Self-consistent Peer Ranking for Assessing Student Work
Dealing with Large Populations

Kees van Overveld[1] and Tom Verhoeff[2]

[1] *Dept. of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, Eindhoven, Netherlands*

[2] *Dept. of Mathematics & Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands*

Keywords: Large Scale Assessment, Peer Reviewing, Ranking Algorithm.

Abstract: Assessing large populations of students puts a serious burden on teaching staff capacity. For open-format assignments, automation of the reviewing process can offer only limited support. Peer ranking is a partial solution to the problem, with the added benefit that students' critical reading skills are developed. We see two remaining problems, however: (1) for students, it is a major challenge to assign marks on an absolute scale, and (2) students' competence in reviewing may vary significantly—so not all peer reviews should have a similar weight in the process. To remedy these shortcomings, we suggest an approach to peer ranking, inspired by Jon Kleinberg's HITS-algorithm, where both the students' assignment results and the quality of their double anonymous peer reviews are algorithmically ranked. Based on preliminary model calculations, we estimate that this strategy may reduce the required effort for reviewing open-format assignments approximately by a factor of ten. A first large-scale pilot with this method will take place in undergraduate courses at Eindhoven University of Technology, spring 2013. Since this involves about 900 students, automated support is a must. We describe the peer reviewing facilities that were introduced in our web-based education support system named peach[3].

1 MOTIVATION AND PROBLEM DEFINITION

Assessing large populations of students puts a serious burden on teaching staff capacity. This is even more so if strict deadlines need to be observed with respect to providing feedback to students. In a practical scenario, set at Eindhoven University of Technology in early 2013, some 900 students will be submitting elaborations of homework assignments, each corresponding to about two A4 pages of text, in a weekly rhythm, where marks need to be provided no later than two weeks after submission, and no more than two staff members are available for reviewing.

If reviewing a single work is estimated to take 20 minutes, completing the entire correction takes 300 person hours, or 150 hours per individual teacher. Although one week contains $24 \times 7 = 168$ hours, it is obvious that straightforward reviewing is no option.

Peer reviewing, i.e., students reviewing each other's work using a protocol that ensures anonymity, seems a plausible first option (Sadler and Good, 2006; Lu and Bol, 2007). A naive scheme, however, where students give marks to their peers, suffers from two obvious drawbacks:

1. Unless the assignments admit only a single correct answer, there is subjectivity involved in marking. In the current casus, the assignments are deliberately open ended. They contain questions of the form 'give an example for X', 'give some arguments in favor of, and some arguments against Y', or 'what is your substantiated opinion regarding Z'. Although a student can be expected to form a global opinion ('this is quite good'), we ask too much if this opinion should be made quantitative, say, on a 10-point scale.
2. More importantly, not all students can be expected to be equally competent reviewers. This problem could be mitigated by having every work reviewed by sufficiently many students, so that non-systematic errors can be expected to average out. This will not work in practice, however, since it is unrealistic to have students review more than, say, five works each.

Problem 1 is partially solved by having students merely *rank* works, that is, to put the (say) five works they review in order of quality, rather than to give ab-

van Overveld K. and Verhoeff T..
Self-consistent Peer Ranking for Assessing Student Work - Dealing with Large Populations.
DOI: 10.5220/0004352903990404
In *Proceedings of the 5th International Conference on Computer Supported Education* (CSEDU-2013), pages 399-404
ISBN: 978-989-8565-53-2

solute marks. From the methodology of social sciences (Mellenbergh, 2011), it is known that comparative ranking is generally easier than absolute ranking. We use the term "peer ranking" (following (Allain et al., 2006)) for comparative ranking in the context of peer review.

Peer ranking, however, does not completely solve Problem 1: as part of the assessment process, our students need an absolute marking.

The research question of this paper, combining Problems 1 and 2, is now stated as:

> 'How can peer ranking be used, taking differences in students' reviewing competences into account, in order to obtain absolute marks in assessments?'

For peer ranking that accounts for differences in reviewing competences among peers, we coin the term 'self-consistent peer ranking'.

In Section 2, we formally define self-consistent peer ranking and an approach to it, loosely based on Jon Kleinberg's HITS algorithm (Kleinberg, 1999). Some implementation details are described in Section 3. Prior to the actual implementation in a real-life setting, we want to gain some feeling for the merits of the approach. Therefore, we performed a model study; this is discussed in Section 4. Section 6 lists a number of possible variations of the method, Section 5 discusses the conditions for application of the algorithm in an educational context, and Section 7 discusses the web-based support facility peach[3]. Finally, in Section 8 we summarize our conclusions and indicate directions of future work.

2 PROPOSED APPROACH

The problem of ranking the quality of submitted works, based on judgments by reviewers with unknown and varying reviewing competence, somewhat resembles the problem that Google is solving by means of *page ranking*:

- a web page is *good* if many web pages link to it;
- not every link should contribute equally to the 'goodness' of a webpage;
- a link from a *good* webpage should contribute more;
- this gives a cyclic definition of what constitutes 'good' for web pages.

In the case of peer reviewing, the reasoning goes:

- a student's work is *good* if peers have a high esteem of it;
- not every peer's opinion should contribute equally to the 'goodness' of a work;
- the opinion of a *competent* peer should contribute more;
- this gives a cyclic definition of what constitutes 'good' (for works) and 'competent' (for peers).

The definitions for the goodness of a work and the competence of a peer can now be given formally.

Students have a *reviewing competence*, called c_i for student number i, $i = 1 \ldots N$. Competences are initially unknown.

Works have a *quality* ('goodness'), called q_j for work number j, $j = 1 \ldots N$. Note that a q_j is not necessarily a final grade; that is, once we have an estimate for q_j, we still have the problem of converting it into a grade. Qualities are initially unknown. Review competence and quality of work are assumed to be independent variables.

An assessment where student i reviews work j produces an *indicator*, called a_{ij}. A larger a_{ij} value means that student i rates work j as better. The indicator a_{ij} gives information both about student i and work j. Again, this is not necessarily a grade. When a collection of a_{ij} is known, the challenge is to recover the c_i and the q_j.

For a first, naïve approach, we treat c_i and q_j symmetrically; we scale them between -1 and 1; we assume a full set of a_{ij} (that is, every student has reviewed every work), and we prepare the a_{ij} so that they are also scaled between -1 and 1. The values c_i, q_j, and a_{ij} are called *self-consistent*, when (a) the q_j are the weighted averages of the a_{ij}, where the c_i are the weight factors, i.e. $q_j = \sum_i a_{ij} c_i$, and (ii) similarly with the roles of c_i and q_j reversed, i.e., $c_i = \sum_j a_{ij} q_j$. The following algorithm, if it converges, produces a set of c_i and q_j that are self-consistent for given a_{ij}.

1. Initialize all c_i to random values between -1 and $+1$.
2. Calculate first estimate $\forall_j : q_j^0 = \sum_i a_{ij} c_i^0$.
3. Update $\forall_i : c_i^{n+1} = \sum_j a_{ij} q_j^n$.
4. Update $\forall_j : q_j^{n+1} = \sum_i a_{ij} c_i^{n+1}$.
5. Renormalize c_i and q_j to keep them between -1 and $+1$.
6. Repeat steps 3 through 5 until convergence, that is, $c_i^n \approx \sum_j a_{ij} q_j^n$; and $q_j^n \approx \sum_i a_{ij} c_i^n$.

This algorithm is in fact a so-called *power iteration* (Golub and Van Loan, 1996). Power iteration converges under weak conditions. Indeed, in case of convergence, $q = AA^T q$ holds, where q is a vector of q_j, and matrix A holds all a_{ij}. We see that q is an eigenvector of the positive-definite AA^T; the repeated scaling ensures that the largest eigenvalue is 1, and power iteration is a well-known stable route to find the eigensystem with the largest eigenvalue for positive-definite matrices.

The above algorithm has the same structure as Jon Kleinberg's HITS algorithm (Kleinberg, 1999), used for self-consistent ranking of scientific citations. In the next section, we examine the modifications and additions needed to make the algorithm work for self-consistent peer ranking.

3 IMPLEMENTATION DETAILS

To apply the algorithm from the previous section to reviewing students' works, we have to resolve three issues.

i. If students' reviewing comprises *ranking* instead of *marking* their peers' works, we have to *construct a numeric value* for a_{ij} for every pair (student i, work j) from all orderings on the collection of works as found by all students;

ii. Since students will review and rank no more than, say, five works each, the majority of a_{ij} is unknown. If an unknown a_{ij} is represented by 0 (encoding a neutral judgment for work j by student i), the matrix A is sparse. We must *cope with the sparseness* of A;

iii. We demand that, eventually, students receive marks for their works on some given scale, say 0 through 10. The q_j only carry information in their ordering; hence, we have to *convert ranks to absolute marks*.

The resolution of these three issues is closely related. We start with item iii, then i, and finally item ii.

3.1 From Ordered q_i to Marks

After completion of the algorithm, we re-order the q_j so that they are monotonically increasing in j. Now that we have obtained the vector q, we know the order of the quality of the works. This means that the eventual marks should be such that the work $j = 1$ should receive the lowest mark, and the work with $j = N$ receives the highest mark. The marks of the other works could be obtained, for instance, by linear interpolation between these two. The mark m_k for work k then is given by

$$m_k = m_1 + (m_N - m_1)(k-1)/(N-1) \qquad (1)$$

So, with merely correcting two works, we can assign marks to all works.

To obtain a more reliable set of marks, however, we may prefer to have a few more works corrected and marked by teaching staff. In case more works are marked by hand, the interpolation could be more advanced: with four hand-corrected works, we might choose the numbers 1, $N/3$, $2N/3$, N and use a piecewise linear function or a spline in k for the interpolation instead of (1).

3.2 From Ranking Results to a_{ij} Values

Students each rank a small collection of works. The result of ranking by student i is equivalent to a set of relations, $a_{ij_1} < a_{ij_2}$ for j_1 and j_2 in the set of indices of works, reviewed by this student. We may optionally allow ex aequo ranking, that is $a_{ij_1} = a_{ij_2}$ for some maximum number of pairs (j_1, j_2). Ranking information can be encoded in an anti-symmetric $N \times N$ matrix, say S_i, where $+1$ occurs in entry (j_1, j_2) when, according to student i, $a_{ij_1} < a_{ij_2;}$, and -1 occurs in entry (j_2, j_1). All other entries are 0.

For example, if student i ranked $a_{ij_3} < a_{ij_1} < a_{ij_2}$, then we will have

$$S_i \quad = \quad \begin{array}{c|cccc} & j_1 & j_2 & j_3 & \dots \\ \hline j_1 & 0 & +1 & -1 & 0 \\ j_2 & -1 & 0 & -1 & 0 \\ j_3 & +1 & +1 & 0 & 0 \end{array} \qquad (2)$$

Next, all matrices S_i need to be aggregated to obtain the matrix A for the algorithm.

This aggregation is not trivial. For instance, the S_i need not all be mutually consistent. That is, an entry (j_1, j_2) may contain $+1$ in one of the S_i, whereas it is -1 in another $S_{i'}$. Now, prior to running the algorithm, the weights c_i are unknown. Still, it seems that the c_i are necessary to resolve conflicts due to inconsistencies. Therefore, for full self-consistency, the construction of A should take place simultaneous with obtaining c and q.

Although we plan to derive a fully self-consistent aggregation algorithm to obtain A from the matrices S_i in the future, we intend to run first trials with a simple approximation to this scheme. This approximation amounts to setting $a_{ij_1} = -1$ and $a_{ij_2} = +1$ for respectively the lowest and highest ranking works j_1 and j_2, according to student i, and to give the other works a_{ij} values that linearly interpolate these values. So, for five reviewed works per student, the a_{ij} are set to the sequence $-1, -0.5, 0, 0.5, 1$, *irrespective of any rank assignments by other students to these works.*[1] Constructing the a_{ij} from the initial ranking inputs in this way is obviously ad-hoc, and we will use it for a first trial only to see if the approach is promising.

[1] When we admit ex aequo ranking, one or more of the values may be left out of the sequence $-1, -0.5, 0, 0.5, 1$.

3.3 Sparse A

Convergence of the algorithm can be proven for full rank matrix A. Due to sparseness, however, A is highly rank deficient. Fortunately, power iteration is relatively robust. This means that, as long as a minimum percentage of a_{ij} is known, the algorithm still can approximately recover c, and, more importantly, q from A. There are two considerations, however, that we need to take into account.

- Obviously, the fraction of non-empty entries in A cannot be arbitrarily low. Therefore, given that each student reviews five works, the total population of students (to be called 'cluster') in one peer-ranking trial cannot be too high. In order to estimate the size of the largest allowable cluster, we perform a model study, described in the next section.
- With increasing cluster size, the convergence of our algorithm becomes increasingly problematic. 'Problematic convergence' implies the following.
  - We need more iterations (perhaps infinitely many) until convergence. This is no fundamental issue: it is easy to detect convergence; by admitting a maximal number of iterations, we can conclude if convergence fails.
  - With full rank, the solution of the power iteration algorithm is unique. This can no longer be proven for rank deficient A. This again is no fundamental issue, however: when we run the iteration several times with different starting conditions, we can easily verify if converged solutions are sufficiently close.[2]
  - If A gets increasingly rank deficient, the obtained vector q will contain increasingly more noise. This means that the *accuracy* of the algorithm decreases, where the accuracy is defined as the extent to which the found order of the works matches with the order as it would be found with hand-correction. The match between the hand-corrected order and the order found by the algorithm can be empirically assessed by doing a hand correction of the entire cluster. Small mismatches—that is, mismatches where the rank position of any q_j does not differ too much from a rank position as would be found with hand correction—can be partially compensated for by doing a larger fraction of hand corrections—to the extreme where all works are corrected by hand, and there is no added value of peer ranking. We plan to find the optimal cluster size, such that the accuracy of the algorithm is sufficient, by means of empirical assessment prior to full-scale implementation of the algorithm.

4 MODEL STUDY

To get a first, global, idea of attainable maximal cluster size, and hence the maximal efficiency improvement that can be attained by self-consistent peer ranking, we perform a model study. In this model, we postulated a relation between the c_i (student's reviewing competence) and the a_{ij} (the scores, attributed to works j by student i) as follows.

- A student with higher c_i contributes values for a_{ij} that are closer to the true q_j. By the 'true q_j' we mean the q_j that would result if a teacher would have reviewed work j.
- A student with lower c_i inputs values for a_{ij} that are closer to a uniform random number between -1 and $+1$. That is, failing competence is modeled as an unbiased noise term.

Next, to test the algorithm, we set up a collection of size N of works, every work with a *known* quality q_j, and a collection of size N of students, every student with a *known* reviewing competence c_i. Cluster size N will be varied to see what cluster sizes give acceptable accuracy, where the number of reviewed works per student is kept fixed to five. Known qualities and reviewing competences are taken randomly between -1 and $+1$. The known c and q are called c_{known}, q_{known}, respectively.

With c_{known} and q_{known}, the matrix a_{ij} is computed as follows. For every i, five random j's are selected such that every work j is 'reviewed' by exactly five different students i. The scores a_{ij} are calculated as

$$a_{ij} = \frac{q_{\text{known } j}(1 + c_{\text{known } i}) + \mathcal{R}(1 - c_{\text{known } i})}{2}, \quad (3)$$

where $\mathcal{R} = \text{rand}(-1,1)$ is a uniformly distributed random number between -1 and $+1$. All other a_{ij} are set to 0.

With matrix A set up in this way, the algorithm is run, and, if convergent, the resulting q and c are plotted against q_{known} and c_{known}. For ideal reconstruction, the graph should be monotonically increasing. The precise shape is determined by the normalization

[2] There is one curious subtlety. If q is a solution to $q = AA^T q$, then so is $-q$. Since the elements of q are scaled between -1 and 1, we cannot distinguish q and $-q$ beforehand. If the teacher reviews both extreme works (that is, after renumbering, the works with $j = 1$ and $j = N$), however, it should be immediately clear which of the two is the best and which is the worst. This unambiguously fixes the sign of q.

used. In our case, the normalization is a Euclidean distance norm, resulting in a roughly sigmoid shape for the graph.

Experiments reproduce the predicted behavior, where increasing sparseness in A causes increasing deviations from purely monotonic. If we find 5% deviations acceptable (that is, 5% of the works receive an out-of-order q_j), the matrix A can be as sparse as 10%. In other words, for a cluster size as large as 50, with five reviews per work, the algorithm is capable to find an approximation to the correct order with no more than 5% errors.

It turns out, however, that the outcome of these trials is sensitive to the assumptions with respect to the precise form of (3). If we assume students are slightly more competent in reviewing, the performance of the algorithm is drastically better; if students are less competent, the performance is considerably worse—which is not too unexpected. Therefore, although these model trials suggest a cluster size of 50 with five reviewed works per student, we may want to be a bit more conservative when we actually implement the scenario for the first time.

5 DISCUSSION

An algorithm for calculating students' reviewing competence and the quality of students' work may be a necessary ingredient of peer reviewing, but it is definitely not sufficient. In (van Zundert, 2012), educational considerations regarding peer reviewing are studied. In this section, we list a number of assumptions that should hold for an algorithm like the present one to be trustworthy.

- Peer groups should be unbiased and uncorrelated so that every assessment can be seen as an independent measurement of each student's performance. Careful randomization helps to remove correlations; bias is more subtle to deal with, though. For instance, in case of misconceptions, shared by a majority of the students ('homework is boring'), correct answers (such as 'homework is exciting') may score systematically low, and the algorithm has no means to detect this error. It will manifest itself in that the order, calculated by the algorithm, consistently differs from the order obtained by staff. In preparing assignments, therefore, questions with likely answers that are objectively 'wrong', but that could result from collectively shared misconceptions, should be avoided. Rather, assignments should be such that students can base their scores on how much detail is provided, how elaborate an answer is, how clearly the answer has been written, how convincing the answer is, et cetera.
- Peer ranking should be applied to a series of assignments rather than a single assignment, so that statistical evidence can be used to assess the reliability of the final outcome. Statistical evidence could be, e.g., the standard deviation σ of the marks over a series of assignments in one term. If σ decreases as one over the square root of the number assignments, N, it may be the case that the outcome indeed measures students' performance during that term. In case σ does not decrease with increasing N when averaging over the series of assignments, the per-assignment scores apparently do not measure the actual performance level of a student, and the peer review gives no information about this level. There could be various reasons for such inconclusive outcomes: perhaps the assignments do not accurately measure students' performance levels, or students performance levels vary wildly over the term. From a methodological point, it would be good to include the σ's in the final marks.

6 POSSIBLE VARIATIONS

We briefly present three possible variations.

1. The algorithm calculates both c and q from scratch, using the matrix A as only input. We may expect, however, that the students' reviewing competence will not vary much over time. This suggests to bootstrap the algorithm with the results in the first week, and use the found c as a first estimate in the next week. We may even consider to use the running average of the c's over subsequent weeks, representing the intuition that we get increasingly more accurate estimates of the individual students' reviewing competence.
2. Teachers may consider to have one or more 'example elaborations' to be, unknowingly, reviewed by the students. Since works are reviewed anonymously, students will not know that they review a teacher's work instead of one of their peers. Assuming that teacher's works have insurmountable quality, the associated q_j must keep a constant value of 1 during the iterations. Therefore, they serve to further stabilize the algorithm.
3. Despite the efficiency improvement offered by the algorithm, reviewing still requires works to be assessed by teachers, which takes time. To reduce waiting time for students, feedback can be given in three tiers. The first tier is immediately after the

raw ranking: a student then can be informed about the ‘five relative rankings among four other works that this student’s work received. Although this carries no absolute information, the difference between ‘five times number one’ or ‘five times number five’ is probably significant. The second tier is immediately after running the algorithm: students then can get a percentile score (‘85% of your cluster has lower scores than you’). Only the third tier feedback, where a student receives an absolute mark, needs to wait until teachers correct the few representative works per cluster.

7 WEB-BASED SUPPORT: peach^3

At Eindhoven University of Technology, we use a web-based education support system peach^3, since 2001 (Scheffers and Verhoeff, 2012). Students submit their work for deadlined assignments to peach^3 through a web browser. peach^3 monitors the deadlines, stores submitted work, performs configurable automatic checks on the content, disseminates it to those involved in the course, and allows entering of manual feedback and grades. Recently, we added support for peer reviews, including peer ranking.

To carry out a peer review of an assignment Z, a new assignment is created that is designated as a peer review of Z. Students who submitted work for Z are allocated random works by other students within their cluster, in such a way that each work is reviewed by a configurable number of students (in this paper, we have used five as bundle size). They read anonymized versions of work under review in a browser, and provide review reports, grades, and/or a ranking with respect to each other, through a web GUI. All review results can then be exported, processed, and imported back into the system as grade. Afterwards, if so desired, students can see anonymized review reports, grades, and rankings of their work.

8 CONCLUSIONS, FUTURE WORK

We propose a strategy for reducing the amount of reviewing, to be done by teachers, for open-ended assignments. An algorithm, called *self-consistent peer ranking*, requires students in a cluster of peers to anonymously rank, say, five peer works. The differences between students’ ranking competence (the c_i in the algorithm) are estimated, and used to compute a weighted final rank score (the order of the q_j in the algorithm). Next, teachers review the highest and lowest ranking work (and perhaps few more for increased reliability) in a cluster, to establish the absolute marks; marks of works not reviewed by teachers are found by interpolation.

A preliminary model study suggests that clusters can contain some 40 to 50 students, which would indicate a factor of 8 to 10 reduction of manual correction work, if students rank five works each, while the amount of out-of-order errors of the algorithm is no more than 5%. A group of 1000 students would then be split into 20–25 clusters.

A first field trial will take place early 2013 at Eindhoven University of Technology, involving about one thousand students. This will involve our web-based education support system peach^3, that provides support for peer reviews and peer ranking. If the results are promising, we will fine tune the cluster size and other parameters in the algorithm to get the optimally achievable efficiency improvement. Also, we will develop the algorithm further so that the matrix A can be obtained from the individual ranking inputs without having to resort to the ad-hoc assignment of a range of numerical values to the a_{ij} for given i.

ACKNOWLEDGEMENTS

We would like to acknowledge the helpful feedback of colleagues on a draft version of this paper.

REFERENCES

Allain, R., Abbot, D., and Deardorff, D. (2006). Using peer ranking to enhance student writing. *Physics Education*, 41(3):255–258.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. JHU Press.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Lu, R. and Bol, L. (2007). A comparison of anonymous versus identifiable e-peer review on college student writing performance and the extent of critical feedback. *J. of Interactive Online Learning*, 6(2):100–115.

Mellenbergh, G. J. (2011). *A Conceptual Introduction to Psychometrics: Development, Analysis, and Application of Psychological and Educational Tests*. Eleven International Publishing.

Sadler, P. M. and Good, E. (2006). The impact of self- and peer-grading on student learning. *Educational Assessment*, 11(1):1–31.

Scheffers, E. and Verhoeff, T. (Accessed Nov. 2012). peach^3. http://peach3.nl/.

van Zundert, M. J. (2012). *Conditions of Peer Assessment for Complex Learning*. PhD thesis, Maastricht University.