Teaching statement

Vincent Conitzer

December 20, 2005

I believe that teaching constitutes an important part of a researcher’s career. In fact, I believe that making a clear-cut distinction between research and teaching is fallacious. Rather, there is a spectrum of expertise that any given person may have on a particular topic; and in order for research on this topic to have any significant societal value, it must be communicated to people everywhere on this spectrum—from co-authors and researchers in the area, to industrial practitioners and students. I greatly enjoy communicating at all of these different levels. I believe that teaching students is especially important and rewarding, because many of them will have opportunities to change the world using what they have learned, and some will eventually replace the people that taught them.

1 Experience

I have served as a teaching assistant for two courses at Carnegie Mellon University: Advanced Artificial Intelligence Concepts (a Ph.D. level course), and Machine Learning (an undergraduate level course). My responsibilities in these courses included teaching recitations and occasional lectures, creating and grading homeworks and exams, and helping students outside of class. I received an overall evaluation of excellent in both courses (the highest possible rating). In addition, I served as the (full-time, on-site) vertical mentor for the Negotiation task in the eBusiness Technology M.S. program on CMU’s West Campus, guiding the students through their projects. On occasion I have provided some support for the (Ph.D. level) course Foundations of Electronic Marketplaces (which has no official teaching assistant). As an undergraduate at Harvard, I served as a teaching assistant for Linear Algebra and Multivariable Calculus, and, to complete my financial aid package, spent a large amount of time serving as a paid tutor for students on a variety of topics (this included successfully completing challenging assignments such as teaching calculus to a congenitally blind student, and statistics to a student who had never learned basic algebra).

At CMU, I have given numerous guest lectures on various topics in game theory and machine learning, even in courses that I was not otherwise associated with. Moreover, I have been fortunate enough to be given a large number of opportunities to present my own research during graduate school (in department seminars as well as at conferences), and I have given numerous presentations in industry, all of which have further improved my presentation skills. As for advising students, I supervised Zhijian Lim (an undergraduate student) during a year-long senior research project in which he built an expressive negotiation server for donations to charities. I have also given occasional advice to many other undergraduate students on their research projects. Finally, I have held two leadership positions in sports organizations: I was vice president and instructor at Harvard Judo, and I am currently president of CMU’s School of Computer Science soccer club.

2 Future plans

I intend to put together a comprehensive course on issues in the intersection of computer science and economics, such as electronic marketplaces and computational game theory. Because computer systems are increasingly used by multiple self-interested parties, awareness of such issues has increased tremendously, not only in artificial intelligence but across the computer science community (for example, in areas such as networks, resource/task allocation, and scheduling). This has led to widespread demand for such courses. More generally, I would love
to teach courses on artificial intelligence, search/optimization, machine learning, reasoning under uncertainty, and related areas. If necessary, I can also teach basic courses on algorithms, data structures, formal systems, computability, and complexity.

3 Personal philosophy

I have always loved tutoring people one-on-one (and have been doing so consistently since high school, both informally and in an official capacity). When I first started giving lectures to large audiences, I found the experience somewhat uncomfortable: the empathetic skills I had developed while tutoring allowed me to see when some members of the audience had missed something, but the setting made it impossible to reestablish common ground with individual audience members (as one would in a one-on-one conversation). The first lesson that I learned from this was to perfect the presentation beforehand, anticipating as many possible confusions as possible, and revising the slides and delivery to prevent them (usually through clear examples). I still believe that this is the most important part of being a clear presenter—an ounce of prevention is worth a pound of cure. Nevertheless, more recently I have learned to become more adaptive to the audience again, even when its members are initially reluctant to ask questions. Needless to say, I love teaching in discussion format and engaging students one-on-one outside of class. For example, while I served as a teaching assistant for Machine Learning, I voluntarily doubled my office hours to meet the large demand for them from students (and continued helping students even outside of office hours).

I look forward to putting together my own course materials. I greatly enjoy constructing specific examples that illustrate the power of a general framework, or demonstrate the limitations of a particular result. (In fact, much of my research is based on such examples.) When these examples are put in the form of assignments to students, the students are able to build up an intuitive understanding of the material that is much more valuable than rote memorization of the abstract theory. I gained experience in this by putting together many of the assignments in the courses for which I was a teaching assistant. Also, after being a member of Harvard’s ACM Collegiate Programming Contest team (we placed tenth in the world), I created many new problems for future teams to practice on. Occasionally, I create puzzles for fun.

I especially look forward to advising students. Even more so than teaching a course, advising a student provides an opportunity to help a person shape his or her thinking over an extended period of time. One lesson that I learned from my advisor, Tuomas Sandholm, is that advising should be a very personalized process, in which the advisor pays close attention to the student’s interests and strengths. Tuomas easily could have imposed any of a multitude of research topics on me, but instead he allowed and helped me to find my own research style and problems. It is often said that students become “clones” of their advisors. However, by the time students reach graduate school (or even college), they have already formed their own set of beliefs and skills, and attempting to force them into a mold that does not fit them will inevitably result in failure. I do not know any productive researcher who can seriously be considered a clone of his or her advisor. That being said, I already have a plethora of unexplored research topics that students could use as a starting point.

More generally, I believe that it is important to encourage students’ creativity. Whereas artists and writers are commonly perceived as having creative professions (and correctly so), this is not the popular perception of computer scientists. At best, the perception is that we solve specific problems that present themselves, which perhaps sometimes requires a modicum of creativity. Computer scientists themselves know better and (should) think bigger, but I believe that the way we teach computer science does not always reflect that. This is certainly not to say that we should overhaul the curriculum, but I do believe in creating additional opportunities for students to see the open-ended nature of their field of study.

I believe that in the future, computer science will be advanced most significantly by integrating it into society in new and useful ways. This requires creative and more widely educated computer scientists. It also requires that people in other fields have a greater awareness of computer science. I am excited to contribute to both.