Teaching Statement

Lirong Xia, Rensselaer Polytechnic Institute, xial@cs.rpi.edu

1 Undergraduate-level courses

Introduction to AI (2014S, 2017S, 2018S). I have developed and taught the 4000-level *Introduction to AI* since 2014 spring. This course aims at offering a comprehensive overview of important techniques and ideas in AI. The course covers a wide range of topics, ranging from classical AI topics such as search, constraint satisfaction, planning, classical machine learning, and Markov decision processes, to relatively new content such as reinforcement learning, as well as emerging fields such as algorithmic game theory and blockchains. I used U.C. Berkeley's Pacman project as the course project, and use Submitty developed by Prof. Cutler at RPI for auto-grading.

Introduction to AIgorithms (2016S, 2017F). I co-taught the 2000-level *Introduction to Algorithms* in 2016 spring (200 students) with Prof. Elliot Anshelevich. Then in 2017 fall I taught it again by myself. This course aims at offering an introduction to fundamental principles and techniques in algorithmic design. Topics covered in the class include analysis of runtime, graph algorithms, divide and conquer, dynamic programming, and basic computational complexity theory.

Computational Social Processes (2016F). The course is the undergraduate version of the graduate-level Computational Social Choice described below.

In the future, I plan to develop an undergraduate-level multidisciplinary course called *Economics and Computation*, which covers fundamental ideas and techniques in the burgeoning field of the same name. I believe that this course will attract students from both science and engineering, which also helps improve the diversity in computer science.

2 Graduate-level course

Computational Social Choice (2013F, 2014F, 2016S). This is a multi-disciplinary course that was developed to help new Ph.D. students and senior undergraduate students learn general methodologies for independent research in computer science in the context of computational social choice. In the first half of the course I gave lectures on important topics in the field, and the second half was organized as a seminar: students presented selected papers and actively participated in discussions. Each student also completed a personalized project targeting either a submission at a top-tier peer-reviewed conference or an improvement of real-world systems.

The course attracted students from various departments including CS, ECSE, economics, ISE, and math. Overall, students liked the organization of the course, as one of them commented: "*The balance of lectures + presentations is excellent*". Students also enjoyed the critical-thinking environment of the course, as one of them commented: "*He is an excellent teacher. He treated students as independent researchers. He* *was never authoritative. His research advice is succinct and to the point. He covered different topics very efficiently. Honestly, I cannot find any bad point. He is one of the best professors I have ever met.*"

The course projects are rewarding to both students and me. For example, five course projects led to publications at top AI conferences (one at each of AAAI, AAMAS, ICML, IJCAI, and UAI). One project improved a course recommender system for RPI students. Moreover, the Online Preference Reporting and Aggregation System (OPRA) was initialized as a course project in 2014 fall, improved in another course project in 2016 spring, and became an active research project since then.

In the future, I plan to develop a graduate-level course on using machine learning to help with decision-making. The course will be based on an invited book of mine that will be published in the Synthesis Lectures on Artificial Intelligence and Machine Learning by Morgan & Claypool.

3 OPRA: Online Preference Reporting and Aggregation System

As mentioned above, OPRA is an open-source project developed by RPI undergraduate students since 2014 summer. It currently offers the following functionalities.

1. **Voting**, where anyone with RCS ID can login and cast vote on any topics setup by another user, e.g. the best office hours or the topic to cover in recitation. OPRA was used to host opinion polls for GM week 2017.

2. **In-class e-quiz**. I used OPRA to take in-class quizzes in Introduction to Algorithms 2017 fall and Introduction to AI 2018 spring. OPRA is a much more flexible, convenient, and economic replacement of iClicker. Students can easily submit their answers on smartphones or laptops. Their answers greatly helped me to identify potential confusions from students.

3. **Computer Science Best Poster Award selection**, where students vote on posters to decide the winner of the official best poster award.

4. **Computer Science Colloquium registration**, which keeps students' records of attending CS colloquia, a key requirement for graduate students.

OPRA has been used in my classes since 2016 fall, including Computational Social Processes (2016), Introduction to Algorithms (2017), Introduction to AI (2017, 2018). It has been proven to be effective, as a student wrote in a compliment letter: *"By making a poll every class, you increase the class participation rate and make more students focus on class material you talk about. You are also a very nice professor because you let us choose our favorite time slots for office hour."*

4 Teaching philosophy

When teaching undergraduate-level courses, I focused on explaining the fundamental concepts using concrete, step-by-step examples, while also briefly mentioning advanced materials for interested students. To engage students in the classed, I used the following strategies. First, I tried to design examples that would intrigue RPI students, instead of using standard text-book examples. For example, when teaching introduction to AI, I showed students a game between two human beings in a popular Chinese TV Show called "The Brain". Then, I told the students that designing an AI to play the game is closely related to multiple topics covered in the class, such as game theory, minimax search, machine learning, and reinforcement learning. In fact, some students participated in the bonus project to setup the framework for other student groups to design AI programs to compete with each other. Second, as discussed above, I used OPRA to take in-class quizzes. In this way, students are more motivated to engage in the class.

When teaching graduate-level courses, in addition to helping students learn new materials, I put more emphasis on helping students develop skills to do research in computer science. My graduate-level course can be seen as a short version of a Ph.D. in CS. Students are asked to first learn basic material, then read state-of-the-art papers, write quick summaries of these papers, generate new ideas, setup meetings with me to discuss research plans, carry out research, write up a report and go through peer review, and finally, present and defend the research. I believe that after taking my class, students will have a better idea about how to be a successful Ph.D. in CS.