JAG: Joint Assessment and Grading

Igor Labutov IIL4@CORNELL.EDU
Cornell University, Ithaca, NY

Christoph Studer STUDER@CORNELL.EDU
Cornell University, Ithaca, NY

Abstract

Test-taking (assessment) and grading are commonly treated as two separate processes and performed by two different groups of people. Recently, the boundaries between test-taking and grading have started to blur. *Peer-grading*, for example, leverages student knowledge and effort to grade other students' answers. Despite the considerable success of peer-grading in massive open online courses (MOOCs), the process of test-taking and grading are still treated as two distinct tasks which typically occur at different times, and require an additional overhead of grader training and incentivization. In order to overcome the drawbacks of peer-grading, we propose *joint assessment and grading (JAG)*, a novel approach that fuses test-taking and grading into a single, streamlined process that appears to students in the form of an explicit test, but where everyone also acts as an implicit grader. We demonstrate the effectiveness and limits of JAG via simulations and a real-world user study.

1. Introduction

Multiple-choice questions (MCQs) are a common way to overcome the so-called *scaling problem* in assessment: grading a large number of submissions from many students in an efficient manner (Roediger III & Marsh, 2005). The immense scaling potential of MCQs to large classroom settings such as massive open online courses (MOOCs), however, comes at the cost of rigidity and considerable sensitivity to its design—questions and options that are designed without the consideration of the students' likely misconceptions, for example, can yield uninformative questions (Haladyna, 1997; Rodriguez, 2005) (e.g., when the distractors are easy to eliminate). As a consequence, the effort saved in grading is often offset by the effort required to design an effective MCQ test.

More recently, the practical realization of MOOCs opened another opportunity to solving the problem of scaling in large-scale assessments, one that leverages the size of the classroom to its advantage: *peer-grading*. In its traditional form, peer-grading assigns to each student a secondary role of a grader. Students are responsible for validating the correctness of other students' solutions in order to assign a score, typically in accordance with a rubric provided by an instructor. The advantage of peer-grading is its flexibility to students' submissions, which may range from short answers, to essays, diagrams, code, or entire projects (Kulkarni et al., 2015). Peer-grading, however, introduces several challenges to guarantee a successful deployment. First, each student differs in their ability to grade, and the grades assigned by different students must be reconciled in a reasonable way. Second, grading becomes an additional burden on the students, and mechanisms must be put in place that not only incentivize participation and effort, but prevent students from "gaming" the process. Recent research in peer-grading has started to address these challenges (Piech et al., 2013; Raman & Joachims, 2014; Wu et al., 2015)

1.1. JAG: an alternative to Peer Grading

We present a novel alternative approach to peer-grading that naturally resolves the challenges of grade aggregation and incentivization. We propose *joint assessment and grading (JAG)*, which fuses grading and assessment into a single, streamlined process by re-framing grading as additional testing. Our approach is motivated by the fact that a "grader" that has no answer key, when presented with the listing of other students' answers, is no different than a test-taker facing a multiple-choice question (with multiple possible correct or incorrect answers). In other words: a student selecting what they believe to be the correct answer in an MCQ constructed from the open-response submissions of other students is in effect simultaneously (i) grading the other students and (ii) being assessed by his or her ability to select the correct answer. In peer-grading, we already face the challenge of noisy inputs from the (potentially unmotivated) graders. By re-framing the act of grading as that of MCQ testing, the source of the apparent noise in grading becomes distributed according to the ability of the students in the class.

The proposed mechanism of JAG combines the advantages of both worlds: the structure of multiple choice questions and the flexibility to general response types offered by *peer-grading*. First, by constructing the MCQs directly from students' open-response submissions, the questions naturally capture the distribution of misconceptions present in the population of students being tested, requiring little to no instructor input. Second, our framework offers a mechanism for automatically grading open-response submissions, thus facilitating greater student engagement and higher-order thinking characteristic to open-response questions (Haladyna, 1997). Third, by re-framing the task of grading as that of testing, the students are incentivized in the context of a familiar task: namely by expending their effort towards correctly answering an MCQ, they are implicitly directing that effort towards grading other students' submissions. At the same time, the students are not burdened with (what they may perceive as) a "thankless" job of grading, but instead in the process of answering the additional MCQs, the students are provided with an additional opportunity to demonstrate their knowledge.

In this paper, we formalize the process of JAG as a statistical estimation problem. At the heart of our approach is the traditional Rasch model that captures the interaction between student abilities and question difficulties in determining the likelihood of a student answering a question correctly (Rasch, 1993). We develop an expectation maximization (EM) algorithm for estimating the parameters of the proposed model in an unsupervised setting (i.e., in absence of an answer key), and demonstrate the effectiveness of our framework through a real-world user-study conducted on Amazon's Mechanical Turk. Additionally, we investigate the key properties and limitations of our approach via simulations.

2. Related Work

Our work builds on the recent progress in two distinct areas: *crowd-sourcing* and *peer-grading*, that we unite and extend within our proposed framework for *joint assessment and grading (JAG)*.

2.1. Crowdsourcing

An important task in *crowdsourcing* is known as *label-aggregation*, and is concerned with the problem of optimally recovering some underlying ground truth (e.g., image class label) from a number of (unreliable) human judgements. See (Hung et al., 2013) for a detailed review. In the context of education, the task of automatically identifying the correct answers from open-response submissions is closely related to the task of label aggregation. Within the field of crowd-sourcing, the work of (Dawid & Skene, 1979; Whitehill et al., 2009; Bachrach et al., 2012) are the most related to our approach. (Dawid & Skene, 1979) was the first to suggest an expectation maximization (EM) algorithm for label aggregation, motivated by a clinical setting of making a diagnosis. More recently, (Whitehill et al., 2009) extended this approach to model the variation in task difficulty in the context of image labeling. In the context of education, (Bachrach et al., 2012) has proposed a statistical model for aggregating answers from "noisy" students, with the goal of automatically identifying the correct answers to MCQs. They deploy an EM algorithm for Bayesian inference, and demonstrate the ability to infer correct answers accurately in a setting of an IQ test. Our work can be seen as a generalization of (Whitehill et al., 2009; Bachrach et al., 2012), where we explicitly model the dependence among question choices and students that generate those choices in the context of answering open-response questions.

2.2. Peer-grading

Much of the recent research in *peer-grading* addresses a related problem of aggregating a number of "noisy" grades submitted by students in a statistically principled manner. Models such as the ones in (Piech et al., 2013; Raman & Joachims, 2014) pose the problem of peer-grading as that of statistical estimation. Since traditional grading assumes that graders are in possession of a grading rubric, statistical models of peer-grading are concerned primarily with accounting for the reliability and bias of graders in evaluating assignments against a gold-standard. In contrast to such "explicit" models of grading, we view grading as an implicit process that results as a by-product of students' genuine attempt to answer MCQs constructed from the open-response submissions of other students. As such, we do not require additional "grader-specific" parameters, as grading in our framework is subsumed by the response model (model of how students answer questions as a function of their ability and question difficulty). We do note, however, that one of the proposed models in (Piech et al., 2013) explicitly couples grading and ability parameters in an attempt to capture the intuition that better students may also be better graders. This intuition can be viewed as being taken to its extreme in our setting: blurring the boundaries between grading and test-tasking ensures that betters students are more reliable graders by construction.

2.3. Clustering submissions

We also take note of an emerging area of work focused on clustering open-response submissions (not necessarily for peer-grading). An important by-product of scaling in the context of assessment is the inevitable increase in similarity between student open-response submissions, which results in redundancy during grading. Moreover, a recent theoretical result of (Shah et al., 2014) indicates that without some way of reducing dimensionality of submissions, there will always be a constant number of misgraded assignments (assuming certain scaling properties of the classroom). In an effort to reduce the workload of the instructors (or peers), there has been a number of successful attempts to cluster responses in specific domains, e.g., language (Basu et al., 2013; Brooks et al., 2014) and mathematics (Lan et al., 2015). Answer clustering is even more critical in the framework of JAG, where the practical constraints of testing limit the number of options that can be shown in a multiple choice question. To generate effective questions, the presented options must offer a representative sample of the diverse open-response submissions in a large classroom. In Section 7, we demonstrate that the pattern of selected options alone provides the necessary signal to perform domain-agnostic clustering of submissions (i.e., without considering the content of the answers). This observation demonstrates the tremendous versatility of this framework to jointly assess, grade and cluster open-response submissions.

3. Model

3.1. Fully observed setting

We start by reviewing the classic IRT Rasch model that will serve as the foundation of our approach. Consider a set of students S and a set of questions Q, where a student $i \in S$ is endowed with an ability parameter $s_i \in \mathbb{R}$, and each question $j \in Q$ is endowed with a difficulty parameter $q_j \in \mathbb{R}$ (note that we capitalize all sets in our notation). By abuse of notation, we will often overload s_i to refer to both, the student index i and their ability, depending on the context; the same applies to q_j, which we use to refer to the question itself as well as its difficulty. The well-established 1-PL IRT Rasch model (Rasch, 1993) expresses the probability that the student s_i answers question q_j correctly via the following likelihood function:

$$P(z_{i,j} \mid s_i, q_j) = \frac{1}{1 + \exp\left(-z_{ij}(s_i - q_j)\right)}, \qquad (1)$$

where $z_{i,j} \in \{+1, -1\}$ is the binary outcome of student s_i's attempt of question q_j; we use $+1$ and -1 to designate correct and incorrect responses, respectively. If we are in the possession of an answer key for each question, then we also know $\{z_{i,j}\}$, $\forall i, j$ (we will refer to this as the *fully observed* setting). This allows us to estimate the ability of each student and the difficulty of each question by maximizing the likelihood of all outcomes under our model:

$$\{s_i, \forall i, q_j, \forall j\} = \operatorname*{argmax}_{s_i, q_i} \prod_{z_{i,j} \in D} P(z_{i,j} \mid s_i, q_j), \qquad (2)$$

where $D = \{z_{ij}\}$ is the set of outcomes (e.g., of a test).

3.2. Partially observed setting

Consider now the setting where some (or all) of the outcomes $z_{i,j} \in D$ are not observed. In practice, this is the case, for example, when the answer key to some of (or all) the questions is not available. In our setting, where the choices in the multiple choice question are in fact other students' submissions, the correctness of these submissions are not known a priori. Let A_j be the set of open-response answers submitted by a subset of students in $S_{\text{open}} \subseteq S$ in response to the question q_j . At some later time, a student $s_i \in S_{\text{mcq}} \subseteq S$ is presented with the same question q_j, but in the form of a multiple-choice question, with the options being exactly the answers in A_j (note that S_{mcq} need not be disjoint with S_{open}). The student s_i is informed that there may be zero or more correct answers in the set of options in A_j and they are instructed to select "all that apply." The student s_i goes through each option in A_j and submits a response to that option. Let $y_{i,j} = \{y^k_{i,j}\}$ be the set of such responses made by student s_i on the set of answers A_j, where $y^k_{i,j} \in \{+1, -1\}$ is the student s_i^{th} selection on the k^{th} answer (option) in A_j. In other words the variables $y^k_{i,j}$ are the observations of whether the student s_i "clicked" on answer k to question j (i.e., that student judged that particular answer to be correct). In what follows, we describe the statistical model that relates the student and question parameters which we are interested in estimating, to the set of response observations. Our model consists of two components: (i) the *open-response* component that models the students (and their responses) that generate open-response answers, and (ii) the *multiple choice model* component that models the students (and their responses) that are presented with the multiple choice version of each question.

Open-response model: Because we do not know whether the submitted open-response answers are correct, we treat the correctness of each submission as a hidden variable $z_{i,j} \in \{+1, -1\}$; this allows us to express the component of the overall likelihood of our data, responsible for the open-response answers only as follows:

$$P(\{z_{i,j}\} \mid S_{\text{open}}, Q) = \prod_{z_{i,j}} P(z_{i,j} \mid s_i, q_j),$$

where $P(z_{i,j} \mid s_i, q_j)$ is the Rasch likelihood given in (1). Note that we drop the k-superscript notation for the $z_{i,j}$ variables because each student is assumed to provide at most one open-response submission to each question (since k indexes the answers to a specific question). The observed responses to the multiple-choice version of each question (described next) will provide the necessary data to estimate the parameters in the model, including the hidden variables $z_{i,j}$, i.e. the correctness of each open-response submission.

Multiple choice model: Now consider the setting where each question is presented in the form of a MCQ. Recall that the student answering the multiple choice question is presented with multiple options, each generated by some (other) student in the set S_{open}, and where several options (or even no options) may be correct. The intuition that we want to capture in our model is that a student of great relative ability (i.e., $s_i \gg q_j$) will select ($y_{i,j}^k = +1$) the option (i.e., judge it as being correct) *if* that option is actually correct ($z_j^k = +1$). The same student will *not* select that option ($y_{ij}^k = -1$) if that option is incorrect ($z_j^k = -1$). At the same time, a student of poor relative ability (i.e., $s_i \ll q_j$) will not not be able to identify the correct answer, regardless of whether the option is correct, i.e., they will guess. This intuition can be captured by the following function that parametrizes the likelihood of student s_i selecting the option k to question q_j to be correct:

$$P\left(y_{i,j}^k \mid s_i, q_j, z_j^k\right) = \frac{1}{2}\left(\frac{1}{1+\exp\left(-y_{ij}^k z_j^k (s_i - q_j)\right)} + 1\right). \quad (3)$$

One can easily verify that this likelihood satisfies the requirements outlined above by considering every combination of the assignment to $y_{i,j}^k$ and z_j^k, and taking the limits of $s_i - q_j \to \infty$ (great relative ability) and $q_i - s_j \to \infty$ (poor relative ability). Note that this time we drop the index i (index of the student who generated the option k in question q_j) in z_j^k, as in the above, we use s_i to refer to the student answering the multiple choice version of the question. Note that the above likelihood follows the same intuition as proposed by (Bachrach et al., 2012), but in a setting with an arbitrary number of choices and one correct answer. In Figure 1 we illustrate both components of the likelihood (the *open-response* and the *multiple choice* component) as a graphical model. In this illustration, we use the notation $s_{i'}$ to refer to the student that generated the answer and s_i to refer to the student that observes the answer as a choice in a multiple choice version of question q_j.

If we make a leap of assuming conditional independence between the student s_i^{th} responses to each option in a multiple choice question (conditional on s_i, q_j and z_j^k), then we can express the likelihood of observing every response to every multiple choice questions as follows:

$$P(\{Y_j\} \mid S_{\text{mcq}}, Q, \{Z_j\}) = \prod_{s_i \in S_{\text{mcq}}} \prod_{q_j \in Q} \prod_{\substack{y_{i,j}^k \in Y_j \\ z_j^k \in Z_j}} P(y_{i,j}^k \mid s_i, q_j, z_j^k).$$

The assumption of conditional independence requires some additional justification in our setting. Intuitively, we are justified in claiming conditional independence when we believe that the set of conditioning variables accounts for everything that may be shared across observations, such that the only remaining source of the variance is noise. For example, observations of different students answering the same question on the test are conditionally independent given the difficulty of that question. In modeling the likelihood of a student selecting each option in a multiple choice question, however, we overlook the potential for the options to be related. In an extreme example, two options may be identical or paraphrases of each other, which we expect to be common-place when these options are generated by students in a large classroom. In this case, conditional independence no longer holds without an introduction of additional conditioning variables that group the related options in some way. To some extent, this problem can be mitigated by pre-processing and clustering similar answers before displaying them as options in a multiple choice question. This is a strategy that we take in this work. In Section 7, we discuss our ongoing work in automatically clustering answers based on the response patterns to multiple choice questions.

To complete our model, we combine the *open-response* and the *multiple-choice* components:

$$P(\mathbf{y}, \mathbf{z} \mid \boldsymbol{s}, \boldsymbol{q}) = \underbrace{P(\mathbf{y} \mid \mathbf{z}, \boldsymbol{s}, \boldsymbol{q})}_{\text{multiple choice}} \underbrace{P(\mathbf{z} \mid \boldsymbol{s}, \boldsymbol{q})}_{\text{open response}}, \quad (4)$$

where we adopt vector notation for the variables and parameters in our model that will facilitate the development of the learning algorithm in Section 4. In order to give the dimensions for each of the variables in (4), assume that each student in S_{open} provides an open-response answer to each of the questions in Q and that each student in S_{mcq} also answers each question in Q (which entails providing a response to each option contained in a given question). Under these assumptions then, $\mathbf{z} \in \{+1, -1\}^{|S_{\text{open}}||Q|}$, $\mathbf{y} \in \{+1, -1\}^{|S_{\text{open}}||S_{\text{mcq}}||Q|}$, $\boldsymbol{s} \in \mathbb{R}^{|S_{\text{mcq}} \cup S_{\text{open}}|}$ and $\boldsymbol{q} \in \mathbb{R}^{|Q|}$.

4. Parameter Learning

We now derive the expectation maximization (EM) algorithm for obtaining an approximate maximum likelihood estimate (MLE) of the parameters $\boldsymbol{s}$ and $\boldsymbol{q}$ of the model in (4). We briefly outline the key steps in obtaining the algorithm.

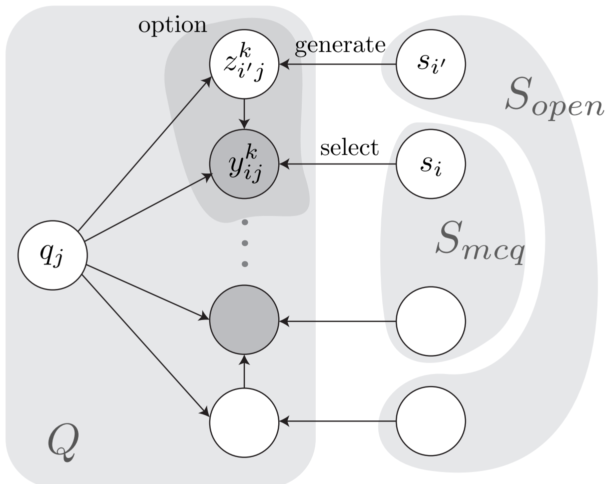


Figure 1. A probabilistic graphical model that summarizes the proposed *joint assessment and grading (JAG)* framework. Our model jointly captures the statistical dependencies between the abilities of students that generate open-response answers (S_{open}), the abilities of students that select the correct answers when they are presented in a form of a multiple choice test (S_{mcq}), the underlying question difficulty q_j and the observed responses ($y^k_{ij} \in \{+1, -1\}$). The correctness of each open-response answer $z^k_j \in \{+1, -1\}$ (omitting the index of student that generated the answer) is a hidden variable, the state of which is inferred in addition to the remaining parameters during inference. Note that the model is able to "grade" the open-response submissions of students by integrating the response patterns of other students who were presented a multiple choice version of the question.

E-step: We compute the expectation of the log-likelihood (the logarithm of Equation 4) with respect to the unobserved variables **z** which yields a function $f(\boldsymbol{s}, \boldsymbol{q})$ of the parameters $\boldsymbol{s}$ and $\boldsymbol{q}$ only. The expectation is performed with respect to the posterior distribution of **z** given a previous estimate of $\boldsymbol{s}$ and $\boldsymbol{q}$ (or an initial guess).

M-step: We obtain an updated estimate of parameters $\boldsymbol{s}$ and $\boldsymbol{q}$ by maximizing $f(\boldsymbol{s}, \boldsymbol{q})$ obtained in the E-step.

The above procedure iterates until convergence. Below we give both steps explicitly in the context of the *joint assessment and grading (JAG)* framework.

E-step: Let $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{q}}$ be an intermediate estimate of the parameters. Conditioning on these estimates, the posterior of z^k_j (correctness of answer (option) k to question q_j) is a Bernoulli random variable with the probability of being correct given by (up to a normalizing constant):

$$P(z^k_j = 1 \mid \hat{\boldsymbol{s}}, \hat{q}_j) \propto \underbrace{P(z^k_j = 1 \mid \hat{s}_{i'}, \hat{q}_j)}_{\text{open response}} \underbrace{\prod_{s_i \in S_{\text{mcq}}} P\left(y^k_{i,j} \mid \hat{s}_i, \hat{q}_j, z^k_j = 1\right)}_{\text{multiple choice responses}}. \tag{5}$$

The posterior over the answer correctness z^k_j naturally integrates two sources of information: (i) the likelihood that the student who generated the answer was correct, and (ii) the likelihood that the students answering the multiple choice version of the question "picked" this answer as correct (note that $s_{i'} \in S_{open}$ and $s_i \in S_{mcq}$). Each likelihood is parametrized by the model's current estimate of the students' abilities and question difficulties, and as a consequence gives more weight to the signal coming from the more able students.

M-step: The expectation of the log-likelihood with respect to **z** yields the following expression:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}}[\log P(\mathbf{y}, \mathbf{z} \mid \boldsymbol{s}, \boldsymbol{q})] = f(\boldsymbol{s}, \boldsymbol{q}) = \\
&= \sum_{D_{\text{mcq}}} P\left(z^k_j = +1 \mid \hat{\boldsymbol{s}}, \hat{q}_j\right) \log \underbrace{P\left(y^k_{i,j} \mid s_i, q_j, z^k_j = +1\right)}_{R_1} \\
&+ \sum_{D_{\text{mcq}}} P\left(z^k_j = -1 \mid \hat{\boldsymbol{s}}, \hat{q}_j\right) \log \underbrace{P\left(y^k_{i,j} \mid s_i, q_j, z^k_j = -1\right)}_{R_1} \\
&+ \sum_{D_{\text{open}}} P\left(z^k_j = +1 \mid \hat{\boldsymbol{s}}, \hat{q}_j\right) \log \underbrace{P\left(z^k_{i',j} = +1 \mid s_{i'}, q_j\right)}_{R_2} \\
&+ \sum_{D_{\text{open}}} P\left(z^k_j = -1 \mid \hat{\boldsymbol{s}}, \hat{q}_j\right) \log \underbrace{P\left(z^k_{i',j} = -1 \mid s_{i'}, q_j\right)}_{R_2}
\end{aligned}$$

where we introduce the short-hand D_{open} and D_{mcq} to refer to the sets of students, questions and responses that were involved in (i) generating open-response submissions and (ii) multiple-choice responses respectively. The above expression is a weighted linear combination of (log-) Rasch-likelihoods (R_1 and R_2 are given in (3) and (1) respectively), and can be easily maximized with a small modification to an existing Rasch solver to account for the constants. We use the L-BFGS algorithm (Zhu et al., 1997) to perform this optimization step.

Initialization: Note that while the M-step is convex, the joint optimization problem in **z**, $\boldsymbol{s}$, and $\boldsymbol{q}$ is not convex, and in general the EM algorithm will only yield an approximate solution and may get trapped in local optima. The problem becomes more pronounced in datasets with few interactions, e.g., small classrooms. As such, initialization plays an important role in determining the quality of the obtained solution. A natural heuristic for initializing the posteriors over **z** is with the fraction of "votes" given to the answer (i.e., fraction of students that identified the answer as correct). Observe that in computing the posterior over the answer correctness during the E-step (Equation 5), the posterior becomes exactly the fraction of "votes" given to the answer if we assume equal difficulty and ability parameters for all students, suggesting that initializing the posterior in this way provides a reasonable starting point for the algorithm. This heuristic is also suggested in (Dawid & Skene, 1979). We demonstrate the effectiveness of this heuristic in Section 6.

5. Experiments with Synthetic Data

In order to understand the behavior of our framework in a hypothetical classroom, we evaluate the model on a series of synthetically generated datasets. As our model attempts to infer the correctness of each answer entirely from the choices made by students in answering multiple choice questions, an important concern is the limitation of inference on difficult questions. Difficult questions are questions where we can expect the majority of students to be unable to identify the correct answers, and present a challenge to any model that relies on aggregating judgements. The model's ability to recover the correct answer despite the majority being incorrect, fundamentally requires the model to leverage its estimates of students' abilities so as to weigh the judgements of better students proportionally higher. Also note that we are concerned with questions of great *relative difficulty* (with respect to the ability of the students in the class), not absolute difficulty.

We can simulate an entire spectrum of regimes that present a varying degree of difficulty to inference, and evaluate the model's performance in correctly inferring the correct answers in each regime. We accomplish this by generating a synthetic population of students and questions with a fixed expected *relative competency* (i.e., $\mathbb{E}[s-q]=k$, where $s \sim p(s)$ and $q \sim p(q)$), performing inference with our model on the generated observations, and computing the fraction of correctly inferred correct answers (accuracy) for different $\mathbb{E}[s-q]$. See Figure 2 for an illustration. Note that $\mathbb{E}[s-q]$ is a quantity that conveniently summarizes the classroom in terms of its "competency" relative to the testing material. Large values of $\mathbb{E}[s-q]$ indicate that the students are well-prepared, and most will answer the questions correctly.

5.1. Simulation procedure

We let $p(s)=\mathcal{N}(\mu_s, \sigma=2)$ and $p(q)=\mathcal{N}(\mu_q, \sigma=2)$. We generate a synthetic classroom with the following parameters $|S_{\text{open}}|=10$, $|S_{\text{mcq}}|=10$, $|Q|=15$, where every student in S_{open} submits a open-response to every question in Q, every student in S_{mcq} responds to every question (which entails providing a respond to every option) and $|S_{\text{mcq}} \cap S_{open}|=\emptyset$. We then sample hidden ($\mathbf{z}$) and observed ($\mathbf{y}$) variables from Bernoulli distributions parametrized by (2) and (3) respectively.

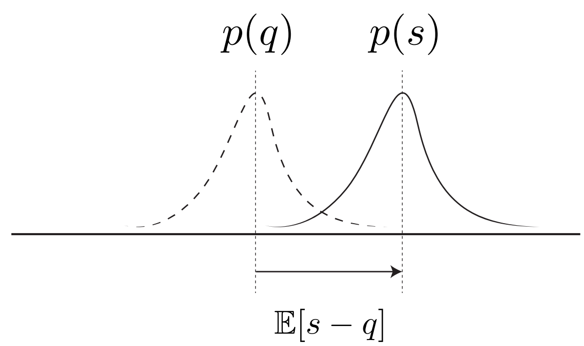


Figure 2. Distributions used in generating synthetic data, where $p(q)$ and $p(s)$ are the distributions of question difficulty and student ability respectively. The quantity $\mathbb{E}[s-q]$ represents the average relative competency of the classroom: a large value of $\mathbb{E}[s-q]$ indicates that the majority of students will answer most of the test items correctly, and vice-versa. See Figure 3 for the effect of the class distribution on performance.

Figure 3 illustrates the performance of the model as a function of the expected relative competency of the students ($\mathbb{E}[s-q]$). We compare the performance of our model to a simple *majority* baseline (i.e., label the answer as correct if the majority of the students select it). As expected, the majority baseline works best when the relative competency of the class is high (since most students will correctly identify the correct answers). The performance degrades significantly in the regime where the relative competency is negative (i.e., most students are expected to answer the questions incorrectly). Observe that the model is able to maintain a significant performance margin (>10%) over the baseline even in the regime of low relative competency.

6. Real-World Experiments

We emulate a classroom setting on the Amazon Mechanical Turk platform by soliciting Mechanical Turk workers to participate in a reading comprehension task. The study was conducted in two separate phases with a different set of workers in each: (i) the *open-response task* and (ii) the *multiple choice task*. In each task, a worker was presented with an article[1], followed by a set of 15 questions. In the *open-response task*, the questions were displayed in an open-response format, and the workers were asked to type in their response. In the *multiple choice task*, the same 15 questions were presented in a multiple-choice format,

[1] Unit 7.2 (Language) from the OpenStax Psychology textbook

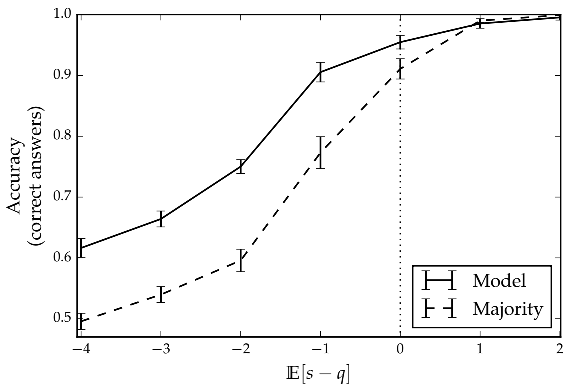


Figure 3. Accuracy in predicting the correct answers on synthetic data, as a function of the average relative competency in the classroom (measured in the multiples of standard deviations of the distributions). The simple majority-vote baseline performs comparably with our model for class distributions with large relative competency (since the majority of the students answer most questions correctly). The model significantly outperforms the baseline in the regime of lower relative competency (i.e., when most questions are too difficulty for the majority of the students).

with the choices aggregated from the open-response submissions obtained in the *open response task*. The answers collected in the *open response task* were clustered semi-automatically before being displayed as choices in the *multiple choice task*. The clustering step aggregated identical answers or answers within a few characters in difference (for example due to spelling errors), and semantically identical answers were then grouped manually (e.g., paraphrases). Clustering answers is a critical pre-processing step as it ensures that a reasonable number of choices is shown as part of the multiple choice question, as well as that the conditional independence assumption discussed in Section 3 holds. In Section 7, we will outline our ongoing work in extending the model to automatically cluster open-response submissions based on the multiple choice response signal alone.

In total, 15 workers participated in the *open-response task* and 82 workers participated in the *multiple choice task*. A total of 225 open-response submissions were generated in response to the total of 15 comprehension questions, resulting in 101 distinct choices after clustering.

6.1. Results

We evaluate the effectiveness of our algorithm on the data collected via Amazon's Mechanical Turk using two performance metrics: (i) accuracy in predicting the correctness of each answer and (ii) quality of the predicted ranking of the students. We evaluate our algorithm in a *semi-supervised* setting where we provide a set of partially labeled items, i.e., we label correctness for a subset of the answers. This represents a practical use-case of our framework—instead of being entirely hands-off, an instructor may choose to manually grade a subset of the students' answers to improve the performance of automatic inference. We evaluate two versions of our model: **EM +open** and **EM -open** in addition to the majority baseline described in Section 5:

- **EM +open:** The full model as described in Section 3 and Section 4.
- **EM -open:** A subset of the **EM +open** model lacking the *open-response* component described in Section 3.2. In other words, during inference the model does not leverage any information about the ability of the answer generator, and relies entirely on the multiple choice responses to infer the correctness of the answers.

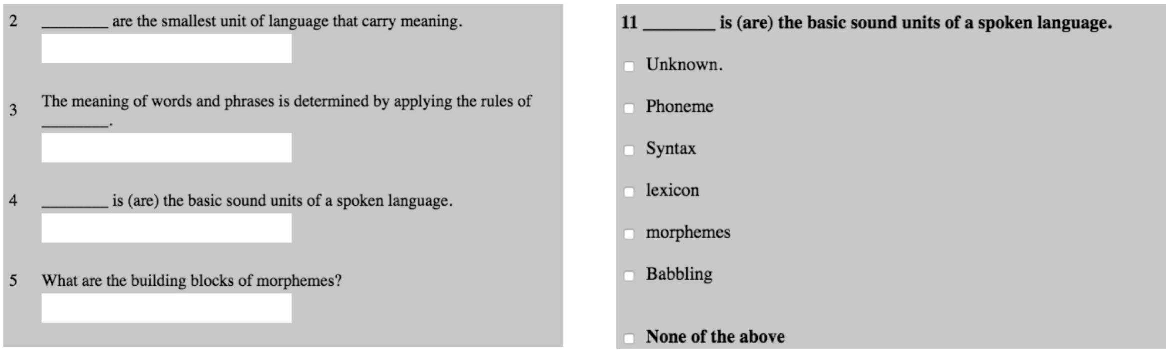
6.1.1. Predicting answer correctness

Figure 5 depicts accuracy as function of the amount of labeled data (accuracy was computed with respect to a gold-standard annotation of correctness for each answer, performed by one of the authors of the paper). From it we conclude that (i) the full model (**EM +open**) significantly outperforms both the majority baseline and **EM -open**, (ii) the **EM +open** performs very well without any labeled data ($\approx 86\%$ accuracy), (iii) adding labeled data improves performance, and (iv) the *open-response* component of the model (one that is lacking in the **EM -open** model) is critical in significantly boosting performance, i.e., incorporating information about the answer creator is valuable in inferring the correctness of each answer.

Initialization: We also note that the initialization heuristic suggested in Section 4 is critical to achieving competitive performance in the regime of little to no labeled data (Figure 6). The performance of the model drops significantly below the majority baseline when a random initialization is used in place of the suggested heuristic.

6.1.2. Predicting student ranking

Although predicting the correctness of each answer is itself a valuable intermediate output, a motivating use-case of our framework is to assess the students' competency. A ranking of the students by their expertise is one example of summative assessment, and may be valuable in identifying students that excel or are in need of additional help. We evaluate the quality of the rankings produced by our model in the following way: (i) use the gold-standard annotation for the correctness of each answer to fit a standard Rasch model, identifying the abilities s_{gold} of each student



(a) *open response task* (b) *multiple choice task*

Figure 4. Screenshot of a segment from each of the two Mechanical Turk tasks. Workers are required to provide an open response answer to each question in the *open response task*, and select (click) all answers that apply in the *multiple choice task*. The choices in the *multiple choice task* are aggregated from the open-response submission of other workers as part of the *open-response task*.

(both in S_{open} and S_{mcq}), (ii) obtain the ability parameters using our model (**EM +open** and **EM -open**) (trained with a varying amount of labeled data) and (iii) rank the students according to each set of parameters and compute rank correlation. We use Kendall-Tau as a metric of rank correlation. Kendall Tau returns a quantity in the range $[-1, +1]$, where $+1$ indicates perfect correlation (every pair of students in both rankings are in a consistent order), -1 when the rankings are inverted, and 0 when the rankings are not correlated.

Figure 7 and Figure 8 depict rank correlation as a function of the amount of labeled answers for the students in sets S_{open} (workers in the **open response task**) and S_{mcq} (workers in the **multiple choice task**) respectively. We observe that (i) incorporating partially labeled set of answers improves rank correlation, (ii) the **EM +open** model performs superior to or on par with the majority baseline (note that **EM -open** is not relevant when ranking the students in the S_{open} set).

6.1.3. Effect of classroom size

In a practical setting, it is important to consider the effect of classroom size on the quality of the inferred parameters. Intuitively, we expect that increasing the number of students answering multiple choice questions $|S_{mcq}|$ will improve performance (accuracy and rank correlation). Figure 9 depicts accuracy as a function of $|S_{mcq}|$ (number of students that answer multiple choice questions), for two conditions based on the amount of partially-labeled answers available. As expected, we observe that the performance of the model (**EM +open**) increases when more students participate in answering MCQs, and the gain becomes more pronounced with a greater number of labeled answers.

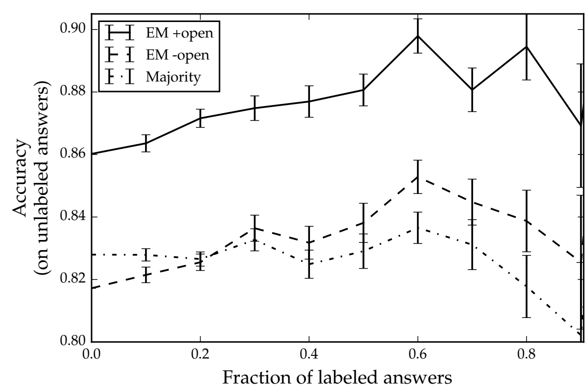


Figure 5. Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk. The model that incorporates both the *open-response* and *multiple choice* components (**EM +open**) significantly outperforms the model that only incorporates the multiple choice component (**EM -open**) and a simple majority-vote baseline.

7. Ongoing work: automatic answer clustering

As we noted in Section 3, in a realistic setting students are likely to submit redundant open-response answers, a significant consequence of which is the violation of conditional independence that is assumed in the existing model. A

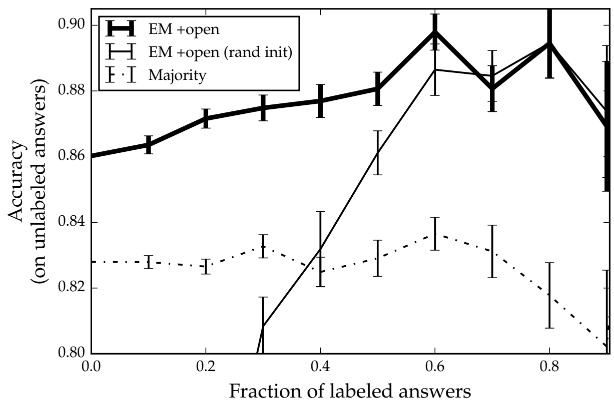


Figure 6. Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk for the model initialized with the heuristic described in Section 4 (**EM +open**) and the model initialized randomly (**EM +open (rand init)**). Good initialization significantly improves performance, especially in the regime of little to no partially labeled data.

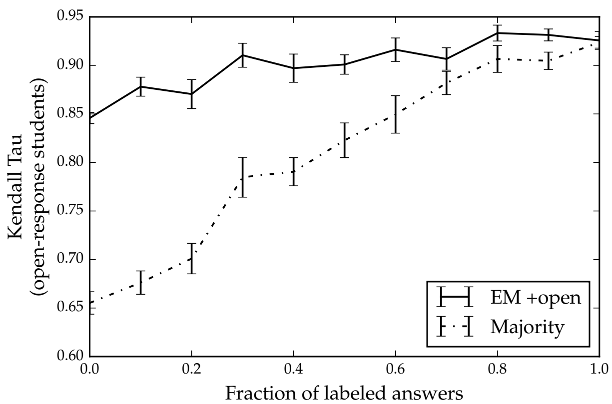


Figure 7. Rank correlation (kendall-tau) for students submitting open-response answers (S_{open}) between the model-inferred ranking (**EM +open**) and the ranking obtained using the gold-standard correctness labels for each answer (via the Rasch model). The model generates high quality rankings with little to no labeled data, significantly outperforming the majority baseline where students are ranked using the parameters obtained from the Rasch model, but where the correctness of each answer is obtained via a majority vote).

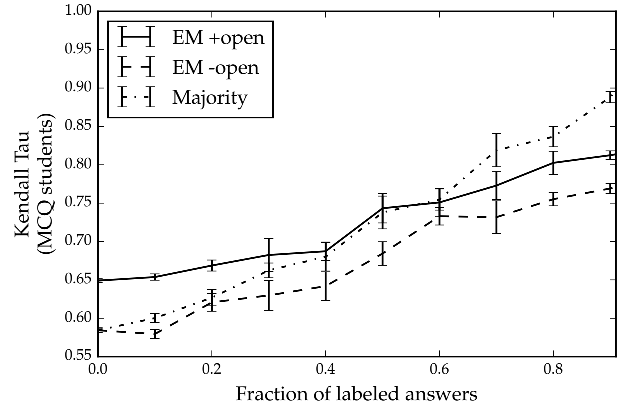
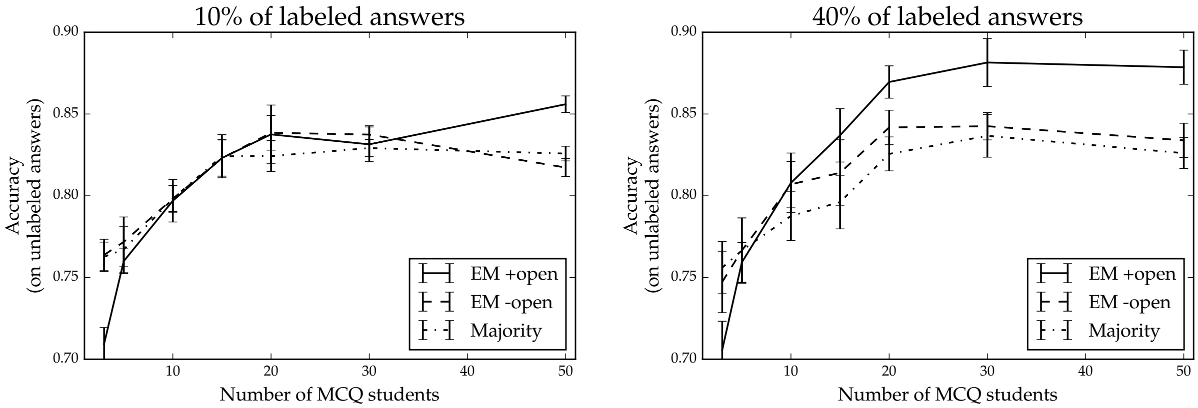


Figure 8. Rank correlation (kendall-tau) for students submitting multiple choice answers (S_{mcq}) between the model-inferred ranking (**EM +open** and **EM -open**) and the ranking obtained using the gold-standard correctness labels for each answer (via the Rasch model). In contrast to the rank correlation for students submitting open-response answers (Figure 7), the rank correlation for multiple choice students is lower.

principled way of identifying related answers and grouping those answers under into a single choice, as a byproduct, would make grading more efficient.

In practice, we expect that the number of truly original answers to a question to be significantly smaller than the number of individual students who answer the question. Answers between students, however, might vary in their phrasing or surface-realization (e.g., simplified vs. unsimplified math expression, short-answer paraphrases, equivalent diagrams drawn in different ways, etc). It is, therefore, natural to think of answer creation from the perspective of a conditional mixture distribution over a smaller set of "latent answers," where for every answer, the student with a certain probability either contributes a variant of an existing answer or an original answer. The Chinese restaurant process (CRP) is one such conditional distribution over "latent answers," with a property that the number of latent answers (clusters) grows logarithmically with the number of observations. This is appealing in our setting where as the number of submitted answers increases, the number of original answers is also expected to grow, but at a diminishing rate (i.e., after many contributions, most submissions will be a variant of an existing answer, rather than an original answer).

We are developing a model that is able to identify the latent clusters of answers based entirely on the pattern of multiple choice responses ("clickthrough"), i.e., without considering the content of the answers (as done in (Basu et al., 2013; Brooks et al., 2014; Lan et al., 2015). This makes the



(a) (b)

Figure 9. Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk as a function of the number of students answering multiple choice questions ($|S_{mcq}|$). More students answering multiple choice questions improves the performance of the model (**EM +open**) in relation to the majority baseline.

approach versatile in its application to any domain, e.g., clustering images, math or language. The model is able to identify answer clusters by relying on the observation of choices that tend to get selected together. We leave a detailed description and analysis of this model for future work. In Figure 10, we illustrate an example set of answers clustered using this model (experiments performed on Mechanical Turk).

8. Conclusion

In this work, we have developed a novel framework for joint grading and assessment (JAG), which—as we believe—offers a powerful alternative to classical peer-grading. The advantage of the proposed JAG framework over traditional peer-grading is that it naturally fuses test-taking and grading into a unified, streamlined process. More importantly, JAG opens the door to a natural way to automatically cluster open-response submissions—a key challenge towards the long-standing goal of scaling assessment to large-scale classrooms.

Acknowledgments

The work of I. Labutov was supported in part by a grant from the John Templeton Foundation provided through the Metaknowledge Network at the University of Chicago. The work of C. Studer was supported in part by Xilinx Inc. and by the US NSF under grants ECCS-1408006 and CCF-1535897.

References

Bachrach, Yoram, Graepel, Thore, Minka, Tom, and Guiver, John. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*, 2012.

Basu, Sumit, Jacobs, Chuck, and Vanderwende, Lucy. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

Brooks, Michael, Basu, Sumit, Jacobs, Charles, and Vanderwende, Lucy. Divide and correct: using clusters to grade short answers at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*, pp. 89–98. ACM, 2014.

Dawid, Alexander Philip and Skene, Allan M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pp. 20–28, 1979.

Haladyna, Thomas M. *Writing Test Items To Evaluate Higher Order Thinking.* ERIC, 1997.

Hung, Nguyen Quoc Viet, Tam, Nguyen Thanh, Tran, Lam Ngoc, and Aberer, Karl. An evaluation of aggregation techniques in crowdsourcing. In *Web Information Systems Engineering–WISE 2013*, pp. 1–15. Springer, 2013.

Kulkarni, Chinmay, Wei, Koh Pang, Le, Huy, Chia, Daniel, Papadopoulos, Kathryn, Cheng, Justin, Koller, Daphne, and Klemmer, Scott R. Peer and self assessment in massive online classes. In *Design Thinking Research*, pp. 131–168. Springer, 2015.

Figure 10. Example of clusters learned from the patterns of multiple choice answers alone. The correlation matrix on the left depicts the individual answers that the model believes belong to the same underlying "latent answer" (red = high probability, blue = low probability). Each answer is listed on the right, grouped according to the maximum a-posterior clustering configuration. Values on the right indicate whether the respective answer cluster is believed to be a correct answer by the model (green = correct (> 0), red = wrong (< 0)).

Lan, Andrew S, Vats, Divyanshu, Waters, Andrew E, and Baraniuk, Richard G. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pp. 167–176. ACM, 2015.

Piech, Chris, Huang, Jonathan, Chen, Zhenghao, Do, Chuong, Ng, Andrew, and Koller, Daphne. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*, 2013.

Raman, Karthik and Joachims, Thorsten. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1037–1046. ACM, 2014.

Rasch, Georg. *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993.

Rodriguez, Michael C. Three options are optimal for multiple-choice items: A meta-analysis of 80 years of research. *Educational Measurement: Issues and Practice*, 24(2):3–13, 2005.

Roediger III, Henry L and Marsh, Elizabeth J. The positive and negative consequences of multiple-choice testing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(5):1155, 2005.

Shah, Nihar B, Bradley, Joseph, Balakrishnan, Sivaraman, Parekh, Abhay, Ramchandran, Kannan, and Wainwright, Martin J. Some scaling laws for mooc assessments. In *KDD Workshop on Data Mining for Educational Assessment and Feedback (ASSESS 2014)*, 2014.

Whitehill, Jacob, Wu, Ting-fan, Bergsma, Jacob, Movellan, Javier R, and Ruvolo, Paul L. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pp. 2035–2043, 2009.

Wu, William, Daskalakis, Constantinos, Kaashoek, Nicolaas, Tzamos, Christos, and Weinberg, Matthew. Game theory based peer grading mechanisms for moocs. 2015.

Zhu, Ciyou, Byrd, Richard H, Lu, Peihuang, and Nocedal, Jorge. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4): 550–560, 1997.