Teaching Statement

Pavel P. Kuksa
Machine Learning Department
NEC Laboratories America
pavel@pkuksa.org

November 20, 2013

I strongly believe that computer science as a discipline is as fundamental as physics and mathematics, given computational models and methods are an integral part of the modern science, technology, and society.

My objectives as an educator in computer science are: (1) to teach students theoretical foundations and underlying principles, sound problem-solving techniques and measurements of their efficacy along with real-world applications; (2) to expose students to hands-on experience with a full cycle of the development and application of computational models and methods to real-world problems; and (3) to teach students how to communicate their results and ideas clearly and to a variety of audiences.

Teaching experience. I have been fortunate to have a number of opportunities as an educator in my career.

During my Masters program I served as a lecturer for the continuing education course, *Systems Modeling*, offered by the Department of Computer and Information Sciences for professional workers as part of the career training curriculum.

In the first two years of my PhD program I served as a teaching assistant for the largest introductory computer science course for business majors (with over 500 students). I was teaching recitation classes for two student sections, and offered lab support and office hours for students. Lab support hours were designed to help students from any section of the course with their programming assignments, and were very interactive and dynamic in nature. Teaching computer science basics for non-major students in this course was an invaluable experience, as I gained new insight into conveying ideas clearly and simply while carrying on discussions with students during recitations and lab hours.

My third, fourth and fifth years in the PhD program afforded me the opportunity to serve as a teaching assistant and a lecturer on selected topics in advanced undergraduate and graduate classes and seminars. In the "Machine Learning", "Pattern Recognition", and "Introduction to Bioinformatics" graduate classes I designed homework, midterm, and final exam questions, offered office hours and by-appointment meetings with the students, as well as served as a lecturer for select topics.

Below I provide a list of computer science classes for which I have had an opportunity to be a teaching assistant or lecturer.

- Undergraduate courses:
  - Introductory computer science classes for non-majors (CS107 "Introduction to Computer Applications"),
  - Fundamentals of discrete mathematics (CS206 "Discrete Structures")
  - Principles of Artificial Intelligence (CS440)
- Graduate courses:
  - Artificial Intelligence (CS530 "Principles of AI")

- Machine Learning (CS536)
- Pattern Recognition (CS535)
- "Systems Modeling", a class for post-graduate professionals
- "Introduction to Bioinformatics", a specialized graduate seminar on application of hidden Markov models in Bioinformatics (CS674).

Teaching philosophy. Teaching computer science classes is an exciting opportunity for both the teacher and the students to explore and unearth fundamental principles underlying computational models and methods and information processing, as well as instill sound problem-solving skills and techniques.

I believe the objective of a computer science class is fully accomplished when the students are both well-versed in all of the fundamentals of the subject areas as well as willing and able to question the status quo in the discipline.

My approach to the assessment of students is manifold. First, students should be expected to master the body of knowledge and demonstrate it through exams, in-class quizzes. Second, students should reflect on the material at a greater length in homework, mini projects, project papers, etc. Third, I would aim to provide office hours, one-on-one conversations, and review sessions to help students master the the subject matter and meet the course expectations.

I also believe in the importance of dynamic and interactive class presentations. I have practiced small in-class active learning exercises in many of my recitations and lectures. For instance, in the algorithms and discrete mathematics classes, I presented topics as problems to solve, first asking students to provide *any* solution to the problem at hand and then interactively worked with them and build up to a correct or optimal solution *or* discuss how the solution can be made better (more efficient, less resources, generalize to other cases, etc). This, in my experience, has made students not only more willing to ask questions and be more involved in class, but also instigated critical thinking in order to approach, analyze and solve problems.

I also believe that what students learn in a particular class should be connected to other subjects and classes. I believe this would further encourage student learning and enforce a greater sense of material relevancy. For example, while teaching compilers and parsers, one can introduce natural language sentence parsing problem from AI.

I also believe in the necessity of different modes of engaging students in learning and exploring subject material. Besides reading exercises and standard homework, I also practiced giving puzzle-type problems, advanced research and open problems that students can think about and fully or partially solve before the next class or recitation.

I also believe that in any computer science subject (including theoretical) it is important to have a class where pen-and-paper homework can be coupled with implementation-based small projects throughout the semester to cement understanding of the subject. These mini projects ideally would centralize around main themes of the course and build on each other. I think breaking down a standard (semester-long) course project in smaller pieces allows students to stay focused throughout the semester and inspires them to complete parts of a larger project as well as the entire project on time.

Course qualifications. I am qualified to teach basic computer science classes (introduction to algorithms, data structures, programming, foundations of computer science) as well as classes in areas of my particular strength including machine learning, bioinformatics, pattern recognition, and natural language processing. As a professor, I would also like to offer research seminars on particular topics of interest which may include recent trends in large-scale learning, machine learning in the biomedical domain, and question-answering systems.