Teaching Statement

Burr Settles (bsettles@cs.cmu.edu)

Computer science and its kin are among the most relevant disciplines of the twenty-first century. As digital technology is more tightly woven into the fabric of society (for example, via the Web and mobile devices), real-world problems rely more and more on computational solutions. Teaching offers a tremendous opportunity to influence the future generations of researchers and engineers — as well as computer-literate artists, entrepreneurs, physicians, and policy-makers — who will address these problems.

Summary of Experience

I have had the opportunity to teach in several different capacities: giving guest lectures in my areas of expertise during my postdoc at Carnegie Mellon, as an instructor and teaching assistant during graduate school at the University of Wisconsin–Madison, and as an informal mentor at both institutions. As an instructor at UW–Madison, I developed curricula, lectures, and assignments for two very different courses. *Artificial Intelligence* (CS 540, summer 2003) is the introductory course in AI for graduate students and advanced undergraduates (enrollment: 30, student evaluation: 4.3/5.0). This course covers intelligent agents, game playing, logic, planning, and introductory machine learning. *Computational Biology & Biostatistics* (summer 2008) is an intimate, fast-paced seminar course for participants in an undergraduate summer research program (enrollment: 5). The goal of the program is to attract more women and minority students to research in computational biology and biostatistics, and the course covers basic molecular biology, dynamic programming for sequence alignment, and statistical models for gene expression and biomedical text analysis. I have also served as a TA for several computer science courses at UW–Madison, including *Programming Languages & Compilers* (CS 536) and *Machine Organization & Architecture* (CS 354).

I would feel confident teaching courses on artificial intelligence, machine learning, and their applications at both the graduate and undergraduate level (particularly natural language processing and bioinformatics). I would also be happy to teach core introductory courses such as data structures and algorithms, where I would work hard to win converts to our field. Given the opportunity to teach a graduate seminar related to my interests, I would likely focus on emerging topics that go beyond supervised machine learning, such as active, semi-supervised, or large-scale interactive learning for language and social data in particular. I also embrace computer science's interdisciplinary nature, and would be open to designing joint courses with other departments, such as biology, psychology, sociology, art, and music.

Teaching Philosophy

My goal as a teacher is threefold. First, to instill in students a sense of excitement for today's scientific and information challenges, and the role that computational methods play in solving these problems. Second, to engage students in meaningful ways, ensuring that they have a solid understanding of the concepts involved. And third, to train and encourage students to think scientifically and independently. I discuss each of these in turn below.

In order to instill a sense of appreciation for the relevance of computational methods, I use compelling real-world illustrations. For example, spam filters and sentiment analysis tools both demonstrate the utility of automated text classification systems. Such examples motivate the task at hand and provide context for the approaches we cover in the course. Once the problem is introduced, it is easier to give a more formal definition of the task, point out why it is difficult, and present algorithmic solutions.

My second goal in teaching is to engage students with course material at a significant level. Because different people learn in different ways, this sometimes requires several strategies. During lectures, I make a practice of working through examples of each major algorithm. Depending on the situation, I sometimes do this by myself on the board, enlist one or more students to walk through the necessary steps, or break the class up into small workgroups to help each other. I also design assignments that contain both written and programming components. Written problems help establish a conceptual understanding of the content, while the programming tasks help students develop a working knowledge of the algorithms in practice. For example, while teaching my AI course, I had students implement a game-playing agent for a computer version of the board game Mancala[1]. Each student's solution was pitted against the others in a class tournament, the results of which determined part of their grade (and earned the winners prizes donated by local businesses). Projects like this motivate students to write software that is efficient and effective, not merely technically correct. I also embrace technology as a teaching tool both in and out of the classroom. For example, I have used the course email list to issue out-of-class discussion assignments, such as "translate your favorite quote into first-order logic," or "choose a real-world problem, and frame it as a supervised machine learning task." Students who are shy and reserved in class are often much more forthcoming in this setting. I am interested in using blogs, wikis, and other online forums to facilitate peer-to-peer learning.

My third goal as a teacher is to encourage students to think scientifically and independently. One way I have done this is by discussing "open" research problems in class, and prompting discussions of how we might attack them. This gives students the opportunity to formulate and articulate their ideas in the company of their peers. I also try to include recent scientific findings in daily course lectures. For example, while covering *feature spaces* during a machine learning segment of my AI course, I spent part of the lecture discussing an article on feature *selection* for natural language tasks that was published only a few months earlier [2], as well as a pre-print of a conference paper on feature *induction* that was to appear within a few weeks [4]. To encourage students to think independently, I believe in letting them conduct a short research project of their own choosing in graduate and advanced undergraduate courses. This gives students the opportunity to apply the knowledge they have acquired in class to a topic that specifically interests them.

Mentoring and Outreach

In addition to teaching and advising students in my own research group, I am interested in mentoring students on an individual basis, e.g., for independent studies and short-term projects. As a graduate student, I had the opportunity to mentor an undergraduate from Stanford University for a summer, which resulted in a publication [1]. As a postdoc at Carnegie Mellon, I have been able to advise students in our research group, and supervised both a PhD student's data analysis requirement, which resulted in a publication [3], and an undergraduate independent study. I would also be interested in participating in outreach programs aimed at attracting new students to research in computer science.

References

[1] T. Brow, B. Settles, and M. Craven. Classifying biomedical articles by making localized decisions. In *Proceedings of the Text Retrieval Conference (TREC)*, 2006.

[2] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[3] E. Law, B. Settles, and T.M. Mitchell. Learning to tag from open vocabulary labels. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 211–226. Springer, 2010.

[4] A. McCallum. Efficiently inducing features of conditional random fields. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 403–410. Morgan Kaufmann, 2003.

[1] This assignment has since been adopted by other tenured faculty in their own AI courses.