Peer Grading in a Course on Algorithms and Data Structures: Machine Learning Algorithms do not Improve over Simple Baselines

Mehdi S. M. Sajjadi
Department of Computer Science, University of Hamburg, Germany

Morteza Alamgir
Department of Computer Science, University of Hamburg, Germany

Ulrike von Luxburg
Department of Computer Science, University of Hamburg, Germany

Abstract
We used peer grading in a course on algorithms and data structures at the University of Hamburg. During the whole semester, students repeatedly handed in solutions to exercises, which were then evaluated both by teaching assistants and by peer grading. We tried different methods from the machine learning literature to aggregate the peer grades in order to come up with accurate final grades for the submitted solutions (supervised and unsupervised, methods based on numeric scores and ordinal rankings). We found that none of them improves over the baseline of using the mean peer grade as the final grade.

1. Introduction
Peer grading refers to a process where students grade the work of their co-students based on a scoring rubric provided by the instructor. It has become increasingly popular in the context of massive open online courses, where thousands of students hand in homework that has to be graded (e.g., Kulkarni et al., 2015). Similar as in other crowdsourcing scenarios, the hope is that even though students might not be perfect graders, it might be possible to come up with a “fair” or “accurate” final grade for each submitted homework by aggregating many such imperfect grades. The challenge of finding good aggregation algorithms has been taken up by the machine learning literature and a number of suggestions have been made in the literature (e.g., Piech et al., 2013; Vozniuk et al., 2014; Raman & Joachims, 2014; Shah et al., 2013). The bottom line of these papers is that statistical models or machine learning algorithms are successful in solving this task.

Intrigued by its idea and potential, we used peer grading in one of the undergraduate courses in computer science at the University of Hamburg: the course on algorithms and data structures (AD). During the course of one semester, the students solved 5 exercise sheets with 3-5 exercises each. We used teaching assistants (TAs) to grade all solutions in the traditional way. Roughly once every two weeks, students got an exercise sheet as homework, resulting in 5 sheets with 17 individual exercises in total. The exercises consisted in little puzzles about algorithms or small proofs about algorithmic properties as seen in many text books. Solving 50% of the homework problems was a necessary requirement to pass the course. Exercises were solved and submitted by groups of three students. This semester we had 79 such
groups in total. For each individual exercise, every group handed in their solution via an online submission system (we used a modification of the moodle open source platform\(^1\)). After the submission deadline, a solution template together with assessment criteria was posted online by the course instructor. The template tried to describe all possible correct solutions of the exercises, point out pitfalls or common errors and discuss details about the grading procedure. Grading took place simultaneously in three different ways: (i) Self grading: All students had to read and understand the solution template and grade their own solution. Because solutions were submitted in groups of 3 students but self grading took place individually, this resulted in 3 self grades per solution. (ii) Peer grading: each student graded 2 randomly chosen solutions of other groups. Grading was double-blind. This resulted in 6 peer grades per submitted solution. (iii) All submitted solutions were also graded 2 randomly chosen solutions of other groups. Grading was performed by several TAs.

In order to ensure that students take their peer grading duties seriously, we made a reasonable peer grading performance a mandatory requirement to pass the course: students were required to participate in the peer grading of all but one exercise sheet and we announced that sloppy grading behavior would not be tolerated. Moreover, the peer grades contributed with a weight of 20% to the overall assessment of the exercise performance of each student. An anonymous questionnaire at the end of the semester revealed that about half the students liked the concept of peer grading, had the impression that it helped them deepen their understanding and would like to do it again in another course. The other half of the students did not like it, mainly because it had taken too much of their time.

3. Algorithms for estimating true grades

From a machine learning point of view, the most interesting questions in the context of peer grading are the following:

Q1 Unsupervised setting (as used in MOOCs): In the complete absence of TA or instructor grades, is it possible to aggregate many “imperfect” peer grades in order to come up with a “fair” or “accurate” final grade for each submitted solution?

Q2 Supervised setting (as in a university context): In a scenario where partial grading is available by an instructor, is it possible to predict or recover the instructor’s grades based on aggregated peer grades? If instead of a single instructor, partial grading is performed by several TAs, is it possible to predict missing grades at least in the quality of TA grades?

Q3 Adversarial setting: Is it possible to come up with a grading scheme that is robust against adversarial behavior of individuals (such as deliberately giving very good grades to everybody, or deliberately downgrading others) or adversarial attacks of groups of students (such as a cartel of students that act together in order to fool the grading procedure)?

We focus on the first two questions. Let us introduce some notation. Over the course of the semester, the students solved and graded exercises \(e_1,e_2,\ldots\). Fix one exercise \(e\) and suppose that solutions \(s_1, s_2, \ldots\) to this exercise have been submitted to the system. Denote the set of all submitted solutions by \(S\). Consider a set \(G\) of graders \(g_1, g_2, \ldots\).

By \(\text{score}(s, g)\) we denote the score given to solution \(s\) by grader \(g\). In the unsupervised scenario, we do not have any data on what the “true” score of each solution should be. Instead, the standard approach is to simply define the true score as the population average over the scores of all potential graders. Because in practice we just observe the scores of very few graders, each of whom might have a particular bias, the goal of an unsupervised algorithm is then to correct for inaccuracies that might have been introduced by the actual set of graders. In the supervised scenario, we assume that there exists a true score \(\text{score}(s, \text{true})\) of the solution (say, given by the instructor). The goal of peer grading algorithms is to give an estimate \(\text{score}(s, \text{estimated})\) of the true scores of all solutions based on the set of all peer grades. Depending on whether the focus is on the actual values of the scores or just on the ranking of the solutions, errors are evaluated by different error functions. We use the \(L_2\) error in order to compare the absolute score values and the Kendall-\(\tau\) error to compare the rankings induced by scores. The Kendall-\(\tau\) error between two rankings counts the number of pairs for which the ranking order is inverted: an error of 0 for agreement, 1 for inversion, and an error of 0.5 if one of the rankings is equality and the other one an inequality. The final Kendall-\(\tau\) error is the mean over the errors for all possible pairs.

In the unsupervised setting, we use the following approaches to estimate the scores:

- **Mean and median.** We simply estimate the true scores of a solution by the mean (resp. median) of all peer grades of this solution.

- **Unsupervised-single-task (UST).** We follow Piech et al. (2013) and make the following model-based assumptions. Fix the data of one particular exercise. We assume that the true scores of all solutions are normally distributed as \(\mathcal{N}(\mu_{\text{score}}, \sigma_{\text{score}}^2)\). Each grader has an inherent bias \(\text{bias}(g)\) and a certain reliability \(\text{reliability}(g)\). The intuition is that the bias models the tendency of a grader to give generous scores or to be very strict.

\(^1\)https://moodle.org/
Peer Grading in a Course on Algorithms and Data Structures

Figure 1. Fitting UST and UMT on artificial data (100 solutions, 100 graders, 5 exercises). Left: Dataset generated according to UMT’s model assumptions (true scores generated with $\mathcal{N}\left(\frac{1}{2}, \frac{1}{6^2}\right)$, biases $\mathcal{N}(0, \frac{1}{8^2})$ and reliabilities $\Gamma(3, \frac{1}{30})$). Right: Skewed true score distribution according to Weibull($\frac{3}{2}, \frac{1}{3}$). For 80 graders, we model biases and reliabilities as before, while 20 graders always draw scores uniformly at random. The panels show the $L_2$-errors and the Kendall-τ errors of the estimated grades with respect to the actual underlying true grades. The box plots show the errors for 100 independently generated datasets with a varying number of peer grades per solution.

whereas the reliability accounts for the variance in the grading performance. Formally, if a solution $s$ has true score $\text{score}(s, \text{true})$, then the grade given by grader $g$ is normally distributed as

$$\mathcal{N}\left(\text{score}(s, \text{true}) + \text{bias}(g), 1/\text{reliability}(g)\right).$$

The bias of all graders is distributed as $\mathcal{N}(0, \sigma_b^2)$, and the reliabilities of all graders are Gamma-distributed as $\Gamma(\alpha, \beta)$. The hyperparameters $\mu_{\text{score}}, \sigma_{\text{score}}^2, \alpha, \beta$ are chosen prior to seeing the data. The goal is then to fit the model to the observed data and to learn the parameters $\text{bias}(g)$, $\text{reliability}(g)$ and $\text{score}(s, \text{estimated})$. We used the EM algorithm for this purpose, which according to Piech et al. (2013) gives results similar to those obtained by more elaborate Gibbs sampling procedures. Evaluating this model on artificial data confirms the finding in Piech et al. (2013) that it does a remarkable job at recovering the true underlying score, see Figure 1 for an illustration. In particular, the model even works reasonably well on artificial data with a model mismatch (data that has not been generated according to the model assumptions) and is not very sensitive to the choice of the hyperparameters.

• Unsupervised-multiple-tasks (UMT). In our course, each student graded about 47 different solutions of 17 exercises over the whole semester. While the UST model just learns from one exercise at a time, the UMT model learns the parameters jointly over all exercises. The assumption that the bias and reliability of a grader are inherent attributes and do not vary over different exercises leads to more accurate estimates as there is much more data to work with.
Ordinal model Bradley-Terry (BT). It has been suggested that instead of estimating the actual numeric grade of each homework, it might be more reliable to just reconstruct the ranking of the solutions from peer grading. A number of models have been described for this purpose (Raman & Joachims, 2014; Shah et al., 2013), we used the Bradley-Terry model. It assumes that the likelihood of switching the ranking order of two solutions depends on the distance of their true scores, see Raman & Joachims (2014) for details. Note that in our case, we use the numeric scores provided by the graders to induce the ranking desired by a grader.

In the supervised models, the goal is to learn using the peer grades to predict the true grades as given by an instructor. In our case, we take the grades provided by the TAs as ground truth. We consider the following two approaches:

Supervised-naive (SN). As baseline we use the following naive idea. For each exercise, we split the solutions into a training set and a test set. We use the TA grades as ground truth for the training sets to estimate the student grader biases: we compute the mean deviation of each student’s grades to the TA grades for the same solutions. On the test set we then estimate the true grades from the peer grades by shifting each peer grade according to the bias of the corresponding student and then compute the mean over all bias-corrected peer grades.

Supervised-multiple-tasks (SMT). Another approach is to incorporate the TA grades directly into the UMT model. For some subset of the solutions, the respective TA grades are put into the model like any other peer grade, but the TA reliabilities are set to a high constant. This automatically corrects the student reviewer biases towards the TA grades and also gives higher reliabilities to students that grade similar to the TAs. To see how much this improves the overall accuracy, the error is only computed on solutions whose TA grades are unknown to the model.

4. Our data and its analysis

4.1. AD data: first observations

Before we actually train the models, let us have a first glance at our raw data. To be able to compare performances across different exercises, we rescaled the scores of each exercise to lie in the interval $[0, 1]$. A different standardization of shifting and rescaling the scores to have mean 0 and variance 1 ($z$-scores) led to very similar results. To get a first impression of our data, consider the plots on the left in Figure 2. In each panel, the figure shows histograms of scores for one particular exercise. These already show a number of interesting facts. Not very surprisingly, we see that self grades are often higher than peer grades, which in turn tend to be better than the grades given by the TAs. For some of the exercises, it looks as if the peer grade histograms are “shifted versions” of the TA histograms but this is not always the case. To the contrary, sometimes the overall characteristics and shapes of the histograms are quite different. In general the scores do not look as if they were normally distributed. The first obvious reason is that the scores are bounded to a fixed interval which can lead to several artefacts. The score distributions are often skewed, for example when the exercise was easy and many students
Peer Grading in a Course on Algorithms and Data Structures

Figure 3. Fitting models to our AD data. The box plots show the mean error with respect to the TA grades of each exercise over the whole semester. In each panel, the three groups refer to the data that was used to fit the models: the 3 self grades only, the 6 peer grades, or both together.

got full marks for their solutions. In many cases the score distribution is clearly bi- or multi-modal: this can arise if a big number of students solved only part of the exercise whereas others solved it completely.

A second aspect illustrated in Figure 2 (right) visualizes the relationship between TA grades and peer grades for individual solutions. While the tendency of peer grades generally being higher than the TA grades is again evident here, we can also see quite a number of solutions that received a score of 0 by TAs but moderate to large scores by peers. Looking into the corresponding solutions reveals that a typical reason is a faulty solution for which many reviewers do not spot the mistake and give full marks e.g., the solution describes an algorithm that does not meet the required run-time complexity or the solution may show an alternative proof that has not been discussed in the template solution and the peer grader does not have the skill to judge whether this proof is correct or wrong. We can also see in the figure that there is a large number of solutions with full TA grades but slightly lower peer grades — this is due to the effect that it is simply unlikely for all 6 peer grades to be full scores and a single lower grade will drag down the mean value.

On a more abstract level, the figure reveals that the “sources of error” for a grader are not only a bias due to different taste or different levels of strictness as suggested in the probabilistic models above, but also a serious lack of understanding or information. This is problematic for statistical algorithms: if a solution gets large scores by most peer graders, there is no way in which we can estimate whether this is really justified because the solution is correct, or whether the reason is that all peer graders have overlooked a crucial mistake in the solution. On the other hand, it is hard to detect cases of otherwise reliable graders getting a score completely wrong, in particular in the realistic scenario where we do not have an abundance of grades per solution.

4.2. Analysis of unsupervised models

We analyze our data using the models described in Section 3. In the unsupervised scenario, we simply take all the grades we have, fit the models, and estimate a “true score”. Note that in the unsupervised setting we cannot optimize the model to fit any ground truth (such as the TA grades). If the histogram of peer grades is shifted with respect to the
Figure 4. Analyzing the effect of using several TAs as baseline. Left: Smoothed histograms of all given grades by the TAs and students. Right: Box plots in the same setting as in Figure 3 (top right), but with white noise added for the evaluation step to simulate the effect of using several TAs: errors were computed against noisy true grades (independent Gaussian noise \( N(0, 0.05^2) \)).

TA grades (for example, as it is the case in the third panel of Figure 2), there is no way for an unsupervised model to correct for this. Hence, comparing the estimated true scores to TA scores by any loss function that compares the scores directly, such as the \( L_2 \) error, might be dominated by the overall bias shift. To cover for this, we use the Kendall-\( \tau \) rank correlation as a second error measure.

The hyperparameters of the models were chosen as follows. The parameters \( \mu_{\text{score}}, \sigma^2_{\text{score}} \) and \( \sigma^2_{\text{bias}} \) for UST and UMT only control the strength of regularization and were found to have little to no impact on the overall accuracy. The reliability parameters \( \alpha \) and \( \beta \) control whether the model gives all students a similar reliability or fits them to a large variety of reliability values. In our evaluations, we used the sample mean and variance of the given peer grades for each exercise to set \( \mu_{\text{score}} \) and \( \sigma^2_{\text{score}} \) and fixed the remaining hyperparameters at \( \sigma^2_{\text{bias}} = 1/36 \), \( \alpha = 3 \) and \( \beta = 1/30 \). For BT, we again used the sample mean and variance as priors and chose \( \alpha = 10 \), \( \beta = 2 \) for the reliability. Note that in a purely ordinal setting, students would only report rankings of solutions, so sample mean and variance would be unknown. In that case, the parameters can be chosen arbitrarily to control the distribution of the resulting scores.

Figure 3 shows the results on our AD dataset. The ordinal algorithm BT cannot be run on the self assessment grades only, as each student only graded one solution, which does not imply any rankings. On the peer grades alone, it performs much worse than the cardinal models. This is due to the fact that the ordinal model needs pairwise rankings, but each peer only graded two solutions per exercise. As a consequence, even though all solutions have been graded 6 times, we can only recover 3 pairwise orderings from these grades. This is clearly insufficient for accurate ranking estimations. Note that although the self grades are more inaccurate than the peer grades, merging self and peer grades decreases the overall errors.

A more surprising finding is the performance of the cardinal models. As opposed to the results on artificial data in Figure 1, where the model-based approaches clearly outperform the baseline, we now see that the model-based approaches UST and UMT provide no improvement over the simple mean. This finding is disappointing from a machine learning point of view and contradictory to the results in the literature. Let us discuss some reasons that could possibly explain it.

- **Amount of data.** We have about 6 peer grades and 3 self grades per solution, collected over 17 exercises. Each student submitted around 47 grades in total. The results on artificial datasets as well as other studies on peer grading suggest that this should be enough in order to get a reasonably reliable estimate, even for the model-based approaches. It is not the lack of data that leads to this problem.

- **Model assumptions mismatch.** As described in Section 4.1, our data typically does not satisfy the model assumptions (normal distributions, etc). However, experiments with artificial data whose distributions do not agree with the model assumptions show that typically the model still works reasonably. We do not believe that the model mismatch is the major source of the problem.

- **Model fitting.** We set the hyperparameters as described above. Simulations with artificial data show that the models are not very sensitive to the choice of the hyperparameters. To actually fit the model we used the EM algorithm. Piech et al. (2013) reported that the results of
the EM algorithm are almost equal to the ones of Gibbs sampling. We believe that while a small improvement might be possible with more clever model fitting strategies, the overall effect cannot be explained by this.

• *TAs as baseline.* We evaluate our errors against the TA grades, that is we consider the scores or orderings given by the TA grades as ground truth. However, these grades have been given by 6 different TAs, so the variance within these grades might make them unsuitable to serve as ground truth (in the extreme case, if we compared against random grades, then none of the models would outperform the others). We first study this effect with artificial experiments. Consider the setting in Figure 1, but we now added noise to the true grades before computing the errors (we used Gaussian noise with standard deviation 0.05, this is a realistic estimate for the TA standard deviation in our data). The results can be seen in Figure 4 (right). The results still look similar to the original results in Figure 1, just the overall performance worsened slightly due to the noise. In particular, the UST and UMT models still considerably outperform the mean estimate. This is even the case if we use an unrealistically large standard deviation for the noise, say 0.2.

As a next step, we look at our actual data to evaluate the consistency among the TA grades in comparison to the consistency among the peer grades. Unfortunately, we did not record data that allows to compare the performance of different TAs on the same solutions. Instead, we consider the following evaluations. We first compared the average reported grade of the TAs to the average grades given by the student reviewers. As can be seen in Figure 5 (left), the TAs have a very low bias amongst each other, in particular it is much lower than the biases amongst the students. Next, we look at the overall histograms of all given grades by each TA, see Figure 4 (left). There is little variance amongst the TA histograms, in particular compared to the variance in the peer histograms. All in all it looks like the TAs grade reasonably consistently, so we believe that the use of different TA grades as ground truth cannot be the major reason for the lack of improvement of the probabilistic models over the simple baseline.

• *Bias vs. reliability.* As seen in Figure 5 (mid, right), the overall biases are not very large, with only few students exceeding a bias of 0.1. On the other hand, the variance in reporting grades is quite high. More than half the students reported grades that deviate from the mean peer grades with a standard deviation of at least 0.14, a variance of roughly 0.02. To gain some intuition on the values, note that a student who theoretically always reported the same score 0.72 for all solutions (the overall mean peer grade on all exercises together) would end up with a variance of 0.05.

Piech et al. (2013) reported that more than 90% of UST’s improvements are due to the fact that the model-based approaches correct for the bias, an effect that we also confirmed on artificial data. However, in our AD data, the errors induced by low reliabilities dominate the errors due to bias. This may be part of the reason why the models do not perform well here.

• *Sources of grading error.* As discussed above, reasons for differences in grading behavior are not only that people have “different tastes” or are “differently strict”. Rather, it is often the case that graders make serious errors due to lack of information or lack of understanding in the mate-
rual. This problem is bound to come up when using peer grading in a course such as algorithms and data structures and might be much less of an issue when grading is used to evaluate project reports (Raman & Joachims, 2014) or design questions (Kulkarni et al., 2015).

To check whether the model-based algorithms improve if we just use “easy-to-grade” exercises, we selected a number of exercises where the errors were low, indicating that the students had no difficulties in grading the solutions.

We found that even in this scenario, the model-based algorithms do not perform better than the mean. This may be due to the fact that in easy-to-grade exercises, the mean algorithm does a good enough job at eliminating the different biases or reliabilities, so there is not much room for improvement left. Similarly, if we just train on “difficult-to-grade” exercises, we do not find an improvement of model-based algorithms compared to the mean.

The reason is that in this scenario, the errors are dominated by the solutions that are graded totally wrong by most students, so none of the algorithms can do a good job.

• Unmotivated students. One might suspect that some proportion of students tried to get away with minimal effort and produced grades that are pretty much random. However, looking closely into the data reveals that we only had a small portion of these students, so the models should succeed at giving them low reliabilities and lead to a better performance.

4.3. Analysis of supervised models

When we consider the results for the supervised learning models in Figure 3, we see a similarly disappointing picture: the supervised models do not improve over the simple mean estimator. While the Kendall-$\tau$ errors do not change much compared to the unsupervised models, the variance in L2 errors over different tasks gets smaller. This is an effect of calculating the student reviewer bias values against the TA grades which improves the overall error in tasks that have a very high error and overfits in tasks that were graded well by most students in the first place. All in all, supervision does not seem to have a significantly positive effect on the errors of our algorithms.

5. Conclusions

We have mixed feelings towards peer grading after evaluating all the data collected in our algorithms and data structures course.

The positive point of view is that even though peer grades tend to be more optimistic than TA grades, the size of this effect is not very large. The peer grades give a reasonably informed picture of the true grades. For this reason, using simple estimates such as the mean grades is competitive to more elaborate model-based algorithms. From an application point of view this finding is helpful: an easy to understand mechanism such as a simple mean is more acceptable to students than a complicated model when it comes to generating their final grades.

From a machine learning point of view, our results are somewhat disappointing. None of the models we tested outperform the simple mean estimator on our data. While we may simply not have been clever enough at fitting the model, our general feeling is that it will be very difficult to come up with algorithms that do a much better job. The reasons for this difficulty might be the heterogeneity of the score distributions of different exercises, the high variance among graders, and the different and rather unpredictable sources of grading errors (“lack of understanding” rather than a “slightly different taste”). It seems very hard to model all these aspects unless one has a much larger amount of grades per solution. However, the peer grading setup does not allow us to simply scale up the number of grades per solution, because students are not willing to grade more than a small number of solutions. Finally, let us mention that our negative findings may be special to our course such as algorithms and data structures. Peer grading could be more successful for tasks where grading is a matter of taste rather than of understanding.

The data set generated by our class has been made publicly available on our homepages. Beyond the material discussed in this paper, the data contains a couple more experiments which we did not report here due to space constraints (e.g., one week we asked for ordinal comparisons instead of cardinal scores, we conducted an experiment to compare the performance of the TAs on some selected exercises, etc.). Moreover, we are currently working on a release of a plugin to the moodle platform that supports peer grading in a group-based scenario as used in the AD course.

Acknowledgments

This work has partly been supported by the German Research Foundation (grant LU1718/1-1 and Research Unit 1735) as well as by the German Federal Ministry of Education and Research (grant number 01PL12033, Universitätspool/TP 16 Lehrlabor at Universität Hamburg). The authors alone are responsible for the content of this publication.
References


