Teaching Theory in the time of Data Science/Big Data

Anna C. Gilbert Atri Rudra

June 17, 2016

1 History and some caveats

The genesis of this post is a conversation between the two authors about a month back. One of them (Anna) was going to give a talk at NSF to talk about the theoretical foundations of Data Science and the other (Atri) was thinking about Computer Science (henceforth CS) curriculum because of the changes that the Computer Science and Engineering (henceforth CSE) depart at University at Buffalo is considering. Anna's talk at NSF, which included some of the data on theory courses at top 20 schools, generated a lot of interest in knowing more about the state of theory courses at various schools. This was followed by more data collection on our part and this post is meant to a *starting* point of discussion on how we teach theory courses, especially in the light of increased importance of data science.

2 Overview: Drivers of change

CS enrollments as well as the number of CS majors have increased exponentially in the last few years. In 2014, Ed Lazowska (University of Washington), Eric Roberts (Stanford University), and Jim Kurose (Univ. of Massachusetts-Amherst) exhibited the trend in increasing enrollment in CS courses (in addition to the increases in the number of CS majors). Their graphs (see Figure 1) show the trend in introductory CS course enrollments at six institutions in the years 2006–2014. See Lazowska's presentation for more detailed statistics and a discussion of the potential implications of these increases. These trends remain valid in 2016. See Figure 2 for the University at Buffalo, for example.

In the past ten years, class sizes have exploded and there have been extra burdens on the teaching staff. Indeed, Lazowska points out that over 10% of Princeton's majors are CS majors, while it is highly unlikely that 10% of Princeton's faculty will ever be CS faculty. At the same time, many institutions are re-evaluating and changing their theoretical computer science (henceforth TCS) course requirements and content. Whether these changing requirements are as a result of the added pressure on teaching staff or simply correlated with, we are shifting priorities in both the material covered and the way in which it is covered. (Anecdotal evidence suggests that proof-based homeworks are significantly harder to grade than a programming assignment for a class with several hundred students and so a traditional proof-based course might move towards more programming-oriented.) The traditional mathematical parts of the CS curriculum are shifting in emphasis (e.g., theory of computation). We do not argue that these shifts in emphasis are inherently good or bad, nor can we demonstrate that these changes are a direct result of the increase in enrollment. We seek merely to point out these changes as they have an impact on a large (and increasing) number of students.

The changes in the course content, the (amount of) emphasis on particular parts of the TCS curriculum, and the changes in the overall requirements (including mathematics and statistics) are all occurring exactly at the time when there is a big move towards "computational thinking" in many fields and a national emphasis on STEM education more broadly. These trends mean changes in the foundational backgrounds of both computer science majors (many of whom take jobs in industry, some of whom go on to graduate work) and other people working in fields that need solid computational foundations. With the increasing role of machine learning and statistics in modern data computations, it is more important than before that students get a deep understanding of the mathematical foundations of data science. Traditional TCS has been discrete math but "continuous" math, especially

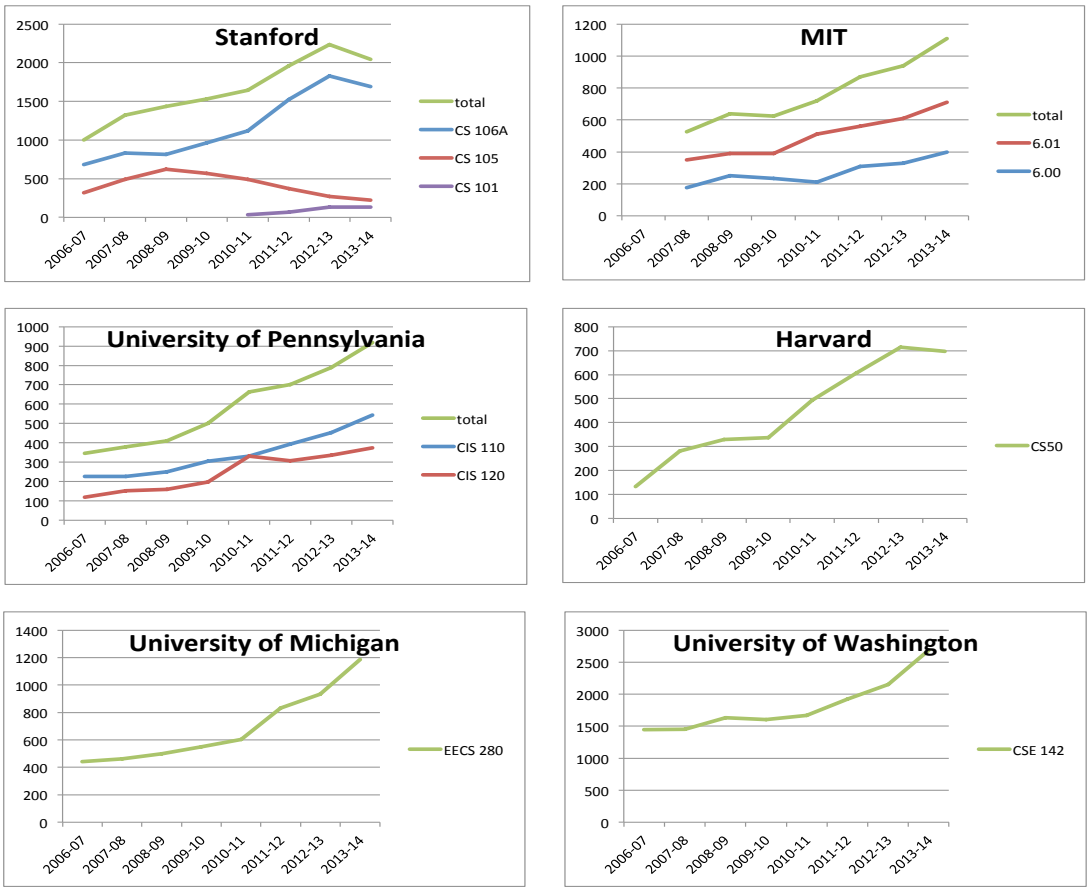

Figure 1: Enrollment trends in introductory CS sequences at six institutions (Stanford, MIT, University of Pennsylvania, Harvard, University of Michigan, and University of Washington) from 2006–2014.

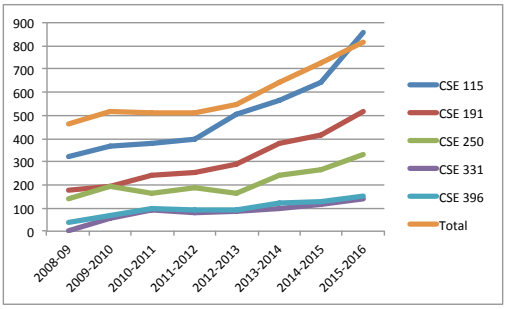

Figure 2: Enrollment trends at University at Buffalo from Fall 2008 to Spring 2016. In addition to total number of CSE majors, the chart also shows the enrollment in CSE 115 (the introduction to CSE course), CSE 191 (Discrete Math), CSE 250 (Data Structures), CSE 331 (Algorithms) and CSE 396 (Theory of Computation), all of which are required of all CS majors.

as related to statistics, probability, and linear algebra, is increasingly important (this is also true in the cutting edge of research in TCS).

3 Survey and data analysis

We considered the top 20 CS schools according to US News ranking for CS programs (all 24 of them). We recognize that it may be inappropriate to use the *graduate* program rankings to consider the *undergraduate* program requirements and that the graduate rankings cover all of the graduate program, not just the TCS component of a department's undergraduate or graduate program. We sent colleagues a short survey and collected data on these 24 schools. (This spreadsheet has all the data.) Below we will use CSE to talk about a Computer Science (and Engineering) department in each school.

3.1 What constitutes a theory course?

We counted the total number of theory courses that *all* CS majors have to take *within* the CSE department and we calculate what fraction of the total number of required courses theory courses make up. We categorized the theory courses under these bins:

1. Discrete Math
2. Data Structures (and Algorithms)
3. Algorithms (and Complexity)
4. Theory of Computation
5. Probability/Statistics
6. Number of theory electives that all CS majors have to take

We would like to clarify four things:

- Data Structures course is, in most schools, primarily a programming course with little (or no proofs) and almost always taught by non-core theory faculty. For the analysis, we still counted the data structures course as a theory course (after all data structures did come from theory so we should own it!) but we note that some (many?) of our colleagues do not consider this a theory course.
- Many schools have Probability/Statistics requirement outside of the CSE department. We will come back to this in Section 3.3.
- We have not specifically addressed parallel or distributed algorithms which are necessary in much of data science or data-enabled science.
- All the discussion below is based purely on TCS or math courses that **all** CS majors have to take. In particular, when we report numbers we do not consider cases where (even most) students take a specific TCS course but students have the option of not taking that course.

3.2 Current theory requirements

We begin with statistics on the total number of semesters of theory courses that are currently required of all CS majors.[1] The basic statistics are in Table 1.

The median number of semester long courses was three. All but one school requires a discrete math course, all but two schools require a data structures (and algorithms course) and all but 9 schools require an algorithms (and complexity) course. Eight schools require a theory of computation course. All schools have a significant programming component to their data structure course. As of now, only Cornell has non-trivial programming assignments in their required algorithms course in addition. We would like to remind the reader that we are only considering TCS courses required of all CS majors (e.g. CS 124/125 at Harvard has programming assignments but it not required of all CS majors).

[1] We use the well established equivalence of 3 quarters to 2 semesters.

Table 1: Total number of semesters of theory courses and as a fraction of all required CS courses

	Total TCS req'd courses	**Fraction of all req'd courses**
mean	2.94	0.24
median	3	0.24
maximum	5	0.5
minimum	2	0.125

3.3 Current Mathematics/Statistics requirements

For the current Mathematics/Statistics requirements, we considered only those courses outside of CSE that were (1) required of all CS majors and (2) were Probability/Statistics or Linear Algebra. We chose to focus on these two courses since they are most relevant to data science.

Probability/Statistics. Of those surveyed, 19 schools required a Probability/Statistics course, while five did not. Five had developed a specific required course within the CSE department (Stanford, Berkeley, UIUC, Univ. of Washington, and MIT), three had a choices among courses both inside and outside the CSE department, and 11 required a course from among those outside the CSE department. See Figure 3 for the breakdown of what percentage of departments provide such courses. Of the five institutions that did not require a Probability/Statistics course, two (Univ. of Wisconsin and Harvard) listed such a course among elective courses in Mathematics. Princeton, Yale, and Brown do not list such a course.

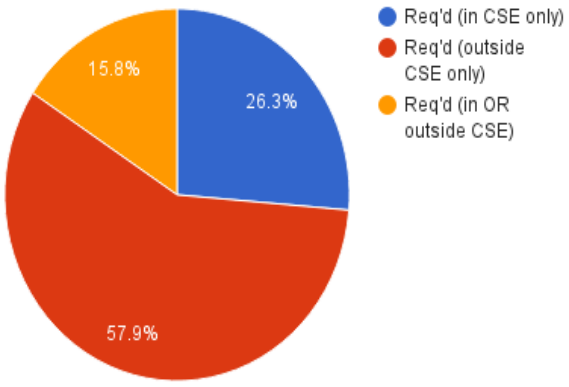

Figure 3: Probability/Statistics requirements, courses within CSE only, courses outside CSE only, and those departments with options both inside and outside CSE.

Linear Algebra. Sixteen surveyed schools require a Linear Algebra course and, of these 16, only Brown and Columbia have a linear algebra course in CSE that satisfies the linear algebra requirement (though both allow for non-CSE linear algebra courses). See Figure 4 for the percentage of surveyed CSE departments that require Linear Algebra and those that do not.

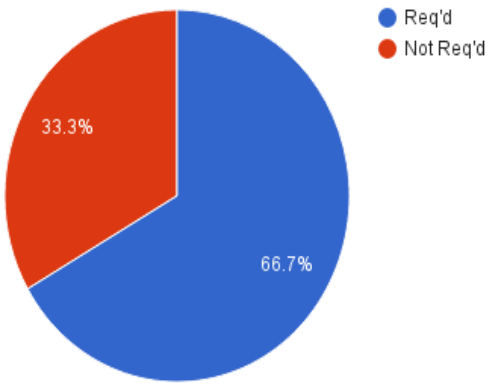

Figure 4: Linear Algebra course requirements.

3.4 Changes: past and future

As we observed in the beginning of this survey, there have been tremendous increases in CSE enrollments and increasing emphasis on computational thinking across many disciplines. These changes are simultaneous with what we perceived to be changes in the TCS curriculum. We dug a little deeper in our data and asked people more questions about the changes they have seen or are discussing at their institutions. A limited survey tells us that of the eight departments responding (as of June 10, 2016), seven have either kept the number of theory requirements the same and changed emphasis on content or have decreased the total number of requirements (three institutions decreased– all of these were accompanied by an overall decrease in number of required courses for CS majors), while only one has increased the number of requirements.

Four universities changed their Mathematics requirements in the last 10 years. These changes are primarily to require fewer semesters of 2nd year Calculus (e.g., some no longer require Ordinary Differential Equations) and, instead, require Linear Algebra and/or Probability/Statistics (whether inside the CSE department or not). Two institutions plan to make changes in the future; they are likely to require Linear Algebra.

4 Starting points for discussion

We suggest that now is the time to re-think some of the theory curriculum, to work with our colleagues in Mathematics and Statistics, to develop some mathematical foundations classes that are appropriate for all CS majors (and STEM majors, more broadly), so that all CS majors get exposure to such material (no later than junior year). We have outlined a series of starting points for this discussion.

- *Require students to choose one out of five different foundational courses offered by several different departments; e.g., CSE, Statistics, and Mathematics.*
- *Team teach courses that cover a combination of material across several departments.*
- *Develop small exploratory or project-based courses (with hands-on programming) that complement our theoretical foundations classes, similar to lab-based courses in the sciences. Or add programming assignment to TCS courses.* In addition to programming assignments in the data structures courses (which is universal),

perhaps the best place to add such assignments would be in the algorithms course. The idea behind these assignment would be to show how algorithms taught in the lectures can be applied to real life data. With good auto-graders having programming assignments will help in being able scale the grading to increasingly large classes. Anecdotally, students who are not that mathematically inclined (who form a majority of CS majors) would appreciate the material better if they code up the algorithms they see in the class. We believe proof based assignments are relevant and should remain in the algorithms courses but it might be better to complement them with some programming assignments.

- If your department is looking into reducing the total number of required CS courses, instead of dropping the theory of computing course, *examine carefully and discuss what pieces of the Theory of Computation curriculum to retain, which elements and modes of analysis enable a variety of applications in the rest of computer science.* See an earlier post on Dick and Ken's blog titled *Complexity as an Enabler* for a broader discussion.

Our goal is to educate the different students at our respective institutions as best as we can, by working with our colleagues at our home institutes and by having a dialogue with our theory colleagues across the country.

5 Sociological phenomena

When we sent out emails initially to friends in our social networks to gather or to confirm this data, we noted that we asked only three women total. We then mused if we could have increased that number by thinking a bit harder about which women were in our social network and whether or not the institutions we collected figures for had women theorists. We found that, upon reflection, we could have asked eight more women in our social networks, for a total of 11 women theorists, each at a different school, among the top 24 institutions. There are certainly more than 11 institutions with women theorists but either the women faculty are in areas we are not familiar with or they are women in our areas whom we do not yet know personally (e.g., new, junior faculty). In other words, a ten minute reflection yielded an almost four-fold increase in representatives from an under-represented group.

6 An appeal for help

We recognize that we surveyed only 24 institutions. This was done mostly to reduce work on our part since the first data was collecting by reading the relevant curricula webpages. Needless to say to gain a better picture of TCS and math requirements for CS degrees in schools in the US, more data will be helpful. We are hoping that readers of this blog at many more institutions can make valuable contributions to our data collection and discussion. Interested readers can contribute their institution's information to this survey by filling in a Google form. We will periodically update the master spreadhseet with information that we get from this Google form.

7 Acknowledgements

We would like to thank Dick and Ken for allowing us to do a guest post on their blog.

Thanks to the following for their input on related matters that greatly helped us in writing this article:

- At University at Michigan: Stephen DeBacker and Martin Strauss
- At University at Buffalo: Geoff Challen, Jesse Hartloff, Roger He, Andrew Hughes, Shi Li, Russ Miller, Ken Regan, Jinhui Xu and Jaric Zola.
- Petros Drineas (Purdue), Xaoming Huo (Georgia Tech) and Piotr Indyk (MIT).

Finally, we are grateful to the following folks for contributing data to the master spreadhseet:

Jim Aspnes, Paul Beame, Avrim Blum, Mark Braverman, Shuchi Chawla, Chandra Chekuri, Jeff Erickson, Bill Gasarch, Elena Grigorescu, Sudipto Guha, Anupam Gupta, Venkat Guruswami, Russell Impagliazzo, Piotr Indyk, Sampath Kannan, David Kempe, Bobby Kleinberg, Shachar Lovett, Seth Pettie, Lenny Pitt, Prasad Raghavendra, Tim Roughgarden, Madhu Sudan, Rocco Servedio, Chris Umans, Paul Valiant, David Zuckerman.