Teaching Statement
Austin R. Benson

Teaching philosophy and experience
I have had the opportunity to teach in a variety of capacities:
• teaching assistant for a large course (200+ students)
• teaching assistant for a small course (5 students)
• instructor for and creator of a new course on Python programming for scientific computing
• instructor for a short course on discrete mathematics for incoming PhD students
• mentor for undergraduate, master’s, and PhD students
These experiences have shaped my teaching philosophy around a few core concepts. First, I value opportunities for hands-on learning. For example, a course might teach an algorithm, and I would like students to implement an algorithm and run it to analyze a real-world dataset. I find that simply implementing an algorithm can help one learn how the method works. Second, I place an emphasis on team-based problem solving. Both academic research and industry problems are team-focused, so we should prepare students accordingly. Finally, I strive to continuously improve the learning experience. Along these lines, I developed a new Python programming course offering at Stanford and participated in a workshop for shaping the future curriculum of my department. For all of my teaching efforts, I was designated a Teaching Fellow by Stanford’s Institute for Computational and Mathematical Engineering in 2016.

I enjoy teaching and the most convincing evidence of this is simply the number of times I have volunteered to teach. In just my second quarter at Stanford, I volunteered to create and teach a new introductory short course on Python programming.\(^1\) In order to encourage hands-on and team-based learning, I structured each lecture with two parts: first, I led a discussion on some new topics about Python programming; second, the students worked through exercises in the classroom. The exercises allowed students to work together and develop skills immediately after learning new material. I also created coding-based homework that reinforced the class material and highlighted Python’s advantages for solving scientific problems. For homework, I encouraged students to work in teams provided that they write their own solutions.

I strive to continuously improve my teaching, and I taught the Python programming course the following quarter with several updates: I reformatted the lecture material, exercises, and homework based on student feedback; I improved student assessment with an automatic homework grading system; and I incorporated the course into a new Stanford initiative in online learning. Through the online learning initiative, I recorded several 5-minute video tutorials on specific topics in the course, which students could view outside of class. These additional efforts had a positive effect on my classroom, as evidenced by an increase in average evaluation scores for both the course content and my instruction:

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Response rate</th>
<th>Course content score</th>
<th>Instructor score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter 2012/2013</td>
<td>39 / 51</td>
<td>3.93 / 5.0</td>
<td>4.10 / 5.0</td>
</tr>
<tr>
<td>Spring 2013</td>
<td>38 / 40</td>
<td>4.21 / 5.0</td>
<td>4.35 / 5.0</td>
</tr>
</tbody>
</table>

\(^1\) http://stanford.edu/~arbenson/cme193.html

Table 1: Teaching evaluation scores from the two quarters that I taught a Python programming course. After making several improvements to the class following the first quarter, my scores increased.
Currently, I am a TA for one of the largest advanced computer science courses at Stanford: Social and Information Network Analysis (CS 224W). To encourage team-based learning, I give students opportunities to teach each other by working in small groups or by writing on the whiteboard in office hours. This collaborative experience has been a successful teaching strategy. Another one of my TA duties is mentoring student teams for their final projects. I enjoy seeing how students connect the ideas from class to a project idea matching their personal interests, and I have found this to be an effective alternative to exam-based assessment. I plan to incorporate similar group-based final projects into my own courses.

Mentoring and advising
My mentors and collaborators have had a tremendous impact on my personal development. Consequently, I place enormous value in becoming an effective mentor myself. At Stanford, I have mentored one master’s and one PhD student through research rotations in our group. The master’s student’s project led to a publication to appear at a premier data mining conference, and he has since joined Stanford Computer Science as a PhD student. I am actively mentoring the other student, who has already produced new theoretical results that will result in publication. As a teaching assistant for the “Projects in Applied and Computational Mathematics” course at Stanford, I also mentored an undergraduate student on a machine learning project for predicting loan defaults. This student has since joined Google to work on data mining and machine learning full time.

Through this mentoring, I have found that relationships based on collaboration (and not directives) is the right way to encourage growth. I plan to foster many new relationships in this manner when I become a professor. I have also found that the diversity of student skill sets provides a major challenge for mentoring in data science. Typically, my research projects in this domain have five components: (1) idea generation, (2) theory, (3) software development, (4) empirical analysis and experiments, and (5) presentation (writing and talks). Most students excel in only a subset of these areas, and I strive to develop their skills in all of them. For example, to help idea generation, I recommend a few research papers and then hold a discussion on new research directions. For theoretical problems, I develop some clean and concise questions for the student to work on and gain confidence.

Teaching interests
My research touches several areas of computer science, so I am comfortable teaching introductory courses in data mining, machine learning, network analysis, and algorithms. Through more advanced courses, I would like to teach about computational methods in network science and data mining that are related to my research interests. Specifically, I want to teach a course on computational methods for network analysis that would get students comfortable with network algorithms, software for graph analytics, and analysis of real-world network data. The wide availability of public network data from various disciplines provides a great opportunity for students to get hands-on experience and also perform analysis on datasets that interest them personally.

\[^{2}http://web.stanford.edu/class/cs224w/