How Effective Can Simple Ordinal Peer Grading Be?

IOANNIS CARAGIANNIS, University of Patras
GEORGE A. KRIMPAS, University of Patras
ALEXANDROS A. VOUDOURIS, University of Patras

Ordinal peer grading has been proposed as a simple and scalable solution for computing reliable information about student performance in massive open online courses. The idea is to outsource the grading task to the students themselves as follows. After the end of an exam, each student is asked to rank — in terms of quality — a bundle of exam papers by fellow students. An aggregation rule will then combine the individual rankings into a global one that contains all students. We define a broad class of simple aggregation rules and present a theoretical framework for assessing their effectiveness. When statistical information about the grading behaviour of students is available, the framework can be used to compute the optimal rule from this class with respect to a series of performance objectives. For example, a natural rule known as Borda is proved to be optimal when students grade correctly. In addition, we present extensive simulations and a field experiment that validate our theory and prove it to be extremely accurate in predicting the performance of aggregation rules even when only rough information about grading behaviour is available.

Additional Key Words and Phrases: Social choice; peer grading; MOOCs; rank aggregation

1. INTRODUCTION

Educational platforms such as Coursera, Udacity, and EdX provide easy access to high level education to everyone who has a decent Internet access. In September 2015, these platforms had more than 24 million users — essentially, students attending the offered courses — and this number is expected to further increase in the near future. The term "massive open online course", or simply MOOC, is very descriptive of the service these platforms offer. A MOOC is the result of their partnership with a faculty member in a top university, whose role is to design the course and organize the course material so that it takes advantage of the most popular Internet apps that the platform utilizes. Courses offered include literally everything.

Even though the service provided is certainly useful, the viability of MOOCs will strongly depend on their revenue sources. Currently, investments from VCs have secured their survival for a short term, but what about their future? A feature that could be the main source of revenue for MOOCs is the so-called *verified certificate* which the students can get at a cost of a few dozens of dollars. The verified certificate keeps information about the performance of a student in a course (or in a chain of courses) and can be used to justify a student's quality to potential employers. So, the verified certificate should have *reliable information* about the student performance in the courses she has participated in. Even though the means to guarantee this in the traditional University system is well-established, achieving this in a MOOC is a challenge.

The big issue is in the massive student participation. Of course, the Internet provides tools so that organizing exams with huge numbers of students is possible. But what about assessment and grading? As the most popular courses attract $50\,000$ students or more and the vision of MOOCs enthusiasts is for millions of students per course, is grading of assignments or exams possible? No doubt, professional graders would be extremely costly. Organizing the material using multiple-choice questions and answers that could be graded automatically cannot be an option when the students are asked to prepare an essay or a formal mathematical proof or express their critical thinking over some issue. Grading is a typical example of a *human computation* [Law and von Ahn 2011] task in these cases.

The only solution that seems consistent to the MOOCs vision is known as *peer grading* [Kulkarni et al. 2013; Piech et al. 2013; Walsh 2014], according to which the grading task is outsourced to the students that participated to the exam themselves. This approach has been already implemented in some MOOCs and standalone experimental tools such as crowdgrader.org [de Alfaro and Shavlovsky 2014], peergrading.org [Raman and Joachims 2014], and our own co-rank[1] [Caragiannis et al. 2016] are already available. Even though the approach seems straightforward, there are subtle implementation issues. For example, allowing the students to use cardinal scores is problematic, since they participate both in the exam and in grading and they may have incentives to assign low grades in order to improve their personal relative performance. Even if we assume that they grade honestly, their experience in doing so is very limited and the result will most probably be unreliable.

In this paper, we focus our attention on *ordinal peer grading*, which has recently received attention in the AI and machine learning community [Caragiannis et al. 2015; Raman and Joachims 2014; Shah et al. 2013]. Following the setting that we considered in our previous work [Caragiannis et al. 2015], each student gets a bundle of a small number (our favourite number that we have extensively used recently is 6) of exam papers so that each exam paper is given to the same number of students. Each student has to *rank* the exam papers in her bundle (in terms of quality) and an *aggregation rule* will then combine the (partial) rankings submitted by the students and come up with a final ranking of all exam papers; this will be the grading outcome. Information about the position of a student in the final ranking (e.g., top 10% out of $33\,000$ students) can be included in her verified certificate.

In [Caragiannis et al. 2015], we formally proved that a simple aggregation rule, inspired from Borda's rule from social choice theory [Brandt et al. 2016], recovers correctly an expected fraction of $1 - \mathcal{O}(1/k)$ of the pairwise relations in the underlying *ground truth* ranking, when bundles of size k are used and students make no mistakes when grading. The assumption for a ground truth and the comparison of the grading outcome to it is similar in spirit to recent approaches that combine voting and learning [Caragiannis et al. 2013, 2014; Chierichetti and Kleinberg 2014; Conitzer and Sandholm 2005; Lu and Boutilier 2011; Pivato 2013; Xia 2014; Young 1988]. The new aspect in [Caragiannis et al. 2015] (as well as in the current paper) is the relaxed requirement of recovering the ground truth only *approximately*. Experimental results show that Borda has very good performance in an imperfect grading scenario inspired by a noisy model of generating random rankings that has been proposed by Mallows [1957]. Note that, unlike other studies [Raman and Joachims 2014; Shah et al. 2013], we investigate the potential of applying ordinal peer grading exclusively, without involving any professionals in grading.

In this paper, we follow a different approach. To explain our rationale, we remark that theoretical analysis requires to handle with extra care dependencies between sev-

---

[1] Available at co-rank.ceid.upatras.gr.

eral random variables that appear due to the distribution of exam papers to bundles. The analysis of Borda was possible only due to its particular definition; we have not managed to extend the analysis to any other aggregation rule. Also, the $\mathcal{O}$ notation in the theoretical guarantee for Borda above hides large constant terms that constitute the bound of theoretical interest only. We would like to develop a "theory" for determining the performance of Borda with the highest possible accuracy. And, of course, why should we restrict our study only to Borda?

Instead, we define and study a large class of *simple* aggregation rules, which we call *type-ordering aggregation rules*. A type-ordering rule determines the position each exam paper has in the final ranking based only on the ranks each paper has in the bundles that contain it. This class includes Borda. We present a theoretical framework for assessing the performance of each member of this class with respect to a series of performance objectives. A crucial step in our study is that we have completely neglected the dependencies between the random variables that make the rigorous analysis difficult. This sacrifice of mathematical rigor is formally incorrect unless the number of students tends to infinity; this can be justified by the massive participation in MOOCs. But the best justification of our approach is that the theoretical predictions of performance are experimentally shown to be *exact*, which means that the dependencies have no positive or negative impact on performance. Furthermore, once (statistical) information about the grading behaviour of students and the desired performance objectives are known, our framework can serve as an *optimization toolkit* for selecting the optimal type-ordering aggregation rule. This requires an exact solution to an instance of the feedback arc set problem which, albeit NP-hard in general, can be solved exactly for the instances that do arise.

Our theoretical framework allows us to obtain a series of results. For example, we establish that Borda is the optimal type-ordering aggregation rule when students act as perfect graders. This is rather surprising, since Borda is among the simplest aggregation rules in the class we consider. Even though it was not observed to be optimal in any other scenario we considered, its performance is always extremely close to optimality. Furthermore, as mentioned above, the optimization task of deciding the optimal aggregation rule strongly depends on the information about grading behaviour. We study how inaccuracies of this information affect the choice of the optimal aggregation rule and its performance for the Mallows model. The results suggest a very minor impact and, essentially, a tiny sample of a student population is enough for building a fairly accurate model of grading behaviour.

Overall, our approach combines theory, simulations, and experimentation and is presented graphically in Figure 1. The lower chain of the figure describes what one would expect from a simulated exam. There is a student population and some of them participate in an exam. The preparation level of the students that determines their performance in the exam is a random variable following a uniform probability distribution. After the exam, each student acts as the grader of a small number of exam papers submitted by other students. The grading performance can depend on the preparation level as well. The grades are combined using the aggregation rule and the final ranking is compared to the ground truth to come up with the observed performance.

The most interesting part of Figure 1 is the upper chain. First, a *field experiment* can be used to extract information about the student population, translated into a noise model. We have performed such a field experiment with students in our home institution; we describe it in detail and present the collected data later in the paper. These data are used to build a *noise model* which, together with the desired performance objective, are given as input to the optimization engine. The optimal aggregation rule for the particular scenario is then constructed, and a theoretical prediction about the performance the rule is expected to have is reported. The optimal aggregation rule
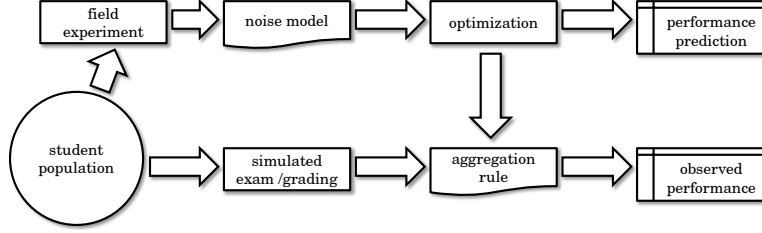
Fig. 1. A graphical overview of our approach.

can also be applied to the grades from our simulated exams (hence, the downward arrow in Figure 1) and a comparison of the theoretically predicted performance with the observed performance of the simulated exam will validate our theory.

The rest of the paper is structured as follows. We begin with preliminary definitions and useful notation in Section 2. The type-ordering aggregation rules and our theoretical framework are presented in Section 3. The field experiment and the validation of our framework are discussed in Section 4.

2. PRELIMINARIES

We assume that n students have participated in an exam and have submitted their exam papers. Our approach to ordinal peer grading has three distinct tasks: the distribution of papers to students, the grading task by each student, and the aggregation of the grades into a final result.

Distributing the exam papers

All students that participated in the exam will have to participate in grading as well. The goal of the first task is to balance their grading load. This is done by distributing (copies of) each paper to the students so that each exam paper is given to exactly k students and each student receives exactly k (distinct) exam papers. The k papers that a student receives form her *bundle*. These are the exam papers which the student has to grade. Crucially, the bundle of a student should not contain her own exam paper.

A k-regular bipartite graph $G = (U, V, E)$ with n nodes on each side of the bipartition (called an (n, k)-bundle graph in [Caragiannis et al. 2015]) can be used to represent the distribution of exam papers to students. Each node of sets U and V represents a student. An edge of the graph G between a node $u \in U$ and a node $v \in V$ indicates that the exam paper of the student corresponding to node u is in the bundle of the student corresponding to node v. The restriction on the degree of the nodes of set U means that each exam paper is given to exactly k students and the restriction on the degree of the nodes of V means that all bundles have size k.

In our previous work, we considered bundle graphs that satisfy a particular structural property, namely they contain no cycle of length 4. This was a technical constraint, required only in our theoretical analysis. Experimental results in that paper indicate that uniformly random k-regular bipartite graphs are almost as good as bundle graphs. These are the bundle graphs we considered in the current work. A random k-regular graph can be built as follows. Starting from the complete bipartite graph $K_{n,n}$ with node sets U and V, first remove the edges between nodes corresponding to the same students in U and V. Then, draw a perfect matching uniformly at random among all perfect matchings of $K_{n,n}$ that do not include previously removed edges. The edges in the k perfect matchings obtained by repeating the above step k times form the bundle graph.

Modelling the grading task

Throughout the paper, we assume that there is an underlying strict ranking of the exam papers, the *ground truth*, which we aim to recover. As it will shortly become apparent, the setting we consider is so restrictive that we should not expect to recover the ground truth exactly. Instead, we aim to recover the ground truth approximately.

A restriction of our setting is that each student is given only k exam papers to grade. Another restriction is that the grading task for each student is simply to rank the exam papers in her bundle, in decreasing order of quality. We consider different scenarios for the grading behaviour of the students. In a first scenario, assume that, after the end of the exam, the instructor announces indicative solutions and gives detailed instructions that the students can use during grading. Here, we assume that students will act as *perfect graders*. Admittedly, this is an unrealistic assumption but we include it as an extreme case in our study together with many others.

Our most realistic scenario poses another challenging restriction. In general, we assume that students make mistakes when grading. For modelling purposes, we assume that a student grades as follows. When she receives a bundle of papers, she draws a random ranking according to a probability distribution that characterizes the grading behaviour of all students participating in the exam. So, the behaviour of a student in an imperfect grading scenario is characterized by 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. Clearly, a noise matrix is doubly stochastic, i.e., the sum of the entries in any column and any row is equal to 1. Observe that the corresponding noise matrix for perfect grading is the $k \times k$ identity matrix. The term "noise model" is often used as a synonym for the term "noise matrix".

Note that a noise matrix provides only aggregate information over all students of a population. Actually, it is not hard to see that a doubly stochastic matrix may correspond to many different probability distributions over rankings. This kind of information will be the only tool we will use in our theoretical analysis. On the other hand, in our simulations, we will use more refined models of grader behaviour, assuming that it strongly depends on the level of preparation of the student for the exam and her success in it. A particular such model is the following. Each student has a quality drawn uniformly at random from the interval $[1/2, 1]$ and affects her position in the ground truth and her ability to grade as well. 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 $1 - q$; the pairwise relation $\succ_b$ will evolve into her ranking of the exam papers in her bundle. 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.

The above process has been implemented and is used in our simulations with bundles of size $k = 6$. In our theoretical analysis, we have computed the corresponding noise matrix P_{mallows} by sampling 10^9 students with uniform qualities and simulating the grading behaviour described above. For ease of reference, we present it here.

$$P_{\text{mallows}} = \begin{bmatrix} 0.6337 & 0.1753 & 0.0824 & 0.0494 & 0.0339 & 0.0253 \\ 0.1753 & 0.5112 & 0.1549 & 0.0768 & 0.0479 & 0.0339 \\ 0.0824 & 0.1549 & 0.4865 & 0.1500 & 0.0768 & 0.0494 \\ 0.0494 & 0.0768 & 0.1500 & 0.4865 & 0.1549 & 0.0824 \\ 0.0339 & 0.0479 & 0.0768 & 0.1549 & 0.5112 & 0.1753 \\ 0.0253 & 0.0339 & 0.0494 & 0.0824 & 0.1753 & 0.6337 \end{bmatrix} \tag{1}$$

Due to its similarities with the well-known Mallows model [Mallows 1957] for generating random rankings, we refer to this grading behaviour as Mallows grading.

Aggregation rules

The third important task is to aggregate the partial rankings provided by the graders into a final output ranking. This is done using an *aggregation rule*. A simple but very compelling aggregation rule is inspired by the Borda count voting rule. In our context, Borda computes a score for each exam paper by examining the positions it has in the rankings of the graders that have this exam paper in their bundles. A first position by an exam paper contributes k points to its score, a second position contributes $k - 1$ points, and so on. The outcome of Borda is a ranking of the exam papers in non-increasing order of their Borda scores. When we use Borda, we assume that ties are broken uniformly at random but other tie-breaking schemes could be considered as well.

In our previous paper [Caragiannis et al. 2015], we also considered several other aggregation rules such as a rule that we call Random Serial Dictatorship (RSD) as well as rules that are based on appropriately defined Markov chains, motivated by early work on rank aggregation on the web [Dwork et al. 2001; Page et al. 1999]. RSD is very slow in the computation of the final outcome and, even though it performs remarkably well with perfect graders, it has a poor performance in simulated exams with Mallows graders. We will not consider it in the current paper; actually, applying it with input from $10\,000$ graders, which is the typical scenario we consider in this paper, is a computational challenge. The aggregation rules that are based on Markov chains were defined in an unsuccessful attempt to distinguish between high and low quality graders and put more weight on the partial rankings of the former. These ideas are not considered in this work either. Instead, we focus on much simpler aggregation rules.

3. TYPE-ORDERING AGGREGATION RULES

We will use the term *type* to refer to the grading result of an exam paper. The grading result for it consists of the ranks it gets from the k graders that have it in their bundles. So, the type is a vector of k integers from $[k] = \{1, 2, ..., k\}$. We follow the convention that the k entries in types appear in monotone non-decreasing order. We use

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

to denote the set of all types for bundle size k. It is not hard to see that $\mathcal{T}_k$ contains $\binom{2k-1}{k}$ different types.

As an 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 Borda would give to both exam papers the same Borda score of 28. Is there some particular reason for which these two exam papers should be very close in the final ranking? 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 are due to poor judgement by 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.

So, it seems that Borda is restrictive; then, one would think that this is due to the particular scores that Borda uses. We will not consider the task of investigating whether different scores could yield better results but will instead define a much

broader class of aggregation rules. A *type-ordering aggregation rule* uses a strict ordering $\succ$ of the 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 different scores, a type-ordering aggregation rule can distinguish between exponentially many (in terms of k) different types. In the following, we use the term *Borda ordering* to refer to any ordering of the types in non-increasing order of Borda score. We also use $B(\sigma)$ to denote the Borda score of an exam paper with type $\sigma = (\sigma_1, ..., \sigma_k)$. Clearly, $B(\sigma) = \sum_{i=1}^{k}(k+1-\sigma_i) = k^2 + k - \sum_{i=1}^{k}\sigma_i$.

We remark that the use of types in the definition of a broad class of aggregation rules has been possible due to the regularity that we imposed on the bundles and the distribution of exam papers to them. Of course, this creates issues related to the theoretical analysis of these rules (such as dependencies between the random variables involved in the distribution to bundles and in grading) and it should be expected that their rigorous analysis will be much more involved than the analysis of Borda (which is already quite complicated; see [Caragiannis et al. 2015]). In the next section, we discuss how to overcome such issues.

3.1. A framework for theoretical analysis

For the analysis of type-ordering aggregation rules, we will assume an *infinite* number of students. This is close to the vision of MOOCs with huge numbers of enrolled students and is the important assumption that constitutes the theoretical analysis possible. So, the positions of students in the ground truth ranking can be thought of as occupying the continuum of the interval $[0, 1]$. We will usually identify an exam paper as a real number $x \in [0, 1]$, i.e., by its rank in the ground truth ranking. Furthermore, we will assume that in each of the k bundles to which exam paper x belongs, the remaining $k - 1$ exam papers are selected uniformly at random from the student population. Our assumption of infinitely many students allows us to ignore subtleties such as the requirement that all students in a bundle should be distinct and also different than the student that acts as the grader of the bundle (the probability that this requirement will not be satisfied in some bundle is zero).

Consider an aggregation rule that uses an ordering $\succ$ of the types defined by bundles of size k and is applied to partial rankings provided by graders whose behaviour follows the noise model P. Let us focus on computing the expected number of pairwise relations in the ground truth ranking that are correctly recovered in the outcome of the rule. It suffices to consider every pair of exam papers $x, y \in [0, 1]$ with $x < y$ (i.e., exam paper x has a better rank in the ground truth compared to exam paper y) and add one point if x has a better type than y according to the ordering $\succ$, and half a point if both exam papers have the same type. In this last case, the tie is resolved uniformly at random and the probability that the correct pairwise relation will be recovered is $1/2$. Hence, denoting by C the expected fraction of pairwise relations recovered by the rule (we will refine this notation in a while) and the event that exam paper x gets type σ after grading by $x \triangleright \sigma$, we have

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

The first sum runs over all pairs of different types σ, σ' with order $\sigma \succ \sigma'$ and the second sum runs over all types. The scary (at first glance) double integral can be hidden under the notation $W(\sigma, \sigma')$ to obtain

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

We will use the term *weight* to refer to the quantity $W(\sigma, \sigma')$. Our assumption for an infinite number of students nullifies any dependencies between the rank vectors that exam papers x and y get after grading. So, the events $x \rhd \sigma$ and $y \rhd \sigma'$ are independent and the definition of the weight $W(\sigma, \sigma')$ becomes

$$W(\sigma, \sigma') \ = \ \int_0^1 \int_x^1 \Pr[x \rhd \sigma] \cdot \Pr[y \rhd \sigma'] \, \mathrm{d}y \, \mathrm{d}x. \tag{3}$$

Let us now compute the probability that exam paper x gets type $\sigma = (\sigma_1, ..., \sigma_k)$. 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 $N(\sigma) = \frac{k!}{d_1! \cdot ... \cdot d_k!}$ 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. Again, 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

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

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,

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

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. We use L_k to denote the set of all k-entry vectors $\ell = (\ell_1, ..., \ell_k)$ with $\ell_i \in [k]$ and, for compactness of notation, we abbreviate $\sum_{i=1}^{k} \ell_i$ by $|\ell|_1$. We have

$$\Pr[x \rhd \sigma] \ = \ N(\sigma) \prod_{i=1}^{k} \sum_{j=1}^{k} p_{\sigma_i, j} \binom{k-1}{j-1} x^{j-1} (1-x)^{k-j} \tag{4}$$

$$= \ N(\sigma) \sum_{\ell \in L_k} \prod_{i=1}^{k} p_{\sigma_i, \ell_i} \binom{k-1}{\ell_i - 1} x^{\ell_i - 1} (1-x)^{k-\ell_i}$$

$$= \ N(\sigma) \sum_{\ell \in L_k} \left(\prod_{i=1}^{k} p_{\sigma_i, \ell_i} \binom{k-1}{\ell_i - 1} \right) x^{|\ell|_1 - k} (1-x)^{k^2 - |\ell|_1},$$

where the second equality is obtained by exchanging the sum and product operators. Using the fact that $(1-x)^m = \sum_{j=0}^{m} \binom{m}{j}(-1)^j x^j$ for $m = k^2 - |\ell|_1$, we obtain

$$
\begin{aligned}
\Pr[x \triangleright \sigma] &= N(\sigma) \sum_{\ell \in L_k} \left(\prod_{i=1}^{k} p_{\sigma_i, \ell_i} \binom{k-1}{\ell_i - 1} \right) x^{|\ell|_1 - k} \sum_{j=0}^{k^2 - |\ell|_1} \binom{k^2 - |\ell|_1}{j} (-1)^j x^j \\
&= N(\sigma) \sum_{\ell \in L_k} \sum_{j=0}^{k^2 - |\ell|_1} \left(\prod_{i=1}^{k} p_{\sigma_i, \ell_i} \binom{k-1}{\ell_i - 1} \right) \binom{k^2 - |\ell|_1}{j} (-1)^j x^{|\ell|_1 - k + j}. \quad (5)
\end{aligned}
$$

Interestingly, $\Pr[x \triangleright \sigma]$ is a univariate polynomial of degree $k^2 - k$. Then, the double integral can be computed analytically. The computation is tedious but straightforward.

3.2. Computing optimal type-ordering aggregation rules

The approach in Section 3.1 suggests a general way of evaluating the performance of any type-ordering aggregation rule. In order to compute the expected number of correctly recovered pairwise relations, it suffices to use equations (2), (3), and (5). Equation (5) can be used to obtain $\Pr[x \triangleright \sigma]$, which is then used in equation (3) to compute the weights (for any possible pair of types σ and σ'). Finally, equation (2) returns the expected number of correctly recovered pairwise relations.

Of course, the expected number of correctly recovered pairwise relations is not the only performance objective one would like to measure. For example, we could simply ignore exam papers that are very close in the ground truth ranking. The ground truth ranking is mostly a modelling assumption and it should not be very restrictive in the evaluation of an aggregation rule. So, we could just measure the expected number of correctly recovered pairwise relations between pairs of exam papers with ranks in the ground truth that differ by at least $a\%$ (for small values such as 5%). Another possibility would be to ignore pairwise relations between pairs of exam papers that have both very low rank in the ground truth. For example, why is it important to recover correctly the pairwise relation between the students that have true ranks 80% and 95%? A general objective in this direction would be to measure the correctly recovered relations between pairs of exam papers that involve one with true rank in the top $a\%$ (e.g., 20%).

Our theoretical framework can be easily extended to handle such cases. In general, a performance objective is defined by a bivariate function $f : [0,1]^2 \to [0,1]$ which returns the importance of measuring a correctly recovered relation between two students x and y with $x \leq y$. In the presentation of our framework in Section 3.1, we have assumed such a function with $f(x,y) = 1$ for every student pair. The two scenarios of the previous paragraph can be captured by the function (i) $f(x,y) = 1$ when $y - x \geq a\%$ and $f(x,y) = 0$ otherwise, and (ii) $f(x,y) = 1$ when $x \leq a\%$ (and $x \leq y$) and $f(x,y) = 0$ otherwise. Many other performance objectives can be defined including ones in which the function f returns fractional values between 0 and 1.

The only modification in the computation of Section 3.1 is in the computation of the weights which should become

$$
W(\sigma, \sigma') = \int_0^1 \int_x^1 f(x,y) \Pr[x \triangleright \sigma] \cdot \Pr[y \triangleright \sigma'] \, \mathrm{d}y \, \mathrm{d}x. \quad (6)
$$

In order to capture the generality of the scenarios considered, we overload the notation for the performance measure C to specify the bundle size k, the aggregation rule $\succ$, the noise matrix P describing the grading behaviour, and the performance objective f.

THEOREM 3.1. *Consider a type-ordering aggregation rule $\succ$ that is applied on k-sized partial rankings from an infinite population of students with grading behaviour that follows a noise matrix P. Then, the fraction of correctly recovered pairwise relations that satisfy the performance objective given by the bivariate function f is*

$$C(k, \succ, P, f) = \sum_{\sigma, \sigma' : \sigma \succ \sigma'} W(\sigma, \sigma') + \frac{1}{2} \sum_{\sigma} W(\sigma, \sigma), \qquad (7)$$

where $W(\sigma, \sigma')$ is given by (6) and $\Pr[x \triangleright \sigma]$ is in turn given by (5).

Note that the weights do not depend on the aggregation rule at all. They depend on the grading behaviour and the performance objective. Instead, the aggregation rule determines only the particular weights that should be summed up in order to compute $C(k, \succ, P, f)$. This means that, once we have information about the bundle size, the grading behaviour, and the desired performance objective, we can seek for the type-ordering aggregation rule that is optimal for this particular scenario. All we have to do is to compute the type-ordering aggregation rule $\succ$ that maximizes $C(k, \succ, P, f)$ which, actually, translates to computing an ordering of the types so that the leftmost summation in the definition (7) is maximized.

It is not hard to see that the problem is equivalent to solving the feedback arc set 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. The next statement should now be obvious.

THEOREM 3.2. *Computing the optimal type-ordering aggregation rule for a scenario involving specific bundle size, grading behaviour, and desired performance objective is equivalent to solving feedback arc set on an edge-weighted complete directed graph.*

Feedback arc set (FAS) is NP-hard even in its very simple variant on unweighted tournaments [Alon 2006]. The particular weighted version we consider here admits a PTAS [Kenyon-Mathieu and Schudy 2007]. Unfortunately, the solutions that such a PTAS can guarantee in reasonable time are sufficiently far from optimality and the resulting type-ordering aggregation rule will consequently have highly suboptimal performance. Fortunately, the FAS instances that we had to solve in order to compute optimal rules have a very nice structure for all the scenarios considered . This structure allows us to compute the optimal FAS solution (almost) exactly by a straightforward algorithm that we present in the following. We strongly believe that this nice property holds in *any* scenario that can appear in practice.

Let us assume that we would like to solve FAS on an edge-weighted complete directed graph $G = (V, E, w)$ and to compute an ordering of the nodes of V so that the total weight of edges in the direction that is consistent to the ordering is as high as possible. First observe that if two opposite directed edges have the same weight, the ordering of its endpoints does not affect the contribution of the consistently directed edge. So, the decision about the relative order of such *non-critical* node pairs can be postponed until the very end of the algorithm and any decision about them will be just fine. Now, consider two nodes u and v of G such that $w(u, v) > w(v, u)$; then, the consistently directed edge that we would like to have in the final solution is (u, v). We will call such pairs of nodes *critical* pairs. Decisions about the ordering of critical pairs of nodes have to be taken first. An ideal situation would be if after deciding the critical node pairs, we came up with a partial ordering of all nodes that participate in at least

one critical pair. The ordering could then be completed by appropriate decisions about non-critical node pairs. And, luckily, this process would have resulted in an optimal solution for FAS since every pair of nodes has the maximum possible contribution to the objective. Of course, things are not as easy in general since the decisions about critical pairs may lead to cycles of nodes, which cannot be part of the final ordering.

Our algorithm proceeds as follows. It takes as input an edge-weighted complete directed graph $G = (\mathcal{T}_k, E, W)$ with $\mathcal{T}_k$ as the node set and weight $W(\sigma, \sigma')$ (computed using (6)) for every directed edge from type σ to type σ'. Our algorithm builds an auxiliary unweighted directed graph $H = (\mathcal{T}_k, A)$ again over the types. For every critical pair of types σ, σ' with $W(\sigma, \sigma') > W(\sigma', \sigma)$, the auxiliary graph has a directed edge from type σ to type σ'. The next step is to compute all strongly connected components of H; two types σ and σ' belong to the same strongly connected component if H contains a directed path from σ to σ' and a directed path from σ' to σ. This computation can be easily done by computing breadth first search trees rooted at every node of H. After this step, the ordering of the types in different strongly connected components is irrevocably decided. In order to decide the ordering of types within the same strongly connected component, we use brute force on the corresponding subgraph of G. If the size of a strongly connected component is so large that brute forcing is prohibitive, we just order the types within the component according to a Borda ordering (breaking ties uniformly at random). As a final step, we decide the order of non-critical node pairs.

The approach to use Borda ordering when brute forcing is very costly in terms of running time might give the impression that the outcome of the above algorithm is always very close to a Borda ordering. Surprisingly, our algorithm returns Borda orderings (or orderings that are very close to Borda) only when this is absolutely necessary. One such situation is presented in the next section where we show that Borda is indeed the optimal aggregation rule in all scenarios that involve perfect graders. For imperfect graders, brute forcing has been proved extremely useful as the vast majority of strongly connected components are very small. We report statistical information from the size distribution of strongly connected components in Section 4.

3.3. Borda is optimal for perfect graders

We will now exploit our theoretical framework to obtain our first concrete result.

THEOREM 3.3. *For every scenario that involves perfect graders, Borda (with any tie-breaking rule) is the optimal type-ordering aggregation rule.*

PROOF. Assume that we have a scenario with a bundle size of k, perfect grading (i.e., a $k \times k$ identity noise matrix), and a bivariate function f that represents the performance objective.

We first compute the probability that exam paper x gets type σ using (4) and the fact that $p_{\sigma_i,\ell} = 1$ if $\sigma_i = \ell$ and $p_{\sigma_i,\ell} = 0$ otherwise. Hence,

$$\Pr[x \rhd \sigma] = N(\sigma) \prod_{i=1}^{k} \binom{k-1}{\sigma_i - 1} x^{\sigma_i - 1}(1-x)^{k-\sigma_i}$$

$$= N(\sigma) \left(\prod_{i=1}^{k} \binom{k-1}{\sigma_i - 1} \right) x^{k^2 - B(\sigma)}(1-x)^{B(\sigma)-k}.$$

Now, consider two exam papers with ranks x and y in the ground truth such that $x < y$ and let σ and σ' be two types. Using the above equality, we obtain

$$\frac{\Pr[x \triangleright \sigma] \Pr[y \triangleright \sigma']}{\Pr[x \triangleright \sigma'] \Pr[y \triangleright \sigma]} = \left(\frac{y(1-x)}{x(1-y)}\right)^{B(\sigma)-B(\sigma')}. \tag{8}$$

Since $y > x$, it is also $1 - x > 1 - y$ and the right hand side of the last equation is above, equal, or below 1 if and only if the quantity $B(\sigma) - B(\sigma')$ is positive, zero, or negative. Hence, the quantity

$$\Pr[x \triangleright \sigma] \Pr[y \triangleright \sigma'] - \Pr[x \triangleright \sigma'] \Pr[y \triangleright \sigma]$$

and the Borda score difference $B(\sigma) - B(\sigma')$ between the two types σ and σ' have the same sign. Now, let $\mathrm{sgn} : \mathbb{R} \to \{-1, 0, 1\}$ be the signum function. Then, we have that

$$\mathrm{sgn}\left(W(\sigma, \sigma') - W(\sigma', \sigma)\right)$$
$$= \mathrm{sgn}\left(\int_0^1 \int_x^1 f(x, y)(\Pr[x \triangleright \sigma] \Pr[y \triangleright \sigma'] - \Pr[x \triangleright \sigma'] \Pr[y \triangleright \sigma]) \, \mathrm{d}y \, \mathrm{d}x\right)$$
$$= \mathrm{sgn}\left(B(\sigma) - B(\sigma')\right)$$

This implies that any Borda ordering $\succ$ of the types maximizes the quantity $\sum_{\sigma, \sigma' : \sigma \succ \sigma'} W(\sigma, \sigma')$ and, consequently, the quantity $C(k, \succ, I, f)$, and the theorem follows. $\square$

The statement of Theorem 3.3 is rather surprising as Borda is among the simplest type-ordering aggregation rules. For example, when $k = 6$, Borda classifies the exam papers into only 31 different levels (based on their Borda scores) while there are type-ordering rules that exploit a more refined classification of the papers into 462 different levels; the gap is much higher for larger values of k. Theorem 3.3 essentially says that this extra power is not at all necessary and Borda is always a better choice when perfect grading is used.

4. VALIDATION OF OUR FRAMEWORK

4.1. Building a realistic noise model using a field experiment

We have run a field experiment with the students that attended the course on Computational Complexity in our home institution during the Spring 2015 semester. This is a course that the first author teaches during the last few years and usually includes an optional mid term exam. As it is typically the case in Greek universities, cardinal integer and half-integer scores between 0 and 10 are used in such exams and they represent how correct the answers of the students to the exam questions are. Hence, these cardinal scores represent the success of the students in the exam in absolute terms.

In our experiment, our goal has been to investigate how effective the students can be in ordinal grading. For this purpose, we created a hypothetical exam with three questions and prepared several answers for them. In particular, we prepared 16 different answers to question 1, 12 answers to question 2, and 8 answers to question 3. Combinations of these answers into all different ways resulted in a pool of 1536 different exam papers. We created bundles of size 6 from this pool. Each student was given a bundle of exam papers which was asked to rank (for a bonus grade). Note that the selection of papers in each bundle was not arbitrary. The answers for the questions belonged to different levels of correctness and included excellent ones, almost excellent ones with a minor issue not fully resolved, answers in the right direction but with sloppy write-up, completely incorrect answers, no answer at all, etc. Specifically, we had 7, 6, and 5 different levels of correctness for the answers in questions 1, 2, and

3, respectively. When bundles were formed, we imposed the following constraint for any pair of exam papers A and B in a bundle: if the correctness level of paper A in an answer is strictly higher than that in paper B, then paper B cannot have a strictly higher correctness level than A in any other answer. Furthermore, there was at least one question for which the answers had different levels of correctness. This guaranteed a strict ranking of the exam papers in each bundle and, furthermore, that this ranking would be well-defined and independent of any assumptions about the importance of the different questions.

In addition to this ranking exercise, the students also participated in the traditional mid term exam. This allowed us to quantify the correlation between their grading behaviour and their success in the traditional exam. The results are depicted in Figure 2(a). Even though students that were excellent in the traditional exam did very well as graders, the grading performance of the majority of the remaining students seems to be uniformly distributed between average and excellent, with just a few under-performing outliers.[2]
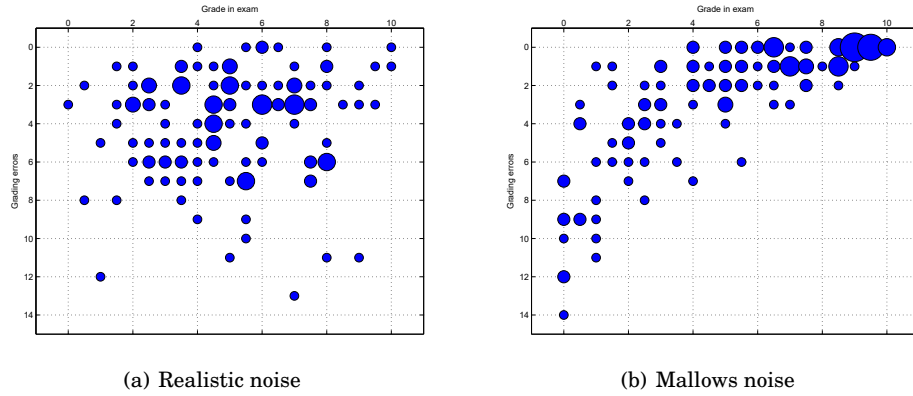

(a) Realistic noise   (b) Mallows noise

Fig. 2. Correlation between (cardinal) grade of students in the traditional exam and grading error (Kentall-tau distance from the correct ranking). Data refer (a) to the 136 students that participated in our study and (b) to 136 random students/graders drawn from the Mallows distribution. Each bubble corresponds to a number of students that is proportional to its area.

For comparison, we have plotted the same small number of randomly chosen students according to the Mallows distribution in Figure 2(b). Here, in contrast to Figure 2(a), the correlation between student quality (to be thought of as equivalent to the success in the traditional exam) and grading performance is clear. The data depicted in Figure 2(a) have been used extensively in "realistic" exams. In these exams, we simulate a large population of students, whose quality and grading behaviour is determined by randomly drawing a bubble from those in Figure 2(a) with probability proportional to the area of the bubbles. Essentially, each of our 136 students in the experiment serves as the support of the realistic distribution while the quality is slightly perturbed to result in a strict ground truth ranking. The above information has also been distilled in the noise matrix $P_{\text{real}} = (p_{i,j})_{i,j \in [k]}$ with

---

[2]An explanation for this grading behaviour is that, even though the students have participated in many exams like the mid term in the past and have a very good idea of what they are expected to do, this was the very first time they were asked to rank.

$$P_{\text{real}} = \begin{bmatrix} 0.463 & 0.257 & 0.102 & 0.058 & 0.058 & 0.058 \\ 0.205 & 0.316 & 0.227 & 0.110 & 0.066 & 0.073 \\ 0.161 & 0.191 & 0.257 & 0.205 & 0.132 & 0.051 \\ 0.102 & 0.117 & 0.191 & 0.242 & 0.279 & 0.066 \\ 0.044 & 0.066 & 0.139 & 0.220 & 0.301 & 0.227 \\ 0.022 & 0.051 & 0.080 & 0.161 & 0.161 & 0.522 \end{bmatrix} \tag{9}$$

which represents a "realistic" noise model. The information in the matrix was obtained by measuring the frequency that the i-th ranked exam paper by students should be correctly ranked at position j. For example, 28 out of 136 students ranked third an exam paper in their bundles which should have been ranked fourth; thus, $p_{3,4} = 0.205$. This matrix is used in the computation of the optimal type-ordering aggregation rules for realistic scenarios according to the methodology developed in Sections 3.1 and 3.2. Observe that matrix P_{real} does not include any information about the correlation between the grading behaviour and the quality of the student that acts as grader; actually, this is a feature that our framework completely neglects. Surprisingly, the experimental results that we present in the following indicate that storing information about this relation is not necessary for making accurate performance predictions.

4.2. On the accuracy of theoretical performance predictions

Our next step was to compute the optimal type-ordering aggregation rules for several scenarios. We have kept the same bundle size of $k = 6$ in all scenarios and distinguish between the realistic and Mallows[3] noise models by using the corresponding matrices P_{real} and P_{mallows} defined in equations (9) and (1), respectively. As performance objectives, we have considered the following:

— all2all: the total number of all correctly recovered pairwise relations;
— th-10% and th-50%: the total number of correctly recovered relations between pairs that include an exam paper that is ranked in the top 10% and top 50% in the ground truth, respectively;
— acc-2% and acc-5%: the total number of correctly recovered relations between pairs with positions that differ by at least 2% and 5% in the ground truth, respectively.

We have used the theoretical framework presented in Sections 3.1 and 3.2; of course, computations have been automated.[4] A first remark is that the algorithm that we have used to solve FAS exactly is very fast. This is due to the fact that strongly connected components have very small size. In all cases we considered, among the 462 different types that we can have for bundles of size 6, more than 97% of them form singleton components and the maximum component size never exceeded 26 (for the 100-sample approximation of Mallows that we discuss in Section 4.3). Their distribution for scenarios with realistic and Mallows noise models for all performance objectives is depicted in Table I.

Recall that our theoretical framework assumes an infinite number of students and, hence, neglects apparent dependencies between random variables that certainly exist in scenarios with finite number of students. So, the information in Table II is rather

---

[3]Notice that our framework could easily host any non-Mallows synthetic noise matrix, e.g., a noise matrix following some generalized random utility model (see [Azari et al. 2012]).

[4]All our computational results have been obtained using an Intel 12-core i7 machine with 32Gb of RAM running Windows 7. Our methods have been implemented in C using the GNU Multiple Precision Arithmetic Library (GMP) and in Matlab R2013a. High precision is absolutely necessary in order to compute the weights even for bundles of size 6 since, by inspecting equations (5) and (6) carefully, we can see that the computations involve products with more than 30 factors and factorials of integers up to 30.

Table I. Distribution of the size of strongly connected components. Results for th-10% are not shown since all strongly connected components are singletons for such scenarios.

size	realistic model				mallows model			
	all2all	th-50%	acc-2%	acc-5%	all2all	th-50%	acc-2%	acc-5%
1	448	460	449	451	453	459	449	449
3–7	13	2	12	10	6	3	10	12
8–11	1	0	1	1	2	0	2	0
≥ 12	0	0	0	0	1	0	1	1
max	10	3	10	10	20	4	20	20

surprising and shows that, besides our assumptions, our theory provides extremely accurate predictions for the performance of aggregation rules in practice. Note that the values in Table II are percentages and we never observed differences beyond the second decimal point between the theoretically predicted value and the experimental one. Also, note that we have used $10\,000$ students in our experiments. This number is lower than the vision for the most popular courses that will be offered by MOOCs in the near future; the predictions become even more accurate for higher numbers of students.

Table II. Performance of Borda and optimal type-ordering aggregation rules for scenarios with perfect, realistic, and Mallows grading with respect to the five different objectives. The values presented are theoretical predictions (theory) and experimental measurements with $10\,000$ students ($n = 10^4$).

noise	perfect grading		realistic grading				mallows grading			
setting	theory	$n = 10^4$	theory		$n = 10^4$		theory		$n = 10^4$	
method	borda	borda	opt	borda	opt	borda	opt	borda	opt	borda
all2all	92.01	92.02	80.01	79.57	80.09	79.57	85.15	84.38	85.16	84.39
th-10%	96.94	96.95	87.61	87.18	87.60	87.17	92.05	90.52	92.07	90.54
th-50%	94.13	94.14	83.62	83.43	83.62	83.43	88.39	87.80	88.40	87.81
acc-2%	93.57	93.57	81.27	80.73	81.27	80.74	86.52	85.72	86.52	85.73
acc-5%	95.47	95.47	82.97	82.42	82.97	82.42	88.42	87.61	88.42	87.62

Table II contains the expected values (from 1000 simulations) of the performance measure for each grading scenario and corresponding optimal aggregation rule (as well as for Borda). Even though Borda is never optimal in any imperfect grading scenario that we considered, its performance is always very close to the optimal rule. In particular, Borda is never more than 0.55% worse than the optimal rule in realistic scenarios (for the acc-5% performance objective); such differences can be up to 1.53% in scenarios with Mallows noise (and the th-10% objective). Figure 3 reports detailed information for all these simulations. Clearly, the performance of the aggregation rules for all the objectives that we considered is sharply concentrated around the expected values. The clouds of points are very close to the diagonal (corresponding to values with equal coordinates) in the realistic scenarios and only marginally further from it in the Mallows scenarios.

4.3. The effect of inaccuracies in the noise model

The realistic noise model that we built in Section 4.1 is, by definition, an approximation of the students in our home institution. Besides limitations that have to do with our modelling assumptions, it has the obvious drawback that it has been built using a very small fraction of our students. So far, the reader should have been convinced that the aggregation rules we have built are indeed optimal for a large population that inherits the quality and grading performance of this small fraction of students; this has been the focus of our experiments with realistic grading. What is far from clear is whether these aggregation rules will perform equally well for the whole population of
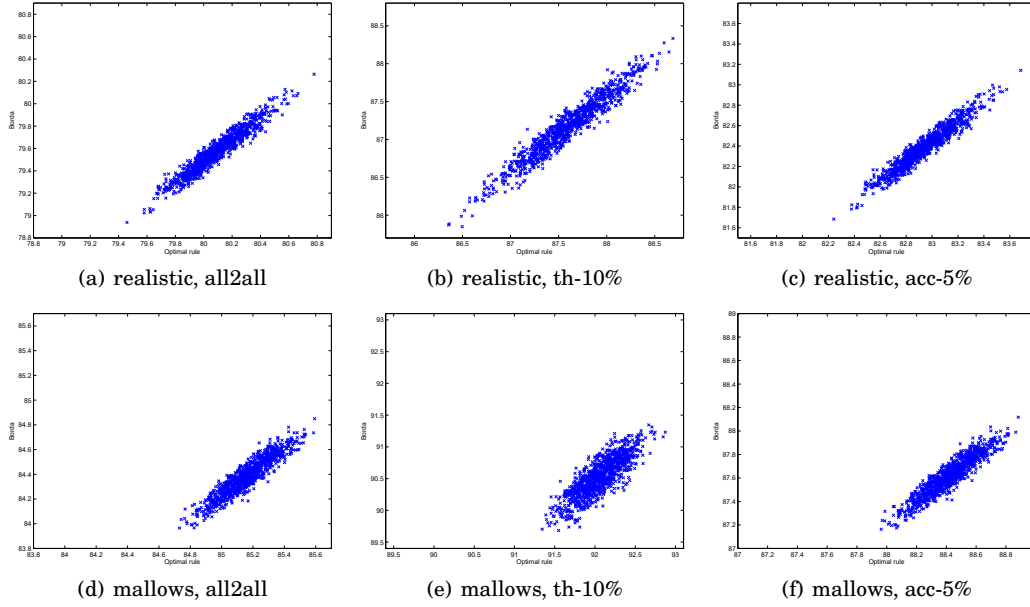
Fig. 3. Performance of Borda compared to the optimal rule for the realistic and Mallows noise scenarios and the performance objectives all2all, acc-5%, and th-10%.

the students in our home institution. To see the importance of this question, imagine it in the planetary scale that MOOCs envision. Can we make safe predictions for huge student populations by sampling a tiny fraction of them, building a noise model as we did in Section 4.1, and then select optimal type-ordering aggregation rules as we did in Section 4.2?

We give a positive answer to this question by considering Mallows grading scenarios. With Mallows, we have the luxury of a well-defined noise model for the grading behaviour of a huge student population and we have used it in order to compute optimal aggregation rules for this model. This information will be used only for assessing the approach presented in the following. Instead, we will pretend that no information about grading behaviour is available and all we can do is to apply (actually, to simulate) a field experiment like the one we presented in Section 4.1 on a tiny fraction of the students in order to come up with a noise matrix. In this way, we will compute an approximation of the true noise model.

We have followed this approach using samples of Mallows graders of size 100 and 1000. The two noise model matrices we obtained are as follows:

$$
P_{100} = \begin{bmatrix}
0.59 & 0.19 & 0.07 & 0.08 & 0.06 & 0.01 \\
0.19 & 0.44 & 0.18 & 0.09 & 0.04 & 0.06 \\
0.10 & 0.19 & 0.43 & 0.19 & 0.07 & 0.02 \\
0.05 & 0.05 & 0.15 & 0.45 & 0.19 & 0.11 \\
0.06 & 0.10 & 0.09 & 0.14 & 0.46 & 0.15 \\
0.01 & 0.03 & 0.08 & 0.05 & 0.18 & 0.65
\end{bmatrix}
\quad
P_{1000} = \begin{bmatrix}
0.639 & 0.186 & 0.066 & 0.058 & 0.031 & 0.020 \\
0.193 & 0.534 & 0.150 & 0.055 & 0.032 & 0.036 \\
0.073 & 0.149 & 0.501 & 0.147 & 0.076 & 0.054 \\
0.039 & 0.075 & 0.155 & 0.497 & 0.147 & 0.087 \\
0.033 & 0.038 & 0.071 & 0.163 & 0.517 & 0.178 \\
0.023 & 0.018 & 0.057 & 0.080 & 0.197 & 0.625
\end{bmatrix}
$$

The matrices have been used to compute the optimal type-ordering aggregation rules for the five performance objectives. Interestingly, the instances of FAS that we had to solve were slightly harder now, particularly for the 100-sample noise model, where strongly connected components of size up to 26 emerged in the auxiliary graph. Still, our methodology was applied smoothly and allowed us to compute optimal rules.

Table III. Perfomance of the optimal type-ordering aggregation rules for approximations of the Mallows model. The data for Mallows are presented again here for direct comparison.

# samples	100		1000		Mallows	
setting	theory	$n = 10^4$	theory	$n = 10^4$	theory	$n = 10^4$
all2all	84.95	84.95	85.14	85.15	85.15	85.16
th-10%	91.82	91.85	92.05	92.04	92.05	92.07
th-50%	88.21	88.21	88.39	88.38	88.39	88.40
acc-2%	86.31	86.31	86.51	86.51	86.52	86.52
acc-5%	88.19	88.20	88.41	88.41	88.42	88.42

Table III shows the theoretical prediction and observed performance values of these rules (by applying the rules on 1000 simulated exams with 10 000 Mallows students). The performance of the aggregation rules that were computed using the 100-sample approximation of Mallows are already amazingly close to those for Mallows. For the rules that we computed using the 1000-sample approximation, it is almost impossible to distinguish them from the Mallows-optimal ones in terms of performance.

A more refined graphical representation of these findings is given in Figure 4 (best viewed in color). Each plot contains a blue and a red cloud of 1000 points each corresponding to a single simulated exam with 10 000 students. The blue points (respectively, red points) measure the performance of the optimal rule for the 1000-sample (respectively, 100-sample) Mallows approximation versus the Mallows-optimal rule. The blue cloud almost coincides with the diagonal in each plot, indicating an optimal approximation of the Mallows-optimal rule. The red cloud is distinct (there is a negligible "intersection" of blue and red points only for the th-10% performance objective) but very close. To realize how close the two clouds are, almost the whole cloud of points for Borda (from Figures 3(d), 3(e), and 3(f)) would be located outside the plot area of Figure 4 (if we attempted to plot it).

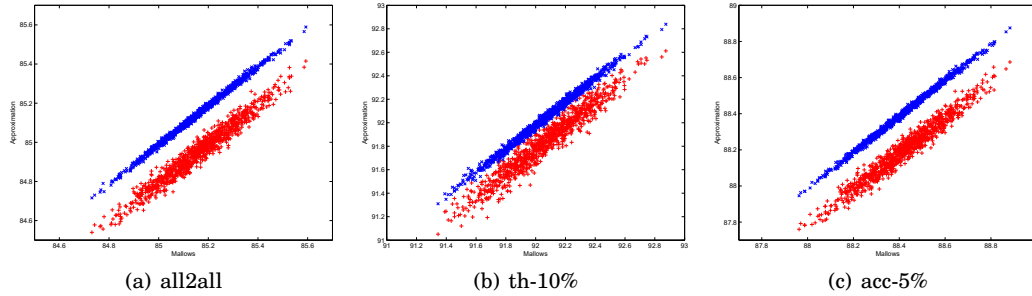(a) all2all   (b) th-10%   (c) acc-5%

Fig. 4. A comparison of the optimal rule for Mallows and its approximations with respect to the objectives (a) all2all, (b) acc-5%, and (c) th-10% (experimental results from 1000 executions of the aggregation rules on 10 000 students).

REFERENCES

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

H. Azari, D. Parkes, and L. Xia. 2012. Random utility theory for social choice. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*. 126–134.

F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

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

I. Caragiannis, G. A. Krimpas, and A. A. Voudouris. 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.

I. Caragiannis, A. D. Procaccia, and N. Shah. 2013. When do noisy votes reveal the truth?. In *Proceedings of the 14th ACM conference on Electronic commerce (EC)*. 143–160.

I. Caragiannis, A. D. Procaccia, and N. Shah. 2014. Modal ranking: A uniquely robust voting rule. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 616–622.

F. Chierichetti and J. M. Kleinberg. 2014. Voting with limited information and many alternatives. *SIAM J. Comput.* 43, 5 (2014), 1615–1653.

V. Conitzer and T. Sandholm. 2005. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 145–152.

L. de Alfaro and M. Shavlovsky. 2014. CrowdGrader: A tool for crowdsourcing the evaluation of homework assignments. In *SIGCSE*. 415–420.

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

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

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

E. Law and L. von Ahn. 2011. *Human Computation*. Morgan & Claypool.

T. Lu and C. Boutilier. 2011. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 145–152.

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

L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. The PageRank citation ranking: bringing order to the web. Stanford InfoLab technical report. (1999).

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

M. Pivato. 2013. Voting rules as statistical estimators. *Social Choice and Welfare* 40(2) (2013), 581–630.

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

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

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

L. Xia. 2014. Statistical properties of social choice mechanisms. In *Proceedings of the 5th International Workshop on Computational Social Choice (COMSOC)*.

H. P. Young. 1988. Condorcet's theory of voting. *The American Political Science Review* 82(4) (1988), 1231–1244.