As CS Enrollments Grow, Are We Attracting Weaker Students? A Statistical Analysis of Student Performance in Introductory Programming Courses Over Time

Mehran Sahami
Computer Science Department
Stanford University
Stanford, CA 94305, USA
sahami@cs.stanford.edu

Chris Piech
Computer Science Department
Stanford University
Stanford, CA 94305, USA
piech@cs.stanford.edu

ABSTRACT

In recent years, enrollments in undergraduate computer science programs have seen tremendous growth nationally. Often accompanying such growth is a concern from faculty that the additional students choosing to pursue computing may not have the same aptitude for the subject as was seen in prior student populations. Thus such students may exhibit weaker performance in computing courses. To help address this question, we present a statistical analysis using mixture modeling of students' performance in an introductory programming class at Stanford University over an eight year period, during which enrollments in the course more than doubled. Importantly, in this setting many variables that would normally confound such a study are directly controlled for. We find that the distribution of student performance during this period, as reflected in their programming assignment scores, remains remarkably stable despite the large growth in enrollment. We then explain how the notion of having "more weak students" and the fact that the distribution of student ability is unchanged can readily co-exist and lead to misperceptions about the quality of incoming students during an enrollment boom.

Keywords

Enrollment growth; mixture modeling; student performance; introductory programming.

1. INTRODUCTION

The past decade has seen tremendous growth in enrollments in computer science programs. Indeed, the Computing Research Association 2014 Taulbee Survey [12] shows a near doubling of the number of newly declared CS/CE undergraduate majors from 2007 to 2014. The increase in majors is correlated with a sharp increase in the number of students taking introductory computer science courses, which has received much attention of late in the popular press [9, 11].

SIGCSE '16, March 02 - 05, 2016, Memphis, TN, USA

ACM 978-1-4503-3685-7/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2839509.2844621

The increasing numbers of students taking CS courses has led some faculty to lament the quality of students entering such programs. Indeed, commentary along these lines is readily found in public online forums frequented by computing faculty. For example, consider the following quote from a former EECS professor at a well-known university [8]:

> *Based on my recent teaching experience, there are definitely many students signing up for CS majors who aren't prepared enough and have difficulty getting through foundational material. Some of them manage to learn through perseverance (retaking courses several times), some don't. They won't be nearly as efficient as median students…*

More anecdotally, we have often heard colleagues questioning whether the students in CS courses today are as strong as those in the past. The thinking along these lines being that when enrollments were lower only the students who had both the aptitude and interest in computing were enrolling. Now, with CS becoming a "hot" major, we are potentially attracting students who have interest, but perhaps less aptitude than before.

Rather than allowing such anecdotal statements to continue unchecked, in this work we seek to provide a statistical analysis of student performance in introductory programming courses during the recent period of large enrollment growth (2007-2014) to quantify the extent to which we may be attracting weaker students into such courses. It is especially important to counter anecdotes with rigorous analysis in such a context. Unsubstantiated statements about "weaker" students can potentially be even more damaging and off-putting to students from under-represented populations who may be particularly susceptible to stereotype threat [3, 7, 10].

Based on building robust statistical models, we find that, contrary to the perception of attracting weaker students, the quality of student work in the introductory programming course we analyze has remained remarkably consistent during the entire eight year period we consider. Indeed, through a mixture modeling approach to fitting the distribution of grades on students' assignments, we find clear evidence that the distribution of student quality (as reflected in assignment scores) is virtually unchanged even in the face of an over 100% increase in enrollment. By examining the implications of the model in more detail we are able to reconcile the quantitative results regarding consistent student populations over time with the reasons that potentially give rise to anecdotes about "more weak students" taking CS courses. In this way, we seek to not only provide a quantitative basis for understanding the implications of enrollment growth, but also help to provide educators with insight about potential misperceptions that can arise in such situations.

The remainder of the paper is organized as follows. In Section 2 we describe the introductory programming course and data we analyze in our study. Section 3 provides details of the statistical methods we use to model the data and gives the results obtained from our statistical analysis. In Section 4 we discuss the results in this study further, highlight our main contributions, and present directions for future work.

2. THE DATA

Analyzing the performance of student populations over time can potentially suffer from many confounding factors, such as the efficacy of different instructors in a course, the impact of different pedagogies on learning, and the use of different assessment instruments. Luckily, in our setting, all these factors are directly controlled for. Specifically, we consider CS106A, an introductory programming course using Java (equivalent to CS1) taught at Stanford University. This course was taught by the same instructor (the first author) every Autumn term for the past eight years (2007-2014). During this time the contents of the course (e.g., textbook, handouts, lectures, section materials) remained largely unchanged (by design) and the same instructional pedagogy was employed. Moreover, the course used the same seven programming projects in every Autumn term offering and employed the same grading rubrics for their evaluation.

The main variable that changed during the period from 2007 to 2014 is the size of the course, which grew from less than 300 students to more than double that size. As a result, the data on student performance in this course (as measured by programming assignment grades) over time provides a highly controlled setting to test hypotheses related to the performance of students enrolling in the course as enrollments have increased.

It is important to be precise about the data that we consider in our study. In order to use a performance measure that is directly comparable across years, we focus on students' aggregate (weighted sum numeric) grade on the seven programming assignments/projects in the course, as these are consistently evaluated across years. In aggregate, the seven programming assignments make up 50% of the overall course grade. We do not include exam data (the course has a midterm and final exam) as the exams are different each term. Thus, exam scores are not directly comparable across terms as difficulty across exams is not directly calibrated. Trying to equate means/variances of exam scores across classes would create statistical distortions in our otherwise controlled data. We recognize that the exclusion of exam scores may present a potential threat to the validity of our results as assignment grades are only one measure of students' ability in a programming course. However, we believe that the distortive effects of exam data normalization across courses would present an even greater threat to the validity of cross-term comparisons. Thus, exam data is not included in our analysis.

We start by considering the full set of students officially enrolled in the course – that is students who have included the course on their study list by the "add" deadline, which is the end of the third week of our 10 week term. Students who then "withdrew" from the course (i.e., dropped the course before the end of the eighth week of the term) are excluded from our dataset of assignment scores (as their set of assignment submissions are intentionally incomplete), but such students are then examined separately so we can determine if there is a significant increase in the percentage of such students over time. This allows us to separately analyze the performance of those students who completed the course from those who withdrew, to have a finer level of resolution in examining student performance.

Next, any students with documented academic integrity issues are excluded from the dataset. The reason for this is that we want the data to be representative of students' *actual* performance, and thus we try to minimize the influence of plagiarism/cheating to the extent that it can be identified. It is important to note that Stanford is an Honor Code school and students were given the opportunity to retract assignments before the end of the course if they felt that the work they submitted was not entirely their own. Any student with one or more such assignment retractions or who was otherwise found to have an academic integrity issue (through the use of plagiarism detection software) was removed from the dataset as their assignment scores no longer entirely reflected their own work. We note that information regarding academic integrity issues at Stanford must be treated with great delicacy as these cases are considered highly confidential. For these reasons, we cannot provide more details with respect to that aspect of the data (e.g., the actual number of data points removed as a result of this filtering, separate analysis of students involved in such issues, etc.), other than to say that the number of data points filtered as a result was generally small and that the filtering process does not qualitatively impact the results we report here.

Finally, to further improve data integrity, we did a manual inspection of all the remaining data to remove any obviously "dirty" data (e.g., students with 0 scores on all assignments/exams who had mistakenly enrolled in the course or forgot to drop, but clearly had no intention of actually trying to complete the class). This final stage resulted in the filtering of very few data points. The remaining data was used in the analysis we discuss presently.

Figure 1 shows the number of students completing the course (i.e., data points after the filtering process above was applied) in the Autumn term of each year as well as the number of students who withdrew from the course in the respective term. The actual numbers are included in the graph above each data point.

We note that the number of students withdrawing from the course in any given offering is very small on a relative basis – never greater than 3.5% of the starting population of the class. In fact, the percentage of students withdrawing from the class in 2007 is essentially the same as that in 2014 (2.8% in both cases) when enrollments were more than double. Since the percentage of withdrawals remains very low and stable during the period in which enrollments grew, it provides no indication—as far as withdrawals are concerned—that the student population enrolling in the course as time goes on is less capable on a relative basis, as we would have expected to see an increase in the *percentage* of withdrawals if that were the case.

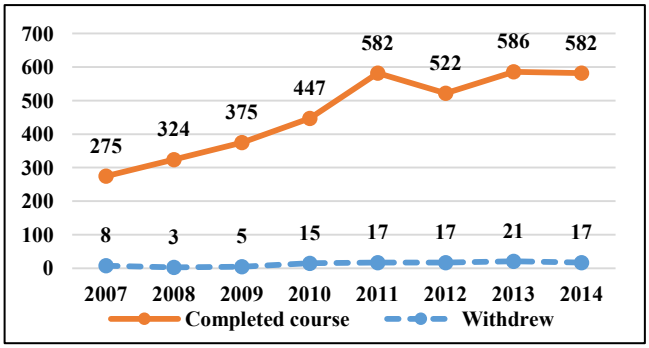

Figure 1. Students completing and withdrawing from course

More importantly, the number of withdrawals is quite small on an absolute basis, and thus would not provide a robust indicator of potential changes in the aptitude of students taking the class even if we saw slightly more fluctuation in this measure. Thus, to see whether the student population has become "weaker" as enrollments have grown we need to consider a detailed analysis of the assignment scores of students who completed the class.

3. RESULTS

3.1 Initial Comparison of Classes

Perhaps the most obvious initial test to consider in determining whether the performance of students in our introductory programming course has changed as enrollments have grown is to consider a T-test of mean assignment scores in the course over time. Setting the 2007 class as the baseline (as it has the smallest enrollment in our controlled data), we perform T-tests between this class and the class in every subsequent year. Essentially, we are looking to see if there is any evidence to reject the null hypothesis, which is that the mean assignment score in the 2007 class is the same as any other class we compare against. We report the p-values from these T-tests (two-tailed, heteroscedastic tests) between the 2007 class and all other years in Table 1. As a side comment, we note that performing ANOVA across all the classes simultaneously would not accurately reflect differences in class means *as enrollments grow* since the test statistic would also be influenced by differences among classes with comparable enrollment levels (e.g., the 2011 and 2014 classes).

Table 1. p-values of T-tests comparing to 2007 class

Class	2008	2009	2010	2011	2012	2013	2014
p-value	0.69	0.69	0.82	0.10	0.11	0.46	0.82

The T-test results in Table 1 do not lead us to reject the null hypothesis at the $\alpha = 0.05$ level for *any* class we compare against, indicating that there does not appear to be a statistically significant difference in the mean scores between the 2007 class and any other class compared against (i.e., classes with much higher enrollments). We do note that the comparison with the 2011 class reaches the $\alpha = 0.10$ significance level, but this is tempered by the fact that *seven* T-tests are being performed, so we would expect some T-tests to result in lower (and potentially close to statistically significant) values even when there is no real difference in the means of the data sets. Interestingly, in comparing the 2007 class with the 2014 class, we find a p-value of 0.82, indicating that it is in fact extremely likely that the means of the two classes are the same, despite the fact that enrollment in the former class is less than half that of the latter class.

To see the stability of class means over time, Figure 2 provides a graph of the assignment means (actual mean values are also given in the graph), with error bars showing one standard deviation around the mean, respectively for each class. Aggregate assignment scores are on a 50 point scale, although a few points of extra credit are possible.

We caution, however, that a lack of statistical significance in the difference of the *means* does not necessarily imply that the *distributions* of the assignment scores are also similar. Indeed, in some cases it is possible for two *distributions* to be quite different, but still yield similar *means* using a T-test (for example, if the distributions have different higher order central moments, such as variance or skewness). To address this point, we next consider analyzing the *distribution* of scores in each class via mixture modeling.

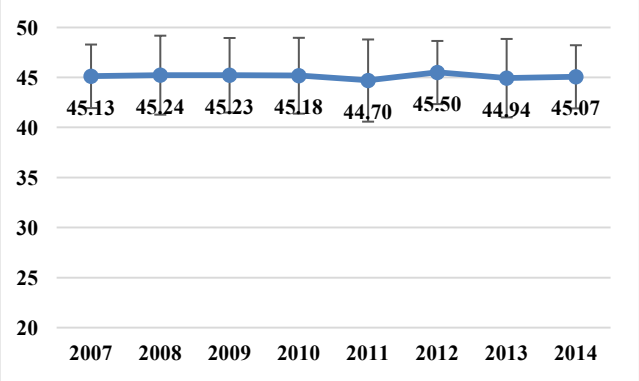

Figure 2. Means of student assignment scores (values labeled), with error bars indicating standard deviation in each class

3.2 Mixture Modeling

At a high level, *mixture modeling* is the task of modeling a probability distribution as a weighted sum of two of more component distributions [5]. Such modeling is often used when a probability distribution may not be fit well by a single parametric distribution (e.g., a single Gaussian), but is more accurately captured by combining two or more such models – for example, a multi-modal distribution.

More formally, a mixture distribution $f(x)$ with K components is a weighted sum of component distributions f_1, f_2, …, f_K, with respective component weights w_1, w_2, … w_K, defined by the equation (eq. 1):

$$f(x) = \sum_{i=1}^{K} w_i f_i(x)$$

In our analysis, we consider parametric mixture models, where the component distributions f_i are all Gaussians with different respective means μ_i and variances σ^2_i. The mixture models are fit to the data using the EM (Expectation Maximization) algorithm [1, 2, 4], which is guaranteed to converge to a local optimum of the likelihood function of the data (i.e., it finds parameters μ_i and σ^2_i for all the component Gaussian distributions as well as component weights w_i that jointly maximize the likelihood of the data being fit). While EM is an iterative algorithm that can be sensitive to the starting point chosen (as it is only guaranteed to converge to a local optimum of the likelihood function), we note that given how few parameters are being fit in our models, our results were extremely stable with EM producing exactly the same set of parameters for a given data set when initialized at a variety of different starting points.

To illustrate why mixture models are needed (rather than just fitting a single Gaussian to the data) let us consider the case where we try to model the distribution of student assignment scores with a single Gaussian (which would be equivalent to a mixture model with only one component (i.e., K=1)). Figure 3 shows a histogram of the actual data (student assignment scores) for the 2014 class compared with the values that would be expected using a single Gaussian model with mean and variance fit to the data.

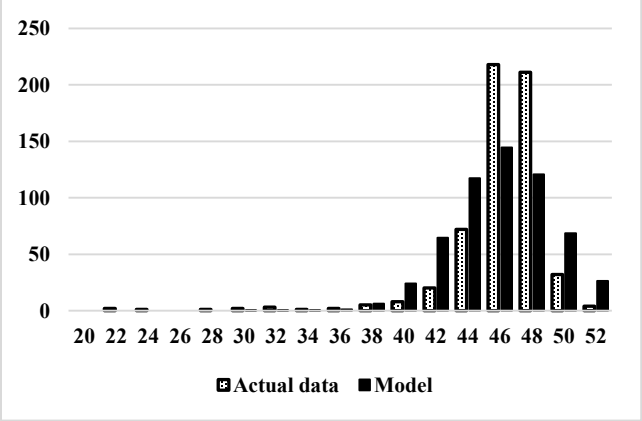

Figure 3. Actual scores compared to a single Gaussian model

As is clear in Figure 3, the single Gaussian provides a poor fit to the actual data as it underestimates the most common score ranges (i.e., 46-48, 48-50) and overestimates other ranges.

Alternatively, we can consider a mixture of two Gaussians (i.e., K=2) fit to the same data (2014 class). We call this the 2-Gaussian mixture model. Figure 4 illustrates the two Gaussian components that are found in the data via the EM algorithm, showing one component that captures much of the lower tail of the distribution (Component 1) and another much more peaked component that reflects the most common score ranges (Component 2).

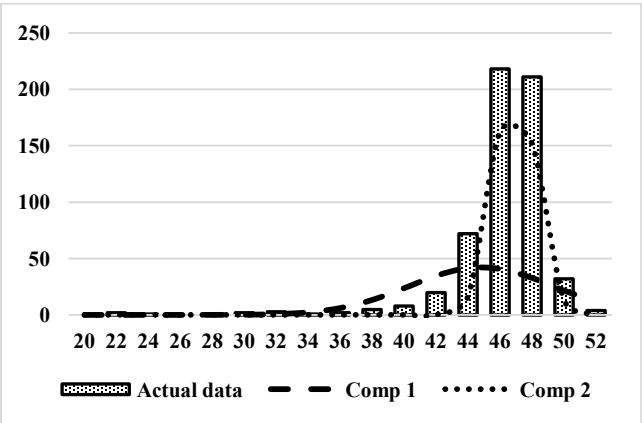

Figure 4. Components in Gaussian mixture model with K=2

We can combine these two components into a single distribution using the mixture model equation (eq. 1) along with the respective component weights also fit using EM. The resulting 2-Gaussian mixture model is graphed alongside the actual data in Figure 5. As can be seen in the figure, the 2-Gaussian model provides a much better fit to the data.

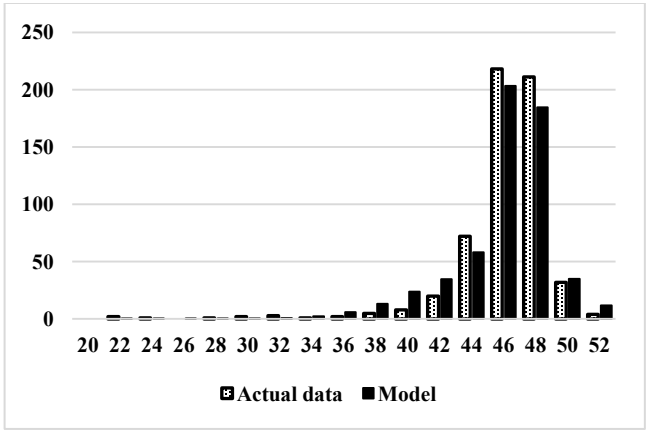

Figure 5. Actual scores compared to a 2-Gaussian model

More importantly than just the accuracy of the model, the 2-Gaussian model gives us greater insight about the distribution of students in the class. Namely, there is one group of students (reflected in Component 2) that is generally doing well and shows a solid understanding of the course assignments as reflected in their scores. There is another group of students (the bottom half of Component 1) which appears to be comprised of "weaker" students with regard to the assignments. This group makes up the tail of the class distribution. We turn our attention to analyzing these two groups over time momentarily.

Here, we pause to observe that it is also possible to fit mixture models with more than two components to the data. Indeed, we considered three-component (i.e., K=3) mixture models and found that they led to qualitatively similar results as the two-component models, but were harder to directly interpret as they had greater model complexity and more parameters. Others [6] have also found that two-component mixture models provide a compelling fit to educational data related to student activity, albeit in a different context than ours. While we could have engaged in Bayesian model selection to determine the number of mixture components more automatically, this would have required the use of a *subjective* prior distribution over the number of model components, which would be open to argument, and would further complicate potential interpretation of the model. Thus, we simply note that it is possible to construct more complex models using the methodology presented here, but for the remainder of our discussion we focus on two-component models for clarity as they provide an excellent fit to the data without overly complicating the interpretation of the model.

3.3 Mixture Components Over Time

After discovering that a two-component model provides both a good fit to the data as well as giving us insight about the subpopulations of students in the course (as reflected in assignment scores), we can look at the evolution of these two components over time by fitting a two-component model to the data set of assignment scores for each respective year.

Figures 6 and 7, graph the respective assignment means for Component 1 and Component 2 in the 2-Gaussian model for each class over time (the actual mean values are also given in the graphs). Error bars in both figures show one standard deviation associated with each respective mean.

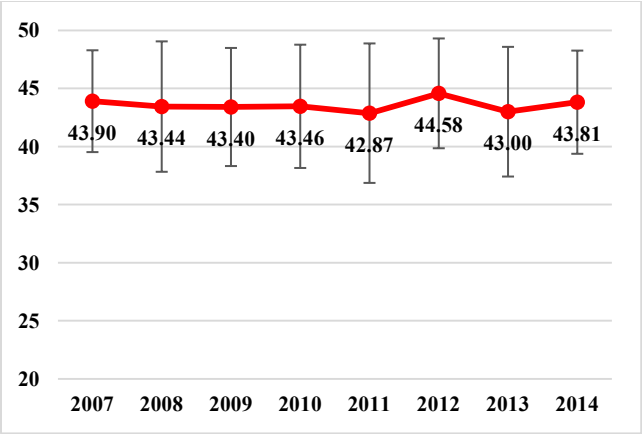

Figure 6. Component 1 means of student assignment scores (values labeled), with error bars indicating Component 1 standard deviation for each class

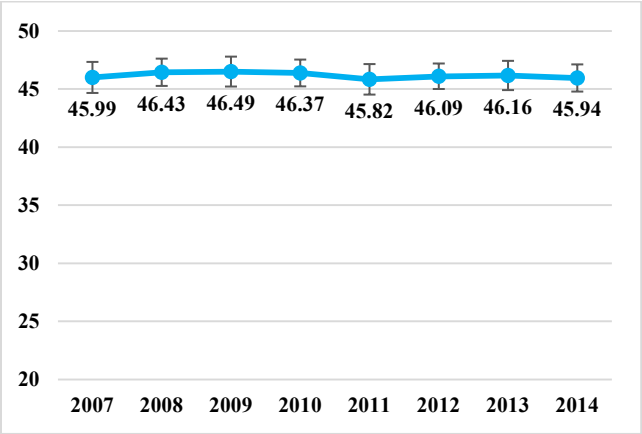

Figure 7. Component 2 means of student assignment scores (values labeled), with error bars indicating Component 2 standard deviation for each class

We see that the Gaussian components, fit *independently* for each data set (year) are remarkably stable over time. Indeed, comparing the 2007 class to that in 2014 shows that the means for both respective components differ by less than one-tenth of one point on an absolute 50-point scale or less than 0.2% on a relative basis. This gives us compelling evidence that the subpopulations of students in the course (at least with respect to assignment scores) have not changed substantially as enrollments have grown over time.

Nevertheless, there is yet one more critical factor we need to examine in such a model. Knowing that the component means (and standard deviations) of the distribution of students in the class has not changed over time reflects that there are two clear and stable subpopulations in the course. However, we must also recall that a mixture model weights the component distributions (with weights w_i) in the sum that forms the mixture distribution. Thus, we must examine if there is a trend in the mixture *weights* over time, which could reflect, for example, that a larger proportion of students might be coming from Component 1 as enrollments increase.

We must pay careful attention to the nuances of such an analysis, as Component 1 overlaps with Component 2 in the mixture model for every class year we fit (see Figure 4 for a graphical example of this phenomenon). Notably, the larger variance of Component 1 allows it to capture two ends of the distribution—both the left-hand tail of "weak" students as well as the right-hand tail of "very strong" students. To address this issue, we consider the proportion of students in each class whose probability of being assigned to Component 1 in that class is greater than that of being assigned to Component 2 (i.e., they are captured by Component 1) and have scores *below the mean* of Component 1. This essentially captures just the left-hand tail of students in each class (i.e., the lower half of Component 1). The graph of the percentage of such students in each class is given in Figure 8 (actual percentages are also given in the graph).

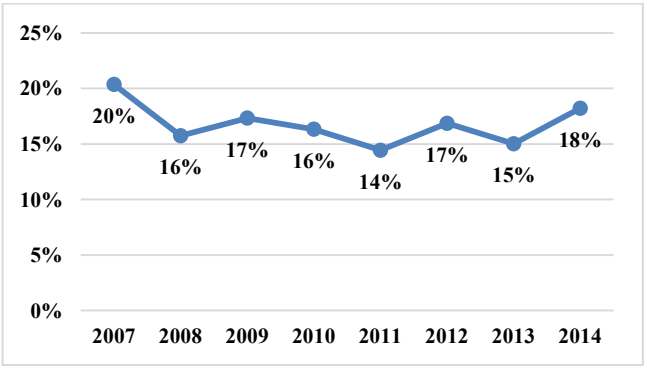

Figure 8. Percentage of students assigned to Component 1 in each class with an assignment score below component mean

Here we find that while the percentage of students in Component 1 who are below the component mean shows some fluctuation over time, there is no positive correlation with the growth in class size. Indeed, a correlation analysis of this percentage with class size reveals a *negative* correlation, but there are so few data points (eight) in measuring this correlation that we cannot conclude that an increase in class size leads to a greater proportion of "strong" students in the course, only that there is no evidence that the proportion of "weak" students grows as enrollments increase.

4. DISCUSSION AND CONCLUSIONS

The series of increasingly refined statistical analyses we conducted on student assignment data showed no evidence for an increase in the proportion of "weaker" students taking the course in the face of significant enrollment increases. Indeed, the statistics bore out a remarkable stability in the student population (and subpopulations) over time. This may lead one to wonder where a perception of an increase in "weak" students came from to begin with. The answer here is somewhat obvious in retrospect, but can now be understood from a rigorous prospective. While the *proportion* (and distribution) of "weak" students remains relatively stable as enrollments grow, that in turn implies that the number of "weak" students grows *on an absolute basis* with enrollments in a linear fashion.

Such students, who tend to struggle more with assignments, are more likely to seek out course staff for assistance, such as attending office hours, sending questions via email, posting to discussion groups, etc. Since such activity is often the most direct communication instructors receive from students, they perceive

greater numbers of students struggling as enrollments grow. Naturally, this can lead to the conclusion that more "weak" students are being drawn into the course. In fact, on an absolute scale, more such students are enrolling in the course, but that is simply to be expected from having greater numbers of students in the class overall. As shown in the analysis above, such anecdotal interactions with students should not be construed as indicating that the *population* of students taking the course has somehow shifted in distribution and a greater percentage of "weak" students are enrolling in the course. Thus, our analysis helps to resolve a perceptual paradox that we have seen arise among several faculty in the face of growing enrollments: it is simultaneously possible to see more students struggle in a class and also have there be no change with regard to the *proportion* of "weak" students taking the course. We would encourage instructors to be more cognizant of this fact when making public statements about students choosing to pursue computing courses in the face of growing enrollments so as not to alienate students who may not be as confident in their abilities, especially if they have no prior experience in computing or are part of a population susceptible to stereotype threat.

The main contributions of this work are three-fold. First, we highlight a growing concern among faculty regarding potential changes in the student body taking computing courses in the context of quickly-growing enrollments. Second, we provide a statistical analysis and general methodology, employing the use of mixture models, in a controlled setting to show that no evidence exists in our data to imply that increased enrollments have led to an increase in the proportion of "weak" students. Lastly, we provide an explanation for why faculty may perceive changes in the student population as enrollments grow, when in fact there is little change in the *distribution* of students' facility with the subject.

Importantly, our analysis, while showing results in the limited context our own CS1 course, provides a general methodology that we believe others can use to analyze data at their respective institutions to more rigorously understand the evolution of student performance in their courses. Indeed, our analysis is not meant to be the final word on student performance in computing courses in the face of enrollment increases. Far from it, our goal is to provide a methodology that others use as a starting point in analyzing their own student populations.

In future work, we seek to apply this sort of analysis to other courses at our own institution and elsewhere, as more data with controlled factors becomes available. For example, is the stability we see in student performance in the face of large enrollment growth a more universal phenomenon (at least in introductory CS courses)? We are also interested in seeing how other factors, such as more informed teaching or greater resources for help (e.g., discussion forums, course helpers, etc.) may impact the stability of student performance in computing courses over time.

We are particularly interested in performing such analyses on courses further downstream in our program to better understand the evolution of student populations not only over time, but also through course content. For example, we can try to identify what factors (e.g., prior experience, help-seeking behavior, intended major, etc.) might impact the subpopulation component that a student is associated with. By examining students longitudinally, we can seek to find factors that might be indicative of students moving between subpopulation components as they progress through a series of courses.

On a more theoretical note, this work has pushed us to consider extensions to the EM algorithm applied to multiple disjoint data sets in order to induce mixture models where some of the mixture components share parameters across all the data sets. For example, we could build a 2-Gaussian model where the mean and variance for one of the Gaussians is required to be same across all data sets examined whereas the second Gaussian component is allowed to vary for each data set separately. In the educational setting, this would allow us to see how the "lower" component varies given an "upper" component that is the same across classes.

5. ACKNOWLEDGMENTS

We thank the faculty in our department and at other institutions for discussing issues related to student performance in their computing courses in light of growing enrollments. Such discussions and the attendant views regarding potential changes in the population of students provided the motivation for this work. We also thank Michelle Friend for early discussions of this topic.

6. REFERENCES

[1] Cheeseman, P., *et al*, 1988. AutoClass: A Bayesian Classification System, *Proceedings of the Fifth International Conference on Machine Learning*, 54-64.

[2] Cheeseman, P. and Stutz, J. 1996. Bayesian Classification (AutoClass): Theory and Results. In *Advances in Knowledge Discovery and Data Minin*g, AAAI Press, 153-180.

[3] Cheryan, S., Plaut, V. C., Davies, P., and Steele, C. M. 2009. Ambient belonging: How stereotypical environments impact gender participation in computer science. *Journal of Personality and Social Psychology*, 97(6), 1045-1060.

[4] Dempster, A. P., Laird, N. M., Rubin, D. B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1-38.

[5] Everitt, B.S. and Hand, D.J. 1981. *Finite Mixture Distributions*. Chapman & Hall.

[6] Koller, D., Ng, A., Do, C., and Chen, Z. 2013. Retention and Intention in Massive Open Online Courses: In Depth. *EDUCAUSE Review Online*, June 3, 2013. URL=http://er.educause.edu/articles/2013/6/retention-and-intention-in-massive-open-online-courses-in-depth.

[7] Kumar, A.N. 2012. A Study of Stereotype Threat in Computer Science. In *Proceedings of ITiCSE '12*, 273-278.

[8] Quora.com. *Higher Education: Are there too many students going into Computer Science?* URL=http://www.quora.com/Higher-Education/Are-there-too-many-students-going-into-Computer-Science. Retrieved on August 18, 2015.

[9] Soper, T. 2014. *Analysis: The exploding demand for computer science education, and why America needs to keep up*. GeekWire.com, June 6, 2014. URL=http://www.geekwire.com/2014/analysis-examining-computer-science-education-explosion/.

[10] Spencer, S.J., Steele, C.M., and Quinn, D.M. 1999. Stereotype threat and women's math performance, *Journal of Experimental Social Psychology*, 35(1), 4-28.

[11] Taylor, C. 2014. *From The Ivy League To State Schools, Demand For Computer Science Is Booming*. TechCrunch.com, May 25, 2014.

[12] Zweben, S. and Bizot, B. 2015. 2014 Taulbee Survey. *Computing Research News.* Vol. 27, No. 5 (May 2015), 25.