Mathias Lécuyer – Teaching Statement

Given my experience working on the security and privacy challenges raised by machine learning ecosystems, I am excited to teach classes in the underlying topics, such as **security and privacy, distributed systems, operating systems, machine learning, or systems for data science**. I would also value the opportunity to design and teach interdisciplinary **courses at the crossroads between systems and causal inference, differential privacy, or reinforcement learning**. The following describes the philosophy I developed through my teaching and mentoring experience, and my plans to further improve it.

A defining feature of computer science is the ability, and often inclination, that people have to experiment and discover by themselves at all skill levels. I have grown to recognize how powerful this aspect can be for facilitating student learning, and how rewarding it is to watch students develop new interests and acquire the tools to pursue them. For these reasons I am excited about the opportunity, and responsibility, of teaching and mentoring.

Teaching

During my Ph.D., my approach to teaching shifted from directed content delivery – where I explain a concept in a logical way – to an active learning[1] process involving students. The basis is simple: I guide students in experimenting with concrete tools to pique their curiosity, provide context, and guide the more formal discussion of overarching concepts. What makes this approach effective is what comes next. I help students draw links between their experimentation and the formal knowledge: the goal is that they be able to identify what they have learned, why learning it was important, and how they have learned it so they proceed in their education with practical strategies for reproducing this process.

A good example of this shift in my approach to teaching comes from the workshops I taught for the Society of Women Engineers. The goal of these workshops is to promote engineering disciplines among high school female students, and I was introducing the architecture of the Internet and web applications. My first teaching experience, in Spring 2014, was disastrous. I delivered a lecture that logically proceeded through the different components with examples using command line tools, but students quickly disconnected from the relatively dry structure. When I taught the same workshop two years later, it was much more successful. I first polled attendees about their favorite website, which happened to be Tumblr. I showed them how to display the website's browser code and guided them in exploring how changes to this code would visually impact the web page but be lost on a reload. Tailoring the experiment to their interests helped me connect with them and provided a segue towards exposing the different components of a website's architecture. At the end of the workshop, we talked about how the experiment helped them understand concepts subsequently exposed, and how they could continue this type of exploration on their own. Some were already linking the concepts to some visual customization code they had written for their blog and describing what they would try next. Helping students understand the big picture and connect it to what they know is a key joy of teaching!

This strategy is also useful for helping students advance from understanding a specific example to drawing more generally applicable rules, an important mental exercise that requires practice. When I taught a lecture on data scraping at Microsoft Research's data science summer school, I first stated the learning objectives for the lecture and then delved right away into a hands-on exercise on parsing data from a website, using this example to describe the architecture of a web service. This got us to think about mobile applications, which typically access more structured data through an API. We then saw how to use a man-in-the-middle proxy to discover those access points and write code to query them. By the session's end, we talked about how an abstract, architecture-level understanding let us find a better solution to a practical problem. Incorporating active learning strategies in my teaching helps students connect theoretical knowledge to practical applications. I find this skill particularly relevant to increasingly prevalent data-driven systems.

During my teaching experience at Columbia, I have learned to adapt to a variety of student backgrounds and levels of preparation. Whether teaching high school students, or acting as a teaching assistant for computer science or journalism majors, I have had to adapt my language, examples, and assumptions about what students already know.

[1] Here, active learning refers to an instructional method, not the sub-field of machine learning.

I realize, however, that these settings have been privileged in that I was interacting with a small number of students, making it easier to tailor a lesson. Going forward, I want to pursue the more ambitious goal of achieving inclusivity when teaching bigger groups of students. How can I identify, and relate to, their expectations for the class? How do I best provide different entry points for students with different levels of preparation? How do I maintain an open channel for feedback throughout the course in order to adjust the content, activities, or pace to actual student learning?

Mentoring

I had the opportunity to mentor several junior Ph.D. students in my lab, as well as a master's student. When doing so, I try to communicate the long-term vision of the project in a compelling way and identify well-scoped parts of the project that match the student's strengths and interests. My goal is to allow them to make tangible progress in the short-run even as they are becoming familiar with the broader project and its open challenges. In the context of interdisciplinary research, this hands-on involvement helps students learn new tools outside their core field, such as in machine learning or causal inference. With the big picture in mind, they are then equipped to make connections between these tools and existing problems, or find original formulations to unresolved research questions. This approach has allowed junior students in my lab to make meaningful contributions to some of my projects, published in top conferences. One student also went on to successfully lead his own interdisciplinary project at the intersection of systems and causal inference.

I plan to use a similar approach to help my students discover their interests and refine their ability to formulate interesting research problems, which I believe to be key roles for an advisor. In particular, I want to combine a classic lab meeting structure, where students share weekly updates on their research or discuss papers within thematic sequences, with a more informal one designed to cultivate broad interests. In this informal meeting, we would discuss intellectually stimulating questions and observations that are not necessarily linked to ongoing projects, but came to their mind while reading books, attending seminars, or every day news and conversations. To fuel these discussions, I would encourage them to keep a "thoughts diary", where they document such questions or observations a couple times per week. Overall, I will strive to foster a collaborative and supportive environment where students can develop their curiosity and become accomplished researchers.