Reducing Difficulty Variance in Randomized Assessments

Paras Sud, University of Illinois, Urbana-Champaign

Paras Sud led this work as his thesis project for his B.S. in Computer Science from the University of Illinois at Urbana-Champaign. He's currently working in industry.

Prof. Matthew West, University of Illinois, Urbana-Champaign

Matthew West is an Associate Professor in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign. Prior to joining Illinois he was on the faculties of the Department of Aeronautics and Astronautics at Stanford University and the Department of Mathematics at the University of California, Davis. Prof. West holds a Ph.D. in Control and Dynamical Systems from the California Institute of Technology and a B.Sc. in Pure and Applied Mathematics from the University of Western Australia. His research is in the field of scientific computing and numerical analysis, where he works on computational algorithms for simulating complex stochastic systems such as atmospheric aerosols and feedback control. Prof. West is the recipient of the NSF CAREER award and is a University of Illinois Distinguished Teacher-Scholar and College of Engineering Education Innovation Fellow.

Prof. Craig Zilles, University of Illinois, Urbana-Champaign

Craig Zilles is an Associate Professor in the Computer Science department at the University of Illinois at Urbana-Champaign. His research focuses on computer science education and computer architecture. His research has been recognized by two best paper awards from ASPLOS (2010 and 2013) and by selection for inclusion in the IEEE Micro Top Picks from the 2007 Computer Architecture Conferences. He received the IEEE Education Society's Mac Van Valkenburg Early Career Teaching Award (2010), campus-wide Excellence in Undergraduate Teaching (2018) and Illinois Student Senate Teaching Excellence (2013) awards, the NSF CAREER award, and the Univerisity of Illinois College of Engineering's Rose Award and Everitt Award for Teaching Excellence. He also developed the first algorithm that allowed rendering arbitrary three-dimensional polygonal shapes for haptic interfaces (force-feedback human-computer interfaces). He holds 6 patents.

Reducing difficulty variance in randomized assessments

Abstract

When exams are run asynchronously (i.e., students take it at different times), a student can potentially gain an advantage by receiving information about the exam from someone who took it earlier. Generating random exams from pools of problems mitigates this potential advantage, but has the potential to introduce unfairness if the problems in a given pool are of significantly different difficulty. In this paper, we present an algorithm that takes a collection of problem pools and historical data on student performance on these problems and produces exams with reduced variance of difficulty (relative to naive random selection) while maintaining sufficient variation between exams to ensure security. Specifically, for a synthetic example exam, we can roughly halve the standard deviation of generated assessment difficulty levels with negligible effects on cheating cost functions (e.g., entropy-based measures of diversity).

Introduction

At many universities, introductory STEM courses are taught as large (200+ student) lecture courses which presents many challenges, but summative assessment is one of the most significant. While lectures and web-based auto-graded assignments scale gracefully, traditional pencil and paper exams present challenges in the form of reserving space, printing exams, proctoring, timely grading, and handling conflict exams [1–3].

To address this challenge of scale, some faculty are exploring alternative strategies to give exams. Some universities have developed computer testing centers [3, 4] where students can reserve a time to take their exam in a secure, proctored computer lab. Other faculty have elected to use a commercial online proctoring service for their exams. Because of their geographically distributed student populations, most MOOCs use online computerized exams as well. One advantage of these exams is that they are offered by computer, which can both improve the authenticity of the assessment (e.g., students can be asked to write code on a computer where they have access to a compiler and debugger, unlike on paper) and the student work is provided in a digital format which facilitates machine scoring.

In addition, all of these approaches generally offer *asynchronous* exams, where students can choose when to take their exams. This is a very popular feature for students, as it gives them the flexibility to take the exam at a time that is convenient for them, and it eliminates the need for the course to manage conflict exams. It does, however, create an opportunity for *collaborative cheating* [5], where a student taking the exam early provides information about the exam to a student taking the exam later to give them an advantage. In the context of MOOCs, sometimes it

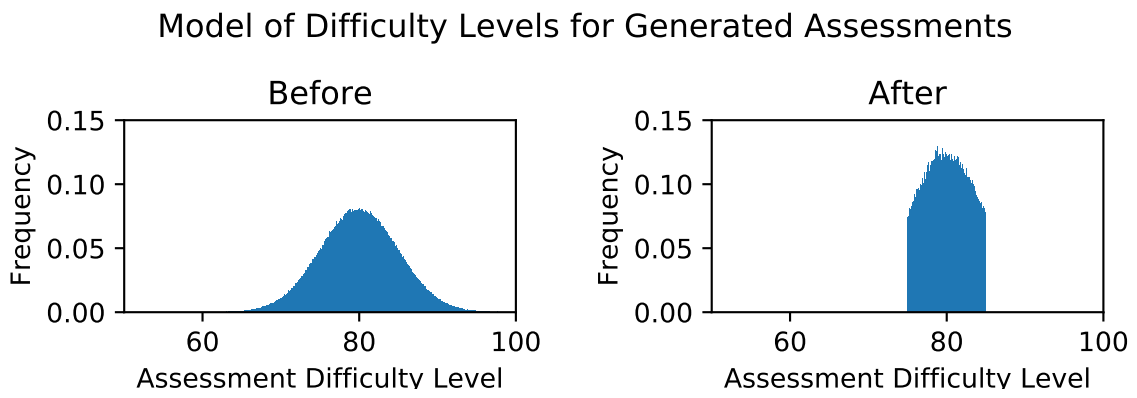

Figure 1: Exams generated randomly from pools of problems are expected to have difficulty that is normally distributed, if the problems are not exactly the same difficulty (left). The fairness of the exams can be improved by discarding the exams from the tails of the distribution (right).

is the same student taking the exam both times using a strategy designated as Copying Answers using Multiple Existences Online (CAMEO) [6].

Recent work has shown that the potential informational advantage from collaborative cheating can be largely mitigated by introducing randomization into the exam [5]. Randomization results in students getting different exams, so that the information passed from student to student is less likely to be relevant. Chen et al. find that it isn't sufficient to randomize just the parameters of a problem, but having a small number of different versions of each problem (along with random parameterization) appears to make the informational advantage statistically insignificant. Randomization of problem selection and order has been previously used in the generation of multiple-choice exams to prevent copying on pencil and paper exams (e.g., [7, 8]).

A concern with randomized exams, however, is fairness; we want to give each student an exam with problems of roughly similar difficulty. Problems can be binned into *pools* by topic coverage and difficulty, but it is challenging to generate problems of identical difficulty on the same topic that are different enough so that having seen one doesn't give you a significant advantage on the other. It is thus desirable to have a mechanism that permits the generation of fair exams in the presence of pools of problems with small difficulty variations.

If we naively generate exams via random draws from a collection of problem pools, we expect that the distribution of exam difficulties will approximate a normal distribution as the number of questions grows, as illustrated in Figure 1 (left-hand side). That is, there will be many exams where the lucky draws (i.e., getting an easier than average problem from a pool) largely compensate for the unlucky draws. The problematic exams are the ones at the tails of the distribution, where an exam consists almost entirely of lucky or unlucky draws.

In this paper, we propose that the fairness of randomized exams can be improved by discarding exams at the edges of the distribution, as illustrated in Figure 1 (right-hand side). Furthermore, this filtering can be done without significantly impacting exam security, which was the reason that the randomization was introduced in the first place. Specifically, we make three main contributions:

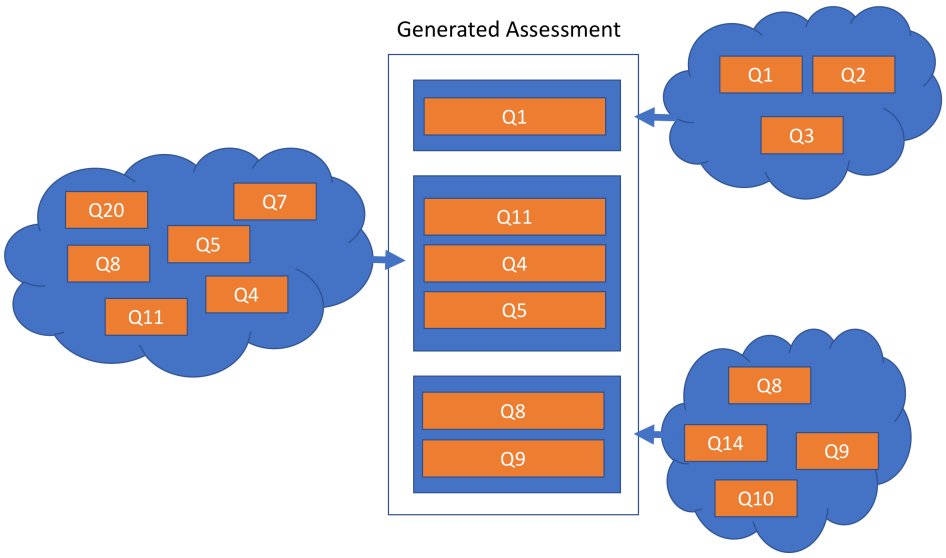

Figure 2: Generating an example exam in PrairieLearn from three alternative groups. The first exam slot selects a single question (Q1) from an alternative group containing three questions: Q1, Q2, and Q3. The next three slots draw questions from a pool of six questions, and the final two slots are filled with questions from a pool of four questions. The selection process happens independently for every student.

1. we frame the *fair random-exam generation* problem in a manner that recognizes that the fairness of an exam is a function of the student's capabilities (in the next section),
2. we describe a straight-forward algorithm for reducing the difficulty variance in randomized exams (in the Algorithm Section), and
3. we explore the trade-off between difficulty variance and measures of exam security of a hypothetical exam using historic student data (in the Experimental Results Section).

We conclude with a discussion of related and future work.

The Fair Exam Generation Problem

In this work, we focus on a formulation of exam generation that models an implementation in widespread use (e.g., around 20 courses) at our university. In this formulation, an exam is specified as a series of slots, as shown in Figure 2. Each slot is associated with an *alternative group* or *pool* of problems. A slot of size n contributes n questions to the exam, the same number for every student. The questions for a slot are chosen from its alternative group, without replacement. All of the problems in a slot are assigned the same point value and partial credit schedule.

Fairness is a property of a collection of exams in relation to a group of students. We define an exam as *completely fair* with respect to a given student if their expected score is the same for any exam in the collection. Intuitively, the *unfairness* for a given student should grow with the

variance of that student's expected scores across the whole collection of the exams. That is, the more the student's score depends on exactly which exam they receive the less fair the exam is. Likewise, the unfairness of the exam as a whole should intuitively increase as the unfairness for individual students increases.

In order to develop a useful fairness metric, it is important to recognize that the true variance of a student's expected score across the whole collection of exams is practically unknowable. No student is going to be willing to take a enough exams from the collection for us to compute a statistically significant variance. Furthermore, we want to estimate an exam's fairness before the students from a given class even take it the first time. To address these shortcomings, we make two simplifying assumptions.

First, we assume that the current student population for a given class is not so different from previous student populations. Said another way, the fairness metric for a collection of exams for a previous population of students is a good predictor of its fairness for the current students. This assumption allows us to use historical student exam performance data to estimate fairness of an exam. Such historical data is often available in the context of randomized asynchronous exams, as the randomization mitigates much of the traditional downside of re-using exam questions.

Second, we're going to assume that exam fairness is a relatively smooth function of student ability. That is, the fairness of a collection of exams for a strong student is well predicted by the fairness perceived by other strong students, and the fairness for a weak student is well predicted by the fairness perceived by other weak students. This assumption allows us to cluster students into groups and estimate the fairness for that group, which allows us to use the exam score data of that cluster to estimate the variance for individuals in that cluster even though they each only have a single attempt on one exam from the collection.

It is important to note, however, we are not assuming that an exam collection is fair for all students if it appears fair to the "average" student. It is well known in psychometrics that the expected score on an item as a function of student ability (for example, as modeled by Item Response Theory [9]) can vary wildly from question to question. Two items can have the same overall average score, but one may have a much lower average score for the weakest students. Which of these two problems is picked can have a significant impact on the expectation of a weak student's score and this variance should be interpreted as unfairness.

To account for this variation in fairness between student abilities, we compute fairness metrics at the granularity of quintiles. Quintiles are commonly used for item analysis and strike a good balance between separating distinct behaviors and not losing too much statistical power by creating too many clusters. For a given quintile, we can compute the expected score for a specific exam by summing up the expected scores of each item[1], as shown in Algorithm 1. To compute the expected score for each item, we first break the students into five groups based on their overall exam score (class quintile). We then calculate predicted scores for each question for each quintile using data only from students in that quintile.[2]

[1]For the purpose of this paper, we are going to ignore problem ordering, although that has been shown to impact performance in some cases.

[2]If an item has not been previously used on an exam its average scores by quintile can be predicted from either student average scores on homework if the problem was used for a homework or manually estimated by the instructor.

Algorithm 1 Computing the expected score for a given exam.

```
1: function MEAN_EXAM_SCORE_BY_QUINTILE(exam, quintile)
2:     points = 0
3:     max_points = 0
4:     for q in get_questions(exam) do
5:         mean = question_score_by_quintile(q, quintile)
6:         points = points + mean
7:         max_points = max_points + get_max_points(q)
8:     end for
9:     return points/max_points
10: end function
```

We then define the *unfairness of a collection of exams for a given quintile* as the standard deviation of the expected scores for that quintile across all of the exams.

To be clear, a collection of exams is not necessarily unfair if there is high variance in the student scores when students are given different exams from this collection. We expect such a variance in score resulting from a variance in student abilities. We consider an exam as unfair if the students would have scored significantly differently had they received a different random exam.

Furthermore, in our opinion, it is not necessary to entirely eliminate unfairness from randomization. It is sufficient to make it small enough so as it doesn't dominate the other sources of uncertainty involved in testing (e.g., uncertainty resulting from the particular selection of topics for the exam from all of the topics that were covered in the class).

Algorithm

With a clear definition of unfairness, it is rather straightforward to develop an algorithm that reduces variance relative to naive random exam generation by discarding outlier exams. Our algorithm uses a Monte-Carlo approach to estimate the mean and standard deviation of each quintile's expected score of the full collection of exams, as specified by the exam specification. Using these computed standard deviations as a guideline, the instructor sets a threshold for the allowable exam variance. If a randomly-generated exam lies outside these thresholds, it is discarded and new exams are generated until one meets this acceptance criteria.

Pseudocode for the algorithm is shown as Algorithm 2 for clarity. First, the `estimate_exam_collection_means_and_sds` method generates a sample of n random exams. Using the previously described `mean_exam_score_by_quintile` method, the mean scores are computed for these exams by quintile. Then the mean and standard deviation of this sample are computed and returned.

When we filter exams, we only keep an exam if, for all quintiles, it has a predicted score that is within a specific range of the mean generated exam score for students in that quintile. The range of valid exams is specified as a fraction of the original standard deviation and this fraction is an instructor specified parameter. Pseudocode is shown as the `is_exam_valid` method of Algorithm 2.

Algorithm 2 Assessment Filtering Algorithm

```
1: function ESTIMATE_EXAM_COLLECTION_MEANS_AND_SDS(n)
2:     E = generate_random_assessments(n)
3:     scores = []
4:     avgs = []
5:     sds = [] // standard deviations
6:
7:     // calculate mean scores for randomly generated exams
8:     for i = 1 to 5 do
9:         for j = 1 to n do
10:            scores[i][j] = mean_exam_score_by_quintile(E[j], i)
11:        end for
12:        avgs[i] = avg(scores[i][1..n])
13:        sds[i] = std_dev(scores[i][1..n])
14:    end for
15:
16:    return avgs, sds
17: end function
18:
19: function IS_EXAM_VALID(e, avgs, sds, num_sds_parameter)
20:    for i = 1 to 5 do
21:        score = mean_exam_score_by_quintile(e, i)
22:        if |score − avgs[i]| > sds[i] · num_sds_parameter then
23:            return false
24:        end if
25:    end for
26:    return true
27: end function
```

Selecting the *number of standard deviations* (`num_sds`) parameter represents a trade-off between fairness and security. As one decreases this parameter, the fraction of exams that are considered outliers (and hence discarded) increases, which in turn increases fairness. But as exams are discarded, there is potentially less and less variation between exams, reducing the benefit of randomization in mitigating the informational advantage of collaborative cheating. In the limit, all exams but one are discarded and complete fairness is achieved at the expense of no mitigation of collaborative cheating.

To aide instructors in setting this parameter, our implementation provides an interactive tool to visualize how changing this parameter changes the resulting distribution of exam difficulties and fraction of exams that are retained (shown in Figure 3). If the instructor is not satisfied with the trade-off between fairness and security, they can modify the alternative groups in the exam specification to give the algorithm more choices.

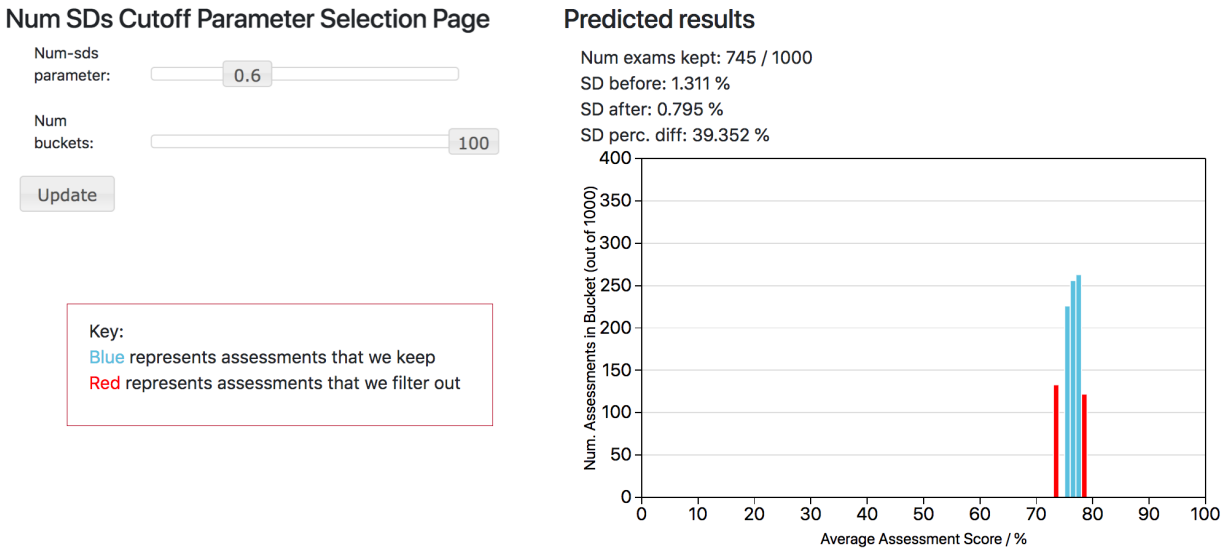

Figure 3: The `num_sds` (cutoff) parameter selection page, used by the instructor to select an appropriate value for the `num_sds` parameter for a specific assessment. In the above diagram, we are filtering out about 25% of exams that have a difficulty level (averaged over all students) below and above the average.

Experimental Results

In an ideal world, we would be able to significantly reduce the unfairness of randomized exams with minimal impact on exam security. In this section, we explore the relationship between these two variables using a synthetic exam constructed from real student data and sweeping the `num_sds` parameter from one extreme to the other.

The data that we use for our synthetic exam is drawn from a large-enrollment sophomore-level engineering class at a public U.S. university. In this course, the computerized exams [10] are made up of auto-graded problems that take numerical answers; students are given multiple attempts to answer questions with partial credit given if answered correctly on a second or later attempt [11]. Some of the exam questions were given to every student (i.e., an alternative group size of 1). By constructing our synthetic exam out of only these questions, we can compute the actual exam scores for all students for every possible exam that could be generated from the exam specification.

We constructed an exam specification that included 5 slots, each of size 1, yielding a 5 question exam. Exams of this length are common in this class, which offers 1-hour exams every two weeks. Each slot draws from its own distinct alternative group and the alternative groups range in size from 3 to 8 problems. All slots were assigned the same number of points.

No effort was made to match the difficulty of the problems in the alternative groups, so this workload represents a challenging scenario for an algorithm attempting to ensure fairness. If the algorithm can be successful in this scenario, then it reduces the pressure on instructors to come up with many questions of equal or similar difficulty to put in an alternative group.

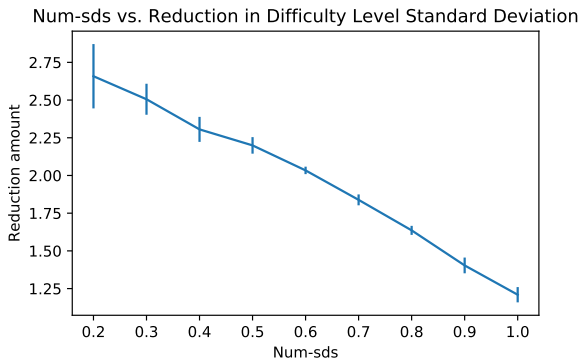

Figure 4: The reduction in the standard deviation (`SD_diff`) grows linearly as we reduce our control parameter `num_sds`.

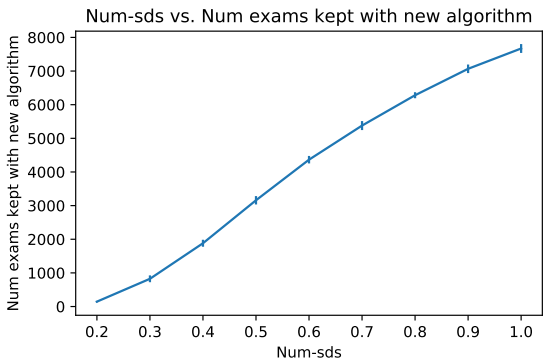

Figure 5: As we reduce the `num_sds` control parameter, the number of exams remaining decreases almost linearly.

Improved Fairness (`SD_diff`)

Our main success metric will be called `SD_diff`, which is the decrease in the standard deviation of the filtered collection of exams relative to the complete set of randomized exams. Specifically, `SD_diff` is computed as follows.

Let E be a set of 10 000 generated exams, and E' be the subset of these exams that remain after filtering. That is, the exams in E' are those for which `IS_EXAM_VALID()` (Algorithm 2) returns `true`. Now take X to be a random variable giving the difficulty of an exam drawn at random from E, and X' the corresponding random variable for E'. Then we define

$$\texttt{SD\_diff} = \sigma(X) - \sigma(X'), \tag{1}$$

where $\sigma(X)$ is the standard deviation of X.

As shown in Figure 4, our parameter sweep of our control parameter `num_sds` demonstrates that it has a roughly linear relationship with the reduction of the standard deviation (`SD_diff`). Moreover, we find that the lowest performing quintiles have the largest variance and that the largest contributions to the `SD_diff` come from mitigating this variance.

Number of exams kept

Figure 5 shows the relationship between our control parameter, `num_sds`, and the number of the original 10 000 exams that are retained. We see a roughly linear relationship between the `num_sds` parameter and the number of exams remaining after filtering, with 0.2 effectively being the limit for `num_sds` because at that point almost all of the exams have been eliminated.

The number of exams remaining, however, is not a good proxy for the security of the exam. Exam security can be high even with a small number of exams if all of the problems in each alternative group are still well represented across the exams.

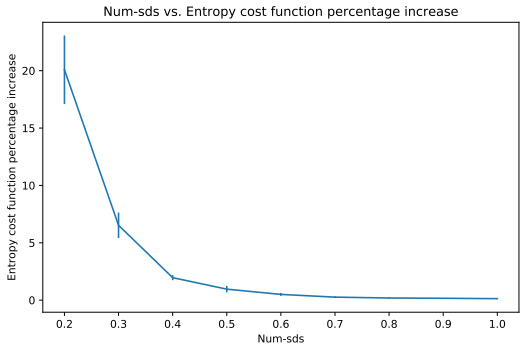

Figure 6: Reductions in `num_sds` initially have little impact on entropy, but the entropy cost function grows rapidly as `num_sds` is decreased below 0.4

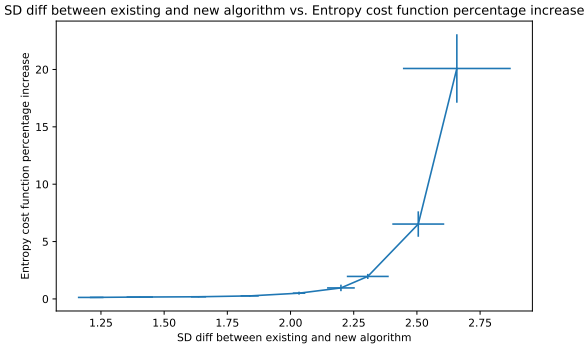

Figure 7: The standard deviation can be significantly reduced (up to `SD_diff` of 2.25) with no loss of entropy, but further reductions quickly increase the entropy cost function.

Security (Entropy)

A better metric for the extent that exam variation has the potential to mitigate the informational advantage of collaborative cheating is a metric like entropy. The entropy of a probability distribution is a measure of the average surprise that we incur when taking a sample from that distribution. So if we maximize the entropy, then we have maximized the surprise that a student will experience when seeing a randomly generated exam. If a distribution has low entropy, then a student could effectively predict which questions are likely to be on their exam.

We compute entropy as follows. Consider an exam specification with n slots and m_i questions in the alternative group for the i^{th} slot, where $i \in \{1 \dots n\}$. Let S_i be a discrete random variable with possible values $\{q_1, q_2, \dots q_{m_i}\}$ representing the questions that a student could get in slot i. Then a total entropy value across all slots can be defined as:

$$H = \sum_{i=1}^{n} \sum_{j=1}^{m_i} -1 \cdot P(S_i = q_j) \cdot \log_2(P(S_i = q_j)). \tag{2}$$

This can be simplified to:

$$H = -\sum_{q \in Q} P(q) \cdot \log_2(P(q)). \tag{3}$$

However, since we want entropy to be high (more randomness in the generated exams), rather than low, we can define an entropy cost function as

$$C = \frac{1}{H} = -\frac{1}{\sum_{q \in Q} P(q) \cdot \log_2(P(q))}. \tag{4}$$

We find that, initially, removing exams by lowering the `num_sds` control parameter has almost no impact on entropy, but at around 0.4 there is a knee in the curve and entropy decreases drastically (our entropy cost function increases drastically), as shown in Figure 6. We explored

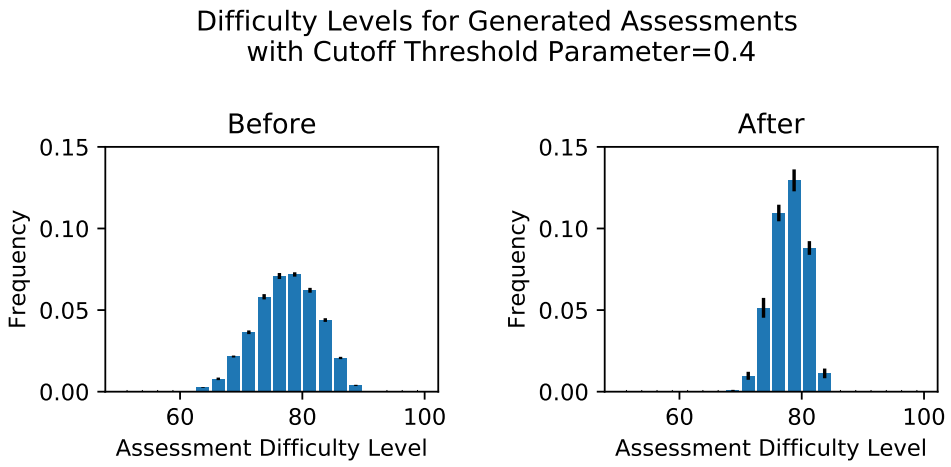

Figure 8: With a control parameter value of num_sds $= 0.4$, there is a striking difference in the distribution of difficulty levels for randomly generated exams. The variance in assessment difficulty levels has been reduced significantly.

other potential security metrics (e.g., metrics based on the probability of overlap between a pair of exams and based on the expected benefit of memorizing the n most frequently encountered questions) and found our results to be highly insensitive to the choice of metric. All of the metrics consistently had a knee in the curve around num_sds $= 0.4$.

Figure 7 shows the trade-off between SD_diff and inverse entropy. From this plot, we find that setting num_sds to 0.4 seems to be a sweet spot for this workload, maximizing the increase in fairness that can be achieved (SD_diff $= 2.3$) for a negligible loss of security. Figure 8 compares the distribution of difficulties for the original collection of exams to that of the distribution of difficulties for the exams filtered using num_sds $= 0.4$. It can be seen that the standard deviation is shrunk to less than half of its original value.

Related Work

Two streams of related work are worth noting, but they are focused on very different contexts. The first is the breadth of work using Item Response Theory (IRT) [9]. Using IRT, it is *not* necessary to give students exams of equivalent difficulty, as long as all of the items have been calibrated. Instead, given the item response functions for each of the items and a student's score on each of the items, a maximum likelihood estimate for the student's latent ability can be computed, which can be mapped into a score for the exam. IRT is commonly used for high-stakes standardized exams like the SAT and ACT.

The main drawback of IRT models is the lack of transparency that they have for students in understanding their scores. Unlike common college exams, where each question has a point value and the students score is the sum of the points on the individual questions, items on an IRT exam do not have an obvious point value and the influence a given question has on a student's score is dependent on whether they correctly answered other questions. While we don't doubt the effectiveness of the IRT methods, we believe that scoring college exams using IRT is politically

untenable with students.

The other line of research that is tangentially related are algorithms for composing exams from large banks of test items (e.g., [12, 13]). These algorithms are generally provided items that have rich meta-data indicating the difficulty, average solution time, and coverage of different topics along with an exam specification indicating the desired minimum coverage of each topic, the target difficulty, and a minimum and maximum time length. The proposed algorithms perform heuristic searches for optimal exams, because exact solutions are NP-hard. Again, this work is more suited for a high-stakes standardized exam context where items can be characterized in the necessary manner.

Conclusion

In this paper, we have demonstrated a straight-forward algorithm for reducing the difficulty variance in a collection of randomized exams. Our synthetic exam experiment suggest that unfairness can be reduced by a factor of two, with only a small reduction in the security of the exam, as predicted by the exam's entropy. We find this result to be rather exciting, because the potential benefit is significant for the code complexity involved. We hope our implementation will go into production this coming semester, where it can positively impact tens of courses on our campus.

References

[1] R. Muldoon, "Is it time to ditch the traditional university exam?" *Higher Education Research and Development*, vol. 31, no. 2, pp. 263–265, 2012.

[2] E. Lee, N. Garg, C. Bygrave, J. Mahar, and V. Mishra, "Can university exams be shortened? an alternative to problematic traditional methodological approaches," in *Proceedings of the European Conference on e-Learning*. Kidmore End: Academic Conferences International Limited, 06 2015, pp. 243–250.

[3] C. Zilles, R. T. Deloatch, J. Bailey, B. B. Khattar, W. Fagen, C. Heeren, D. Mussulman, and M. West, "Computerized testing: A vision and initial experiences," in *2015 ASEE Annual Conference & Exposition*, no. 10.18260/p.23726. Seattle, Washington: ASEE Conferences, June 2015, https://peer.asee.org/23726.

[4] R. F. DeMara, N. Khoshavi, S. D. Pyle, J. Edison, R. Hartshorne, B. Chen, and M. Georgiopoulos, "Redesigning computer engineering gateway courses using a novel remediation hierarchy," in *2016 ASEE Annual Conference & Exposition*, no. 10.18260/p.26063. New Orleans, Louisiana: ASEE Conferences, June 2016, https://peer.asee.org/26063.

[5] B. Chen, M. West, and C. Zilles, "How much randomization is needed to deter collaborative cheating on asynchronous exams?" in *Learning at Scale*, 2018.

[6] C. G. Northcutt, A. D. Ho, and I. L. Chuang, "Detecting and preventing "multiple-account" cheating in massive open online courses," *Computers & Education*, vol. 100, pp. 71–80, 2016.

[7] J. Šnajder, M. Čupić, B. D. Bašić, and S. Petrović, "Enthusiast: An authoring tool for automatic generation of paper-and-pencil multiple-choice tests," in *Interactive Computer Aided Learning: The Future of Learning–Globalizing in Education*, 2008.

[8] M. West, M. S. Sohn, and G. L. Herman, "Randomized exams for large STEM courses spread via communities of practice," in *Proceedings of the 2015 American Society for Engineering Education Annual Conference and Exposition*, Seattle, WA, 2015.

[9] S. E. Embretson and S. P. Reise, *Item response theory*. Psychology Press, 2013.

[10] C. Zilles, M. West, D. Mussulman, and T. Bretl, "Making testing less trying: Lessons learned from operating a computer-based testing facility," in *Frontiers in Education*, October 2018.

[11] M. West, G. L. Herman, and C. Zilles, "Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning," in *2015 ASEE Annual Conference & Exposition*, no. 10.18260/p.24575. Seattle, Washington: ASEE Conferences, June 2015, https://peer.asee.org/24575.

[12] G.-J. Hwang, B. M. Lin, and T.-L. Lin, "An effective approach for test-sheet composition with large-scale item banks," *Computers & Education*, vol. 46, no. 2, pp. 122–139, 2006.

[13] G.-J. Hwang, H.-C. Chu, P.-Y. Yin, and J.-Y. Lin, "An innovative parallel test sheet composition approach to meet multiple assessment criteria for national tests," *Computers & Education*, vol. 51, no. 3, pp. 1058–1072, 2008.