Interaction Beyond the Classroom

Eric Wong (wongeric@mit.edu)

Teaching for me is an engaging, rewarding, and valuable part of academia. Indeed, I believe that success in teaching goes hand-in-hand with success in research—both require a strong ability to convey and explain ideas. I am excited to inspire and guide the next generation of computer scientists and data scientists, and provide an education that they can rely on to do great things in the future.

Teaching experience

Throughout my academic experience, I have taught middle schoolers, undergraduates, masters students, and PhD students from a diverse set of backgrounds, of which I will highlight parts that have most influenced my teaching style and philosophy.

As an undergraduate, I joined the course staff for Algorithm Design & Analysis in 2014, one of the core computer science courses for CMU. This class was different from other computer science courses in that it had an oral evaluation component: rather than submitting written solutions, students would derive and explain their answer on a whiteboard in front of a teaching assistant. As a teaching assistant, I was proud to see my students becoming more confident and clear in their explanations as the semester went on, reminiscent of my own past experience in this course.

During my PhD, I was a teaching assistant for Advanced Introduction to Machine Learning at CMU, an advanced course intended for Machine Learning PhD students. However, due to the iterative and approximate nature of many machine learning algorithms, students could not always tell if the solution they produced was actually correct. I was the main lead in implementing an autograder with the goal of providing useful diagnostics that streamlined the student feedback loop. I continued to solicit feedback from the students and by the end of the course, had developed an informative, robust testing toolkit that provided usable student feedback on their machine learning algorithms.

My most ambitious teaching experience was as the head teaching assistant for Practical Data Science, a brand new course that I developed with my PhD advisor. Two aspects stand out here to me: the first was the challenges of scaling teaching to larger classes—over 300 enrolled students for 2 teaching assistants. I used the techniques and skills that I had previously built in autograding to provide diagnostic feedback for data science problem sets. This enabled hundreds of students to get real-time feedback on their assignments without being constrained by the limits of office hours. The second aspect was the creation of hybrid course notes that naturally incorporated runnable code examples alongside the course material. These notes served not only as an interactive learning resource, but also as a valuable code reference for homework assignments and course projects.

Teaching philosophy

I strive to make the learning process *interactive* rather than passive. While faculty and teaching assistants can encourage interaction during office hours and the classroom, a significant part of student learning occurs outside of these settings. I aim to take interactivity in learning beyond direct conversations with the course staff to the rest of the learning process. Throughout my teaching experience, I have pursued this through three main mechanisms.

First, students looking to review course material on their own typically only have access to static resources such as lecture slides, course notes, or recorded videos. This results in a passive learning experience outside of the classroom. I rely on *executable* course write-ups and assignments that integrate code and text together into a hybrid medium for conveying course materials. Importantly, the code is fully runnable alongside the text in an interactive environment such as Jupyter Notebook. This allows students to tweak and re-run code on the fly as they study, and supports the course content with visible, interactive examples.

Second, students seeking assistance for assignments can be delayed or discouraged due to limited availability or overcrowding of office hours. I have developed and employed *diagnostic autograders* that enhance and accelerate the feedback loop for students working on assignments. These are autograders that go beyond simple verification of whether an answer is correct or not: they provide useful feedback that points the student in the right direction based on the provided answer.

Third, clear communication of ideas is a vital skill for students that extends well beyond the academic setting. However, most academic assessments are conducted via graded assignments or exams, which check whether a student can solve a problem. Instead, I have used different types of evaluations that assess not only correctness, but also *whether the student can communicate their solution and reasoning* via an oral examination or written report. These experiences help students develop confidence in their public speaking and writing skills, and promotes diversity in evaluations for the learning process.

I have benefited from these principles in previous courses as a student, and employed all of these techniques in previous teaching assignments. However, I believe that teaching as a topic will continue to evolve and develop over time—what works best today may no longer be applicable ten years from now. I have participated in teaching workshops and seminars to stay informed of best practices and techniques, and remain committed to continue refining my abilities and knowledge as an educator.

Advising and mentoring

Throughout my PhD and my postdoctoral experiences, I have had the pleasure of advising and mentoring brilliant students across the the entire spectrum of undergraduate, masters, and PhD students. When directly supervising a student, I typically guided the student throughout the whole research pipeline from the initial research idea to a conference submission. Students that I have directly supervised have gone on to successful PhD programs; a former undergraduate is now a PhD student at the Machine Learning Department at CMU, and a former masters student is now a PhD student at the Department of Computer Science at Stanford. During my time at both CMU and MIT, I have also supported junior PhD students as both a mentor and a collaborator, providing guidance for research skills such as formulating research questions, writing papers, and giving presentations.

Classes

Based on my experience, I would be keen to teach general classes on data science and machine learning, at either the undergraduate or graduate level. I can also offer more specialized, graduate-level courses centered around the real world, covering topics such as privacy, security, and robustness, or a general machine learning course on deep learning. Beyond these, I have a few more courses that I would be interested in developing.

I would like to develop a *fundamentals of machine learning* course for those with a computer science or programming background. Machine learning itself is a highly interdisciplinary field with roots in statistics, mathematics, and optimization—topics that most computer science majors do not have a background in. However, from my past experience, I understand that obtaining such as wide background is both expensive and unnecessarily broad. Instead, I would like to design a course that focuses on introducing the core knowledge at the intersection of these fields, that which is needed to derive and analyze machine learning algorithms. Such a course would provide computer science or engineering students with a grounded understanding of how machine learning works, without requiring an additional degree or major.

Finally, I would like to create a *debugging data and machine learning* class. Models based on data do not always work as intended. However, the actual techniques for finding, diagnosing, and fixing problems in a data-driven pipeline are often overlooked, despite being an incredibly useful skillset for scientific and industrial research. The aim of this course is to introduce the main problems and corrective methods across the entire data science pipeline—the data, model, and predictions. The topics would touch upon collection, cleaning, and biases of data; inspection, interpretation, and interventions for models; and confidences, correlations, and post-hoc analysis of predictions.