOC enthusiasts and will assume that the number of students participating in the exam is huge (our technical assumption will be that it approaches infinity). In this way, we will have taken scale, partial view, and grading inaccuracy to their extreme. Still, we will present an approach —heavily inspired by social choice theory but also of a machine learning flavour— which has been proved recently to be simultaneously simple and effective.

The rest of the chapter is structured as follows. In Section 17.2, we briefly present the challenge of peer grading in massive open online courses, discuss the current practice and introduce the concept of ordinal peer grading. Then, in Section 17.3, we introduce useful notions and discuss the main tasks that typically take place when ordinal peer grading is used. When particular technical characteristics have to be defined, we follow the recent papers by the author (Caragiannis et al., 2015, 2016b). We identify the important parameters and discuss criteria for selecting their values in Section 17.4. In that section, we also define a natural performance objective for ordinal peer grading. In Section 17.5, we present the approach from (Caragiannis et al., 2016b) for assessing the performance of ordinal peer grading methods in a particular class and for selecting the best such method when statistical information about grading behavior is available. Experimental results follow in Section 17.6. We conclude with directions for future research in Section 17.7.

17.2 Massive Open Online Courses

Platforms supporting online courses like Coursera[2] and EdX[3] have emerged as an education trend and have attracted significant funding from venture capitals and support from leading academics. Based on the data for 2015,[4] the total number of students that enrolled in at least one online course exceeded 35 million, a 100% increase compared to 2014. More than 500 Universities worldwide were involved in more than 4200 online courses in 2016, offering courses not only in popular technology-related subjects such as Computer Science, Business and Management, and Engineering, but also in the Social Sciences and Humanities.

Whether MOOCs will become successful depends on whether they will manage to find a suitable business model and secure revenue sources. It seems that the *verified certificate* that a student participating in such course can get for a few dozens of dollars can serve as such a revenue source. The verified certificate keeps information about the performance of the student in a course or in a chain of courses and can be used to justify the student's quality to potential employers. So, it should contain *reliable information* and achieving this in a popular online course is far from trivial. Enrolment data for the 50 most popular MOOC courses[5] suggests that the vision of the MOOC enthusiasts for courses with several millions of attending students is not very far.

[2] www.coursera.org

[3] www.edx.org

[4] www.class-central.com/report/moocs-2015-stats/

[5] www.onlinecoursereport.com/the-50-most-popular-moocs-of-all-time/

How can an exam with over one million of students be graded? With the emergence of MOOCs as a trend a few years ago, and as the number of students started increasing at extremely high rates while the available resources were apparently very limited (e.g., hiring professional graders would be unreasonably costly), a simple first approach that was adopted was to use *automatic grading*, i.e., exams organized around questions with multiple-choice answers that could be graded automatically. Unfortunately, this approach is unsatisfactory in an exam where the student is asked to prepare an essay or express her critical thinking over some issue; exams of this flavour are typical in courses of Social Sciences and Humanities subjects. It is also unsatisfactory in any exam in a Science or Engineering subject where the students are asked to prepare a mathematical proof. Grading in these exams is an inherently *human computation* (Law and von Ahn, 2011) task.

Self-assessment was a next step; each student was asked by the instructor to assess her progress in the course and was also given guidelines on how to do so. Self-assessment may give the student a way to get feedback from her studies but it cannot result to reliable grading information that can be used to compare students in terms of their performance in "class".

Soon, it became apparent that the students should be involved in the grading task. This led to *peer grading* (Kulkarni et al., 2013; Piech et al., 2013; Walsh, 2014), which is widely used in most MOOC platforms today. In addition, several standalone experimental tools such as `crowdgrader.org` (de Alfaro and Shavlovsky, 2014), `peergrading.org` (Raman and Joachims, 2014), and the author's co-rank[6] (Caragiannis et al., 2016a), as well as startup services such as `peergrade.io` are available. The current practice is to use the students as graders, with the traditional meaning of the term. So, each student gets some exam papers by fellow students and grades them by assigning them cardinal scores (and, possibly, giving feedback), in a similar way a professional grader would do this. This results in much noise in the grades. The students are not experienced in assessing the performance of their fellow students in absolute terms and, if they eventually learn how to do so, they will have obvious incentives to use low cardinal grades (hoping that their own grade will look better compared to the student majority). Much of the recent literature in data mining and machine learning studies methods for calibrating cardinal grades; e.g., see the work of Piech et al. (2013), Sajjadi et al. (2016), and Wright et al. (2015).

An approach, that has received much attention recently in the AI and machine learning community (Caragiannis et al., 2015, 2016b; Raman and Joachims, 2014; Shah et al., 2013), does not use cardinal scores and is very close in spirit to voting rules from social choice theory. In a nutshell (the whole process is described in detail in the next section), each student is given a small number of exam papers and is asked to rank them in terms of quality. Then, the partial rankings are combined in an aggregate global ranking that is used as the final grading information that can be stored in the verified certificate of each student. In particular, information of the form "student X was ranked in the top 11% among the 35,000 students that participated in the course Y" can be stored in

[6] `co-rank.ceid.upatras.gr`

the student's verified certificate and already carries much information that can be used by potential employers. If the instructor desires so, the ranking information can be translated to a cardinal score (like 8/10 or A+) using a predefined distribution.

17.3 Organizing Ordinal Peer Grading

We now describe the tasks that are necessary for supporting ordinal peer grading. We will keep the presentation simple by making the simplifying assumptions that *all* students will participate in grading and *no experts* will be used. We do expect however that experts (professional graders or teaching assistants) may be used in practice to calibrate the peer grading outcome; we also expect that some of the students may decide to refrain from grading. These two characteristics will complicate the general structure described below and pose important implementation issues that should be solved when deploying ordinal peer grading in real environments. Since such issues seem to be addressed in *ad hoc* ways until now, we will not incorporate them into the general structure presented below.

So, ordinal peer grading involves three main tasks: First, after the end of an exam, copies of exam papers are distributed to the students. Then, each student acts as grader and ranks the exam papers she received. Finally, the partial rankings are aggregated into a global ranking. Let us denote by n the total number of students participating in an exam (and in its grading).

1. **Distributing the exam papers**. The goal of the first task is to balance the grading load. This can be done by making copies of each exam paper and distributing them back to the students so that each student receives the same number k of exam papers by other students and each paper is given to exactly k students for grading. Following the terminology of Caragiannis et al. (2015), we use the term *bundle* to refer to the set of k exam papers assigned to a student for grading. A *bundle graph* can be used for representing this assignment. In particular, an (n, k)-bundle graph is a bipartite graph[7] $G = ([n], [n], E)$. Both its left and right node sides correspond to the n students; the students are assigned to the integers in $[n]$ randomly. An edge (i, j) indicates that student i's exam paper is in the bundle of student j. Clearly, an (n, k)-bundle graph has all its nodes with degree k while no edge is of the form (i, i).

2. **Grading.** Once the copies of the exam papers have been distributed to the students, each student has simply to rank the k exam papers in her bundle in decreasing order of quality. The instructor may affect the grading process by announcing indicative solutions of the exam and providing detailed grading instructions to the students; in this case, we typically assume that each student acts as a *perfect* grader[8] and ranks the exam papers in her bundle

[7] The notation $[n]$ is an abbreviation for the set of integers $\{1, 2, ..., n\}$.

[8] Admittedly, this assumption is very optimistic as, most probably, in practice some students will make grading mistakes in this case as well. It is useful though as it can be used to obtain upper bounds on grading performance.

correctly. If no such information becomes available from the instructor to the students, it is natural to assume that the grading performance of a student is correlated to her performance in the exam. These assumptions will be made concrete in the next section.

3. **Rank aggregation.** The last task is to take the partial rankings provided by all students as input and compute a global ranking of all exam papers. A very simple way to do this rank aggregation is to use a method inspired by the well-known *Borda* count. Each exam paper gets k points for each appearance in the top position of a partial ranking, $k-1$ points when appearing in the second position, and so on. Its Borda score is simply the total number of points it receives in this way. The final ranking is obtained by ordering the exam papers in terms of their Borda score in non-increasing order, breaking ties randomly. Many other methods can be used as well; a broad class of rank aggregation methods that contain Borda are discussed in the next section.

Despite its apparent similarities to social choice, the ordinal peer grading setting that we just defined has important differences from classical voting rules. First, the rank aggregation task is applied on partial votes only. With the exception of the papers by de Weerdt et al. (2016), Dwork et al. (2001), Sculley (2007), and Caragiannis et al. (2017), this seems to be non-standard in the literature. Second, the decision on the contents of each bundle (and, hence, the exam papers in each partial ranking) is taken by the ordinal peer grading algorithm (and not by each individual providing input as in the four papers above). A third characteristic, which has been used only sporadically in social choice (e.g., see Alon et al., 2011; Holzman and Moulin, 2013), is that the candidates and the voters coincide.

17.4 Problem Parameters and Objectives

From the description of the previous section, the main parameters that have to be decided in order to use ordinal peer grading are the bundle size k, the structure of the bundle graph, and the rank aggregation method.

First, there is a trade-off in deciding the optimal bundle size. On one hand, it should be small so that grading is possible with reasonable effort by each student. As the final grading outcome depends on the quality of the input provided by the students, the number of students that will get frustrated and give grading up or grade at random should be minimized. A small bundle size is an incentive in this direction; additional incentives by the course instructor — such as extra grades depending on the distance between the student's ranking of the exam papers in her bundle and the relative position of them in the final ranking — may guarantee the highest student participation in grading. On the other hand, large bundles imply more grading information that will be given as input to the rank aggregation algorithm. Hence, the larger the bundle size, the more accurate the final grading outcome could be.

Typically, the bundle size will be much smaller than the total number of exam papers. As a result, deciding the structure of the bundle graph is important as well. Note that a global ranking among n exam papers defines the relation between $\binom{n}{2} \in \Theta(n^2)$ pairs. In contrast, each grader provides information for $\binom{k}{2}$ pairs only; this gives a total number of $\Theta(nk^2)$ pairwise relations that can be correctly recovered (with the optimistic assumption that graders provide correct information), plus some additional pairwise relations that can be indirectly inferred by exploiting transitivity (e.g., for exam papers a, b, and c the pairwise comparisons $a \succ b$ and $b \succ c$ by different graders could be combined to conclude that $a \succ c$ as well). However, it should be clear that, as we would like to keep the bundle size small, in order to maximize the amount of information we get from the graders, we do not have the luxury to assign the same pair of exam papers to more than one grader. In graph-theoretic terms, this means that the bundle graph should not contain 4-cycles (as a 4-cycle in a bundle graph would indicate that two different graders have the same pair of exam papers in their bundles). A slightly less restrictive structure is that of a random k-regular bipartite graph as bundle graph (this is guaranteed to contain very few 4-cycles with high probability).

The most important decision is related to the rank aggregation method to be used. A property that sounds highly desirable is to come up with a global ranking of the exam papers that agrees as much as possible with the input provided by the graders. More technically, let us define the distance between the input provided by a grader and a candidate final ranking as the total number of pairwise comparisons among exam papers in which the grader disagrees with the candidate ranking. Then, a global ranking that has minimum total distance from all graders would better aggregate the individual inputs. This is a variation of the well-known Kemeny voting rule, adapted to our setting. Unfortunately, resolving Kemeny (i.e., computing the global ranking with the above property) is a well-known computationally hard problem in voting theory (Bartholdi et al., 1989). In practice, this hardness is magnified by the fact that the total number of exam papers is huge. Hence, simple rank aggregation rules like Borda would be the most desirable from the computational complexity point of view.

But once we have restrict ourselves to simple rank aggregation rule, what is the appropriate objective for selecting the best possible one? There is no single answer here; for simplicity of exposition, we will evaluate rank aggregation rules using as performance objective the expected[9] fraction of corrected recovered pairwise relations between exam papers. Essentially, we assume that there is a true (strict) ranking of the exam papers (i.e., a *ground truth* ranking) and evaluate a rank aggregation rule by measuring the similarity of the ranking produced by the rule to the ground truth.

We are now ready to present a first theoretical statement.

Theorem 17.1 (Caragiannis et al., 2015). *When Borda is used to aggregate the partial rankings provided by perfect graders, the expected fraction of correctly recovered pairwise relations in the final ranking compared to the ground truth is at*

[9]The term "expected" is used since the assignment of students to the nodes of the bundle graph is random.

least $1 - \mathcal{O}(k)$ *when the* (n, k)*-bundle graph that is used for distributing the exam papers does not contain* 4*-cycles, and at least* $1 - \mathcal{O}(\sqrt{k})$ *in general.*

Theorem 17.1 says that performance approaches optimality as the bundle size increases. This is important and suggests that ordinal peer grading can be highly scalable. But, unfortunately, it seems that such a rigorous analysis cannot be more informative than that. For example, fixing a value of k, is Borda the best choice? In other words, is it optimal among simple rank aggregation rules? Theorem 17.1 provides no answer. The constants hidden in the $\mathcal{O}$ notation are rather huge (higher than 50) and, hence, the statement gives only a rough estimation of Borda performance as a function of k. Furthermore, the proof is several pages long and quite involved.[10] It holds specifically for Borda and is based on the particular properties this rule has. It is not at all clear how the analysis could be adjusted to work for other rank aggregation rules and it is even less obvious how imperfect graders could be included in it.

17.5 A Machine Learning Approach

In this section, we present a radically different approach that was originally presented in (Caragiannis et al., 2016b). This approach aims to bypass the limitations of the rigorous theoretical analysis and even get performance estimates of the highest possible accuracy. It can be applied not only to Borda but to the broad class of type-ordering aggregation rules that we will define shortly. Also, it is not restricted to perfect graders but exploits statistical information about grading behavior when computing the performance estimate for a rank aggregation rule. More importantly, following a direction that is typical in modern machine learning literature, the approach can be used to compute the most suitable —the optimal— type-ordering aggregation rule for a given bundle size and statistical information about grading behavior.

17.5.1 Type-Ordering Aggregation Rules

We will use the term *type* of an exam paper to refer to the grading result for it. As each exam paper belongs to the bundles of k different graders, its type is a vector of k integers that contain the position the exam paper has in the k partial rankings provided by the graders that have it in their bundle. We follow the convention that the k entries in a type vector are sorted in monotone non-decreasing order. Then, the set of possible types (for bundle size k) is

$$\mathcal{T}_k = \{\sigma = (\sigma_1, \sigma_2, ..., \sigma_k) | 1 \leqslant \sigma_1 \leqslant ... \leqslant \sigma_k \leqslant k\}.$$

It can be easily seen that the number of different types in $\mathcal{T}_k$ is $\binom{2k-1}{k}$. For example, $\mathcal{T}_6$ contains 462 types.

[10] The proof uses martingale theory and Azuma's tail inequality (see standard textbooks on randomized algorithms such as the one by Mitzenmacher and Upfal, 2005) in order to cope with dependencies between the several random variables involved. These dependencies appear due to the restrictions that both the bundle size and the number of students that grade a single paper are fixed.

As another example with $k = 6$, an exam paper of type $(1, 2, 2, 2, 2, 5)$ is ranked first by one of its graders, second by four graders, and fifth by one grader. Now, consider another exam paper of type $(2, 2, 2, 2, 3, 3)$ and observe that both have the same Borda score of 28. So, Borda does not distinguish any of the two papers as best. Now, consider the two types $(1, 1, 1, 2, 5, 6)$ and $(2, 2, 2, 3, 3, 3)$ of Borda scores 26 and 27, respectively. Borda indicates that an exam paper with the second type is better. But looking carefully at the ranks, we could come up with the following interpretation. The first exam paper is very good (and most probably in one of the two top positions in any bundle) and the two low ranks might be due to poor judgement by some of the graders. In contrast, the second exam paper is just above average and this is reflected in all grades. Of course, such interpretations are valid only when they can be supported by information about the graders. But, certainly, there are cases where such interpretations are indeed valid and, in contrast to what Borda does, it might be a good idea to take them into account.

A *type-ordering aggregation rule* uses a strict ordering $\succ$ of all types in $\mathcal{T}_k$. Then, the final ranking of the exam papers follows the ordering $\succ$ of their types, breaking ties uniformly at random. In general, rules of this class seem to be very powerful. Compared to Borda, which partitions the set of exam papers into only $k^2 - k + 1$ different groups (an exam paper can have a Borda score between k and k^2), a type-ordering aggregation rule can distinguish between exponentially many (in terms of k) different types.

17.5.2 Modelling Students' Grading Behavior

The intuition discussed above suggests that the most suitable type-ordering aggregation rule for a particular exam depends on the grading behavior of the students. A simple way to express statistical information about grading behavior is to use a *noise model* for the average grader. This is done through a $k \times k$ *noise matrix* $P = (p_{i,j})_{i,j \in [k]}$, where $p_{i,j}$ denotes the probability that the exam paper with correct rank j among the k exam papers in a bundle is ranked at position i by the grader. Noise matrices are doubly stochastic: the sum of entries in any column and any row is equal to 1. The noise matrix of perfect graders is simply the $k \times k$ identity matrix. Notice that this modelling of grading behavior is very rough; a noise matrix may correspond to many different probability distributions over rankings. As we discuss in the following, this rough representation of grading behavior is enough in order to get accurate estimates of performance for type-ordering aggregation rules and to decide the most suitable rule for exams with a particular student population.

17.5.3 A Framework for Theoretical Analysis

We now present the main ideas in the theoretical analysis presented in (Caragiannis et al., 2016b). We will consider an exam and, taking the vision of the MOOC enthusiasts to the extreme, we will assume that the number of students participating in the exam is infinite. So, the positions of students in the ground truth ranking can be thought of as occupying the continuum of the interval $[0, 1]$.

In the following, we will identify each exam paper by a real number $x \in [0,1]$ that also indicates the position of the student/paper in the ground truth ranking.

Assume that we have fixed the bundle size to k, we have collected statistical information for the grading behavior of our student population in a noise matrix P, and we use a type-ordering aggregation rule that uses the ordering $\succ$ of the types in $\mathcal{T}_k$. Then, the pairwise relations between two exam papers x and y with ranks $x < y$ (i.e., x is ranked higher than y in the ground truth) is correctly recovered in the final ranking produced by the rank aggregation rule (compared to the ground truth) if both exam papers get the exact same type σ and this tie is randomly resolved in favour of exam paper x, or exam papers x and y get types σ and σ' (we use the notation $x \rhd \sigma$ and $y \rhd \sigma'$ to represent these events) so that $\sigma \succ \sigma'$. Denoting the expected fraction of correctly recovered pairwise relations by $C(k, \succ, P)$ we obtain

$$\begin{aligned} C(k,\succ,P) &= \int_0^1 \int_x^1 \left(\sum_{\sigma,\sigma':\sigma\succ\sigma'} \Pr[x \rhd \sigma \wedge y \rhd \sigma'] + \frac{1}{2}\sum_{\sigma} \Pr[x \rhd \sigma \wedge y \rhd \sigma] \right) \mathrm{d}y\,\mathrm{d}x \\ &= \sum_{\sigma,\sigma':\sigma\succ\sigma'} \int_0^1 \int_x^1 \Pr[x \rhd \sigma \wedge y \rhd \sigma']\,\mathrm{d}y\,\mathrm{d}x \\ &\quad + \frac{1}{2}\sum_{\sigma} \int_0^1 \int_x^1 \Pr[x \rhd \sigma \wedge y \rhd \sigma]\,\mathrm{d}y\,\mathrm{d}x \end{aligned}$$

The assumption of infinitely many students nullifies any dependency between the rank vectors two exam papers x and y get after grading (i.e., $\Pr[x \rhd \sigma \wedge y \rhd \sigma'] = \Pr[x \rhd \sigma] \cdot \Pr[y \rhd \sigma']$). This is due to the fact that the probability that the two exam papers will appear in the bundle of the same grader is zero and different students grade independently. So, by defining the *weight*

$$W(\sigma,\sigma') = \int_0^1 \int_x^1 \Pr[x \rhd \sigma] \cdot \Pr[y \rhd \sigma']\,\mathrm{d}y\,\mathrm{d}x \tag{17.1}$$

for every pair of types $\sigma, \sigma' \in \mathcal{T}_k$, we obtain

$$C(k,\succ,P) = \sum_{\sigma,\sigma':\sigma\succ\sigma'} W(\sigma,\sigma') + \frac{1}{2}\sum_{\sigma} W(\sigma,\sigma). \tag{17.2}$$

So, in order to compute $C(k, \succ, P)$, it suffices to compute the probability $\Pr[x \rhd \sigma]$ that an exam paper with position x in the ground truth ranking gets type $\sigma = (\sigma_1, ..., \sigma_k)$ after grading. We will now devote some space[11] to show that this probability is nothing more than a polynomial of x and, hence, computing the double integral in equation (17.1) is straightforward.

By considering all ways to distribute the entries of the type vector as ranks of an exam paper by the graders that handle it (ignoring symmetries), there are

$$N(\sigma) = \frac{k!}{d_1! \cdot ... \cdot d_k!}$$

[11] The material until the end of this subsection is technical and can be skipped at first reading.

ways that the exam paper can get type σ, where d_i is the number of graders that have the exam paper ranked i-th. Due to our assumption for infinitely many students and the uniform inclusion of them into bundles, the quality of each exam paper included in a bundle does not affect the quality of other exam papers (in the same or different bundles). Clearly, the grading by different students is performed without dependencies either. Denoting by $\mathcal{E}(x,\sigma_i)$ the event that exam paper x is ranked σ_i-th in a bundle, the probability that x is of type σ is

$$\Pr[x \rhd \sigma] \quad = \quad N(\sigma) \prod_{i=1}^{k} \Pr[\mathcal{E}(x,\sigma_i)].$$

To compute $\Pr[\mathcal{E}(x,\sigma_i)]$, it suffices to consider all possible true ranks that exam paper x may have in a bundle and account for the probability of having such a rank and being ranked σ_i-th by the grader handling the bundle. Let us denote by $\mathcal{E}^*(x,j)$ the event that the true rank of x in a bundle is j. Then,

$$\Pr[x \rhd \sigma] \quad = \quad N(\sigma) \prod_{i=1}^{k} \sum_{j=1}^{k} p_{\sigma_i,j} \Pr[\mathcal{E}^*(x,j)].$$

Now, the probability $\Pr[\mathcal{E}^*(x,j)]$ is equal to the number of ways we can choose $j-1$ exam papers to be ahead of x times the probability that all of them will indeed be ahead of x in the bundle times the probability that the rest $k-j$ exam papers in the bundle will have true ranks worse than j. The assumption for an infinite population of students allows to safely infer that each of the remaining $k-1$ exam papers in a bundle where exam paper x belongs is selected uniformly at random from the whole student population. We apply this reasoning, using L_k to denote the set of all k-entry vectors $\ell = (\ell_1, ..., \ell_k)$ with $\ell_i \in [k]$ and abbreviating $\sum_{i=1}^{k} \ell_i$ by $|\ell|_1$ for compactness of notation. We have

$$\begin{aligned}
\Pr[x \rhd \sigma] \quad &= \quad N(\sigma) \prod_{i=1}^{k} \sum_{j=1}^{k} p_{\sigma_i,j} \binom{k-1}{j-1} x^{j-1}(1-x)^{k-j} \\
&= \quad N(\sigma) \sum_{\ell \in L_k} \prod_{i=1}^{k} p_{\sigma_i,\ell_i} \binom{k-1}{\ell_i-1} x^{\ell_i-1}(1-x)^{k-\ell_i} \\
&= \quad N(\sigma) \sum_{\ell \in L_k} \left(\prod_{i=1}^{k} p_{\sigma_i,\ell_i} \binom{k-1}{\ell_i-1} \right) x^{|\ell|_1-k}(1-x)^{k^2-|\ell|_1},
\end{aligned}$$

where the second equality is obtained by exchanging the sum and product operators. Hence, $\Pr[x \rhd \sigma]$ is a univariate polynomial of degree k^2-k. Then, the double integral in the definition of $W(\sigma,\sigma')$ in (17.1) and, hence, $C(k,\succ,P)$ (using equation (17.2)) can be computed analytically with a tedious but straightforward calculation.

The quantity $C(k,\succ,P)$ is the theoretically predicted performance of the type-ordering aggregation rule in an exam with a bundle size of k and student population with grading behavior that is described by noise model P. Crucially, all

the derivations above are equalities. Hence, the only reason that could make this prediction inaccurate is the assumption for an infinite number of students participating in the exam. As we discuss later in Section 17.6, no such inaccuracy has been observed in practice and the theoretical analysis presented above is fully justified.

17.5.4 Computing the Optimal Rule

The analysis of the previous section can be used to compute the most suitable type-ordering aggregation rule for grading exams with students from a specific population. Note that, as defined in (17.1), the weights do not depend on the aggregation rule at all. They depend only on the bundle size and on the grading behavior. Instead, the aggregation rule determines the particular weights that should be summed up in the definition of $C(k, \succ, P)$. This means that, once we have information about the bundle size and the grading behavior, we can calculate the weights for every ordered pair of types first and then compute the ordering of types so that the leftmost sum in the equation (17.2) is maximized.

It is not hard to see that the problem is equivalent to solving the feedback arc set (FAS) problem on an edge-weighted complete directed graph. In particular, the input is a complete directed graph that has a node for each type $\sigma \in \mathcal{T}_k$. A directed edge from a node corresponding to type σ towards a node corresponding to type σ' has weight $W(\sigma, \sigma')$. Now, the objective is to find an ordering of the nodes so that the total weight of "consistently directed" edges from a node to a node of higher rank in the ordering is maximized.

Theorem 17.2 (Caragiannis et al., 2016b). *Computing the most suitable type-ordering aggregation rule for a scenario involving specific bundle size and grading behavior is equivalent to solving feedback arc set on an edge-weighted complete directed graph.*

FAS is NP-hard even in its very simple variant on unweighted tournaments (Alon, 2006). Even though the particular weighted version that has to be solved in our case admits a PTAS (Kenyon-Mathieu and Schudy, 2007), the solutions that such a PTAS can guarantee in reasonable time are far from optimality and the resulting type-ordering aggregation rule will consequently have highly suboptimal performance. Fortunately, the instances that have to be solved in order to compute optimal type-ordering aggregation rules have a very nice structure.[12] This structure allows to compute the optimal FAS solution (almost) exactly by a straightforward algorithm that is briefly described in Section 17.6.

Figure 17.1 summarizes the whole approach described above. Using on input the bundle size and a noise model that describes the grading behavior of a student population, the most suitable type-ordering aggregation rule is computed, together with a prediction of the expected fraction of correctly recovered pairwise relations.

[12]This is not a formal statement but this has indeed been the case for all the scenarios considered by Caragiannis et al. (2016b). So, it is conjectured therein that it holds in *any* scenario that can appear in practice.

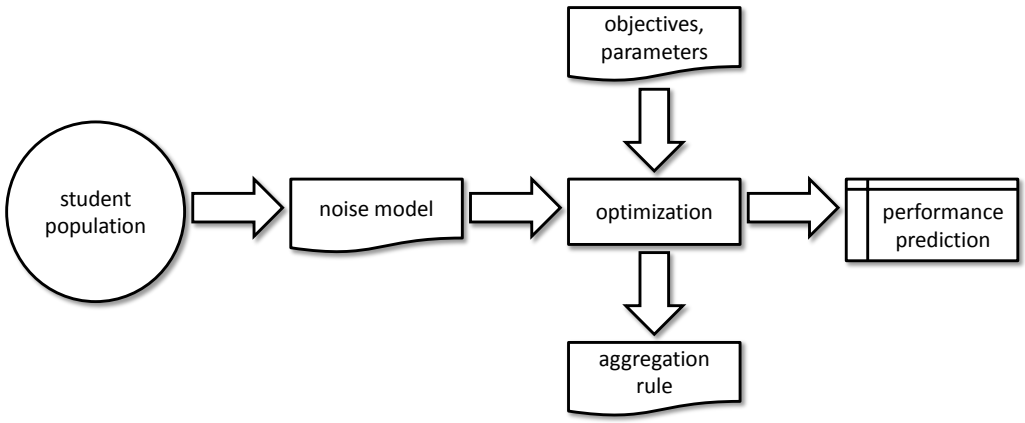

Figure 17.1: A graphical overview of the approach in (Caragiannis et al., 2016b).

Interestingly, the approach described above allows for a general statement that involves Borda. In particular, when perfect grading is used, the noise matrix has 1s only in the main diagonal (and 0s elsewhere). Then, $\Pr[x \rhd \sigma]$ has a nice simplified form that allows to conclude that the most suitable type-ordering aggregation rule is Borda and, actually, the tie-breaking does not affect the expected fraction of correctly recovered pairwise relations at all.

Theorem 17.3 (Caragiannis et al., 2016b). *For perfect graders, Borda (with any tie-breaking rule) is the optimal type-ordering aggregation rule.*

Theorem 17.3 complements Theorem 17.1 nicely and is much more informative. Furthermore, its proof is short and elegant (see Caragiannis et al., 2016b). The statement is rather surprising as Borda is among the simplest type-ordering aggregation rules; essentially, the statement says that the extra power type-ordering aggregation rules may have compared to Borda is not at all necessary when perfect grading is used.

17.6 Experimental Results

We briefly present a very small set of experimental data from (Caragiannis et al., 2016b) here. The data refer to ordinal peer grading using a bundle size k equal to 6 and grading behavior that is correlated with student quality as follows. Each student has a quality drawn uniformly at random from the interval $[\frac{1}{2}, 1]$ and affects both her performance in the exam (i.e., her position in the ground truth ranking) and her ability to grade. The ground truth is the ranking of the students in decreasing order of quality. A student b of quality q performs the grading task as follows: she considers every pair of exam papers x and y in her bundle, such that x appears ahead of y in the ground truth, and temporarily determines $x \succ_b y$ with probability q and $y \succ_b x$ with probability $1-q$. If, after considering all pairs of exam papers in the bundle, the pairwise relation $\succ_b$ is cyclic, the whole process is

repeated from scratch. Otherwise, the ranking of the exam papers in the bundle induced by $\succ_b$ is the grading outcome of student b. Due to its similarities with the well-known Mallows model (Mallows, 1957) for generating random rankings, we refer to this grading behavior as *Mallows grading*.

The noise matrix P_{mallows} that corresponds to the average Mallows grader is:

$$P_{\text{mallows}} = \begin{bmatrix} 0.6337 & 0.1753 & 0.0824 & 0.0494 & 0.0339 & 0.0253 \\ 0.1753 & 0.5112 & 0.1549 & 0.0768 & 0.0479 & 0.0339 \\ 0.0824 & 0.1549 & 0.4865 & 0.1500 & 0.0768 & 0.0494 \\ 0.0494 & 0.0768 & 0.1500 & 0.4865 & 0.1549 & 0.0824 \\ 0.0339 & 0.0479 & 0.0768 & 0.1549 & 0.5112 & 0.1753 \\ 0.0253 & 0.0339 & 0.0494 & 0.0824 & 0.1753 & 0.6337 \end{bmatrix}$$

The noise matrix has been computed by estimating the probability that a Mallows grader ranks at position i an exam paper with correct rank j among the k exam papers in her bundle; the estimate follows by simulating 10^9 Mallows graders.

Once the bundle size k and the noise model P_{mallows} are available, the approach in Section 17.5.3 is used to compute the weights $W(\sigma, \sigma')$ for every pair of types $\sigma, \sigma' \in \mathcal{T}_k$. For $k = 6$, $\mathcal{T}_6$ contains 462 types. Hence, the type-ordering aggregation rule that is optimal for Mallows graders (as defined above) will follow by solving the feedback arc set problem on a complete directed edge-weighted graph G with 462 nodes.

FAS is then solved as follows. First, notice that if we could compute an type-ordering $\succ$ so that $\sigma \succ \sigma'$ for every pair of types with $W(\sigma, \sigma') > W(\sigma', \sigma)$, then this would definitely maximize the sum of weights in the right hand side of equation (17.2). Clearly, the relative order of a pair of types σ and σ' with $W(\sigma, \sigma') = W(\sigma', \sigma)$ does not affect the sum of weights. So, the algorithm we use for FAS begins with an optimistic pseudo-ordering that requires that $\sigma \succ \sigma'$ for every pair of types σ and σ' with $W(\sigma, \sigma') > W(\sigma', \sigma)$ while it leaves any other pair of types undecided. This pseudo-ordering is represented by a directed graph H that has a node for each type in $\mathcal{T}_k$ and there is a directed edge from type σ to type σ' if $W(\sigma, \sigma') > W(\sigma', \sigma)$. If this graph did not contain any cycles, then the pseudo-ordering could be easily extended to a correct complete ordering. For example, this is indeed the case for perfect graders (as the proof of Theorem 17.3 indicates).

In general, and this is the case with the scenario with Mallows graders we consider here, the graph H will contain cycles, which we have to break in order to compute the desired type-ordering aggregation rule. In order to do this, we first decompose the graph into minimal strongly connected components $C_1, C_2, ..., C_t$ with the following properties. For $i < j$, every edge between a node σ of C_i and a node σ' of C_j has direction from σ to σ'. By definition, within each strongly connected component C_i, there are two opposite directed paths connecting every pair of nodes. So, it remains to "correct" the direction of some edges within each connected component in order to break cycles. This has to be done carefully so that the total weight of directed edges of G that appear in H at the end of this process is maximum. And once this is done, we can complete the ordering of types by adding edges with appropriate direction (so that no cycle is introduced) between nodes that are not connected in H yet.

Of course, if H contains huge strongly connected components, we have made no progress at all in this way. But for the particular experiment, H contains 453 components that consist of a single type only, six components that have size between 3 and 7, two more components of size up to 11, and one additional component with 20 nodes. Clearly, there is nothing we have to do for singleton components. For components of size up to 10, an exhaustive search will give the best correction of the direction of edges so that the contribution of corresponding weights to the sum at the right hand side of equation (17.2) is maximized. For larger components, we order their types according to their Borda score (breaking ties randomly). This yields an almost exact solution to FAS with a predicted expected fraction of correctly recovered pairwise relations equal to 85.15%. Compared to the optimistic upper bound that includes all edges of H and edges between tied types, the loss in the predicted expected fraction of correctly recovered pairwise relations is less than 0.001%.

Interestingly, in spite of the assumption for an infinite population of students in our theoretical analysis, simulations with 10,000 students yield essentially identical results. Figure 17.2 contains data from 1,000 simulated exams with Mallows graders. The coordinates of each point are the fractions of corrected recovered pairwise relations when Borda and the optimal type-ordering aggregation rule (for Mallows graders and bundle size equal to 6) is used, respectively. The average values for both rules are 85.16% and 84.39% (these differ by less than 0.01% from the values predicted using the theoretical framework in Section 17.5.3) while all values are sharply concentrated around their expectation. Furthermore, observe that Borda is always suboptimal (by approximately 0.8%) and hence the whole cloud of points in Figure 17.2 is below the main diagonal.

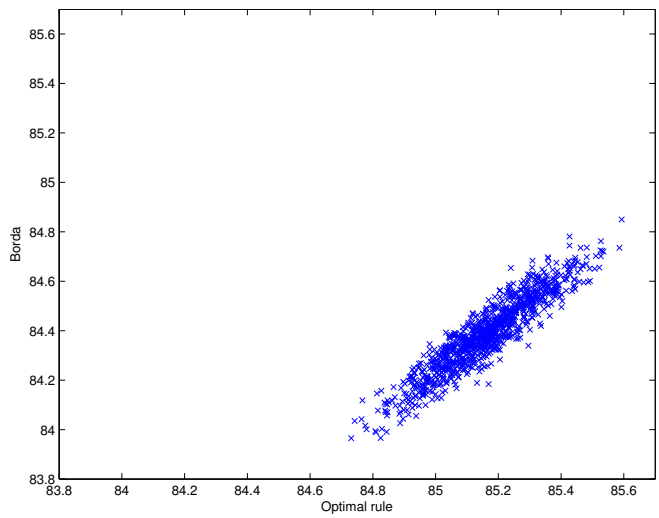

Figure 17.2: Performance of Borda compared to the optimal type-ordering aggregation rule for Mallows graders. Each of the 1,000 points corresponds to a simulated exam with the participation of 10,000 students, whose grading behavior follows the Mallows model.

17.7 Directions for Future Research

We have given a partial view of ordinal peer grading, mostly focusing on recent work by the author. More details about the approach presented in Section 17.5 can be found in (Caragiannis et al., 2016b). Therein, the reader can find a larger set of experiments compared to those presented in Section 17.6. Among other experiments, Caragiannis et al. (2016b) describe an experiment that aims to build a realistic noise model for the grading behavior of real students. This experiment has involved students attending the author's course on Computational Complexity at the University of Patras. A second interesting experiment aims to assess the impact of inaccuracies in the noise matrix to the performance of type-ordering aggregation rules and the selection of the optimal such rule. It turns out that the effect of such inaccuracies is negligible.

There are many interesting open problems regarding theoretical research on ordinal peer grading, experiments, and the deployment of our methods to real MOOCs. Regarding rank aggregation rules, we would like to see efficient implementations of approximations of Kemeny rank aggregation. An implementation in this direction is the random serial dictatorship rule described in (Caragiannis et al., 2015). This rule seems to have amazing performance with perfect graders (clearly outperforming Borda) but is rather poor in the Mallows scenario. Is there a variation of Kemeny that yields good results for imperfect graders?

We have claimed that it is easier for students to come up with a ranking of the exam papers in their bundle compared to assessing their quality in absolute terms. Interestingly, there is an even simpler grading format according to which each student is simply asked to approve a specific number of exam papers from her bundle. Then, a natural rank aggregation rule ranks the exam papers in terms of their approvals, breaking ties randomly. This functionality has already been implemented in our `co-rank` application. It would be very interesting to have a supporting theoretical analysis of it. The forthcoming paper (Caragiannis and Micha, 2017) is in this direction (but in a slightly different context).

Another thread of interesting research questions is related to incentives. Of course, classical impossibilities in social choice theory imply that students may grade strategically in order to improve their own position in the final outcome. Can this strategic behavior be taken into account when deciding the optimal rank aggregation rule? What about malicious behavior (of students that just want to fool the rank aggregation rule)? We believe that the approach presented in Section 17.5 could be adapted to strategic and malicious graders but this requires challenging technical work.

Of course, during the deployment of ordinal peer grading in real systems, there are several issues that need to be addressed. First, a few professional graders may be available. In the language we have used here, this implies a partial knowledge of the ground truth. How should this partial knowledge be combined with rank aggregation of students' grading in order to get an even better final ranking? Another issue that we have completely neglected here is related to student drop out after their participation in an exam but before its grading. Even though we do not believe that such situations invalidate our theory, there

are implementation issues that have to be taken seriously into account in real MOOCs.

Another interesting setting is when grading takes place in steps with all students involved in the first step and only the students that had good performance in the exam involved in the later ones. Besides the obvious implementation issues related to this setting, there are probably nice theoretical questions here. These are open problems that certainly deserve investigation.

Acknowledgments

I am grateful to my co-authors George Krimpas and Alexandros Voudouris for their contribution to our joint related work and for numerous interesting discussions.

Bibliography

N. Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20(1): 137–142, 2006.

N. Alon, F. A. Fischer, A. D. Procaccia, and M. Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 101–110, 2011.

J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

I. Caragiannis and E. Micha. Learning a ground truth ranking using noisy approval votes. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

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

I. Caragiannis, G. A. Krimpas, M. Panteli, and A. A. Voudouris. co-rank: An online tool for collectively deciding efficient rankings among peers. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4351–4352, 2016a.

I. Caragiannis, G. A. Krimpas, and A. A. Voudouris. How effective can simple ordinal peer grading be? In *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*, pages 323–340, 2016b.

I. Caragiannis, X. Chatzigeorgiou, G. A. Krimpas, and A. A. Voudouris. Optimizing positional scoring rules for rank aggregation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 430–436, 2017.

L. de Alfaro and M. Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE)*, pages 415–420, 2014.

M. M. de Weerdt, E. H. Gerding, and S. Stein. Minimising the rank aggregation error. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1375–1376, 2016.

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference (WWW)*, pages 613–622, 2001.

R. Holzman and H. Moulin. Impartial nominations for a prize. *Econometrica*, 81: 173–196, 2013.

C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 95–103, 2007.

C. Kulkarni, K. P. Wei, H. Le, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, and S. R. Klemmer. Peer and self assessment in massive online classes. *ACM Transactions on Computer-Human Interaction*, 20(6):33, 2013.

E. Law and L. von Ahn. *Human Computation*. Synthesis Lecture on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2011.

C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

M. Mitzenmacher and E. Upfal. *Probability and Computing – Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

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

K. Raman and T. Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1037–1046, 2014.

M. S. Sajjadi, M. Alamgir, and U. von Luxburg. Peer grading in a course on algorithms and data structures: Machine learning algorithms do not improve over simple baselines. In *Proceedings of the 3rd ACM Conference on Learning at Scale*, pages 369–378, 2016.

D. Sculley. Rank aggregation for similar items. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, pages 587–592, 2007.

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

T. Walsh. The PeerRank method for peer assessment. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, pages 909–914, 2014.

J. R. Wright, C. Thornton, and K. Leyton-Brown. Mechanical TA: Partially automated high-stakes peer grading. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE)*, pages 96–101, 2015.