Addressing Bias to Improve Reliability in Peer Review of Programming Coursework

Steven Bradley
s.p.bradley@durham.ac.uk
Department of Computer Science, Durham University, UK

ABSTRACT

Peer review has many potential pedagogical benefits, particularly in the area of programming, where it is a part of everyday professional practice. Although sometimes used for formative assessment, it is less commonly used for summative assessment, partly because of a perceived difficulty with reliability. We explore the use of a hierarchical Bayesian model to account for varying bias and precision amongst student assessors. We show that the model is sound and produces benefits in assessment reliability in real assessments. Such analyses have been used in essay subjects before but not, to our knowledge, within programming.

CCS CONCEPTS

• **Social and professional topics → Student assessment**; • **Applied computing** → *Computer-assisted instruction*; *Collaborative learning*.

KEYWORDS

education, programming, assessment, peer review

1 INTRODUCTION

Peer review of code is a core part of professional practice in software development, and research by Wang et al. [17] has shown that introducing peer review during programming language learning has a range of pedagogical benefits, with positive impact on: student programming skills; collaborative learning competence; compliance with coding standards; time management capability; competence of giving and accepting criticism. As part of assessment, peer review is most often used in the context of formative assessment, with both students and faculty expressing concern of the reliability of students as markers [14]. In this paper we examine the reliability of groups of students as peer assessors, and apply statistical techniques that have been applied to peer review in other disciplines to increase reliability, in the context of a first-year higher education (HE) programming course.

We are not examining validity specifically here, but our assumption is that coursework is more valid way of assessing programming than an exam. One of the principal concerns with summative assessment through coursework is that the potential for plagiarism affects reliability. The use of *divergent* (as opposed to *convergent*) assessment has been found to be important in identifying plagiarism in student coursework [1]. The coursework data we examine are based on assignments that have a lot of student choice in them, both to achieve the desired divergence and also to encourage creativity. This divergence makes the reliability of assessment all the more challenging and important.

Our statistical model builds on similar work by Goldin [5], which was applied to the assessment of essays in Intellectual Property Law. Further details are to be found in Sections 2.3 and 3, but it makes sense to give a brief summary of the idea here. We assume that each piece of work to be assessed has an underlying numeric quality score in each of a number of dimensions that are defined as part of an assessment rubric. Each time a piece of work is assessed it is given a score for each dimension, and we model this assessment score as a normally distributed random variable. Exactly how this variable is distributed (i.e. the μ and σ) depends on both the work being assessed (its underlying quality) and characteristics of the assessor.

Firstly, the assessor may be unreliable, in the sense that they would give different marks to two pieces of work with the same underlying quality. Here we are relating reliability (in the sense of reliability of assessment) with the measure of spread in the distribution of assessment marks. In this paper we will use *precision* ($\tau = 1/\sigma^2$) as the statistical measure of spread of assessments by an assessor in the model; other papers use related terms such as 'consistency' and 'spread'. Note that this is not exactly the same as reliability of assessment, not least because assessment reliability does not usually have such a formal definition, but also because the reliability of the assessment process as a whole relies on multiple peer assessors, and a combination of their assessments.

Secondly, the assessor may have *bias* — this is not bias in the sense of favouring a particular gender, ethnic or socio-economic group (we assume that the assessor does not know these characteristics because of anonymity), but rather a propensity to mark higher or lower on average than other assessors. The mean of the assessed score μ is then modelled as the quality of the work plus the bias of the assessor. The bias may be positive or negative, where a positive bias indicates the tendency to award higher marks.

Taken together these give

$$assessed\_score \sim N(quality + bias, \sigma^2)$$

where $\sigma^2 = 1/precision$. An assessor with small bias and high precision will usually give an accurate assessment of the quality of a piece of work — so we use the term *accuracy* to describe the combination of bias and precision. The accuracy of the individual peer assessments contributes to the reliability of the overall assessment process in which they are used. In practice the situation is slightly more complex than this, because marking rubrics often have multiple dimensions on which student work is assessed, which may or may not be related to the structure of the work itself.

What we are interested in is finding the underlying quality of the work based only on the observed values of the assessed scores. We do this by running a Monte-Carlo Markov Chain (MCMC) simulation to find estimates for the quality, alongside the bias and precision of each of the assessors. Exactly how the model is constructed and analysed is described in Section 3 and the context and details of the programming coursework assessments that we analyse are outlined in Section 4. We then evaluate the data in light of the research questions in Section 5 before the concluding Section 6. But first we review the related work in the literature, and on that basis formulate our research questions.

2 RELATED WORK

We are not aware of other work that aims to address bias in peer assessment of programming, but we review other work related to the problem.

2.1 Peer Review

Early work by Topping [14], who reviewed a range of studies of peer assessment, found peer assessment to be both valid and reliable, but noted that 'both assessors and assessees might experience initial anxiety about the process' and that 'student acceptance seemed unrelated to actual reliability'. Another meta-analysis of 48 self and peer assessment studies by Falchikov and Goldfinch [2] found that agreement between teacher marks and peer marks was highest when peers awarded an individual global mark based on specific learning outcomes, as opposed to no explicit criteria, or judged on each dimension separately. None of the studies in these meta-analyses try to take into account individual assessor bias.

Pare and Joordens [8] describe a web-based system, peerScholar, which they applied in large classes (>2000) of psychology undergraduates. They investigated the correlation between 'expert' markers (graduate teaching assistants) and undergraduate peer reviewers. Peer marks were calculated by averaging three peer reviewers. They found moderate correlation between individual expert graders and good correlation between 'average expert' and 'average peer', particularly once accountability for reviews had been included via a 'mark the marker' process. They found average peer marks to be slightly higher (by about 3%) than average expert marks, where marks were given on a 10 point scale. The accuracy of peer grading was also examined by Freeman and Parks [3] in the context of an introductory biology course. They compared peer-awarded grades with 'professional' grading and found that students were in general slightly more generous in their marking, particularly for components that fell within the higher levels of Bloom's taxonomy.

2.2 Peer Review in Computing

Sitthiworachart and Joy [12] describe the use of a web-based system for peer assessment applied to UNIX programming. They used a three-point scale (No/Partial/Yes) for a range of 8 criteria, although they found students were not happy with that and recommended a finer five-point scale. 'For example, more than 50% of students think that they are not qualified to be marker; only the tutor or lecturer is the expert marker.'

Prazina and Okanovic [10] used a bespoke web-based system to carry out double-blind peer review of undergraduate software engineering projects based on ranking, and found that ranks were moderately to highly correlated with final grades. In their study 'no correlation between students points and error could be found'.

Peer assessment for coding assignments in undergraduate programming engineering courses was studied by King [7] who looked at non-anonymous formative peer assessment. She found 'the statistical analyses that compared student performance on the homework assignments to their acceptability of the peer grading session revealed that the peer grading session improved student performance' and 'did not find any statistically significant relationship with student acceptability of peer grading' and 'demographics (sex, low income, race), academic level (year in school based on number of credit units)'.

A study by Sanchez et al. [13] comparing assessment student videos on database administration over six dimensions by professionals (i.e. professional practitioners), academics and (peer) students found 'no significant differences ... between academics and students' in global assessment, although professionals were found to mark harder than academics or students. Some variation was found below the global level, in particular content-related questions had larger differences between the groups than format-related questions.

2.3 Addressing Bias in Peer Review

Goldin [5] introduced the idea of using Bayesian statistics to identify and correct for bias in student grades. Students and an instructor reviewed essays in Intellectual Property law and the results were analysed by building a Bayesian model including bias and precision of assessors, as we have already discussed. A total of 28 students each reviewed four pieces of work along five rubric dimensions on a seven point scale. The analysis of their reviews demonstrates that including bias modelling increased agreement with instructor marks by up to 30%. Two versions of the statistical model were considered, one in which the bias depended only on the assessor, and another version where each assessor had a separate bias parameter for each dimension of the rubric under consideration. Faithfulness of the two models was compared by analysing the error with respect to the instructor marks and the deviance information criterion (DIC), which looks to balance goodness of fit against the number of parameters in the model. Their results showed that having separate bias parameters for each dimension of the rubric gave better models when the rubric was specific to the question, rather than a rubric 'designed to be generally applicable to assessment of legal writing'. They argued that this possibly reflected where students did not have enough knowledge of a particular part of the subject to give an accurate assessment.

Item response theory has also been used in application to peer assessment [15], similarly building a Bayesian model which includes other characteristics, in particular 'severity' and 'consistency' which correspond fairly directly to the notions of 'bias' and 'precision' in our models. They assessed against a rubric with five dimensions for a series of five assignments from 20 students in an e-learning course, as well as performing some statistical experiments to demonstrate the efficacy of the approach on synthetic data.

Garcia-Martinez et al. [4] report on a technique for weighting the marks awarded by different peer assessors in a philosophy MOOC according to their engagement and performance within the MOOC. Each student reviewed three other students on four dimensions using a 1-5 scale. They found that weighting the peer marks improved correlation with instructor marks from moderate positive to strong positive.

Also investigating peer assessment with MOOCs, but this time on HCI courses, Piech et al. [9] investigate a range of probabilistic models for identifying bias and precision. By examining a subset of predefined 'ground truth' assignments they found that staff graders were also subject to inter-rater inconsistency and not as reliable as average peer marks. However their ground truth assignments were reviewed by 160 peer reviewers on average, which is clearly not a feasible approach outside an experimental environment. Their more sophisticated probabilistic models included bias and precision details of previous reviews and the grades achieved by reviewers themselves. All of their models made a marked improvement in the accuracy of the final mark, as compared with finding a median of the original reviews. The more sophisticated models did have a slight improvement in accuracy, but 95% of the improvement was due to statistical modelling of bias.

A theoretical/simulation approach is taken Hamer et al. [6] which looks at the accuracy of peer review, but considers only precision (via weighting) and does not include the idea of bias within their model and algorithm. They conclude that accounting for variation in reviewers (albeit only in their precision) 'provides a robust solution under a wide variety of conditions'.

Another approach to improving reliability is to only include reviews from students that have demonstrated themselves to give accurate reviews according to predefined 'gold-standard'. This is how Calibrated Peer Review [16] works, although evidence supporting claims of its usefulness in improving student writing quality is at best mixed.

2.4 Research Questions

RQ1 Can a statistical model improve reliability of peer assessment of programming coursework?

RQ2 How is peer assessment accuracy of programming coursework related to characteristics of the reviewer?

Several of the papers discuss student opinion of peer assessment that is beyond the scope of this paper, although some comments are included in the conclusions. Quite a lot of quantitative research effort has gone in to examining the agreement between peer assessment grades and instructor grades, but whether this addresses the central question of reliability is moot — in the light of the findings of Piech et al. [9] that 'mean student grade was more consistently accurate with respect to the rubric than the volunteer staff grade'. There is a separate discussion about whether peer assessment is more or less valid than instructor assessment, but as we are focussing on instructor-defined assessments in this case the question of validity is less of an issue, and is probably much more to do with the assignment and rubric that is set, rather than who is doing the marking. In this paper we focus on the question of reliability.

3 MODELLING BIAS

The underlying idea behind the model was explained in the introduction: peer reviews scores are thought of as random variables that are distributed according to characteristics of the work being reviewed (its quality) and the reviewer (bias and precision). The first step is to acknowledge that there are multiple parameters for quality, depending on the work under review and the dimension of the rubric that is being assessed, and each of these is distributed according to its own parameters: it is not safe to assume that all sections of the rubric have the same average score as some things might be harder for students than other parts. On the other hand it is not safe to assume that the average scores of the different sections are unrelated, as the same cohort of students is being assessed under each dimension: if a more capable set of students were being assessed, or they had more time to do the work, then we would expect the average scores for all dimensions to be increased. To balance these competing demands for independence and interrelatedness we use *pooling* in which we assume the averages for the dimensions are themselves normally distributed around a mean with a certain variance. The means for the dimension scores then become identical independently distributed variables — the pooled mean and variance for this distribution are new hyperparameters which themselves need their distribution defined and possibly parameterised. A similar structure is built around the overall quality of the reviewee (the marks for dimensions are independent but related, so are distributed with a pooled mean), and then the variance for each of the distributions is also modelled using appropriate pooling. The quality of a piece of work is distributed with a mean that depends on the student *j* and the dimension *d* like this

$$quality_d^j \sim N(\mu_d + \mu^j, 1/\tau_{quality})$$

where μ_d is the pooled mean for the dimension, μ^j is the pooled mean for the student and $\tau_{quality}$ is the pooled precision (i.e. $1/variance$) for the quality. These are combined into the distribution for the review score given by student *i* of student *j* on dimension *d*:

$$score_d^{ij} \sim N(bias_d^i + quality_d^j, 1/\tau_d^i)$$

The parameters towards the top of the hierarchy are loosely constrained, but it would be possible to include further constraints into the model, for instance if a particular mean or variance of the scores were required. If such a constraint were implemented it would be best for the mean of the biases to be left to float free (with pooling), but we used a mean of zero for the biases in our model as any small adjustments to the marks could be achieved through selection of the mark used for a grade band (see Section 4.3 for more details).

There are many variants of the model that we could try, for instance we could multiply the bias by the quality instead of adding it — we choose to follow the additive approach used in other papers [5, 9, 15]. It is questionable whether we should have a separate reviewer bias parameter for each dimension. Adding more parameters to the model runs the risk of overfitting, where the parameters and hyperparameters vary unnaturally to optimise fit. There are two counters to this: firstly by appropriate pooling, the parameters are constrained. Secondly it is possible, where a 'gold standard' is available to balance the goodness of fit against the number of degrees of freedom. There are various *information criteria* that can be used to compare models for this balance, the most appropriate one in this case being the Deviance Information Criterion (DIC). Goldin [5] compares models similar to ours with unidimensional and multidimensional bias using DIC, and found the best model used multidimensional bias, particularly when the rubric was specific to the content of the question (as ours is). This makes sense intuitively, in that students' ability to assess another piece of work accurately depends on whether they understand the question, and if they have answered a question completely they may not appreciate the value of a partial solution given by another student. Piech et al [9] argue that assigning separate precision/variance to each reviewer is only valuable when reviewers are undertaking large numbers of reviews (10 or more), but we have found it important to identify students that do not grade accurately, not least because they may associate the grading with the wrong work. This is more likely to happen in programming assignments because the files have to be taken away from the system to be assessed, rather than reading and grading in the same web environment — as can be done with essays, which are the subject of nearly all the other peer review bias work.

Once the model is built, values for all of the parameters and hyperparameters need to be found that give the best fit of the observed data. For anything but the simplest model this is impossible to calculate analytically, and with a very high number of parameters the search space has very high dimensionality. Markov chain Monte Carlo simulation is often used via the Metropolis-Hastings algorithm, in what is known as a Gibbs sampler. The basic idea is to construct a high-dimensional random walk (chain) in the space of parameters (including hyperparameters), where the walk is constrained to areas of high probability (good fit) by usually moving to a new state only if the fit is better. Values for each of the parameters are calculated by finding the mean of the parameter over the length of the chain of values of the random walk. To validate that the model is working well, there should be good correlation between the values achieved by chains starting at different positions — we analyse this, along with other characteristics of the model, in Section 5.

It is also worth noting that the scores given by reviewers in our process are integers, whereas the bias and quality scores are continuous. It is often argued that it is inappropriate to use continuous measures or ordinal data, e.g. taking averages of Likert-scale scores. However in our case, although the scores awarded are not continuous they have more structure than ordinal data in that numerical distance (in marks) between each set of subsequent categories is equal in most cases (see Section 4.2). If this were not the case then more care should be taken in justifying the ordinal data to be modelled continuously.

4 PROCESS

4.1 Context

The study took place in a small UK university (14,000 undergraduate students) with entry requirements for computer science placing it in the top ten universities in the UK. Two programming-based assignments in a first-year HE programming module were used. The module was not an introductory programming module — 85% of the students reported that they already had some experience of programming before taking the module. JavaScript was the language used in teaching the module, and this was the first cohort using this language (previously it had been Java). Students who needed it were introduced to basic programming constructs in Python through another module, and the start of the module under study focussed on software engineering tools rather than the programming language itself. In particular git and github were used to work collaboratively on HTML and CSS for web-sites, only moving on to programming in JavaScript once the foundations were covered elsewhere. Most of the students taking the module (140/160) were studying single-honours computer science, with the others taking at least a third of their first-year modules in computer science. 28% of students originated from outside the UK and 17% identified as female.

Both assignments allowed quite a lot of flexibility, to encourage creativity and to ensure divergence of solutions to support plagiarism detection. The first assignment was based around adapting a JavaScript/processing sketch (chosen individually by the students from www.openprocessing.org/), creating a reusable class and demonstrating its use embedded in an HTML page, with controls linked to DOM inputs. There were five equally-weighted dimensions, namely

Usability of code Appropriate parameterisation including defaults; Encapsulation; Useful methods including draw

Development of original Work done in refactoring code to class; Work done in useful parameterisation; Work done in extending scope

Quality of example HTML page is valid; Appropriate on-page instructions; Appropriate on-page controls (form)

Quality of documentation All methods and parameters explained (including constructor); Explanation of example; Source of initial code acknowledged (including licence)

Code quality According to a defined set of rules for ESLint

The second assignment required students to construct a dynamic web-site as a single-page style application with an API to a server written using JavaScript/nodejs. Again there were five equally-weighted dimensions

Client-side functionality User Experience (UX); App complexity; 'Single page' style: asynchronous updates

Client-side quality Standards compliant (HTML5); Responsive to different viewport sizes; Gracefully handles server disconnection; Web site documentation

Server-side functionality More than one entity type; REST API provides each entity with appropriate GET/POST methods; npm to install and start

Server-side quality Successful ESlint; Successful jest tests with good coverage; Testing includes content-type and HTTP code; API documentation

Extensions These were not taught directly in the lecture course, but left for the students to research independently: Successful cloud deployment; Successful integration with remote web service

4.2 Peer Review

Students were each assigned a certain number of pieces of work to review: three for the first assignment and four for the second. This number was selected to balance the reliability of the grade against the amount of reviewing effort required of students. These two features may well be related, as if reviewers have too many pieces of work to review then they may spend less time on each, potentially reducing their accuracy. For the second assignment students were polled anonymously in-lecture (in between the first and second assignments) as to how many reviewees they should each receive, and the results are summarised in Figure 1. The most popular choice was for three reviewees, as used in the first assignment, but the median response was four, so that was the number assigned for the second piece of coursework.

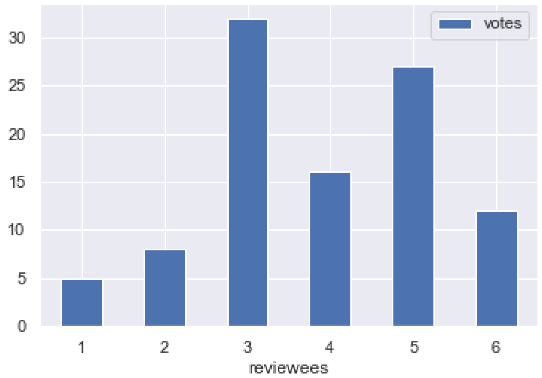


Figure 1: Student opinion on the ideal number of reviewees

After submission of the assignments, students were presented with their allocated reviews via a custom web-based system, and for each review were asked to provide a comment and a grade in each of the rubric dimensions. In the literature there is quite a variety of grading ranges used, and there is no consensus or much evidence to suggest what is best (except that a three point scale was not fine enough [12]). One of the challenges students face is that at university they receive (on average) much lower marks than they had at school. An average of 65% for a module is considered fair, but many of our students will have been used to getting marks in the 90s. In order to help them get accustomed to the new scheme, they were asked to grade according to (a slightly simplified form of) the university marking and classification conventions, which identify grade, mark ranges and generic assessment criteria. For the second assignment a grade of 'Perfect' (100%) was added. The descriptive criteria and mark ranges used are shown in Figure 2. Note that the pass mark for a module is 40% and the 'first class' classification boundary is 70%. As an example, the text of the generic criterion for 'Acceptable' is

> The work examined is acceptable but provides significantly restricted evidence of the knowledge, understanding and skills appropriate to the Level of the qualification. There is also acceptable but significantly restricted evidence showing that all the learning outcomes and responsibilities appropriate to that Level are satisfied.

Students were not required to award a percentage mark for each dimension, but rather one of the grade descriptions, which amounted to a 11 or 12 point scale (for the first and second assignment respectively). Students were given one week to complete their reviews, and they were informed that 5% of the module marks would be available for each set of peer reviews completed (1 set = 1 assignment), so that overall 10% of the module marks relied on the completion and quality of their peer reviews. This was to ensure that students engaged well with the peer review process, and were rewarded for completing high quality reviews. In practice all of the students that submitted work for review also completed their peer reviews. Peer reviews were carried out anonymously, and because each submission contained quite a few files, these had to be downloaded and reviewed away from the peer review system itself, which occasionally led to mistakes by reviewers in associating comments (and presumably grades) with the wrong piece of work.

Score	Grade	Mark range
1	No submission	0
2	Unacceptable	1-19
3	Mostly Unacceptable	20-29
4	Mostly Acceptable	30-39
5	Acceptable	40-49
6	Sound	50-59
7	Good	60-64
8	Very Good	65-69
9	Excellent	70-75
10	Outstanding	76-85
11	Exemplary	86-100

Figure 2: Generic assessment criteria and mark ranges

4.3 Feedback

Once the peer reviews were completed they were first examined to check that the range of marks was reasonable. The distribution of the raw review grades (combined for both assignments) is shown in Figure 3.

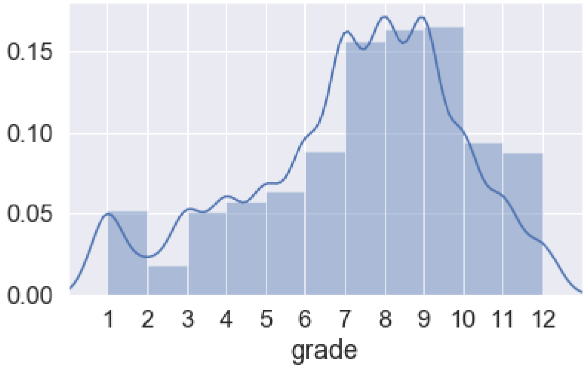


Figure 3: Distribution of raw scores

Marks for each grade were calculated initially using a mid-point of the mark range, and then marks for each dimension were calculated as the average of the marks from all reviewers. Overall the mean mark was just above 62%, comfortably in the reasonable range. Then the grades were analysed using the Bayesian model described in Section 3. Once the model had been checked for convergence (see Section 5) values for quality of work (reviewee), and bias and precision (reviewer) were extracted from the sample chains by finding the mean (after a burn-in period and with thinning).

The distribution of the calculated quality scores is shown in Figure 4.

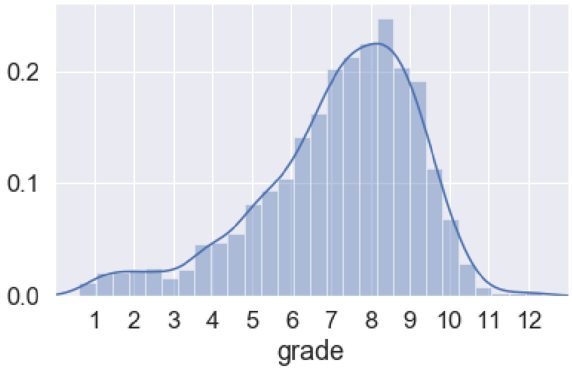


Figure 4: Distribution of quality scores after analysis

It is worth noting that there is a marked smoothing of the adjusted scores at the bottom end of the range, as opposed to the raw marks. Although the original grades were integer scores, the model outputs fractional values for the quality. Also there are some quality scores that are less than 1, which corresponds to a mark of less than 0 (no submission) so the quality scores had to be checked by hand at the bottom end to ensure that none of the marks were negative. Marks close to 'non-submission' were rounded to 0, as slight variations in the bias would give marks of +/- 2% which, while not significant when averaged over the five dimensions, would look strange in the feedback. The mark awarded was a linear combination of the mid-range mark for the grades above and below the quality score. The choice of which point in the range to choose as the representative mark means that there is some flexibility in adjusting the average mark for the cohort as a whole by at least +/- 2.5%, given that that the narrowest grade band is 5%. This fits well with the 3% variation reported by Pare and Joordens [8] when they compared student peer review marks with expert marks. Checking that the overall mark range was correct was done by faculty moderating a sample of (10%) of student submissions.

Each student then received a report showing all of their reviewer comments, plus the grade and the bias score for the reviewer. The aim here was to show that, even if the grade awarded varied between students, any variation between the reviewers had been identified by the system. Finally the mark corresponding to the quality sore for each dimension was shown, along with the overall mark for the assignment, which was just the average of the marks for each dimension.

As well as feedback on their own work, students were given feedback on their reviews. Of the 5% available, 3% was awarded simply on the basis of having completed their peer reviews. The other 2% was awarded on the basis of the average absolute bias over the dimensions. Figure 5 shows the distribution of the biases calculated through MCMC. The vast majority of bias scores have magnitude less than 2. With most scores falling in the central range where the mark band width is 5%, this means that the vast majority of scores fell within 10% of the centre. The lower and upper quartiles for the bias were -0.59 and +0.57 respectively. Based on this distribution, students were awarded full marks (2%) if their average bias was less than 0.5 and half marks (1%) if their average absolute bias was between 0.5 and 1.0.

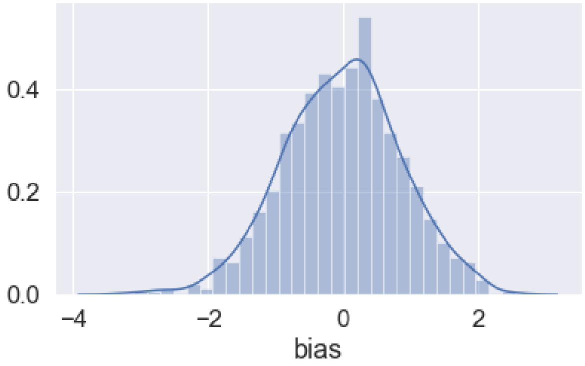


Figure 5: Distribution of student biases

Finally the students were shown their own reviews, including bias scores, alongside reviews done by other reviewers of the same pieces of work, with the bias of their co-reviewers also displayed. Note that although the data presented here are for all peer reviews of both assignments, they were carried out sequentially, with all students receiving their first set of peer feedback before starting their second assignment.

After receiving their marks for the assignment, students were given an opportunity to question the mark they received, on the understanding that if they challenged without good reason they they would put some of their peer assessment marks at risk. Most notable of these were where reviewers had clearly got mixed about about which piece of work they were reviewing, and indicated both by the comments and the grade. It was very pleasing to note that in these cases the reviewer precision was extremely low, so that their grade had virtually no impact on the final mark. However, this was not clear to the students under review, as all they had reported were the grade, the bias and the final mark (derived from the calculated quality). It would be beneficial to have some way of explaining to the students how the precision of the reviewers impacts on the calculated quality, but this is not direct, and challenging to do given the very wide range of precision values.

5 RESULTS AND EVALUATION

RQ1: Reliability improvement through statistical modelling

The first question to address is whether appropriate values can be found for the parameters in the Bayesian model of Section 3 with the review data described in Section 4. A simple way to check this is to look at the correlation of two independent chains, from two different starting points. Figure 6 shows how the correlation between quality scores of the two chains increases as the number of iterations of sampler is increased, and that strong correlation is achieved by 100,000 iterations.

This is substantially more than the 3,000 iterations reported by Goldin [5], but that model is much smaller, based on only 28 students rather than the 160 that we have. Uto and Ueno [15] used 30-50,000 samples, although with a substantial burn-in of 30,000 samples, which is very similar to our results. So we have a model that converges satisfactorily, but does it offer improved reliability?

Without taking into account bias, a natural way to find an estimate for the quality of the work is to take the mean of the peer review scores. With a very large number of reviews we can expect the underlying quality to emerge (as confirmed by Piech et al. [9]) but we would like to get a better indicator of the quality with a smaller number of reviews. By looking at the error between the individual reviews against the mean of all reviews of the same piece of work, we have a measure of how well the individual reviews fit. This is shown in the top line of Figure 7 which compares these errors with the errors for the bias-adjusted means vs. the bias-adjusted scores. Visually what we are looking for here is the vertical spread of the scores with the same mean: the broader the spread the larger the error. Because the raw scores are all integer values, a small amount of vertical jitter has been added to the raw plot, to ease comparison. The individual points are plotted with relatively low opacity so that it is clearer where points are clustered closely together: higher opacity of means higher density of points. The adjusted scores clearly have lower errors — combining all of the dimensions, the RMS error reduces from 1.57 for the raw data to 1.16 for the bias-adjusted data.

What is also clear from Figure 7 is that the errors are different for different dimensions of the rubric, with the spread tending to increase from left to right. For this second assignment the earlier dimensions build more directly on the outcomes of the first assignment (which focussed on client-side JavaScript) whereas the later dimensions are to do with server-side programming that was not covered in the first assignment. Perhaps it is not surprising then that students were more variable in their grading for the later parts. The last section has particularly wide spread, which again is in line with intuition as this is extension material that was not covered directly in lectures but left for the students to research and complete independently. This provides support for our model in which bias and precision are modelled independently for each of the rubric dimensions. In all cases, however, the bias-adjusted score has a narrower spread than the raw score, so we expect to have a better measure of the underlying quality with fewer reviews carried out.

Our model does not only adjust for bias though, it also accounts for the precision of the reviewer. Some quite justifiably argue that more able students, i.e. those with higher scores, are more likely to have good precision as they understand the problem well, and so build in to their model explicitly a relationship between the accuracy of the reviewer and their score. However, it is perfectly feasible for different students to approach review with different levels of diligence independent of the quality of their own work — indeed, when quality of peer review contributes to the final mark for the assignment, students who anticipate receiving low marks for their own work may be motivated to put more effort into their peer reviews.

Because the precision affects the scores only implicitly through the model, it is harder to demonstrate the effect it has on the accuracy of the quality scores. To demonstrate the effect that it has, beyond accounting for bias, Figure 8 demonstrates how the reviewers identified as having higher precision give reviews that have lower error. This implies that the calculated quality score weights the low error scores more heavily. There is a very wide range of values for precision, so to make the detail easier to take in, Figure 8 separates colours the scores according to the precision (τ) quartile of the reviewer. These scores are combined for all dimensions of both assessments, and are included in a plot of bias-adjusted scores against bias-adjusted means, which we have already shown to have lower error than the raw scores vs means, which are included in the top of the figure for comparison.

In summary, we have answered RQ1 affirmatively: our statistical model improves the reliability of peer assessment.

RQ2: How is peer assessment accuracy of programming coursework related to characteristics of the reviewer?

The reviewer characteristic most commonly suggested to influence bias is the capability of the reviewer, which we measure by their quality score. Figure 9 shows student reviewer bias plotted against quality (as a reviewee) for each of the dimensions of the second assignment.

The lines of best fit (with confidence intervals shaded) show that where there is a trend, is it that more able students mark more harshly. The effect is not significant in all cases (significance at 95% is indicated by the confidence areas of the line of best fit both sloping downward), but in the dimensions which are most significant (client-side and server-side quality) the size of the effect is about +/-0.5 bias points, which would usually correspond to a mark difference of about 5% between the highest achieving and lowest achieving students. The difference is real and significant, further justifying the multi-dimensional modelling of bias and precision. It is arguable that a Bonferroni correction should be applied when looking at the significance of each of these. Figure 10a shows the correlation on all dimensions of both assessments combined, where the effect is significant but relatively small at 0.047 bias points per grade, or about 5% mark difference between the highest achieving and lowest achieving students. The result is significant with a p-value of 0.000048.

Looking separately at the precision τ and how it varies across students, firstly it is notable that there is a very wide range of values, so we look at $\log \tau$ instead. There is no observed correlation if we plot the quality of a reviewer's work and their precision, but Figure 10b shows that there is a small positive correlation between the bias and the precision — it is notable that this effect runs in the opposite direction to the assumption made elsewhere that good students are both more precise and harsher in their marking. We found that students who tend to mark harder are less precise, independently of their own ability. Given that we have taken logs

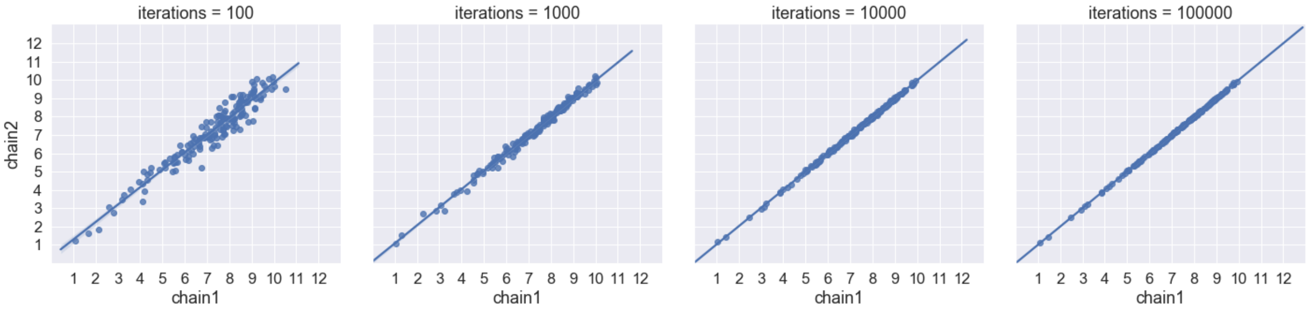


Figure 6: Correlation of quality parameters between separate MCMC chains as number of iterations increases

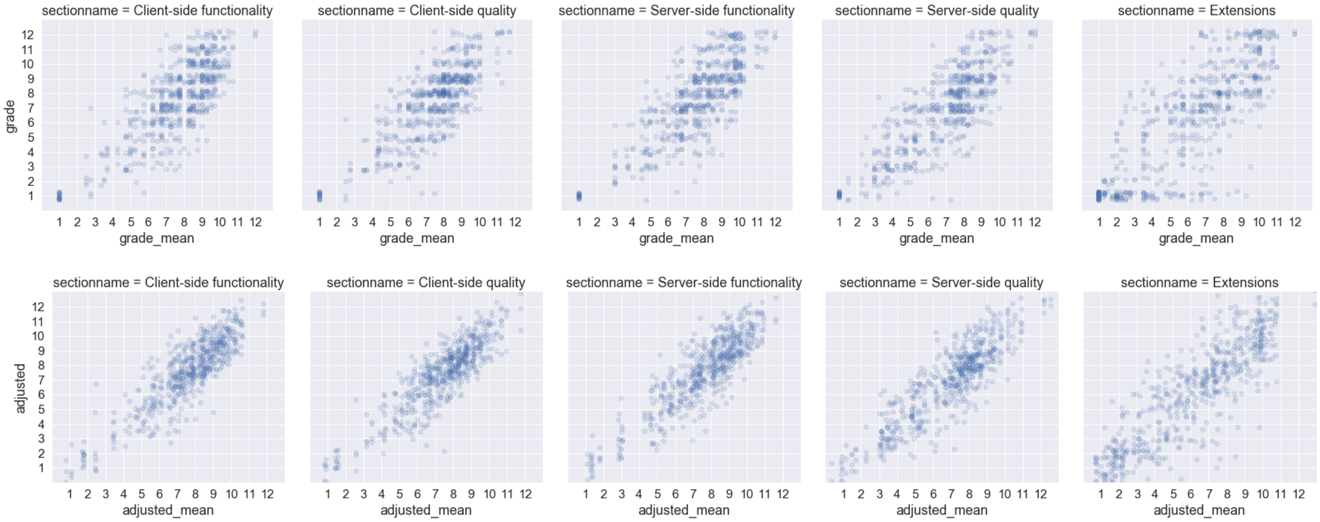
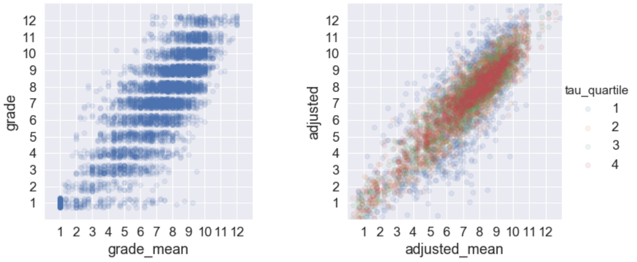


Figure 7: Individual scores plotted against mean scores for assignment 2. The top row is raw marks, the bottom row is bias-adjusted marks. The columns correspond to different dimensions of the rubric for assignment 2.



(a) Raw scores **(b) Bias-adjusted scores,**

Figure 8: Colour-coding of adjusted scores according to precision (tau) quartile

before applying a linear regression, precise measures of significance that we could derive would be of questionable value.

In terms of computing-specific content it is noticeable from the text comments in the peer review that some students interpret code quality more widely than is outlined in the rubric. For example in the first assignment the rubric for code quality referred solely to whether or not the Javascript code passed a defined set of rules under ESlint (a lint tool specifically designed for JavaScript), whereas some students, probably more experienced ones, provided a broader range of comments about variable naming, code commenting etc which were outside the scope of the rubric and hence less precise. This could explain why more able students might be harsher (as expected) but less precise in some cases, alongside our earlier discussion of motivation for less confident students to carry out accurate peer reviews. The effect was more noticeable in the first assignment, so students may have learned from their experience to give more precise reviews.

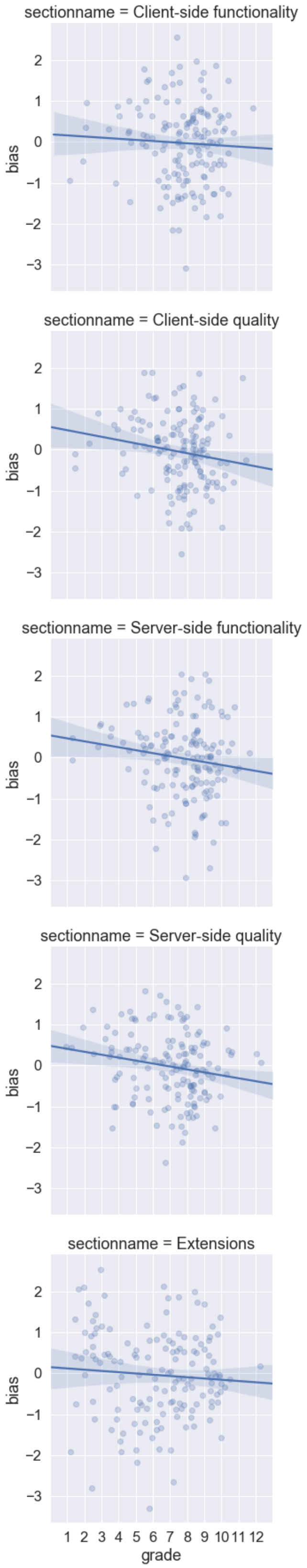
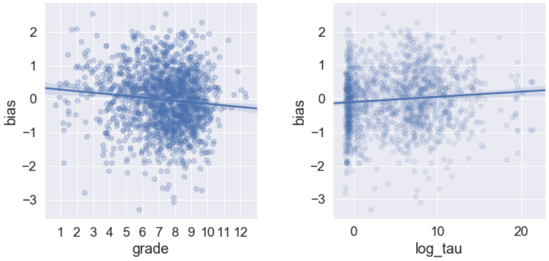


Figure 9: Student reviewer bias against quality of work. The rows correspond to different dimensions of the rubric for assignment 2.



(a) Bias against quality **(b) Bias against precision**

Figure 10: Variation of bias with reviewer characteristics

6 CONCLUSIONS

We have been able to answer both research questions positively. For RQ1 we have demonstrated how our statistical model can effectively increase the reliability of summative peer assessment. Using a Bayesian model, which we have found to converge after 100,000 iterations of MCMC, we have shown that adjusting the scores according to bias reduces the associated errors over the raw scores. Taking into account the precision of the reviewers further enhances the reliability of the measure. Previous work in other subject areas has had different findings as to how the model should be best constructed: earlier work in computing suggested combining bias and precision into a single measure of weighting [6], however we have found that bias and precision should be modelled separately. There are particular reasons from the nature of the subject (i.e. programming) for this, at least in the way that we have assessed it: that the different dimensions of the rubrics do seem to have different characteristics; and that assessing the work away from the web environment makes misattribution errors (easily identified by modelling precision) more likely to occur than when marking essays on screen. In fact we did find that in the few cases that reviewees identified of their reviewers as looking at the wrong piece of work, the corresponding precision mark was extremely low, virtually eliminating their impact.

Looking at RQ2, we have identified that, in common with other studies, students who do better in an assignment tend to mark their peers slightly harder (a small effect with high statistical significance). Contrary to the hypotheses of others (we have not seen any experimental validation of their ideas) we did not find any correlation between grade (as a reviewee) and precision (as a reviewer), but rather that generous markers tended to have slightly higher precision.

Alongside the statistical review of the grades, we also reflected on the process in light of student feedback on the module. A detailed analysis of the comments is beyond the scope of this paper, but some pointers may be helpful to others that are interested in implementing summative peer review within programming.

- Explain to students in advance the process and its pedagogical/ professional benefits.
- Only assign peer reviews to those who have submitted work. Alternative mechanisms have to be in place for assessment of students who submit late for whatever reason.

- Whilst for very large classes (greater than 300) there may be substantial time savings for faculty, our experience was there was little, if any, time saved the first time around. Before sending work out for review it all had to be checked for anonymity (this was very important to the students) and completeness
- Giving students feedback on their feedback is crucial — commonly raised concerns about students not completing their reviews were unfounded in our context, at least in part because the students knew they were being marked on it.
- Make the presentation of feedback (including bias) as transparent as possible to the students
- Students are uncomfortable with having their feedback based solely on other students opinions, as has been found in other studies. This was the main concern that students raised, although there were very few instances of students suggesting that their final mark did not reflect the quality of their work. Combining peer assessment with faculty assessment (perhaps on different dimensions of the same rubric) could help.
- Students like creativity in assignments. Whilst there were a couple of comments about assignment specifications being 'vague', there were more positive comments on creativity.
- Given the chance, a small minority students did want to challenge the marks they received, and offering this opportunity seemed to be welcomed.

6.1 Further Work

Further exploration of the reliability of the statistical approach could be carried out using synthetic data [15], generated in line with the statistical model and based on hyperparameters extracted from our real data. Sensitivity of the model to various types of 'rogue' reviewer [6] could be investigated, as well as a more detailed characterisation of the effect of the number of reviewers. Sensitivity to the number of reviewers could also be carried out by looking at the correlation of derived quality scores between analyses with different numbers of reviewers. We would expect the correlation to converge as the number of reviewers increases. Whilst we argue that both bias and precision should be modelled independently and multi-dimensionally, the more parameters the model has, the more potential there is for overfitting. Statistical measures such DIC could be looked at directly on our models as they have on others [5]. For further verification of reliability, content of text comments could be compared with grades, through natural language analysis [11].

Other studies [7] have found no correlation between student demographics and acceptability of peer review, but with the changing makeup of cohorts this would merit further study.

Finally, we have focussed on reliability, but not addressed validity, on the assumption that using coursework rather than exams is more valid. However it would be useful to carry out a more detailed examination of how expert graders and students differ in the fundamentals of their assessment (validity), through moderation and discussion of a small number of pieces of work, rather than looking at statistical means of reducing noise in the measurement (reliability).

ACKNOWLEDGMENTS

Thanks to Prof Peter Craig for advice on Bayesian modelling. This research was carried out with ethical approval from Durham University COMP-2019-06-12T09:18:26-dcs0spb

REFERENCES

[1] Steven Bradley. 2016. Managing Plagiarism in Programming Assignments with Blended Assessment and Randomisation. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16)*. ACM, New York, NY, USA, 21–30.

[2] Nancy Falchikov and Judy Goldfinch. 2000. Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks. *Review of Educational Research* 70, 3 (Sept. 2000), 287–322.

[3] Scott Freeman and John W. Parks. 2010. How Accurate Is Peer Grading? *CBEâĂŤLife Sciences Education* 9, 4 (Dec. 2010), 482–488.

[4] Carlos GarcÃŋaâĂŘMartÃŋnez, Rebeca Cerezo, Manuel BermÃždez, and CristÃşbal Romero. 2019. Improving essay peer grading accuracy in massive open online courses using personalized weights from student's engagement and performance. *Journal of Computer Assisted Learning* 35, 1 (2019), 110–120.

[5] Ilya M. Goldin. 2012. Accounting for peer reviewer bias with Bayesian models. In *Proceedings of the Workshop on Intelligent Support for Learning Groups at the 11th International Conference on Intelligent Tutoring Systems.*

[6] John Hamer, Kenneth T. K. Ma, and Hugh H. F. Kwong. 2005. A Method of Automatic Grade Calibration in Peer Assessment. In *Proceedings of the 7th Australasian Conference on Computing Education - Volume 42 (ACE '05)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 67–72.

[7] C. E. King. 2018. Feasibility and Acceptability of Peer Assessment for Coding Assignments in Large Lecture Based Programming Engineering Courses. In *2018 IEEE Frontiers in Education Conference (FIE)*. 1–9.

[8] D. E. ParÃľ and S. Joordens. 2008. Peering into large lectures: examining peer and expert mark agreement using peerScholar, an online peer assessment tool. *Journal of Computer Assisted Learning* 24, 6 (2008), 526–540.

[9] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. 2013. Tuned Models of Peer Assessment in MOOCs. In *arXiv:1307.2579 [cs, stat]*. arXiv: 1307.2579.

[10] I. Prazina and V. OkanoviÄĞ. 2019. Methods for Double-Blind Peer Review and Grade Prediction of Student Software Projects. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 693–696.

[11] Juan RamÃşn Rico-Juan, Antonio-Javier Gallego, and Jorge Calvo-Zaragoza. 2019. Automatic detection of inconsistencies between numerical scores and textual feedback in peer-assessment processes with machine learning. *Computers & Education* 140 (Oct. 2019), 103609.

[12] Jirarat Sitthiworachart and Mike Joy. 2004. Effective Peer Assessment for Learning Computer Programming. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '04)*. ACM, New York, NY, USA, 122–126.

[13] Ana SÃąnchez, CÃľsar DomÃŋnguez, Jose Miguel Blanco, and Arturo Jaime. 2019. Incorporating Computing ProfessionalsâĂŹ Know-how: Differences Between Assessment by Students, Academics, and Professional Experts. *ACM Trans. Comput. Educ.* 19, 3 (May 2019), 26:1–26:18.

[14] Keith Topping. 1998. Peer Assessment Between Students in Colleges and Universities. *Review of Educational Research* 68, 3 (Sept. 1998), 249–276.

[15] M. Uto and M. Ueno. 2016. Item Response Theory for Peer Assessment. *IEEE Transactions on Learning Technologies* 9, 2 (April 2016), 157–170.

[16] Mark E. Walvoord, MariÃńlle H. Hoefnagels, Douglas D. Gaffin, Matthew M. Chumchal, and David A. Long. 2008. An analysis of calibrated peer review (CPR) in a science lecture classroom. *Journal of College Science Teaching* 37, 4 (2008), 66.

[17] Yanqing Wang, Hang Li, Yuqiang Feng, Yu Jiang, and Ying Liu. 2012. Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education* 59, 2 (Sept. 2012), 412–422.