

# Teaching Statement

Lev Reyzin

My sophomore year as an undergraduate at Princeton, I became a “lab teaching assistant”; my job was to guide students through assignments for introductory courses. I often stayed in the computer lab long past the scheduled hours, helping students understand recursion and pointers, so I realized quickly that I enjoyed teaching. Since then, I’ve taken many opportunities to teach. As a graduate student at Yale, I was a teaching assistant for 5 semesters (3 more than required): two semesters for discrete math, two semesters for algorithms, and one for a machine learning class intended for both undergraduate and graduate students, the content of which I helped shape. During that time, I had a chance to prepare and give lectures, run sections, host office hours, make and grade homework and exams, and supervise undergraduate projects. Afterwards, at Yahoo! Research, I helped shape a summer intern’s research project. At Georgia Tech, I’ve taken opportunities to guest-lecture for undergraduates and to work with promising graduate students, and in the coming semester, I will co-lead a special-topics graduate seminar.

I would be interested in teaching a variety of undergraduate courses including (but not limited to) *machine learning, learning theory, theory of computation, algorithms, discrete math, artificial intelligence, data structures, and various introductory computer science, programming, data mining, and statistics courses*. I am also enthusiastic to teach courses and seminars on the graduate level on *machine learning, computational and statistical learning theory*, and other areas related to my research, e.g. on the *theory of networks, computational advertising, or interactive learning*. I am excited about the opportunity to mentor graduate students and supervise undergraduate research. Furthermore, the year I spent at Yahoo! Research gave me an understanding of what problems are important in real-world applications, and I believe my students would gain from this perspective.

In teaching, I’ve actively sought opportunities to interact with my students. In my first semester of teaching discrete math, I took the initiative to offer an extra weekly problem-solving session, which half of the class regularly attended. In algorithms and machine learning classes, I asked the professors to allow me to give lectures and observe my teaching. And I helped design the direction of the machine learning course, and to update it to reflect current topics. These experiences made me a better teacher and allowed me to reflect on my own teaching philosophy.

When I began teaching, I was at first disappointed by how many students were more interested in getting a good grade than in learning the material; this is an unfortunate byproduct of grades playing a large role in their future jobs and graduate schools. But as a teacher, I always stressed that understanding the material is more important than simply getting the right answer. And I have tried to have my teaching and grading reflect this view. I would often give partial credit to students who admitted to not knowing how to solve their problem and showed me where they were stuck. I encouraged students to come to office hours and discuss their difficulties with homework, and would happily help them with their assignments if they could show me they put considerable thought into the material – learning was always the goal. The students, having caught on, formed long lines outside my office eager to get some hints, and I think they learned the material better

in the process of discussing their thoughts and ideas with me.

As I became comfortable teaching, I had many rewarding experiences working with my students. It is very exciting to help someone understand something new. And it is especially gratifying to relay your enthusiasm to them. I think there are certain ways of teaching that can make students more engaged and involved. One is to try to make assignments interesting to different types of students; in the machine learning class, we designed assignments with both programming and theory components, and that worked very well. Another way is to expose students to open problems early on, to show them that computer science, like other sciences, is not cut and dried – that we don't have the answers to many questions that we ask. When I was a first-year graduate student taking my advisor's class, at one point she gave us an open problem for homework, without telling us it was open. The problem kept me up for a couple nights and got me hooked on learning theory. I'm sure that many students can similarly be snared into engaging deeply with the material they are learning and into exploring the wonderful world of research.

As a new faculty member, I plan to continue to experiment with different teaching styles and techniques, with the goal of getting my students excited about learning. I view teaching, interacting with, and learning from students to be great benefits of becoming a professor. And I believe that my experience has taught me valuable lessons and helped me become an effective and enthusiastic teacher.