A Factorization Approach to Evaluate Open-Response Assignments in MOOCs using Preference Learning on Peer Assessments

Oscar Luaces[a,*], Jorge Díez[a], Amparo Alonso-Betanzos[b], Alicia Troncoso[c], Antonio Bahamonde[a]

[a] *Artificial Intelligence Center*
University of Oviedo
33204 Gijón, Spain
[b] *Dept. of Computer Science*
Faculty of Informatics
University of A Coruña
15071 A Coruña, Spain
[c] *Dept. of Computer Science*
Pablo de Olavide University
41013 Sevilla, Spain

Abstract

Evaluating open-response assignments in Massive Open Online Courses is a difficult task because of the huge number of students involved. Peer grading is an effective method to address this problem. There are two basic approaches in the literature: cardinal and ordinal. The first case uses grades assigned by student-graders to a set of assignments of other colleagues. In the ordinal approach, the raw materials used by grading systems are the relative orders that graders appreciate in the assignments that they evaluate. In this paper we present a factorization method that seeks a trade-off between cardinal and ordinal approaches. The algorithm learns from preference judgments to avoid the subjectivity of the numeric grades. But in addition to preferences expressed by student-graders, we include other preferences: those induced from assignments with significantly different average grades. The paper includes a report of the results obtained using this approach in a real world dataset collected in 3 Universities of Spain, A Coruña, Pablo de Olavide at Sevilla, and Oviedo at Gijón. Additionally, we studied the sensitivity of the method with respect to the number of assignments graded by each student. Our method achieves similar or better scores than staff instructors when we measure the discrepancies with other instructor's grades.

*Corresponding author: Tel: +34 985 182 028
Email addresses: oluaces@uniovi.es (Oscar Luaces), jdiez@uniovi.es (Jorge Díez), ciamparo@udc.es (Amparo Alonso-Betanzos), atrolor@upo.es (Alicia Troncoso), abahamonde@uniovi.es (Antonio Bahamonde)

Keywords: Peer Grading, Factorization, Preference Learning, Ordinal and Cardinal approaches, MOOCs

1. Introduction

Massive Open Online Courses (MOOCs) have attracted thousands of students from many parts of the world. These courses make university lectures available online and most of the times are free or have a very low cost. It is possible to find MOOCs of almost any kind of subject; even for highly experimental subjects, where students can make real experiments with simulated materials (microscopes, etc) or can take real data from remote controlled devices [1]. The success of MOOCs is based on the fact that several providers are spin-off from the most reputable Universities. This is the case, for instance, of Academic Earth, Coursera, edX, Khan Academy or Novoed.

The evaluation is an important part of the teaching process, and has to be addressed in order to make MOOCs useful to provide a feedback to students and to guarantee the quality of the titles given to graduates. The challenge is to evaluate a very large number of assignments that cannot be automatically evaluated in all cases. In some cases, the assignments include open-response (open-ended) questions whose evaluation requires human intervention.

Since it is unpractical that an instructor evaluates all the assignments, or even a set of Teaching Assistants (TA), researchers have been exploring the possibilities of using methods previously employed in Journals or Conferences: *peer grading* or *peer assessment* [2, 3, 4, 5, 6, 7]. The students of the course are asked to evaluate a small set of anonymized assignments submitted by other students. Additionally, these student-graders receive a set of detailed instructions (called *rubric*) in order to uniform the assessment. However, students typically have no experience in this task and then effective peer grading must deal with the effects of inconsistent subjective evaluation.

Roughly speaking there are two kinds of methods to use peer grading in practice: *cardinal* and *ordinal* approaches. In cardinal peer grading, each grader returns a cardinal-valued assessment for each assignment. Grades are then numbers or categorical labels with a straightforward numerical semantics. This type of grades is really very useful since they transmit an accurate assessment. Graders classify the assignments in an ordered scale that it is hopefully assumed universally. The final grade given to an assignment is usually determined as the average (or median) of the corresponding peer-grades [3] given by the evaluators.

However, the cardinal approach has severe shortcomings. Firstly, we have to expect that assessments will be affected by some graders' bias that would deviate them with respect to the *ideal ground truth*. The presumed universality of the semantics of cardinal grades is not so general. Some students tend to give very high grades, while others (probably with different academic backgrounds) are less generous with their assessments. A way to overcome this problem is to get many grades for each assignment; then, the *correct* grade could be approximated

averaging all available grades. Moreover, assuming that each assignment was graded by a big amount of students, it has been reported [4] that averages are more consistently accurate with respect to the rubric than the staff grades. We have confirmed this fact in the experiments reported at the end of the paper. But unfortunately, we only can obtain a reduced number of assessments for each assignment.

In addition to the existence of different scales, there is a second shortcoming that has to be addressed in cardinal approaches: the *batch effect*. It has been observed [8, 9, 10] that an item tends to receive a higher score when it is evaluated in a batch of worse items than when it is evaluated in a group of better items. Fortunately, despite the graders' biases, the ranking entailed by their assessments is coherent with the ground truth. In other words, the grades can be unreliable but the order is, in general, correctly assessed [11, 8, 9, 2].

The basic axiom of *Ordinal* peer grading methods is that the essential knowledge provided by inexpert graders is the ordering of their bunch of assignments. Notice that these orderings indicate a relative but not absolute quality assessment. Thus, ordinal approaches are committed to take advantage of the robustness of ordering against the alleged accuracy of cardinal grades.

In this paper we present an approach that tries to combine the strengths of both views. The student-graders are asked to give cardinal grades. The whole set of evaluations of each grader is considered as a relative ordering; from them we obtain a set of pairs of preferences (also called preference judgments or comparisons). In this way we avoid ties: we only consider pairs of assignments with different grades. But in addition to student-graders we include a new grader, we call it the *Gaussian grader*. This grader compares assignments with *significantly* different average grades.

We would like to emphasize the role of the artificial Gaussian grader. On the one hand, its preferences include somehow a calibration in the pure ordinal approach. The idea is that some assignments are clearly better than others. Then their comparison has to be present in the learning task, even if the assignments involved have not had any common grader.

On the other hand, the comparisons induced by the Gaussian grader make harder the possibility of having islands in the grading network [4]. Notice that any ordinal grading algorithm needs to have each assignment connected to the rest through a chain of comparisons, otherwise the grades of each connected component are unrelated with the others.

Finally, with the set of preference pairs of all graders we learn an *utility* function that returns higher values to better assignments. This function can be learned using preference learning methods like those presented by Herbrich et al. [12] or by Joachims [10]. In this paper, as it was done in the work by Díez et al. [7], we use a factorization method [13] to learn the utility function. This function estimates the consensus ordering of the assignments computing a ranking that can be easily transformed into a grading function for each assignment.

An additional advantage of this approach is that it is straightforward to include additional features of the assignments or students involved in the evaluation process. Both assignments and graders are represented by feature vectors.

If no other information is available, these features are just a binary codification of their identity. But the representation can be enriched with any information about the characteristics of the assignment, or previous academic history of the students. About this issue we have to be aware of ethic considerations, for instance not including features like gender or ethnicity; see [4]. But additional features may be crucial for the success in some cases; see [14] and [15].

Other approaches perform preference learning by taking into account the degree of preference of one item over another, as the work of Wang and Fan [16]. This method starts from preference matrices given by users, where they express their degrees of preference with respect to a pair of products. Thus, it is, to some extent, a cardinal approach in contrast to ours, which is mainly an ordinal approach that takes into account cardinal differences only when they are statistically significant. Despite the drawbacks of cardinal approaches discussed previously, this method is also hindered by a common characteristic of the peer assessing tasks: each grader evaluates only a few assignments so the preference matrices given by the graders will be very sparse. The consensus matrix, computed as a weighted sum of the preference matrices, will be nearly zero and the underlying ranking will hardly be better than any other chosen at random.

In the rest of the paper we review some related work, and then we make a formal presentation of our approach. The paper is closed with the report of the results obtained with two real word dataset obtained from a common assignment for Computer Science students of 3 Universities of Spain: A Coruña, Pablo de Olavide at Sevilla, and Oviedo at Gijón. Each dataset has 1327 grades given by 160 students to 175 assignments.

The experiments compare the performance of the factorization method proposed here with a baseline that simply averages the grades given to each assignment. The comparison is established against the grades given by the staff instructors of each university, that assessed the whole set of assignments, not only those of their own Universities. Additionally, the experimental section presents a study of the sensitivity of the method with respect to the number of assignments that graders received to evaluate. We observe that the scores improve with this number, and that our method achieves similar or better scores than staff instructors when we measure the discrepancies with other instructor's grades.

2. Related Work

There are a number of related work in this area; some of them have just been mentioned in the Introduction. However, probably the most similar work is reported by Raman and Joachims [6]. In this paper, the authors present a case study with real data from a Cornell University course. The assignments are 42 posters and 44 reports done by groups of students. Each poster received and average of 23.71 grades, while reports received 13.32 grades. The authors propose to use the ordinal approach casting the learning problem as a *rank aggregation* learning task.

The paper compares the performance of several probabilistic aggregation algorithms and acknowledges that *simply averaging the cardinal scores of the peer graders performs surprisingly well*. Probably the reason is that each assignment receives a high number of grades in coherence with results reported for instance by Piech et al. [4].

A very interesting result reported by Raman and Joachims [6] is that the accuracy of the models learned are compared with the rankings achieved by a set of TAs. The conclusion is that *there is no evidence that TA grading would have led to more accurate grading outcomes than peer grading*.

Another probabilistic learning algorithm has been proposed for peer grade estimation in [4]. The paper presents also a case study with 63199 peer grades of a Coursera course about Human Computer Interaction. The method proposed requires self-grading of the students and the evaluation of some assignments that were previously graded by the instructor in order to estimate grader reliability. These constraints were not included in the experiments reported in this paper.

Both papers, [4, 6], emphasize the relevance of assessing the accuracy of graders. In fact, it is crucial to incentive students to make a good evaluation if we want to obtain reliable data. A way to do this is to include the assessment of students as graders as part of their final grade. On the other hand, we think that the evaluation process itself may be an additional way for students to get insight into the field covered in the assessment.

Shah et al. [5] use also a dataset of Human Computer Interaction (HCI) on Coursera; in this case from the third offering of the course. The dataset used has assignments submitted by 1879 students, and 7242 numerical grades were collected by a peer grading experiment.

The authors acknowledge as a desideratum to seek for a *trade-off between the precision of cardinal scores and the robustness of ordinal evaluations for peer grading*. The computational method proposed in the paper is an ordinal approach that searches for the solution of a non-convex optimization problem that uses a logistic sigmoid. The experiments reported achieve a performance similar to the performance of a method that simply computes the median of the grades given to each assignment.

Another relevant paper about peer-grading in MOOCs was also written by Shah et al. [15]. The authors present formal proofs about the errors in peer-gradings when the grade is estimated averaging the grades given by student-graders. There is a constant proportion of assignments erroneously graded. The amount of assignments may become too high in MOOCs, and therefore the procedure is unacceptable.

The proposal is to use methods that include some kind of *dimensionality reduction*; in particular, the authors discuss clustering and featuring. Although the proposals are very abstract, it is interesting to underscore that the factorization method proposed in this paper is a suitable framework to implement both approaches.

There is other kind of approaches to obtain computed aided assessments. The focus is on providing an automatic feedback to students that have just written a free-text answer in an online course. The general idea is to compare

students' answers with some reference answers provided by the instructors. The comparison method is borrowed from the field of information retrieval. Rodrigues and Oliveira [17] compute a cosine similarity after a preprocess. A modification of the BLEU algorithm [18] is used by Noorbehbahani and Kardan [19]. An ensemble of Latent Semantic Analysis (LSA) and n-gram co-ocurrences is the method proposed by He et al. [20]. In turn, Pérez et al. [21] propose a combination between BLEU and LSA.

Probably the use of information retrieval methods with a shallow natural language processing would improve the approach presented in this paper. However, the peer-assessment strategy has some pedagogical advantages since it requires an additional reflexion of students about the contents of the assignments. Thus, we think that it is valuable to explore this approach.

Finally, let us recall that factorization approaches have been used widely since their success in recommender systems. The Netflix price was won by a team that proposed a factorization algorithm, [22]. The scalability of these algorithms was emphasized in [23]. However, recommenders was not the only application tackled with a factorization approach; in [24], the authors presented an algorithm for dimensionality reduction in Machine Learning. All these application fields underscore the advantages of factorization versus other alternative approaches in solving some learning tasks.

3. Learning Method

Let $\mathcal{G}_{PJ}$ be a set of *graders* that provide their *preference judgments*, and let $\mathcal{A}$ be a set of *assignments*. Each grader g has received a subset $D_g \subset \mathcal{A}$ of assignments to evaluate. The initial data to infer a grading function is the collection of grades given by graders:

$$g(i) \in [0, 10], \forall g \in \mathcal{G}_{PJ},\ i \in D_g. \tag{1}$$

From these data we build a set of preference judgments $\mathcal{D}_{PJ}$ given by triples of a grader, and a pair of assignments evaluated by the grader and ordered by its grade. In symbols,

$$\forall g \in \mathcal{G}_{PJ}, \forall i, j \in D_g, g(i) > g(j) \Rightarrow [g, i, j] \in \mathcal{D}_{PJ}. \tag{2}$$

Pairs of assignments with the same grade are not used to build a preference judgment, since only those assignments with different grades are useful to infer a ranking. For example, if grader g assesses 4 assignments, p, q, r and s, giving the following grades: $g(p) = 7$, $g(q) = 5$, $g(r) = 9$, and $g(s) = 5$, we will construct his/her preference judgments as $\{[g, p, q], [g, r, p], [g, p, s], [g, r, q], [g, r, s]\}$. Notice that there is no pair including q and s since neither one is better than the other.

The first step to learn a model for peer grading consists in adding some preference judgments. For this purpose we invent a new grader called in the following the *Gaussian grader*, Gg. This artificial grader considers pairs of assignments (i, j) such that the average of grades received by the first, μ_i, minus

the typical deviation σ_i is greater than the average of j grades plus the corresponding deviation. In symbols, we are going to consider the triple $[Gg, i, j]$ whenever

$$(\mu_i - \sigma_i) > (\mu_j + \sigma_j).$$

Then, we define an additional set of preference judgments with two more triples for each assignment fetched from Gg judgments, we call this set $\mathcal{D}_{Gg}$. Exactly one assignment better (if exists) and one assignment worse (if exists).

$$\begin{aligned} &\forall i \in \mathcal{A}, \exists j_{-1}, (\mu_i - \sigma_i) > (\mu_{j_{-1}} + \sigma_{j_{-1}}) \\ &\qquad\qquad \Rightarrow [Gg, i, j_{-1}] \in \mathcal{D}_{Gg}, \qquad (3) \\ &\forall i \in \mathcal{A}, \exists j_{+1}, (\mu_{j_{+1}} - \sigma_{j_{+1}}) > (\mu_i + \sigma_i) \\ &\qquad\qquad \Rightarrow [Gg, j_{+1}, i] \in \mathcal{D}_{Gg}. \qquad (4) \end{aligned}$$

After this step we have a dataset given by

$$\mathcal{D} = \mathcal{D}_{PJ} \cup \mathcal{D}_{Gg}. \qquad (5)$$

To complete the step, we add Gg as a member of the set of graders $\mathcal{G}$:

$$\mathcal{G} = \mathcal{G}_{PJ} \cup \{Gg\}.$$

The next step in the learning process is to map both assignments and graders into a common Euclidean space

$$\begin{aligned} \phi_g(g) : \mathcal{G} \to \mathbb{R}^k, \quad g \mapsto \boldsymbol{W}_g, \\ \phi_a(i) : \mathcal{A} \to \mathbb{R}^k, \quad i \mapsto \boldsymbol{V}_i. \end{aligned} \qquad (6)$$

From this dataset we could try to learn a utility function defined as:

$$\begin{aligned} f(g, i) &= -||\phi_g(g) - \phi_a(i)||^2 = \\ &= -\langle \phi_g(g) - \phi_a(i), \phi_a(g) - \phi_a(i) \rangle \\ &= -\left(\langle \boldsymbol{W}_g, \boldsymbol{W}_g \rangle + \langle \boldsymbol{V}_i, \boldsymbol{V}_i \rangle - 2\langle \boldsymbol{W}_g, \boldsymbol{V}_i \rangle \right) \\ &= 2\boldsymbol{W}_g{}^T \boldsymbol{V}_i - \boldsymbol{W}_g{}^T \boldsymbol{W}_g - \boldsymbol{V}_i{}^T \boldsymbol{V}_i. \end{aligned} \qquad (7)$$

This function is an estimation of a grade normalized for every grader and every assignment. Somehow, this utility function allows us to complete the assessment matrix. From a geometric point of view, $f(g, i)$ is the opposite of the distance from the Euclidean representation of the grade g and the assignment i. Nearer representations mean higher assessments.

We define the *final grade* for each assignment as the average grade given by every grader. In symbols, we extend the definition of the utility function f as

follows.

$$
\begin{aligned}
f(\mathcal{G}, i) &= -\frac{1}{|\mathcal{G}|} \left|\left| \sum_{g \in \mathcal{G}} \phi_g(g) - \phi_a(i) \right|\right|^2 \\
&= -\frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \langle \phi_g(g) - \phi_a(i), \phi_g(g) - \phi_a(i) \rangle \\
&= -\frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \Big(\langle \boldsymbol{W}_g, \boldsymbol{W}_g \rangle + \langle \boldsymbol{V}_i, \boldsymbol{V}_i \rangle - 2 \langle \boldsymbol{W}_g, \boldsymbol{V}_i \rangle \Big) \\
&= \frac{2}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} {\boldsymbol{W}_g}^T \boldsymbol{V}_i - \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} {\boldsymbol{W}_g}^T \boldsymbol{W}_g - {\boldsymbol{V}_i}^T \boldsymbol{V}_i. \qquad (8)
\end{aligned}
$$

The aim of the learning process devised to make the assessment is to optimize the mapping functions (6) in such a way that the final grades given by the average (8) be as coherent with graders orderings as possible. We will follow a maximum margin approach. Then, we define

$$
err(\boldsymbol{W}, \boldsymbol{V}) = \sum_{[g,i,j] \in \mathcal{D}} \max(0, 1 - f(\mathcal{G}, i) + f(\mathcal{G}, j)). \qquad (9)
$$

We use matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ to collect the columns used in mappings (6). The idea is to ensure that the difference of grades is at least 1. To learn these parameters we use a Stochastic Gradient Descent (SGD in the following) algorithm to minimize the previous equation adding a regularization term

$$
\underset{\boldsymbol{W}, \boldsymbol{V}}{\operatorname{argmin}} \ \left(err(\boldsymbol{W}, \boldsymbol{V}) + \nu r(\boldsymbol{W}, \boldsymbol{V}) \right). \qquad (10)
$$

In this case, to implement regularization we use the square of the Frobenius norm.

$$
\begin{aligned}
r(\boldsymbol{W}, \boldsymbol{V}) &= \left\| \boldsymbol{W} \right\|_F^2 + \left\| \boldsymbol{V} \right\|_F^2 \\
&= \sum_{g \in \mathcal{G}} \left\| \boldsymbol{W}_g \right\|^2 + \sum_{i \in \mathcal{A}} \left\| \boldsymbol{V}_i \right\|^2. \qquad (11)
\end{aligned}
$$

The parameters of the model are updated in the SGD as follows:

$$
\begin{aligned}
\boldsymbol{W}_g &\leftarrow \boldsymbol{W}_g - \gamma \left(\frac{\partial err}{\partial \boldsymbol{W}_g} + \nu \frac{\partial r}{\partial \boldsymbol{W}_g} \right) \\
\boldsymbol{V}_a &\leftarrow \boldsymbol{V}_a - \gamma \left(\frac{\partial err}{\partial \boldsymbol{V}_a} + \nu \frac{\partial r}{\partial \boldsymbol{V}_a} \right). \qquad (12)
\end{aligned}
$$

Let us remark that each $[g, i, j] \in \mathcal{D}$ fetched by the SGD gives rise to an update of the representation of grader g ($\boldsymbol{W}_g$), and of the assignments i and j ($\boldsymbol{V}_i, \boldsymbol{V}_j$).

These updates are carried out only when the margin is violated, see (9). The partial derivatives needed can be built from:

$$\begin{aligned}
\frac{\partial f(\mathcal{G},a)}{\partial \boldsymbol{W}_g} &= \frac{2}{|\mathcal{G}|}(\boldsymbol{V}_a - \boldsymbol{W}_g),\\
\frac{\partial f(\mathcal{G},a)}{\partial \boldsymbol{V}_a} &= \frac{2}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} \boldsymbol{W}_g - 2\boldsymbol{V}_a,\\
\frac{\partial r}{\partial \boldsymbol{W}_g} &= 2\boldsymbol{W}_g, \qquad \frac{\partial r}{\partial \boldsymbol{V}_a} = 2\boldsymbol{V}_a.
\end{aligned} \tag{13}$$

In the preceding derivation of the learning algorithm it is implicit the performance measure that we are using. The goal is to obtain a ranking with $f(\mathcal{G},i)$ (8) as coherent as possible with the rankings provided by real graders. We exclude here the artificial Gaussian grader (Gg). That is, our performance measure is the proportion of judgment pairs correctly orderer by $f(\mathcal{G},i)$ using the student-graders as target.

The performance measure defined above is the area under the ROC curve (AUC). It is also known as the *concordance index* (C-index), or the pairwise ranking accuracy. This measure is called *Kendall-τ* in the paper of Raman and Joachims [6]. In symbols, the performance in $\mathcal{D}_{PJ}$ is given by its AUC defined by

$$\begin{aligned}
err(f,[g,i,j]) &= \mathbb{I}[\![f(\mathcal{G}_{PJ},i) < f(\mathcal{G}_{PJ},j)]\!]\\
&\quad + \frac{1}{2}\mathbb{I}[\![f(\mathcal{G}_{PJ},i) = f(\mathcal{G}_{PJ},j)]\!]\\
AUC(f,\mathcal{D}_{PJ}) &= 1 - \frac{\sum_{[g,i,j]\in\mathcal{D}_{PJ}} err(f,[g,i,j])}{|\mathcal{D}_{PJ}|}.
\end{aligned} \tag{14}$$

Notice that the AUC can be used to compare two different rankings. However, it is not a symmetric measurement. One of the rankings has to be taken as the *ground truth*; in the previous equations, the ordering of $\mathcal{D}_{PJ}$ was considered as the *right* order.

4. Detailed Description of the Assessment Process

Although the method presented in this paper has been formally described in the preceding sections, let us now collect the steps followed in the real experiment reported in the next section.

- All the students had to write an essay answering some questions. The assignment included 3 basic questions about *search* in the course *Intelligence Systems* for undergraduate Computer Sciences students. The first and third questions were open-response, while the second one asked the students to fill a table of scores and can not be considered open-response.

- All students had to anonymize their assignments and submit them using EasyChair. The event created was called *JRLO2014 (Joint Research in Learning to Order 2014)*.
- Then each student, acting as a reviewer or grader, received about 10 assignment to evaluate. The assignation was done at random.
- At the same time, the students received a detailed rubric, spelling out how to assess the assignments.
- Each question had to be graded in a numeric scale of integers form 0 to 10.
- Finally the students received the feedback from the anonymous reviewers of their assignments.

The computational process followed by the data so collected can be summarized as follows.

- From grades given by student-graders (1) we build the learning task $\mathcal{D}_{PJ}$ using (2).
- The learning algorithm that has two steps:
  - adds some new preferences suggested by the so-called Gaussian graders,
  - uses a SGD to find a couple of matrices, $\boldsymbol{W}$ and $\boldsymbol{V}$.
- The evaluation of the performance is computed with the whole set used for training, the learning task $\mathcal{D}_{PJ}$.
- The performance measure is the AUC (14).
- The ranking computed by (8) can be calibrated to transform percentiles into valid grades. This can be done using a table of equivalences or using grades provided by the staff for some assignments to make an interpolation.

5. Experimental results

In this section we report an experimental comparison of the method introduced in this paper. First we present the datasets used in the experiments, and then we show the results obtained in a comparison with a baseline and the evaluations of the instructors of the courses in our Universities.

# of graders	160
# of assignment	175
# of evaluations	1326
evaluations per grader	8.29 ± 1.45
evaluations per assignment	7.58 ± 2.02

Table 1: Datasets description

5.1. Datasets

The data used in the experiments gathers assignments of 175 students from our three Universities. A subset of 160 students participated as graders (see Table 1) with a total of 1326 grades from 0 to 10. Each student received an average of 8.29 assignments to evaluate, and a maximum of 11. Each assignment received in average 7.58 grades. Let us point out that we had only 4.74% of the total possible assessments.

The assignment given to the students comprises 3 questions. The topic covered was searching methods, both informed and uninformed. We were using the handbook by Russell and Norvig [25]. The first question was taken from this handbook, and it asked the students to formalize 3 academic search problems: graph coloring, monkey and bananas, and a water-jug puzzle. This was an easy question for the students both to answer and to evaluate after the rules given in the rubric.

On the other hand, the second and third questions used the *Tools for Learning Artificial Intelligence* [26]. Given a small graph of the Neighborhoods of Vancouver, the students were asked to use a searching prototype to find the shortest paths from a set of pairs of nodes. They had to use 3 beforehand fixed algorithms. The second question of the assignment requested the students to fill a table with the lengths of the best paths and the number of nodes expanded in each search by each algorithm. The third (and last) question asked to discuss the results achieved justifying the scores according to the optimality or not of the algorithms used.

In these experiments we considered 2 datasets, those formed with the grades given to the first and the third questions, we call it $Q1$ and $Q3$ respectively. We built the sets of preference judgments as was explained in Section 3. We avoided the second question since it has a mechanical assessment and thus it is out of the scope of this paper. Table 2 shows the average and typical deviation of grades for each question. Notice that the average grade in Q1 is considerably higher than in Q3. Additionally, this table displays the average range of grades received by the assignments and given by graders.

Let us highlight the extension of the range of grades received by the assignments; this is an indication of the difficulty of the peer grading tasks. Figure 1 depicts the histograms of grades received by the assignments for each question.

Average	Q1	Q3
grade	6.37 ± 3.08	4.84 ± 3.39
range per grader	6.50 ± 2.61	7.31 ± 3.00
range per assignment	7.01 ± 2.45	6.67 ± 3.25

Table 2: Description of grades in each dataset

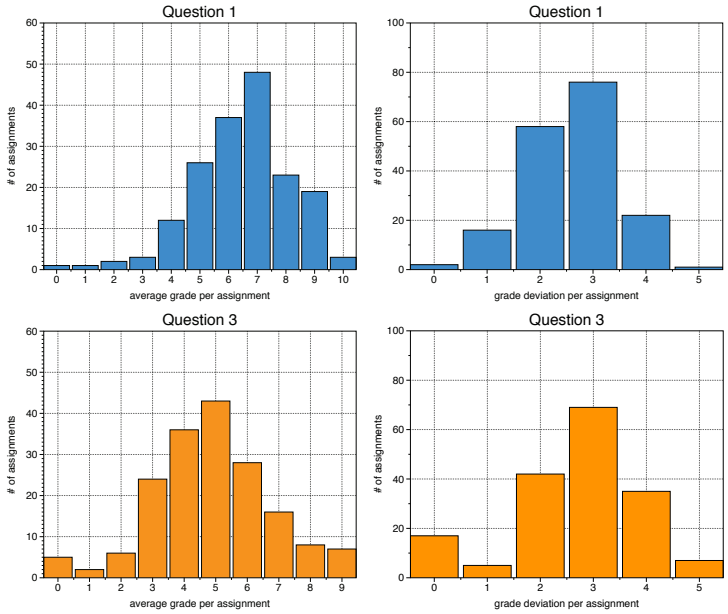

Figure 1: Average grade and deviation per assignment

5.2. Scores and discussion

The SGD (12) was applied using a learning rate defined in terms of the iteration it by

$$\gamma \leftarrow \frac{1}{(\gamma_s \cdot it) + 1}.$$

The parameters used were the results of a search of the best scores in

$$\begin{aligned} k &\in \{2, 10, 50\} \\ \gamma_s &\in \{10^e : e = -7, \ldots, 0\} \\ \nu &\in \{0\} \cup \{10^e : e = -4, \ldots, -1\}. \end{aligned}$$

To evaluate the quality of the results, we used the AUC defined in (14). We compared the rankings obtained by the method proposed in this paper with a baseline algorithm and the ranking induced by the assessments given by the

Instructors		
Uniovi	0.649	
UDC	0.659	
Maximum assignments per grader	Models	Averages
5	0.673	0.650
6	0.676	0.645
7	0.681	0.645
8	0.676	0.658
9	0.695	0.662
10	0.699	0.671
11	0.706	0.671

Table 3: Comparative AUCs (the higher the better) with respect to UPO in Q1

instructors of the Universities involved. The instructors evaluated the whole set of assignments, not only those of their own students. As baseline, we employed, as usual, the average of grades received by each assignment.

In the figures and tables of results, we use the acronyms of the Spanish names of our Universities that appear in the email addresses: Pablo de Olavide University (UPO), University of A Coruña (UDC), University of Oviedo (Uniovi).

Let us recall that the AUC is not symmetric and then we need to fix a ranking to compare with. In all cases we considered the *ground truth* the ranking provided by the instructor of UPO. The reason is that she achieved better scores with respect to the other instructors' rankings than any other.

Additionally, we wanted to check the sensitivity of our method with respect to the number of assignment supplied to student-graders to evaluate. For this purpose, we built new datasets for each question sampling the original data with different maximum number of assignments per grader: from 5 to 11.

The results are shown in Tables 3 and 4, and they are depicted in Figure 2. The scores may seem low; however, they are in line with other published results. Raman and Joachims [6] consider two datasets; they received an average of 23.7 and 13.3 grades respectively. These numbers justify that simply averaging the cardinal scores of the peer graders (the baseline used in this paper) performs well. The ordinal methods discussed in the paper were compared with TA (Teaching Assistant) grades. The AUCs range from 0.778 to 0.657. Notice that in our case we only have an average of $1327/175 = 7.58$, and the AUC with respect to the best instructor ranges from 0.661 to 0.706.

Another important issue is that in general we can observe that the increment in the number of assignment given to each student-grader gives rise to a better performance both in the baseline and in the models learned.

The scores achieved with Q1 are clearly better for the models achieved by our method than for the baseline. On the other hand, in Q3 the scores are

Instructors		
Uniovi	0.795	
UDC	0.650	
Maximum assignments per grader	Models	Averages
5	0.661	0.658
6	0.633	0.646
7	0.676	0.679
8	0.670	0.689
9	0.685	0.688
10	0.693	0.696
11	0.703	0.694

Table 4: Comparative AUCs (the higher the better) with respect to UPO in Q3

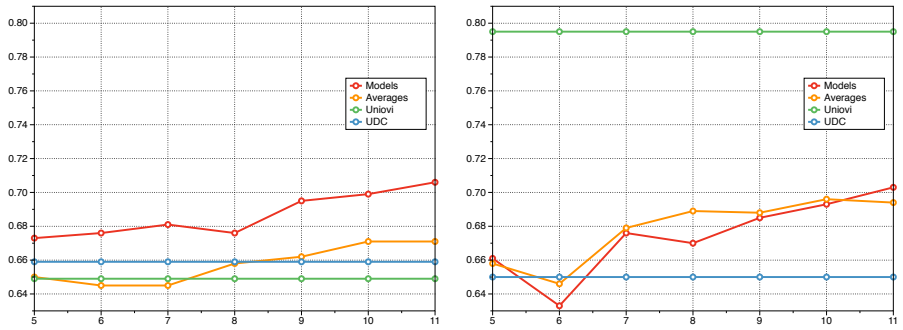

Figure 2: AUCs with respect to UPO assessments of the other instructors, models and baselines using 5 to 11 maximum evaluations per student-grader. Left hand side, Q1, right Q3

indistinguishable, there is not a clear winner. The reason for this behavior is the difficulty of the questions. Q1 is easier for the students than Q3. To express formally a discussion of scores seem to be harder than giving a complete problem formulation for academic toy search problems. Even the rubric provided to student-graders was difficult to understand in the case of Q3. The consequence is that the assessments of Q3 were more incoherent and therefore more difficult to generalize than in the case of Q1.

This is an interesting point to be considered in peer-grading. The assignments and rubric must be clear enough so as the student-graders can accomplish their task satisfactorily.

6. Conclusions

We have presented a factorization method to implement peer assessment. Our approach seeks a trade-off between cardinal and ordinal approaches. The

algorithm presented in Section 3 learns from preference judgments to avoid the subjectivity of numeric grades. But in addition to preference judgments expressed by student-graders, we included other preferences: those induced from assignments with significantly different average grades.

The algorithm presented uses a maximum margin approach solved using a SGD optimizer. It is fast and can be easily scalable to a large number of students.In addition, our method lacks some constraints present in other state-of-the-art approaches, such as the need of students' self-grading or previous grading by instructors. In fact, the last constrain makes those approaches inapplicable to a very large number of assignments, which is one of the main characteristics found in MOOCs.

The paper includes a report of the results obtained using this approach in a real world dataset collected in 3 Universities of Spain, see Section 5 for details. We compare the ranking obtained by our method with those given by a baseline and the staff instructors. The baseline was the average of the scores of student-graders. We found that when the assignments are not too hard for the students and they thoroughly understand the rubric, the performance of our models is better than the baseline and comparable with the rankings of the professional graders. In other case, the baseline and the models are similar and still can be comparable to those of the instructors.

We also checked that the number of assignments given to each student-grader is important. In all cases the performance improves as this number increases.

7. Acknowledgments

The research reported here is supported in part under grants TIN2011-23558, TIN2012-37954 and TIN2011-28956-C02 from the MINECO (Ministerio de Economía y Competitividad, Spain), all partially supported with FEDER funds. We would also like to thank our students for grading the assignments of other colleagues in the Universities of A Coruña, Pablo de Olavide, and Oviedo.

[1] M. M. Waldrop, Education online: The virtual lab, Nature 499 (2013). doi:10.1038/499268a.

[2] P. M. Sadler, E. Good, The impact of self-and peer-grading on student learning, Educational assessment 11 (2006) 1–31.

[3] C. Kulkarni, K. Pang-Wei, H. Le, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, S. R. Klemmer, Peer and Self Assessment in Massive Online Classes, Technical Report, Stanford University, 2013.

[4] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, D. Koller, Tuned models of peer assessment in MOOCs, in: Proceedings of the 6^{th} International Conference on Educational Data Mining (EDM'13), International Educational Data Mining Society, 2013, pp. 153–160.

[5] N. B. Shah, J. K. Bradley, A. Parekh, M. Wainwright, K. Ramchandran, A case for ordinal peer-evaluation in MOOCs, in: NIPS Workshop on Data Driven Education, 2013.

[6] K. Raman, T. Joachims, Methods for ordinal peer grading, in: ACM Conference on Knowledge Discovery and Data Mining (KDD), 2014.

[7] J. Díez, O. Luaces, A. Alonso-Betanzos, A. Troncoso, A. Bahamonde, Peer Assessment in MOOCs Using Preference Learning via Matrix Factorization, in: NIPS Workshop on Data Driven Education, 2013.

[8] A. Bahamonde, G. F. Bayón, J. Díez, J. R. Quevedo, O. Luaces, J. J. del Coz, J. Alonso, F. Goyache, Feature subset selection for learning preferences: A case study, in: Proceedings of the International Conference on Machine Learning (ICML '04), 2004, pp. 49–56.

[9] J. Díez, G. F. Bayón, J. R. Quevedo, J. J. del Coz, O. Luaces, J. Alonso, A. Bahamonde, Discovering relevancies in very difficult regression problems: applications to sensory data analysis, in: Proceedings of the European Conference on Artificial Intelligence (ECAI '04), 2004.

[10] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2002.

[11] W. Barnett, The modern theory of consumer behavior: Ordinal or cardinal?, The Quarterly Journal of Austrian Economics 6 (2003) 41–65.

[12] R. Herbrich, T. Graepel, K. Obermayer, Support vector learning for ordinal regression, in: Proceedings of the Ninth International Conference on Artificial Neural Networks, Edinburgh, UK, 1999, pp. 97–102.

[13] J. Weston, S. Bengio, N. Usunier, Large scale image annotation: Learning to rank with joint word-image embeddings, Machine learning 81 (2010) 21–35.

[14] V. Aggarwal, S. Srikant, V. Shashidhar, Principles for using Machine Learning in the Assessment of Open Response Items: Programming Assessment as a Case Study, in: NIPS Workshop on Data Driven Education, 2013.

[15] N. B. Shah, J. Bradley, S. Balakrishnan, A. Parekh, K. Ramchandran, M. J. Wainwright, Some scaling laws for MOOC assessments, in: KDD Workshop on Data Mining for Educational Assessment and Feedback (ASSESS 2014), 2014.

[16] Y.-M. Wang, Z.-P. Fan, Fuzzy preference relations: Aggregation and weight determination, Computers & Industrial Engineering 53 (2007) 163–172.

[17] F. Rodrigues, P. Oliveira, A system for formative assessment and monitoring of students' progress, Computers & Education 76 (2014) 30 – 41. doi:http://dx.doi.org/10.1016/j.compedu.2014.03.001.

[18] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002, pp. 311–318.

[19] F. Noorbehbahani, A. Kardan, The automatic assessment of free text answers using a modified BLEU algorithm, Computers & Education 56 (2011) 337 – 345. doi:http://dx.doi.org/10.1016/j.compedu.2010.07.013.

[20] Y. He, S. C. Hui, T. T. Quan, Automatic summary assessment for intelligent tutoring systems, Computers & Education 53 (2009) 890 – 899. doi:http://dx.doi.org/10.1016/j.compedu.2009.05.008.

[21] D. Pérez, A. M. Gliozzo, C. Strapparava, E. Alfonseca, P. Rodríguez, B. Magnini, Automatic assessment of students' free-text answers underpinned by the combination of a BLEU-inspired algorithm and latent semantic analysis., in: FLAIRS Conference, 2005, pp. 358–363.

[22] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30 –37. doi:10.1109/MC.2009.263.

[23] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowledge-Based Systems 27 (2012) 271–280.

[24] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Unsupervised feature selection via maximum projection and minimum redundancy, Knowledge-Based Systems In press (2014).

[25] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 2010.

[26] B. Knoll, Kisyński, G. Carenini, C. Conati, A. Mackworth, D. Poole, AIspace: Interactive tools for learning artificial intelligence, in: Proceedings of the AAAI 2008 AI Education Workshop, Chicago, IL, 2008.