625-600 AI Project

- Instructor: Yoonsuck Choe
Basic Rules

• Team of up to 3 people.

• Deliverables:
  – Proposal: all or none grading (must submit); feedback provided.
  – Interim report: all or none grading (must show progress); feedback provided.
  – Presentation: graded.

• Apply AI to your own research area or to a benchmark AI competition.

• May stack with other course projects – need consent of both instructors.

• Must be innovative in one way or another (next slide).
Innovation Requirement

At least one of the following:

- New algorithm.
- Novel paradigms.
- Significantly improved performance of existing algorithm by modification.
- New applications of existing algorithm.
- Use of AI algorithms to analyze data in novel ways: discover new structure, discover novel dependencies, etc.
- Explore novel fundamental questions: “Why X?”
- Challenge existing assumptions: “X is known.,” “X,Y are independent.,” etc.
- Novel uses of AI (e.g., for visualization).
Proposal

- Team members
- What is the research problem? Why is it important/interesting?
- What are other people’s approaches?
- What are other people’s assumptions? (optional)
- What are the limitations of those approaches?
- What are the problems with those assumptions? (optional)
- What is your approach?
- How will you relax the assumptions and why? (optional)
- What experiments will you do?
  - What data set will you use (if applicable)
  - What code base will you use
  - What kind of experiments will you do with the data/code?
- What are the expected results?
Interim Report

About 1 page, single spaced. PDF or hardcopy.

- Basic idea
- Model you used
- Data set you used
- Experiments you ran
- Preliminary results
- Plan for wrapping up
Presentation

6 to 10 slides

- Introduction (research problem and significance)
- Background (existing approaches)
- Proposed approach
- Experiments and Results
- Discussion (novelty, contribution, issues)
- Conclusion
Existing Code Bases

• Learning
  – Choe’s backprop code (C++, Octave).
  – Choe’s neuroevolution code (Octave).
  – Ken Stanley’s NEAT code (several versions, including Java).
  – Your own code,
  – or any other “reasonable” code base.

• Environments
  – Berkeley EECS Pacman project [http://www-inst.eecs.berkeley.edu/~cs188/pacman/pacman.html]
  – Eddie Kohler’s XBrantenberg [http://www.lcdf.org/xbraitenberg/]
  – Marcin Pilat’s Khepera [http://www.pilat.org/khepgpsim/index.html]
  – or any other “reasonable” environment.
Data Sources

- Your own research data.
- UCI ML Repository
- or any other “reasonable” data source.