Teaching Statement

Prateek Sharma

Good teaching can have a far-reaching impact. I strive to be the type of educator whose classes I have been excited by, and who have motivated me to pursue research. My teaching philosophy and teaching strategies are guided by my experience as student and as a teacher. As a student, the most effective and memorable courses have been the ones that have excited me about the subject matter, where I never stopped thinking about the course material. Three semesters as a teaching assistant at UMass for both undergraduate and graduate level courses has allowed me to formulate a nascent teaching strategy to keep students engaged and excited both in and outside the classroom.

Teaching Philosophy. My class-room teaching strategy has two facets: 1) interactive, well-organized lectures, and 2) connecting theory to practice. In the undergrad operating systems course at UMass, I held a weekly discussion section (50 students) that involved recapping material taught in prior classes, and answering questions that students had. To stimulate discussion and participation, I started each discussion with small quizzes to recap concepts and gauge common misconceptions. In-class discussions can be extremely valuable for the entire class (including the instructors!). As a TA for undergrad algorithms, walking around the class while students discussed and solved the problems during the discussion sections provided a glimpse into common stumbling blocks that students faced. Watching concepts "click" and students improve in their skills during the semester, and sometimes even during a single class, is immensely gratifying. Interactive classrooms also allow more immediate feedback and tuning of future class-room lectures, and thus lead to flexibility in the curriculum.

Highly interactive classrooms may not scale to larger class sizes, which are a consequence of increasing interest and enrollments in computer science. As a student, I have found that well-organized lectures can also be highly effective in presenting new and complex material. I shall place a strong emphasis on teaching fundamental concepts in a clear and lucid way, and use practical examples and applications to reinforce their understanding. This is especially important in larger classrooms where scaling student participation is hard. I am also keen on trying techniques such as "flipped" classrooms that can provide interactive environments even in larger classrooms.

While classroom teaching is important, most learning happens outside of the classroom. Creating interesting and challenging home work assignments is a crucial component of my teaching. We are fortunate in computer science that we can design homework assignments where students can learn by doing. Projects where students design and implement software can be a great learning experience, and can be used to teach both theoretical concepts as well as essential practical software skills.

Systems Teaching. My background in systems research will help me not only in teaching systems courses, but will also help in teaching about systems thinking. One of the great things about computer science, and especially systems, is the intricate interconnection between different sub-areas. This allows presenting new concepts from different angles, so that students from different backgrounds can grasp the material. I have frequently guest lectured the undergrad and graduate operating systems classes, and have focused on connecting theory and practice. In particular, showing students the design of the Linux kernel and the manifestation and implementation of OS ideas, is illuminating for many students. Students have consistently appreciated the connection between concepts and their application in real operating systems (that they use on their laptops, phones, etc.), and expressed interest in exploring Linux internals.

My research on cloud computing finds broad application in course-work. Hands-on exercises in using public cloud platforms to run applications remotely can help students learn about the evolving computation and storage capabilities in cloud platforms, and also enhance practical skills. Furthermore, cloud platforms can also serve as an environment to motivate several fundamental concepts. For example, resource allocation and isolation problems such as bin-packing can be explained in the context of cloud providers running multiple virtual machines on their servers. As a TA for the graduate-level distributed operating systems course, I helped design such a cloud-based assignment. The shared nature of the cloud also provides an easy platform for motivating software security and privacy. Such cloud-based assignments can also simplify course logistics by offering easy to use, consistent environments for evaluating student projects. These assignments can be facilitated by the use of education credits offered by the major public cloud providers (Amazon, Google, and Microsoft).

As a systems researcher, teaching systems can be a highly rewarding experience. Many systems research projects often are easily applied in practice, which allows students to learn more about new directions in systems research, and even provide new research directions. More importantly, revisiting fundamental CS concepts and techniques is often a very rewarding experience, especially in systems where old ideas are often forgotten and then rediscovered and reimplemented.

Teaching Interests. With my systems background, I currently feel most comfortable with teaching undergraduate and graduate courses on operating systems, networking, and distributed systems. Apart from these classic systems courses, I would also be interested in teaching systems for data-science, and other courses at the intersection of systems, data-science, and cloud-computing. I am also interested in teaching advanced, seminar-type courses that involve reading and discussing papers, in areas closely aligned with my research. These seminar courses can give students a glimpse into cutting edge research, and also serve as a source of new research ideas and research projects that can often lead to publications.

Diversity and Inclusion. There are ever growing concerns about the lack of diversity in computer science, all the way from high-school to industrial appointments. Academia is a crucial stage in this pipeline, as it can influence both the representation of different groups, as well as influence the next generation of industry leaders about the importance of diversity and inclusiveness. The lack of diversity can be addressed by carefully designing courses that have a broad appeal. Integrating humanities and other sciences, through the use of "CS+X" courses, is a powerful tool in showcasing the magic and utility of computer science. Through my teaching, I will strive to present many inter-disciplinary examples of CS theory and practice, by emphasizing that CS is not just engineering and "hacking", and that many ideas in CS have deep roots and applications in mathematics, economics, and other disciplines. Thus I believe that through organic curriculum redesign and careful teaching techniques, we can broaden the appeal of CS and bring in different perspectives, backgrounds, and skillsets.

Advising. Advising and mentoring students is one of the great perks of academia, and I am eager to advise and work with both undergraduate and graduate students. Towards the end of my masters thesis, I was involved with helping newer and less experienced masters students with the research process, by helping them find new research directions and projects, as well help with the technical aspects of hardware virtualization. During my PhD, I have also mentored junior students in other aspects of the research process, such as formulating research problems, and guiding them successfully through the publication process.

As an adviser, I will help students discover and refine research problems that they are most

interested in solving. One of the crucial aspects of systems research that I've learnt from my adviser, Prashant Shenoy, is the importance of directing engineering effort to maximize research impact. In systems, it is easy to lose sight of research goals when mired in implementation difficulties, and I shall advise my students to keep the bigger research picture in mind. I shall work closely, especially with less experienced students, and offer guidance, motivation, and support to solve problems, as well as give detailed feedback on writing and presentations.

Finally, students often are immune from dogma and ossification of ideas, and can provide fresh ideas and perspectives which are crucial for research. Working with students from diverse academic backgrounds is a perk unique to academia, and can provide a life-time of learning and discovery—I look forward to it very much.