CS-0311: Research in Artificial Intelligence

COURSE INFORMATION

Instructor(s):

Lee Spector
lasCCS@hampshire.edu
Office Extension: x5352
Office Hours:
Regular office hours: Tuesdays and Thursdays, 2:30-3:45, and Wednesdays 1:00-2:30. I am available at many other times as well; feel free to contact me in person or via email to arrange other meeting times. Sign up for regular office hours and advising day meetings here.

Term:

2018F

Meeting Info:

Tuesday 10:30 AM - 11:50 AM Adele Simmons Hall (ASH) 126

Thursday 10:30 AM - 11:50 AM Adele Simmons Hall (ASH) 126

Description:

Students in this course will become members of research teams focusing on projects designated by the instructor. Projects will involve open research questions in artificial intelligence, artificial life, or computational models of cognitive systems. They will be oriented toward the production of publishable results and/or distributable software systems. Students will gain skills that will be useful for Division III project work and graduate-level research. Prerequisite detail: Strong computer programming skills

Course Objectives:

- To engage in scientific/technological inquiry.
- To work collaboratively with classmates, the professor, and the larger research community.
- To understand and to be able to navigate current research literature.

Evaluation Criteria:

- On-time attendance. Necessary lateness or absences must be explained, preferably in advance.

- Active participation in class, including regular service as class scribe and regular participation in show and tell. You should always be prepared to serve as scribe, taking notes on announcements and major topics/activities in class, and emailing me notes at the end of the class session. You should also always have something to show and/or tell about your current work for the class.
- Active participation in research clusters, both in and out of class. At every point in time you will be a member of a cluster of 2-4 students, who will have to meet at least once per week for at least half an hour outside of class, although more cluster meetings are strongly encouraged and some may take place in class. Cluster membership may change from week to week, according to an algorithm that will not be disclosed. Toward the end of the semester, clusters will converge to reflect group project partnerships.
- A portfolio of research-related notes/writing/code/results that grows weekly over the course of the semester, and must be resubmitted each week. By the end of the semester your portfolio should reflect significant project work along with growth in research skills and knowledge. Your portfolio must include a new status report each week, which must describe cluster meetings along with other activities and plans.

Additional Info:

MATERIALS

All necessary materials will be distributed in class or on this site.

COURSE STRUCTURE (ROUGHLY, WITH SOME OVERLAPS)

1. Reading and discussion
2. Project ideation and pitching
3. Project development
4. Project reporting and presentation

WORK TIME EXPECTATIONS

In this course, students are expected to spend six to eight hours per week in preparation and work outside of class time. This time includes reading course materials, engaging in project work, and meeting with classmates.

HOW TO GET AN EVALUATION FOR THIS COURSE

1. Do all of the reading, always before the day for which the reading is assigned.
2. Attend every class and participate in all discussions and other activities, including cluster meetings.
3. Re-submit an appropriately-updated portfolio each week, including progress reports and the products of other assignments, and demonstrating continuous engagement and growth.

You should not expect to receive an evaluation unless you have met these expectations, or unless the ways in which you fall short are: 1) minor and/or unavoidable (e.g. because of illness), AND 2) well-explained both when the lapses occur and in your final self evaluation. If you are ever in doubt about your status in the class

vis-a-vis evaluation then come talk to me.

INCOMPLETES

Course incompletes are restricted and governed by College policy, and will be negotiated on an individual basis.

DIVISION I DISTRIBUTION CREDIT

Successful completion of this course can be used to satisfy the Division I distribution requirement in Mind, Brain, and Information, **but only by special request**. Contact me as soon as possible if you would like to request this. This course also provides opportunities for satisfaction of Division I cumulative skills requirements in Quantitative Skills and Independent Work.

ILLNESS

If you have a fever, please stay home, take good care of yourself, and contact me by email or phone. If your illness makes it impossible for you to meet the course deadlines then contact me and we will negotiate an accommodation.

A "RESPONSIBLE EMPLOYEE"

Text from the Dean of Faculty

A "Responsible Employee" is any Employee who is not a Confidential or Private Employee. Responsible Employees include Faculty, Staff and Resident Advisors, Teaching Assistants, EMTs and all other student employees when disclosures are made to any of them in their capacities as employees. As a faculty member, I am required to immediately report to the College's Title IX Coordinator all relevant details (obtained directly or indirectly) about Sexual Misconduct Violations or potential violations that involve a College Student or Employee as a Complainant or Respondent, including dates, times, locations, and names of parties and witnesses. If a Complainant requests (a) that personally-identifying information not be shared with the Respondent, (b) that no investigation be pursued, and/or (c) that no disciplinary action be taken, the College will seek to honor this request unless there is a health or safety risk to the Complainant or to any member of the College community. Section VI IA. of the Policy provides additional information about remedial and protective measures.

Responsible Employees are not required to report information disclosed (1) at public awareness events (e.g., "survivor speak-outs", candlelight vigils, protests, or other public forums in which students may disclose Sexual Misconduct Violations; collectively "Public Awareness Events"; or (2) during an individual's participation as a subject in an Institutional Review Board-approved human subjects research protocol ("IRB Research"). The College may provide information about Title IX rights and about available College and community resources and support at Public Awareness Events, however, and Institutional Review Boards may, in appropriate cases, require researchers to provide such information to all subjects of IRB research.

ACADEMIC DISHONESTY: PROCEDURES FOR DEALING WITH VIOLATIONS

Text from the Dean of Faculty

Academic dishonesty (plagiarism (presentation of another's work as one's own), fabrication, or falsification of data) is a breach of the ethics of scholarship and a violation of one of the central norms of an academic community. Because reports of academic dishonesty are most likely to arise from work done in a course or for a divisional project, a member of the college faculty usually brings forward the report. When such a report is brought forward, the procedure is as follows:

> The faculty member will inform the student and the School dean that a violation of academic honesty may have occurred. The School dean will inform the dean of advising of the violation. The faculty member will provide all documentation to the dean of advising, who will meet with both the student and faculty member, and recommend a course of action. If the dean of advising determines that it is more likely than not that academic dishonesty has occurred and determines that it is a first offense, the dean of advising will:
>
> - Write a letter of warning to the student, to remain in the student's academic file;
> - In consultation with the faculty member and the School dean, determine academic consequences that may include but are not limited to submitting a revised or new assignment; no evaluation given for the course regardless of add/drop/withdrawal deadlines or, in the case of Division III work, a decision to set aside the project in question and require the student to do an alternative project on a different topic with a different committee (unless the committee concerned agrees to continue working with the student).

Additional text specific to this course

In this course we will often be sharing and borrowing code. This is an important aspect of the course and an important aspect of modern programming practice. This does not mean, however, that it is acceptable to submit code that is not your own without acknowledging sources. Sources should be clearly and explicitly provided in everything that you produce.

OFFICE OF ACCESSIBILITY RESOURCES AND SERVICES (OARS)

Text from the Dean of Faculty

Hampshire College offers services on an individual basis to students with documented disabilities. The College recognizes its obligation under Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990 to provide reasonable accommodations for individuals with disabilities so they may participate as fully as possible in the College's academic programs. Disabilities may include, but are not limited to, sensory impairments, mobility impairments, chronic illnesses and medical disabilities, learning disabilities, developmental disabilities and psychological disabilities. The director of OARS is responsible for the

coordination of services and accommodations for students with disabilities. Accommodations may be provided by OARS directly, but often accommodations are implemented in collaboration with faculty or other relevant campus offices and services. OARS may be reached at 413.559.5498 or via email: accessibility@hampshire.edu.

Announcements

Open Forum

An open forum for any discussions related to work in the course

---

Assignments

For Tuesday, September 11

- Read "And now, digital evolution," by Spector
- Read "Evolution of artificial intelligence," by Spector
- Browse geneticprogramming.com
- Read the Executive Summary of "Artificial Intelligence and National Security," by Greg Allen and Taniel Chan. Optionally, read more.
- Skim "Evolutionary Image Synthesis Using a Model of Aesthetics," by Ross, Ralph, and Zong

For Thursday, September 13

- Watch "An Even Quicker Introduction to the Push Programming Language," (or the older "A Quick Introduction to the Push Programming Language"), by Spector
- Read "Program Synthesis using Uniform Mutation by Addition and Deletion," by Helmuth, McPhee, and Spector (see below)
- **Turn in** your portfolio, which must include a progress report, which must in turn include a cluster meeting report

For Tuesday, September 18

- Read "Evolution Evolves with Autoconstruction" (see below) by Spector, McPhee, Helmuth, Casale, and Oks
- **Bring to class** descriptions of two ideas for possible projects for the class. These could be sketchy and you don't have to have any idea of whether they're feasible or of how we would accomplish them; they're just for generating discussion about how we might want to proceed. Aim for one involving evolutionary computing, and one not necessarily involving evolutionary computing. Your ideas should be written down,

and could be as brief as a sentence each, or maybe a paragraph. You don't have to turn them in on Tuesday -- just be ready to read them to the class for our discussion. You **should** include them in your portfolio, for later submission.

- Optionally check out our paper on the software synthesis benchmark suite (below), Tom Helmuth's dissertation that used and investigated these problems, and/or the associated technical report with even more detailed problem descriptions.

For Thursday, September 20

- **Turn in** your portfolio, which must include a progress report, which must in turn include a cluster meeting report
- **Bring to class** results of research into one or more of the project ideas discussed in class on Tuesday
- Optionally check out "Local projects of interest" below

For Tuesday, September 25

- Read "What's in an evolved name? The evolution of modularity via tag-based reference" by Spector, Harrington, Martin, and Helmuth
- Optionally watch Lee Spector interview by Sentient on the future of AI and evolutionary algorithms (or on Youtube)
- Optionally read "No Strategy Can Win in the Repeated Prisoner's Dilemma: Linking Game Theory and Computer Simulations" by Garcia and Veelen (see email)
- Optionally preview slides for my presentation on Autoconstruction

For Tuesday, October 2

- Check out my pitch on lexicase selection efficiency (see below)
- Pitches are due, and we will discuss them.

For Tuesday, October 16

- Read "Lexicase Selection Beyond Genetic Programming" by Metevier, Saini, and Spector (see below)

For Tuesday, October 30

- Read "search-worksheet.clj.pdf" by Spector (see below)

For Tuesday, November 13

- Check out "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations" by Muro, Escobedo, Spector, and Coppinger (see below)
- Check out "Emergence of Collective Behavior in Evolving Populations of Flying Agents" by Spector, Klein, Perry, and Feinstein

- Check out "Division blocks and the open-ended evolution of development, form, and behavior" by Spector, Klein, and Feinstein
- Check out the "Cellular Automaton Wikipedia page

For Thursday, November 15

- Check out "ε-Lexicase Selection for Regression" by La Cava, Spector, and Danai
- Check out Bill La Cava's GECCO presentation on epsilon lexicase selection (see below)
- Read my post on epsilon lexicase selection

For Tuesday, December 4

- Read Chapters 1-3 of **Automatic Quantum Computer Programming: A Genetic Programming Approach**, by Spector (see below)
- Optionally read "Extending Program Synthesis Grammars for Grammar-Guided Genetic Programming," by Forstenlechner et al (see below)
- Optionally read "Stepping Stones to Inductive Synthesis of Low-Level Looping Programs," by Rosin

For Tuesday, December 10

- Optionally read "Analyzing a Decade of Human-Competitive ("HUMIE") Winners: What Can We Learn?," by Kannappan et al.
- Optionally read "General Program Synthesis Benchmark Suite ," by Helmuth and Spector (linked page includes links to detailed problem descriptions, presentation slides, code, and data)

Ongoing: in project mode. Mostly check class notes.

- Program Synthesis using Uniform Mutation by Addition and Deletion
- General Program Synthesis Benchmark Suite
- Evolution Evolves with Autoconstruction
- Local projects of interest
- Pitch: Lexicase Selection Efficiency
- Additional Pitches
- Lexicase Selection Beyond Genetic Programming
- search-worksheet.clj.pdf
- Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations
- Bill La Cava's GECCO presentation on epsilon lexicase selection

Automatic Quantum Computer Programming: A Genetic Programming Approach

Extending Program Synthesis Grammars for Grammar-Guided Genetic Programming

Class notes (recorded by student scribes)

September 11

September 13

September 18

September 20

September 25

October 2

October 4

October 11

October 16

October 18

October 25

October 30

November 1

November 6

November 8

November 13

November 15

November 20

November 27

November 29

December 4

December 6

Clusters

- September 6 - September 13
- September 13 - September 20
- September 20 - September 27
- October 4 - October 11
- October 11 - October 18
- October 18 - October 25
- October 26 - November 1

Portfolios

Submit updated portfolios here

Submit pitches here

Your pitch, due before class on October 2, should be 1 full page (total) and contain:

- A name for the pitched project
- A section called "Why?" that provides the motivation for the project, in terms of the research question that it will aim to answer, and/or the specific need that it will attempt to fulfill.
- A section called "What" that describes the project itself, what you will aim to produce, test, demonstrate, determine, etc.
- A section called "Roles" that lists between 2 and 11 distinct roles that participants will take on. People will probably end up collaborating and mixing up roles to some extent as we proceed, but you should specify some number of distinct roles all the same.
- A section that lists, for each of the roles described above, what the person in that role will do at the start of the project (to get to work right after the projects/teams/roles are determined), what that person will do for the middle portion of the project (when most of the research/development is being done), and what that person will do for the ending portion of the project (which will focus on presentation). Some tasks may overlap between time periods, but it should be clear what the person in each role will be doing in each phase of the project.

Clojure Resources

- Editing and running code: Many Clojure editing and development environments are available. The workflow that I use myself, and that I recommend to others unless they have reasons to prefer something else, is to use Leiningen from the command line to create and run projects, and to use Gorilla REPL to edit and experiment with code. The Gorilla REPL site has videos that will give you a sense of how it works, and a section called "Secrets," with content provided by me, that I think will also help to get you going.
- Clojure.org, the main site for the Clojure language; note especially their Clojure resources page
- Clojure cheatsheet
- Clojure style guide
- Clojure TV
- Two blog posts on dealing with Clojure error messages (stack traces): 1 and 2
- Lee's tutorial materials:
  - Clojestions, suggested exercises for learning Clojure
  - Clojinc, a saved REPL session intended to support semi-independent learning of Clojure
- Other tutorial materials:
  - 4Clojure, interactive Clojure problems
  - Learn X in Y minutes, Where X=clojure, a brief introduction to Clojure
- A good free textbook: **Clojure for the Brave and True** by Daniel Higginbotham. No Starch Press, 2015. ISBN-10: 1593275919, ISBN-13: 978-1593275914. Available free online
- Lee's Clojure GP code:
  - Propel, a relatively simple, easy-to-understand but nonetheless fully functional PushGP implementation
  - Clojush, my group's main research PushGP system, with a zillion features but also a lot of complexity
  - gp, simple Clojure code for traditional, "tree-based" genetic programming (not Push), which is also explained in this talk
- Clojure MXNet, for the Apache MXNet deep learning framework
- Simple Clojure neural network code by Hampshire student Julian Oks
- A few of Lee's other Clojure projects of possible interest:
  - Rock Paper Stuff, a game designed for artificial intelligence experiments and education
  - Pucks, an environment for experiments and education in artificial intelligence and artificial life
- Quil, a Clojure/ClojureScript library for creating interactive drawings and animations

---