Teaching Statement

Katherine A. Keith
University of Massachusetts Amherst

1 Teaching philosophy

Our society has entered the information age in which most private and public organizations have large quantities of data that they need experts to work with and analyze. Thus, I believe it crucial to train the future work force in computing and data literacy, and provide students with the opportunity to take crucial electives such as *data science*, *machine learning*, and *natural language processing*. These courses are in high demand from employers, can cross disciplinary boundaries, and encourage broader participation in computer science. To the last point, Lewis et al. found that students often felt discouraged from majoring in computer science if they felt they did not fit conventional computer science stereotypes [6]. My mission as a professor is to teach technically rigorous computer science courses while expanding the domains in which students can see themselves applying these technical skills.

My personal teaching philosophy is that success in computer science is derived from a combination of carefully designed learning environments and individual hard work. I approach teaching with a *growth mindset* in which abilities are not fixed but can be developed over time [2]. I see effective, inclusive teaching as one of the most important tools for recruiting and retaining underrepresented students in computer science (see diversity statement), and as such I aim to adapt my teaching to students' needs through pedagogical methods such as active learning to engage students in class, differentiated instruction for students who arrive with less experience, and actively decreasing bias in the classroom (§3). On a personal note, I enjoyed the high-touch environment of a small, primarily undergraduate institution when I taught at Mount Holyoke College during the Spring of 2020, and I look forward to working in an environment in which I can get to know my students personally and adapt my teaching and mentoring to their individual needs.

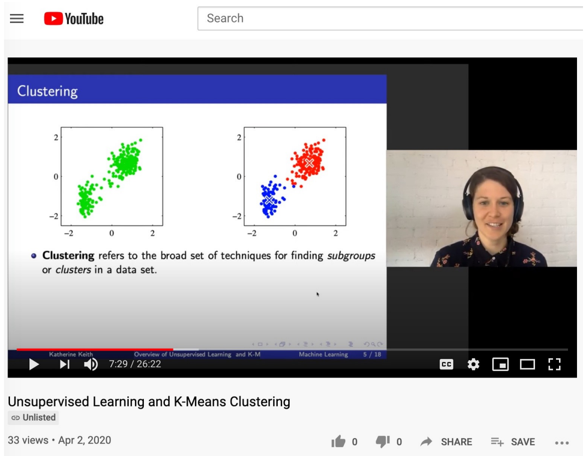

Figure 1: A video lecture I uploaded to support asynchronous learning while teaching *CS335: Machine Learning* at Mount Holyoke College during the COVID-19 pandemic. `https://www.youtube.com/watch?v=6sY-hPGRo7c`

2 Past teaching experience

During the spring semester of 2020, I was the sole instructor for a four-credit senior elective, **CS335: Machine Learning**, at Mount Holyoke College. The course built the mathematical foundations of machine learning (ML) with the *learning goals* of having students be able to derive analytical solutions for mathematical fundamentals of ML, implement supervised and unsupervised learning algorithms, evaluate when an algorithm is overfitting, and identify relationships between regularization, training size, training accuracy, and test accuracy. Students were *assessed* on these learning goals via a midterm in-class exam, a final project, and four large homework assignments which were a mix of mathematics and coding in Python. For instance, for their fourth homework, I provided students with a Jupyter Notebook that contained instructions and some starter code. Students were required to write code to implement multi-class classification for hand-written digits and run experiments to find the regularization penalty that provided the best training and test accuracies. A major learning concept required students to write *vectorized* code that used matrix multiplication of *NumPy*, a scientific computing package, in place of for-loops. My favorite parts of teaching a small class of 19 students was being able to engage each student directly in class and having students actively make use of office hours.

During the fall semester of 2018, I taught a one-credit freshman computer science seminar, **Ethical Issues Surrounding Artificial Intelligence Systems and Big Data**, a curriculum I co-designed with two of my labmates. Our *learning goals* aimed for students to answer the following questions: Do computers make decisions in a way that is more fair and less biased than humans? What are the political, legal, social, economic and technological forces that govern the digital world? What role does the government currently play in directing AI technology in the United States, and what role could it play? We assessed students via in-class discussions and weekly writing reflections submitted to Moodle after every reading. In the future, I would be interested in expanding this seminar (see §4).

I was also a teaching assistant for **CS685: Advanced Natural Language Processing** at the University of Massachusetts Amherst during the spring semester of 2018. This was a PhD-level class that covered advanced topics in natural language processing such as probabilistic models of language, linguistic representations for syntax and semantics, neural network models for language, and selected topics in discourse and text mining. As a teaching assistant, I helped design homeworks, held weekly office hours to provide technical assistance to students, graded students' literature reviews and projects, and gave a guest lecture at the end of the semester. Through this experience, I was able to gain skills in challenging students with difficult technical material and the ability to manage class with a large number of students.

3 Pedagogical methods

Active learning. In my teaching, I encourage engaged and broad participation of students via *active learning*. Specifically, I use *cold-calls* and *think-pair-share* exercises anywhere from 3-10 times during a single hour lecture. For *cold calls*, I pose a question, typically a math or coding problem, and call on a student by name to answer. To ensure I am not biased towards picking some students more than others, I choose from a stack of notecards with students' names and randomly shuffle the stack at the beginning of every class. The cold-calls are meant to keep students engaged during lectures in an encouraging way and so I verbally emphasize at the beginning of every class that students can pass if they are unable or unwilling to complete the problem. I also use *think-pair-share* exercises in my classes in which students first work through a problem on their own, share with a partner sitting next to them, and then I select a pair at random from the class to verbally share their answer. This technique encourages participation from all students and allows them to pause and reflect on their own understanding of the material during a dense, technical lecture.

Project-based learning. *Self-transcendent motivation*, the desire to help other people or change the world in some positive way, has been shown to help motivate students [4]. In my teaching, I have students work on projects throughout the semester in order to invoke *purpose* when engaging with technical material. For example, when I taught *CS335: Machine Learning*, students' projects focused on socially impactful machine learning applications such as health, education, climate change etc. One project group, comprising of three female international students who spoke English as a second language, used neural networks to predict the language complexity of documents and automatically provide language complexity rationales to second language learners, a task that clearly affected them personally. Computer science, data science, and machine learning projects are also able to showcase the interdisciplinary study that is so integral to the mission of liberal arts colleges. Another project group from CS335, which included some biology-computer science double majors, used support vector machines and convolutional neural networks to classify pneumonia from X-ray images. In order to ensure student success, I carefully scaffolded project deliverables, and had students submit in succession an idea proposal, academic literature review, weekly progress reports, and a 8-12 page final project paper. In the final five weeks of the students' projects, I met with project groups weekly to give individualized feedback and guidance. In the final **course evaluations**, one student wrote "The machine learning implementation project was a really cool opportunity and [Prof. Keith] gave us plenty of guidance when needed which was highly appreciated." Another commented "Because the professor is ambitious and holds us to high standards, I learned a lot, especially working on the final project."

Differentiated instruction Because my goal as an instructor is to have *every* student meet my course's learning objectives, I provide *differentiated instruction* via additional, optional learning opportunities. For example, when I taught *CS 335: Machine Learning* at Mt. Holyoke College, I scheduled an optional "fourth-hour" lecture time as review of core mathematical and computational principles (calculus, Python, probability and logarithms, and unit reviews before tests). This ensured that the students who needed additional help with these fundamental principles could also be successful in the course.

Science communication. Too often I have seen computer science classes provide students with technical skills but ignore developing students' ability to communicate technical ideas effectively and think more broadly about what they are building. Thus, in my classes I aim to have at least one assignment or project that has students *think and write critically* about computer science. In particular, as a teaching assistant for *CS685: Advanced Natural Language Processing*, we assigned students to write a 8-15 paper that reviewed a subfield of natural language processing. As the teaching assistant, I gave extensive written feedback on every draft and one student used this feedback to turn her survey on semantic parsing into a published paper [3].

Decreasing bias in the classroom. Diversity and inclusion start in the classroom. In my teaching, I aim to follow research-based best practices to decrease bias [5]. First, I use *Gradescope* to have students submit their work so I can grade anonymously. Second, within the first two weeks of teaching, I memorize all students' names and have them teach me how to pronounce their names correctly. Third, I demystify the paths to success in my classes through *transparent teaching*. I explain the underlying reasoning behind my decisions and policies in order to make explicit the path to success. In final course evaluations for *CS 335: Machine Learning* one student wrote, "Professor Keith is one of the most organized professors I've met. She is very clear about her expectations and very prompt in giving feedback on homework." Fourth, I solicit student feedback several times throughout the semester by having students write down on notecards what was going well in the class and what they would like to see changed. This allows me to more quickly adapt my class management to students needs.

Remote learning. When the unprecedented COVID-19 global pandemic emerged, I was in the middle of my semester teaching *CS335: Machine Learning*. Similar to rest of the academic world, I had to quickly adapt and switch from high-touch, in-person teaching to 100% remote learning in the course of two weeks. I redesigned my class to be a *flipped classroom* while still retaining high-quality, personal interactions via Zoom. This meant I pre-recorded and uploaded video lectures one week in advance and had students watch the lectures on their own time (see Figure 1). I also held weekly one-hour recitations in which I met with groups of 3-5 students via Zoom. Scattered throughout the video lectures were conceptual or technical questions that I asked students to pause and work through on their own. I required students to come to recitations with 2-3 questions from the lecture and during recitation I answered students' individual questions and those posed in the lecture. This structure allowed for asynchronous learning since many students were in different time zones, but also held students accountable since they had to develop novel questions that demonstrated they had understood the material from the video lectures.

4 Future teaching interests

In the future, I am interested in teaching the first and second introductory courses to computer science. Because these courses are the gateway into the computer science major, I am especially interested in making sure the learning environments are engaging and encouraging to underrepresented students, especially women and students of color. Since my research is grounded in statistical and mathematical principles, I would also be able to teach Data Structures and Algorithms. For elective courses, I am interested in teaching Machine Learning, Natural Language Processing, and/or a *Data Science Fundamentals* course that would cover topics such as introductory machine learning, causal inference, data visualization, data cleaning, crowd-sourcing, and distributed systems for data science.

I am also interested in expanding my previous seminar to a class entitled *Artificial Intelligence: Power, Oppression, and Justice*. As Catherine D'Ignazio and Lauren F. Klein describe in *Data Feminism* "a broader focus on *data justice* rather than *data ethics* alone can help to ensure that past inequalities are not distilled into black-boxed algorithms" [1]. The course would highlight efforts of recent scholars to examine AI in the context of social justice and students would read selections from *Weapons of Math Destruction* by Cathy O'Neil, *Algorithms of Oppression* by Safiya Umoja Noble, *Automating Inequality* by Virginia Eubanks, *Race after Technology: Abolitionist Tools for the New Jim Code* by Ruha Benjamin, and *Design Justice: Community-Led Practices to Build the Worlds We Need* by Sasha Costanza-Chock, among others. I am passionate about teaching these courses and I look forward to my own learning journey of building more effective courses and teaching methods in the future.

References

[1] Catherine D'Ignazio and Lauren F Klein. *Data feminism*. MIT Press, 2020.

[2] Carol S Dweck. *Mindset: The new psychology of success*. Random House Digital, Inc., 2008.

[3] Aishwarya Kamath and Rajarshi Das. A Survey on Semantic Parsing. In *Automated Knowledge Base Construction (AKBC)*, 2019.

[4] James M Lang. *Small teaching: Everyday lessons from the science of learning*. John Wiley & Sons, 2016.

[5] Colleen Lewis. CS Teaching Tips: Tips For Reducing Bias. `https://www.csteachingtips.org/tips-reducing-bias`.

[6] Colleen M Lewis, Ruth E Anderson, and Ken Yasuhara. "I Don't Code All Day" Fitting in Computer Science When the Stereotypes Don't Fit. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, pages 23–32, 2016.