Slav Petrov - Teaching Statement

Teaching offers a tremendous opportunity to educate the future generations of computer science engineers and researchers. My goal as a teacher is to provide students with the problem solving skills that will be necessary outside of the classroom. Providing intuitions and emphasizing the essentials is often more important than teaching detailed knowledge about a specific domain. For me, teaching is more than just lecturing classes. Teaching, as I see it, extends beyond the classroom setting and encompasses additional guidance and advising.

I would be eager to teach a variety of applied classes, be it at the undergraduate or graduate level. My primary expertise is in artificial intelligence, natural language processing, and machine learning. I would also be very interested in designing and teaching new classes on web search, data mining, and video analysis (potentially co-taught with a vision colleague). Finally, I enjoy teaching introductory classes, and I have been a teaching assistant for three different computer organization or machine structures classes.

While at UC Berkeley, I have been fortunate to assist two young and passionate teachers: Dan Klein and Dan Garcia. I learned from them that there is no one size fits all solution to teaching, as students learn in individual ways and different topics require varying teaching styles. Some students learn mostly from lectures, others prefer learning in small groups, and yet others learn best by studying on their own and asking questions in a one-on-one setting. Teaching a class therefore involves much more than just lecturing. Lectures, sections and office hours need to be coordinated, so that every student can be addressed in his or her preferred way. Written homework assignments and programming projects need to be integrated into the curriculum to reinforce and teach the core material, rather than for assessment.

The undergraduate curriculum puts a lot of pressure on the students, and as a professor, one needs to put in effort in order to get their attention. Lectures should be engaging, well prepared, and should ideally involve demonstrations of important algorithms and concepts in a practical setting. Visualizing the material in intuitive ways is crucial in my opinion. I firmly believe that the right figure or demonstration (especially if interactive), provides a much clearer intuition and illuminates the main idea better than any number of words or lines of pseudocode code. I recently assisted my advisor in his undergraduate artificial intelligence class, where we heavily utilized Pacman for demonstrations. Search algorithms (depth first, breadth first, A*, etc.) are important concepts, but are usually fairly boring for the students. It was fascinating to see the students' excitement when Pacman was used for visualizing the different search strategies. The concept and importance of search functions, and the utility of heuristic functions for informed search algorithms became immediately clear to the students. In addition to lectures, sections should be used to give the students a more interactive venue for asking questions and the opportunity to further familiarize themselves with the material.

When teaching, I intend to adapt the presentation method to the content. Theoretical concepts are best derived by hand on the black board (or, in large classes, on a tablet PC connected to a projector). Algorithms are best explained by showing demonstrations of the algorithm in action, and slides are good for providing an overview. Homework assignments should be tailored to the material as well. I plan to use written assignments to ensure that students understand the theoretical foundations, and I would like to utilize programming projects extensively in applied fields. Games or practical applications are very engaging and useful for teaching students to build practical systems and lead to a better learning experience for the students. I plan to provide the students with a code framework, allowing them to focus on implementing the core functions. This approach can put a heavy workload on the staff, but it ultimately helps the students truly understand the material. Because minor implementational issues can make programming assignments

frustrating, it is important to give students the opportunity to resolve them quickly. A newsgroup (regularly monitored by the professor and teaching assistants), as well as regularly held office hours are very important in general, but especially in this context. One could call this a customer service approach to teaching.

Office hours are another great way of helping students, and also help the instructor identify potential problem areas. In large classes, it is often difficult to get a feeling of whether the students are struggling. Homework assignments can serve as a feedback mechanism, but have a long delay, making it difficult to clarify misunderstandings. Office hours in contrast provide immediate feedback, and one can literally watch the students learn and comprehend the material. I particularly enjoy office hours because one can adapt the way one explains the material and tailor the explanation to the student, something that I learned through numerous years of tutoring during high school and college.

A personal connection and individual approach is also required when advising students. As an advisor, I plan to have an open door policy and be always available for my students in person or via email. I would describe my advising style as fairly "hands-on," and I would like to get involved as much as the student lets me, but also give the student enough freedom to explore ideas on his or her own. As a graduate student, I enjoyed the weekly meetings with my advisor, but also the brainstorming sessions that we had as a group in our office. Many of my research projects sprang out of such group discussions, and were then refined in weekly meetings with my advisor. I would like to recreate this environment for my own research group.

But I see graduate student advising as more than just giving guidance on research. Helping pick good problems to work on, and constantly reminding the student to be rigorous and to strive for excellence are important aspects of being a good advisor, but there are many others that are sometimes neglected. Doing good research is only a first step in having a successful research career. Writing in a clear and concise way and preparing a clean and illustrative presentation, which highlights the main contributions, are equally important for the final perception of a line of work in the research community. I intend to closely collaborate with my students and be strongly involved in the writing of publications, as well as in the preparation and rehearsal of conference talks. It is important to give students the chance to present their work at conferences, but a good advisor goes further and introduces students to other researchers and is a constant advocate of the work. I am excited about the prospect of seeing young graduate students learn and develop from students to colleagues.