633-600 Machine Learning Term 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.
  - Final report: graded.
- Apply ML to your own research area.
- 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 ML algorithm.
- Beyond supervised, unsupervised, reinforcement.
- Significantly improved performance of existing algorithm by modification.
- New applications of existing algorithm.
- Use of ML 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 ML (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 2 pages, single spaced. PDF or hardcopy.

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

Presentation & Final Report

About 4 pages, single spaced. IEEE double column format. PDF or hardcopy.

- 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 (Q learning, etc.)
  - or any other "reasonable" code base.
- Environments
  - Eddie Kohler's XBraintenberg `http://www.lcdf.org/xbraitenberg/`
  - Marcin Pilat's Khepera `http://www.pilat.org/khepgpsim/index.html`
  - or any other "reasonable" environment.

Data Sources

- UCI ML Repository `http://archive.ics.uci.edu/ml/`
- or any other "reasonable" data source.