Mining Student Ratings and Course Contents for Computer Science Curriculum Decisions

Antonio Moretti
Research & Innovation Network, Pearson
antonio.moretti@pearson.com

Jose Gonzalez-Brenes
Research & Innovation Network, Pearson
jose.gonzalez-brenes.com

Katherine McKnight
Research & Innovation Network, Pearson
kathy.mcknight@pearson.com

Ansaf Salleb-Aouissi
Center for Computational Learning Systems
Columbia University
ansaf@ccls.columbia.edu

ABSTRACT

University professors of conventional offline classes are often experts in their research fields, but have little training on educational sciences. Current educational data mining techniques offer little support to them. We describe a methodology to learn insights from online data to improve CS course quality. Analyzing CurrIculum Decisions (ACID) mines online student ratings and course contents to learn student opinions within a statistical framework. We build a mixed model to investigate the choice of a programming language for introductory courses and the grading criteria for all courses. We find that interpreted languages and an even weighting of projects and exams correlate with higher instructional clarity ratings.

Keywords

Student ratings, course contents, curriculum decisions

1. INTRODUCTION

There are thousands of undergraduates in computer science programs throughout the US, roughly 24% of whom will switch majors to non-computing fields [10]. An essential component of retaining students is the quality of instruction that they receive in introductory courses [10]. While clear instruction and good pedagogy are widely acknowledged as fundamental to retention, supports for instructors to improve their educational practice are often based on old data; the languages used in computer science courses quickly evolve and old surveys are not useful. In this paper, we are interested in learning insights from online data to improve CS course quality.

The field of educational data mining has been cultivating a strong interest in creating technologies to mine data collected from sophisticated online systems such as intelligent tutoring systems, virtual learning environments, and recently from Massive Open Online Courses (MOOC). The merits of these complex online systems have been demonstrated empirically [3, 11] with controlled studies. MOOCs are a powerful resource that allow educators to study student behavior and social learning in a controlled environment, however the scope of the impact of such technologies is limited. For example, a recent survey of active MOOC users in 200 countries and territories revealed that an overwhelmingly majority of students on these courses correspond to the most educated elite of their respective countries [5]. It is clear that improving basic education worldwide is necessary before MOOCs can deliver their promise. Moreover, because most education still happens offline, it is important to provide educational technologies that can utilize the power of internet to understand student behavior and to deliver these technologies to traditional offline classes. It is not clear how existing educational data mining technologies can help bridge this divide.

We bring attention to the assessment community our results using the *Analyzing CurrIculum Decisions* (ACID) framework. ACID leverages student ratings and course content to discover patterns in learner feedback and how they relate to courses themselves. In prior work [16, 15] we demonstrate ACID on learning student opinions. Here, we extend our analysis to report details of our statistical model. We use Amazon Mechanical Turk (AMT) to aquire the syllabus corresponding to online student ratings sampled from Rate My Professor (RMP)[1]. We use these datasets for relationship learning, addressing the questions of programming language and grading rubric, and propose a model based on general linear mixed models (GLMM). We address the following questions:

1. **What course activities and grading rubric correlate with clear instruction?** The question of how to design a grading rubric and weight course activities

[1] www.ratemyprofessor.com

determines what students focus on within a course. It is important for instructors to optimize course activities and grading criteria with respect to the student experience.

2. **For introductory classes, which programming language(s) correlate with clear instruction?** Academics and industry professionals disagree as to the programming language that is best suited for beginners [21]. For example, some argue that introductory courses should use interpreted languages that allow for a faster understanding of the applications of programming rather than compiled languages that rely heavily on language-specific syntax. Others believe that developing skill with compiled languages is necessary for future work in computer science. The choice of a first programming language likely affects students' decision to continue education within the field of computer science.

3. **Are students more interested in courses with publicly available online syllabi?** The choice to make a syllabus publicly available adds to information available to prospective students on the Web. We hypothesize that the posting of an online syllabus can be used as a proxy for factors including instructor organization and motivation, and that students will both be more interested in and prefer these courses.

The rest of this paper is organized as follows. § 2.1 explains data sources for the ACID methodology; § 2.2 explains data quality for the ACID methodology; § 3 describes three experimental results of evaluating teaching decisions using ACID; § 4 relates to prior work; § 5 concludes.

2. ANALYZING CURRICULUM DECISIONS

We describe our data sources and methodology for leveraging course contents and student ratings. We assess the quality of the data collected by the crowd sourcing platform.

2.1 Data Sources

To evaluate the relative impact of different course features, we mine the web for data that reflect:

- **Course contents** University professors often upload information about their classes. This information is targeted towards prospective or enrolled students and often includes syllabi with detailed descriptions of course material such as textbooks, projects, home-works and exams. We make use of this data to infer teaching strategies.

- **Student perceptions of the course.** We make use of self-selected student evaluations collected from a third-party website. The validity and usefulness of self-selected online rating systems, have been assessed in the literature [2, 17]. For example, evidence suggests that online ratings do not lead to substantially more biased ratings than those done in a traditional classroom setting [2] and that online ratings are a proxy to measure student learning [17]: student learning can often be modeled as a latent variable that causes patterns of observed faculty ratings. Researchers hypothesize a non-linear or concave relationship between student learning and the perceived difficulty level of a course [17]; students learn most when a course is not too difficult or too easy. Our work relies on self-selected ratings as a metric to study learner opinion.

Table 1: Statistics for the Ratings Sample

	Easiness	Helpfulness	Clarity	Interest
Mean	2.84	3.30	3.24	3.35
Std. Dev.	1.33	1.62	1.59	4.00
Median	3.00	4.00	4.00	1.38

We use publicly available self-selected ratings of professors from a third-party website, *Rate My Professor*[2] (RMP). This site allows students to rate the professors of the courses they have taken. The database contains data from over 13 million ratings for 1.5 million professors. They collect ratings on a 1—5 scale (being 1 the lowest possible score, and 5 the highest) under the categories of "easiness", "helpfulness" and "clarity." Additionally students may fill out an "interest" field in which they indicate how appealing the class was before enrolling, and a 350 character summary of their class experience. We focus on perceived clarity because of the direct link between clarity and quality of instruction.

For the purposes of this paper, we focus on Computer Science courses due to our familiarity with the content. Since we do not have access to the ratings database, we develop a process to sample data from the website. For this, we first select a random sample of 50 international universities that teach Computer Science from the Academic Ranking of World Universities[3] [19]. From this sample we only consider the 41 universities are English speaking.

We find, scrape and parse the reviews of the ratings data-set for all professors within the computer science departments of the universities in our sample. We remove the ratings from faculty that were rated by fewer than 30 students. More than one professor can teach the same course. For our analysis, we describe one course listing taught by two different professors as two separate courses. Table 1 shows the mean, standard deviation and median of the ratings in our sample. Figure 1 shows two sample ratings for one professor from our sample. The professor name and course names are removed for privacy.

We use Amazon Mechanical Turk, a crowdsourcing platform, to find course features for each of the courses in our ratings sample. We do this by asking respondents to fill out a survey. The survey requests to provide the URL for the online syllabus that corresponds to the course and professor from which we have ratings that is closest to the date of the student review online. Then, using the syllabus, respondents are asked to to provide the programming language(s) used, the textbook(s) used, and the percentage of the grade that

[2] `ratemyprofessor.com`

[3] Academic Ranking of World Universities is also known as Shanghai Ranking `shanghairanking.com`

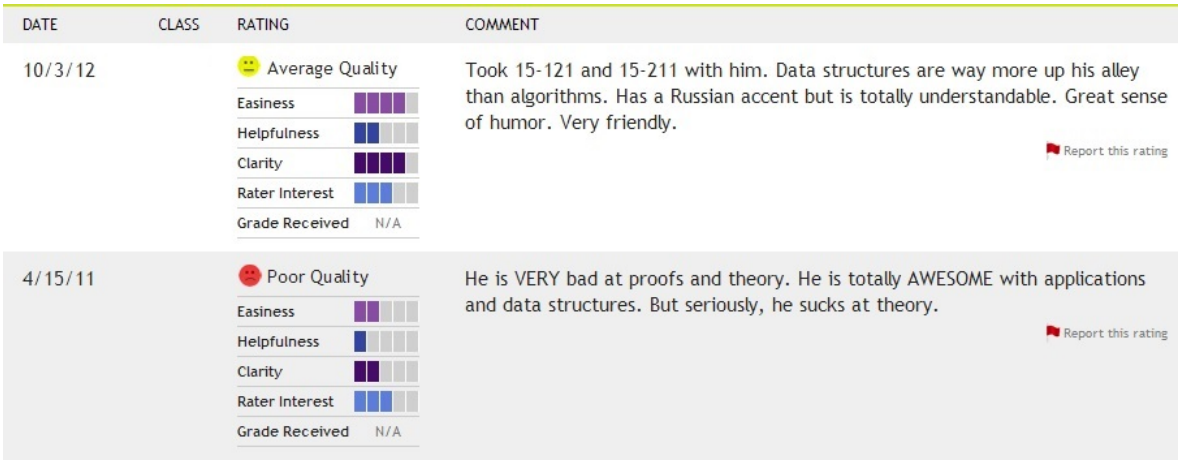

Figure 1: Two Examples from the Ratings Sample

was determined by homework, projects, quizzes, exams and whether the course was taught online or in a blended format (both face-to-face and online). However, when we reviewed the responses to the blended format question, it appeared that most syllabi did not provide enough information by which to make an accurate response.

From our original sample of 1,112 derived tuples (professor, university, course, semester_year), Turkers were able to find syllabi for 342 tuples (31%) We hypothesize three explanations for the missing syllabi: (i) the syllabi may be accessed only with a password through a course management system, such as blackboard, (ii) the syllabi may not be available only, or (iii) the respondents are not able to find the syllabi.

2.2 Data Quality

We now report the how we attempt to collect high-quality data through the use of crowd-sourcing and how we assess the quality of our data.

Mechanical Turk provides a "master" qualification level to respondents that are more reliable. Masters-level respondents require higher compensation for crowd-sourcing tasks than non-masters level respondents although their "acceptance rate," or proportion of approved tasks is much higher. We ran a preliminary experiment, to decide whether respondents on master level qualification provide better quality data for our purposes. We ask respondents to find the syllabus corresponding to a random sample of 30 courses and to answer a set of questions. Table 2 shows the accuracy and interrater agreement of Masters and non-Masters level respondents.

In the pretest we used a screening question to evaluate the accuracy of respondents' data on each task. We asked respondents to find the URL of the website of a randomly selected faculty member at Carnegie Mellon University from a set of 8, from which we knew the answer. We compared the URL they provided with the correct URL to assess accuracy. Of the 13 responses of non-masters workers that did not provide an exact URL match, five responses left the validation question blank. We found that respondents with

Table 2: Respondent Validation

	Accuracy	Interrater Agreement
Masters	100%	96.67%
non-Masters	85.56%	6.07%

master level qualification were significantly more accurate (i.e. answered the validation item correctly) than the non-Masters level respondents (p-value = 0.0002).

Additionally, we tested interrater agreement by asking 3 respondents to carry out the same task, i.e. finding the same URL (for a total of 3x30 or 90 tasks). We used a dummy variable to code whether the three respondents provided the same URL for the course syllabus. Our measure of agreement is calculated by taking the proportion of total responses in which all three respondents provide the same URL. Masters-level respondents agreed (i.e. all three provided the same URL) 100% of the time, whereas the non-Masters level respondents performed much worse – only 6% agreed. As a result of these comparisons, we decided to hire only Masters-level respondents to complete the crowdsourcing experiment.

After collecting the data using Masters level respondents, we performed a post-hoc analysis by examining the responses to the screening question. From the final group of 342 responses that provided a link to an online syllabus, 325 responses (95.03%) provided the correct URL for the faculty website. It should be noted that 13 of the 17 responses that did not provide an exact URL match provided the website for a different faculty member from the set of 8, suggesting that they copied and pasted their previous response without checking to see that the prompt had changed for the new response. Two of the 17 responses provided a link to the directory website for the faculty member rather than the faculty member's personal website. One response provided the correct faculty member's website within the department of Statistics rather than the department of Computer Science (the faculty member is in both departments).

Table 3: VPC and ICC Statistics

	University	Professor	Course
VPC	0.0646	0.3365	0.2355
ICC	0.0728	0.3425	0.1982

3. DATA ANALYSIS: WHAT MAKES A BETTER CLASS?

We report our results of applying the ACID methodology to evaluate teaching decisions. In § 3.1 we discuss the statistical model we use. In § 3.2 we report the results of using ACID.

3.1 Model

We describe our general linear mixed model. We provide descriptive statistics and model selection criteria.

We explore the relationship between student reviews and features collected from online syllabus data using general linear mixed modeling. Student reviews are organized at three levels: by university, professor and course. It is important to note the non-independence of the student reviews due to the hierarchical or clustered nature of the data. We suspect that student ratings within each course, professor and perhaps university are correlated. We begin by estimating the amount of variance attributed to each of these three levels. The simplest multilevel model does not yet include explanatory variables:

$$y_{i,j} = \beta_0 + u_{0,j} + \epsilon_{i,j} \tag{1}$$

The dependent variable $y_{i,j}$ is the clarity rating that student i gave to level j. The term β_0 represents the intercept or mean student clarity rating across all observations. The term $u_{0,j}$ represents the mean clarity rating for level j. The term $\epsilon_{i,j}$ represents the error attributed to student rating i at level j. For comparison we fit a null or single-level model:

$$y_{i,j} = \beta_0 + \epsilon_{i,j} \tag{2}$$

We calculate the percentage of variation in the data set that is separately attributed to each of the three levels of the data. Conventionally the variance partition coefficient (VPC) and intraclass correlation coefficient (ICC) can be interpreted similarly to an R-squared term and are reported in Table 3.

$$\rho = 1 - \frac{\sigma_e^2}{\sigma_e^2 + \sigma_u^2} \tag{3}$$

The VPC and ICC are denoted by ρ, the residual variance is denoted by σ_e^2 and the variance of the effect is denoted by σ_u^2. The ICC is a statistic that is similar to the VPC. However, since the parameter values of the within and between level variance are estimated using sample data, there may be bias due to sampling variation, particularly when there are fewer observations within a given level. The ICC as described by Bartko [1] corrects for this bias by making a small computational adjustment.[4] Observe that the ICC term appears to give slightly less weight to the course effect. It is clear from both statistics that the main effect is the professor effect.

[4]For a description of the computation of the ICC, see the documentation and source code for the R library *lme.*

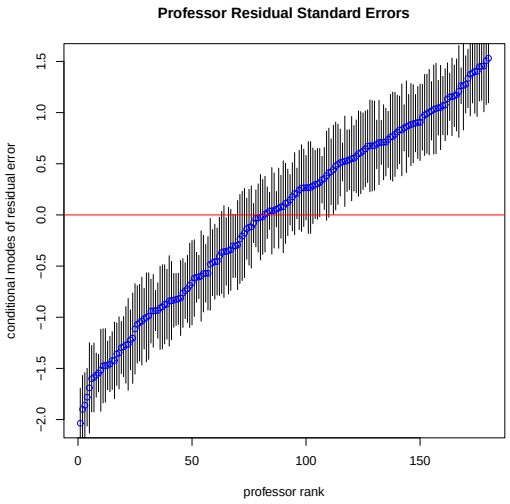

Figure 2: 95% CI for Professor Residual Error

We examine the professor level-residuals and their associated standard errors to look for variation in clarity ratings across professors. The caterpillar plot displays the professor residuals in rank order together with 95% confidence intervals. Wider intervals occur for professors with more student reviews. Observe that the majority of the intervals do not overlap and thus there are significant differences between professors. The blue circles on the far left represent professors who are rated two standard deviations below the mean clarity rating, whereas those on the far right are 1.5 standard deviations higher than the mean clarity rating. The red horizontal line refers to the "average" professor.

We calculate a Chi-squared likelihood ratio statistic by taking the difference between log likelihood values of two successive models. We begin by comparing the null model and the course level model to compare the significance of including the course effect. We continue by adding each of the additional effects. We do not report the values of the test statistic although all additional levels of complexity are statistically significant. We consider the Bayesian information criterion (BIC) and Akaike information criterion (AIC) as model selection tools to avoid over-fitting the data. The BIC and AIC penalize the log-likelihood of a model for the inclusion of extra parameters. The parameters are estimated using restricted maximum likelihood estimation (REML).

We choose the model with the minimum BIC. A two-level mixed model including course effect and professor effect provides the optimal Bayesian information criterion value. Two and three way interaction effects were considered although they did not decrease the AIC or BIC of any of the models. While the log likelihood value is maximized by including the university effect, a simpler model is preferable because it involves fewer parameter estimates and is more likely to generalize. The model can be written in matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\nu} + \boldsymbol{\epsilon} \tag{4}$$

$\mathbf{Y}$ denotes the response variable observations (student rat-

ings). The matrix $\boldsymbol{\beta}$ represents a vector of fixed-effects parameters with a design matrix $\mathbf{X}$. $\mathbf{Z}$ is a design matrix of indicator variables denoting group membership across random-effect levels and $\boldsymbol{\nu}$ is a vector containing random-effect parameters. $\boldsymbol{\epsilon}$ is a vector of error terms. We assume that $\boldsymbol{\nu}$ and $\boldsymbol{\epsilon}$ are normally distributed with:

$$E = \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$Var = \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} \end{bmatrix}$$

This notation allows us to write the variance of test scores, $\boldsymbol{V}$, by using the variance of the random effects and the variance of the residuals. We can now decompose the covariance matrix into the sum of the individual county level variances and the variance of the error terms:

$$\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}' + \boldsymbol{R}. \tag{5}$$

We have four parameters to estimate: $\boldsymbol{\beta}$, $\boldsymbol{\nu}$, $\boldsymbol{G}$ and $\boldsymbol{R}$. The generalized least squares approach is to solve for parameters which minimize the product of the original least squares function and the variance of the test scores.

$$\underset{\beta,G,R}{\arg\min} : (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{6}$$

Since we do not know $\boldsymbol{G}$ and $\boldsymbol{R}$, we begin by estimating these parameters using maximum likelihood estimation. We assume that $\boldsymbol{\nu}$ and $\boldsymbol{\epsilon}$ are normally distributed and solve for the ML estimate using the multivariate normal distribution. Using linear algebra, we can show that the ML estimate of $\boldsymbol{G}$ and $\boldsymbol{R}$ is given by the following:

$$ML : l(\boldsymbol{G}, \boldsymbol{R}) = -\frac{1}{2}log|\boldsymbol{V}| - \frac{1}{2}\boldsymbol{r}'\boldsymbol{V}^{-1}\boldsymbol{r} - \frac{n}{2}log(2\pi) \tag{7}$$

where $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X}) - \boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{y}$, which reflects both $\boldsymbol{\nu}$ and $\boldsymbol{\epsilon}$. After solving for the MLE of $\boldsymbol{G}$ and $\boldsymbol{R}$, we plug these values into (5) and use the result to solve for the parameter estimate $\boldsymbol{\beta}$ that minimizes (6).

3.2 Experimental Results

We show the results of using the ACID methodology to answer three course design questions.

3.2.1 *For introductory classes, which programming language do students associate with clear instruction?*

Professors teaching introductory level courses in computer science choose between a number of programming languages and textbooks. We make use of the data collected to provide insights into which programming languages beginning students associate with clear instruction. We filter the data to only include introductory level courses (one which does not require any prerequisite coursework in computer science). Our restricted sample includes 1,024 reviews; 34.58% of all reviews with syllabus data are of introductory courses. We explore the relationship between clarity ratings and programming language with random professor and course effects. Programming languages with less than 30 student reviews are not reported[5]. Table 4 gives the estimates for student ratings of clarity by programming language and their associated p-values. An intercept is not modeled in order to make the results easily interpretable. The mean clarity rating for introductory courses is 3.599.

[5]SQL is a special purpose programming language used only for relational databases and is not reported.

Table 4: Programming Language Statistics

	Value	Std.Err	t-value	Pr<|t|	n
C	3.38	0.32	10.58	0.0000	109
C++	3.30	0.31	10.65	0.0000	214
Java	3.62	0.19	19.33	0.0000	353
Python	3.70	0.26	14.50	0.0000	133
Scheme	4.06	0.47	8.61	0.0000	32
Scratch	3.91	0.84	4.67	0.0000	49

We found C and C++ had the lowest coefficients (i.e. compiled languages had the lowest perceived clarity ratings). Scheme and Scratch have the highest clarity ratings followed by Python and Java. We note that the standard errors are largest for Scheme and Scratch and smallest for Java and Python. This suggests that results for Java and Python are stronger. Students in our sample associate clearer instruction with interpreted languages rather than compiled languages. Also, both Python and Java are associated with clearer instruction than C or C++.

3.2.2 *What mix of course activities – exams, quizzes, homework and projects – do students associate with clear instruction?*

To assess students' course ratings of clarity based on the percentage of the grade due to exams, quizzes, homework and projects, we created a factor made up of four clusters representing four ways of weighting homework, projects, exams, quizzes and miscellaneous (such as extra credit) for the students' grade. We begin by sorting the data to only include observations in which the grading criteria (percentage of the grade determined by homework, projects, exams, quizzes and miscellaneous) is available and sums to 100. Of the 2,935 observations with syllabus data, there are 2,225 observations with full grading criteria. The difference in these numbers represents 710 ratings for which the respondents were not able to find a complete grade breakdown from the online syllabus.

We use k-means clustering to partition the 2,225 observations with complete grading criteria information based on the five aforementioned variables. We optimize k, our number of clusters, by examining how the BIC and AIC of the mixture model change based on the number of clusters selected. Figure 3 displays the information criterion and Figure 4 displays the log-likelihood values for each number of clusters respectively. A solution involving two clusters minimizes the BIC of the model, whereas a four cluster solution minimizes the AIC. The log likelihood is optimized with the four cluster solution. We consider both two and four cluster models as optimal and we find that they lend themselves to similar interpretation. The cluster means for the four cluster solution are presented in table 5.

The first cluster represents courses that are heavily weighted towards exams with a smaller weight towards homework. The second cluster represents a more even weighting of ex-

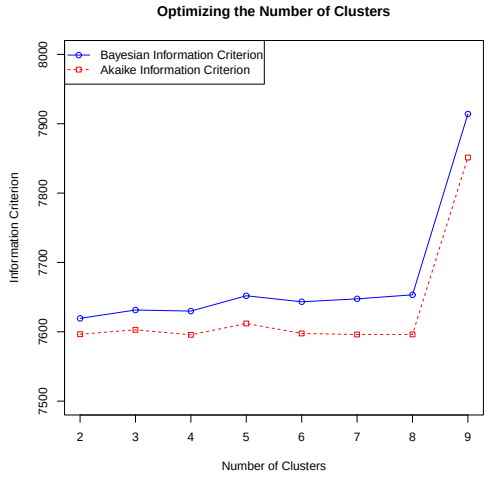

Figure 3: Information Criterion

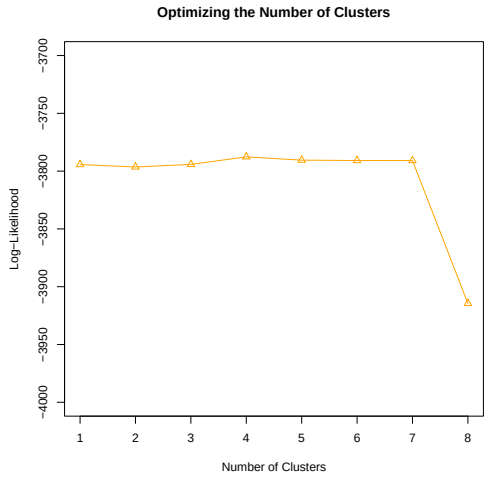

Figure 4: Log Likelihood

Table 5: Cluster Statistics

	HW	Projects	Exams	Quizzes	Other
Cluster1	18.11	2.36	76.66	0.61	2.25
Cluster2	20.59	7.90	48.90	12.46	10.15
Cluster3	7.00	40.18	46.23	3.51	3.08
Cluster4	42.93	0.76	54.61	0.70	2.00

Table 6: Grading Criteria Statistics

	Clarity	Std.Err	t-value	Pr<|t|	n
Exam Heavy	3.23	0.12	26.91	0	726
Equal Mix	3.52	0.14	26.04	0	484
Exam Proj	3.65	0.13	27.76	0	610
Exam HW	3.12	0.13	23.53	0	415

ams, homework, projects and quizzes. The third cluster represents an equal weighting towards exams and projects. The fourth cluster represents courses that are heavily weighted towards exams and homework. The cluster membership is treated as a predictor variable and modeled using equation 4. Table 6 displays the estimated clarity ratings within each group for the four cluster solution.

The exams and projects cluster has the highest estimate of clarity. We find that weighting projects equally with exams is associated with a clearer course experience. The equal mix cluster also is associated with higher clarity estimates. The exam heavy cluster and the exam and homework heavy clusters are associated with lower student clarity ratings. We find that a rubric that weights exams and projects evenly has higher perceived clarity ratings to a rubric which is weighted heavily towards exams and homework. This result extends to both two and four cluster solutions.

Table 7: Online Syllabi

	Clarity	Std. Err	t-value	Pr<|t|	n
Available	3.33	0.07	44.48	0	2953
Not Found	3.26	0.07	46.03	0	7702

3.2.3 Does the posting of a syllabus online translate into higher ratings?

We hypothesize the posting of the syllabus online is a proxy for organization, perhaps motivation or drive of the professor. We make use of all of the data collected to compare student reviews of professors who have a publicly available syllabus and of those who do not. Many professors may choose to only post a syllabus through course management systems that require a password. Potential students of these courses are unable to access the syllabus to determine whether the course would be a good fit. We treat the posting of an online syllabus as a factor and test for differences in clarity ratings between the two groups using our model.

We find statistically significant differences between clarity, helpfulness and interest ratings and report the clarity estimates for the two groups in Table 7. We note that the difference in easiness ratings is not statistically significant. We find evidence that students are more interested in professors and courses in which the syllabus is made publicly available. We note that the parameter estimates for the two groups are within one standard error of one another which suggests that the conclusions are modest.

4. RELATION TO PRIOR WORK

Research has recently focused on online faculty ratings with mixed conclusions. Felton et al. [6] found that online instructor ratings were associated with perceived easiness, and that a "halo effect" existed in which raters gave high scores to instructors perhaps because their courses were easier. We find that student ratings of clarity and easiness are correlated (ρ=0.45) although not as strongly associated as clarity and helpfulness. We do find that student ratings of clarity and helpfulness are highly correlated (ρ=0.84). We chose to focus on clarity ratings as we assumed these were less susceptible to a "halo effect" and other bias relative to the overall ratings of a course or professor. Otto et al [18] found issues related to bias in online ratings stating that online ratings

are characterized by selection bias as anyone can enter faculty ratings at any time. Carini et al [2], Hardy [8], McGhee and Lowell [9] had contradictory results finding that an online format did not lead to more biased ratings. Otto et al. [17] hypothesized that instructor clarity and helpfulness as captured by Rate My Professor are more positively associated with student learning than easiness.

Several approaches have been proposed to synthesize responses using crowd sourcing systems such as Amazon's Mechanical Turk. Majority voting is perhaps the simplest way to combine crowd responses using equal weights irrespective of respondent experience. The results of our preliminary analysis in accessing the accuracy of non-Masters level respondents correspond to the steep drop in respondent accuracy noted by Karger [12] when low-quality respondents are present. Whitehill et al [20] proposed a probabilistic model for combining crowd responses called Generative model of Labels, Abilities and Difficulties (GLAD). The GLAD methodology makes use of the EM algorithm to calculate parameter estimates of unobserved variables including an approximation of the expertise of the rater. Khattak and Salleb-Aouissi compared the accuracy and percentage of bad responses using majority voting, probabilistic models, and their novel approach entitled Expert Label Injected Crowd Estimation (ELICE) [13, 14]. ELICE makes use of a few "ground truth" responses and incorporates expertise of the labeler, difficulty of the instance and an aggregation of labels. Khattak and Salleb-Aouissi found that their approach was robust and outperformed GLAD and iterative methods even when bad labelers were present. Our simple approach was to use Masters level respondents from Mechanical Turk although GLAD and ELICE are alternative methods to reduce the number of expert level respondents required while also obtaining high quality data.

5. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

We demonstrate Analyzing CurrIculum Decisions to discover that interpreted languages and an even weighting of projects and exams correlate with higher instructional clarity ratings. The benefits of our approach are that it is efficient and inexpensive due to its use of publicly available information on the Web. We argue that ACID is an interesting tool to discover patterns in learner feedback for the assessment community. Syllabus data and course ratings data are becoming increasingly available on the Web. This data is used by millions of students and worthy of further research. We also discuss the usefulness of multilevel modeling in the case of hierarchically structured data. These models are increasingly powerful to estimate within and between group variance for complex data. In future work we plan on comparing multilevel models to relational learning methodologies [7, 4].

This study can be expanded in several ways. There is a wealth of text data both in free-form student comments and in the syllabi URLs themselves. Sentiment analysis is a probabilistic approach for categorizing the polarity of student comments. One extension is to regress text sentiment on course features. There is arguably a strong association between comment sentiment and student preference. Another way ACID can be applied is to disciplines other than computer science, or to discover patterns in syllabi text directly across disciplines that can provide insight into learner experiences.

6. REFERENCES

[1] J. Bartko. The intraclass correlation coefficient as a measure of reliability. *Psychological Reports, National Institute of Mental Health*, 19:3–11, 1996.

[2] R. Carini, J. Hayek, G. Kuh, J. Kennedy, and J. Ouimet. College student responses to web and paper surveys: does mode matter? *Research in Higher Education*, 44(1):1–19, 2003.

[3] A. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva, editors, *User Modeling 2001*, volume 2109 of *Lecture Notes in Computer Science*, pages 137–147. Springer Berlin Heidelberg, 2001.

[4] S. DÃ¿eroski. Relational data mining. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 887–911. Springer US, 2010.

[5] E. J. Emanuel. Online education: Moocs taken by educated few. *Nature*, 503(7476):342–342, 2013.

[6] J. Felton and J. Mitchell. Web based student evaluations of professors: the relations between perceived quality, easiness and sexiness. *Assessment and Evaluation in Higher Education*, 29(1):91–108, 2004.

[7] L. Getoor and B. Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

[8] N. Hardy. Online ratings: fact and fiction. *New Directions for Teaching and Learning*, (96):31–38, 2003.

[9] N. Hardy. Psychometric properties of student ratings of instruction in online and on-campus courses. *New Directions for Teaching and Learning*, 2003(96):39–48, 2003.

[10] M. Haungs, C. Clark, J. Clements, and D. Janzen. Improving first-year success and retention through internet-based cs0 courses. *ACM SIGCSE*, pages 549–594, 2012.

[11] S. Jaggars and T. Bailey. Effectiveness of fully online courses for college students: Response to a department of education meta-analysis. *Teachers College: Community College Research Center*, 2010.

[12] S. Karger, D. Oh and D. Shah. Budget–optimal task allocation for reliable crowdsourcing systems. *CoRR*, arXiv:1110.3564, 2011.

[13] F. Khattak and A. Salleb-Aouissi. Robust crowd labeling using little experience. *Discovery Science*, 8140:94–109, 2013.

[14] F. K. Khattak and A. Salleb-Aouissi. Improving crowd labeling through expert evaluation. In *AAAI Spring Symposium: Wisdom of the Crowd*, 2012.

[15] A. Moretti, J. Gonzalez-Brenes, and K. McKnight. Mining the web to leverage collective intelligence and learn student preferences. *EDM*, 2014.

[16] A. Moretti, J. Gonzalez-Brenes, and K. McKnight. Towards data–driven curriculum design: Mining the web to make better teaching decisions. *EDM*, 2014.

[17] J. Otto, D. A. Sanford Jr, and D. N. Ross. Does

ratemyprofessor. com really rate my professor? *Assessment & Evaluation in Higher Education*, 33(4):355–368, 2008.

[18] J. Otto, D. A. Sanford Jr, and W. Wagner. Analysis of online student ratings of university faculty. *Journal of College Teaching & Learning*, 2(7):25–30, 2005.

[19] Shanghai. Academic ranking of world universities. *Retrieved from http://www.shanghairanking.com/*, Accessed at 2013 12 01.

[20] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Neural Information Processing Systems*, pages 2035–2043, 2009.

[21] J. Zelle. Python as a first language. *Retrieved from http://mcsp.wartburg.edu/zelle/python/python-first.html/*, Accessed at 2014 02 23.