CSCI 447: Machine Learning: Soft Computing – Fall 2019

Class meeting time (Fall): MWF, 2:10 – 3:00, NAH 137

Instructor: Dr. John W. Sheppard (pronouns: he/him), EPS 365, 994-4835, john.sheppard@montana.edu

Office hours: MWF, 10:00 – 10:50 or by appointment

Textbook: None. Various readings will be posted in D2L.

Course Outline

This course is an advanced undergraduate/introductory graduate course that introduces students to biologically-inspired approaches to developing advanced, artificially intelligent systems using principles from biology and machine learning. The major topics to be covered include neural networks, evolutionary algorithms (e.g., genetic algorithms, genetic programming, and evolution strategies), and swarm intelligence (e.g., particle swarm optimization and ant colony optimization). Students will be introduced to the fundamental algorithms in each of these topics and will explore recent research. The primary goals of this course include providing the student with the skills to assess methods of soft computing and to build systems incorporating soft computing techniques.

Topics expected to be covered include the following:

- Fundamentals of Machine Learning
- Common Machine Learning Algorithms
- Neural Networks
- Evolutionary and Swarm-based Computation

Learning Outcomes

At the end of this course, the student will be able to:

- Formulate and assess problems in machine learning and soft computing.
- Understand and be able to implement several machine learning and soft computing algorithms.
- Assess the strengths and weaknesses of alternative representations and algorithms in machine learning and soft computing.
- Analyze the expected performance of machine learning and soft computing algorithms based on formal principles from the analysis of algorithms.
- Assess the performance of machine learning and soft computing algorithms based on principles of sound experimental design and empirical analysis.

Homework

The best way to learn about new techniques in artificial intelligence, bio-inspired computing, and machine learning is by implementing approaches to solve different problems. Therefore, the assignments in the course will emphasize implementation and experimentation. Four projects will be assigned, and each project will carry with it the requirement to write a short expository paper describing the results of the associated experiments. All projects will be completed in teams of 3-4 people.

- Project 1 ($\sim$ 3 weeks):
  - Assigned August 26, 2019
  - Project Due September 13, 2019
- Project 2 ($\sim$ 4 weeks):
  - Assigned September 16, 2019
  - Design Document Due September 27, 2019
  - Project Due October 11, 2019
- Project 3 ($\sim$ 4 weeks):
  - Assigned October 14, 2019
  - Design Document Due October 25, 2019
  - Project Due November 8, 2019
- Project 4 ($\sim$ 4 weeks):
  - Assigned November 13, 2019
  - Design Document Due November 22, 2019
  - Project Due December 11, 2019

All assignments are due at the beginning of class on the assigned date. Assignments should be uploaded into D2L using the group submission. Written documents shall be PDF. We will use the finals period to discuss results of the final set of experiments and to synthesize experiences over the entire course. Attendance at this session is mandatory and will be verified through iClicker.

Exams

There will be no exams in this course; however, iClicker exercises will be treated like short quizzes.

Grading

Grading will be based on four programming projects. Each programming project will be split into three parts. The first part will focus on design, the second on implementation, and the third part will focus on writing up experimental results. Note, however, that the first programming assignment will not have a design document.

The projects will be completed with teams of three or four people. Teams may use any language and machine for satisfying the programming requirements, but all members of a team must use the same programming language.

Note that iClickers are required for this course.

- Project 1 (20%)
- Project 2 (20%)
- Project 3 (20%)
- Project 4 (20%)
- iClicker assignments (20%)

For Project 1, the following parts are required.

- Programming (30%) — Must include the following:
  - Source code (one class per file) fully documented with class-level comments and in-line comments explaining logic. This accounts for 1/3 of the Programming grade.
  - Text file identifying which team member was responsible for what portion of the code.
- A video demonstrating proper functioning of the code (30%) — Must be no more than 5 minutes in length and include the following:
  - The process for generating the learned model.
  - The actual learned model.
  - Application of example data to the learned model.
  - NOTE: Do not waste time explaining the structure of the code. That's what your comments are for.
- Paper with results (40%) — Must be no more than 10 pages using JAIR/JMLR format and include the following:
  - Title and authors
  - Abstract
  - Introduction and Background
  - Problem Statement with Testable Hypothesis
  - Experimental Approach
  - Results
  - Analysis and Discussion of Results
  - Conclusion
  - References

For Projects 2–4, the following parts are required:

- Design Document (30%) — Must include the following:
  - UML class diagram.
  - A textual description of the major classes in the UML diagram.
  - An explanation of all major design decisions made.
  - A description of the experimental design to be applied.
  - References for any sources used to aid in the design.
- Programming (10%) — Must include the elements listed above.
- Video (20%) — Must include the elements listed above.
- Paper with results (40%) — Must include all of the elements listed above.

Final Period: Note that the university treats the final period as "instructional time." Therefore, even though there is no final exam, we are required to meet. We will use the final period to discuss the course projects and to hold one last iClicker session.

Group Work: The four projects will be completed in groups of three or four people. Individual projects are not permitted. It is up to the individual groups to ensure fair distribution of labor. All members of the group will receive the same grade. On all projects, all group members will use the same programming language.

iClickers: Attendance will be taken using iClickers on random days. Active learning exercises will also be held using iClickers. Scores will be based on a combination of participation and correctness in responses.

Appeals: With respect to grading appeals, ideally, assignments will be returned within one week of being handed in. Questions on grading should be directed to the TA within one week of receiving your graded assignment. After receiving a decision from the TA, additional appeals may be made to the instructor within one week of receiving that decision. After that, appeals will not be considered. The instructor's decision is final.

Academic Conduct and Academic Integrity

All students are expected to conduct themselves in a professional, honorable, and courteous way. Plagiarism and other forms of academic misconduct, including seeking out solutions to homework assignments or unauthorized collaboration, will have severe consequences. *Example: Googling the solution to a homework problem and submitting what you find as your own work.* Students who engage in any form of academic misconduct will be reported to the Dean of Students and will receive an F for the course. Dropping the course to avoid an F resulting from academic misconduct will not be permitted under any circumstances.

Class Topics (subject to revision): Check schedule above for due dates

- Week 1 (8/26/19 – 8/30/19)
  - Guest Lecturer: Amy Peerlinck
  - *Topic:* Introduction to Machine Learning (Learning Paradigms, First Algorithm, Experimental Methods)
  - *Reading:* S. Kularni and G. Harman, "Introduction: Classification, Learning, Features, and Applications," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* T. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, 10(7)1895–1923, 1998.
- Week 2 (9/2/19 – 9/6/19) — No class on Monday (Labor Day)
  - *Topic:* Learning Theory (PAC and Bayes)
  - *Reading:* S. Kularni and G. Harman, "The Optimal Bayes Decision Rule," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* S. Kularni and G. Harman, "PAC Learning," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* S. Kularni and G. Harman, "VC Dimension," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* S. Kularni and G. Harman, "Infinite VC Dimension," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
- Week 3 (9/9/19 – 9/13/19)
  - *Topic:* Naïve Bayes and Variants
  - *Reading:* T. Mitchell, "Generative and Discriminitive Classifiers: Naive Bayes and Logistic Regression," new chapter intended for *Machine Learning*, 2015.
  - *Reading:* N. Friedman, D. Geiger, & M. Goldszmitd, "Bayesian Network Classifiers," *Machine Learning*, 29:131–163 (1997).
- Week 4 (9/16/19 – 9/20/19)
  - *Topic:* Density Estimation and Kernels
  - *Reading:* E. Alpaydin, "Nonparametric Methods," *Introduction to Machine Learning*, The MIT Press, 2014.
  - *Reading:* S. Kularni and G. Harman, "Kernel Rules," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
- Week 5 (9/23/19 – 9/27/19)
  - *Topic:* Nearest Neighbor, k-Means, and k-Medoids
  - *Reading:* S. Kularni and G. Harman, "The Nearest Neighbor Rule," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* T. Hastie, R. Tibshirani, & J. Friedman, "Unsupervised Learning: Cluster Analysis, *The Elements of Statistical Learning*, Springer, 2009.
- Week 6 (9/30/19 – 10/4/19)
  - *Topic:* Linear Models (Perceptrons, Logistic Regression, SVMs)

- *Reading:* T. Mitchell, "Generative and Discriminitive Classifiers: Naive Bayes and Logistic Regression," new chapter intended for *Machine Learning*, 2015.
- *Reading:* S. Kularni and G. Harman, "Neural Networks: Perceptrons," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
- *Reading:* S. Kularni and G. Harman, "Support Vector Machines," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.

- Week 7 (10/7/19 – 10/11/19)
  - *Topic:* Radial Basis Function Networks, Kernel Machines
  - *Reading:* j. Moody & C. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation* 1, 281–294 (1989).
  - *Reading:* K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, & B. Schölkolpf, "An introduction to Kernel-Based Learning Algorithms," *IEEE Transactions on Neural Networks*, Vol. 12, No. 2, March 2001, pp. 181–201.
- Week 8 (10/14/19 – 10/18/19)
  - *Topic:* Multi-Layer Networks, Autoencoders, Restricted Boltzmann Machines
  - *Reading:* S. Kularni and G. Harman, "Multilayer Networks," *An Elementary Introduction to Statistical Learning Theory*, John Wiley & Sons, 2011.
  - *Reading:* D. Rumelhart, G. Hinton, & R. Williams, "Learning Internal Representations by Error Propagation," ICS Report 8506, Institute for Cognitive Science, University of California San Diego, September 1985.
- Week 9 (10/21/19 – 10/25/19)
  - *Topic:* Stacked Autoencoders, Deep Belief Networks, Convolutional Networks
  - *Reading:* I. Goodfellow, Y. Bengio, & A. Courville, "Autoencoders," *Deep Learning*, The MIT Press, 2016.
  - *Reading:* H. Aghdam & E. Heravi, "Convolutional Neural Networks," *Guide to Convolutional Neural Networks*, Springer, 2017.
- Week 10 (10/28/19 – 11/1/19)
  - *Topic:* Decision Trees, Hierarchical Clustering, Density-based Clustering
  - *Reading:* J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, 1:81–106 (1986).
  - *Reading:* J. Ward, "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, Vol. 58, No. 301, Mar. 1963, pp. 236–244.
- Week 11 (11/4/19 – 11/8/19)
  - *Topic:* Rule Learning (FOIL, Ripper, Apriori)
  - *Reading:* J. R. Quinlan, "Learning Logical Definitions from Relations," *Machine Learning*, 5:239–266 (1990).
  - *Reading:* T. Hastie, R. Tibshirani, & J. Friedman, "Unsupervised Learning: Association Rules, *The Elements of Statistical Learning*, Springer, 2009.
- Week 12 (11/11/19 – 11/17/19) — No class on Monday (Veteran's Day)
  - *Topic:* Genetic Algorithms and Schemata
  - *Reading:* L. Booker, D. Goldberg, & J. Holland, "Classifier Systems and Genetic Algorithms," *Artificial Intelligence* 40 (1989) 235–282.

- Week 13 (11/18/19 – 11/22/19)
  - *Topic:* Evolution Strategies and Differential Evolution
  - *Reading:* H.-G. Beyer & H.-P. Schwefel, "Evolution Strategies," *Natural Computing* 1: 3–52 (2002).
  - *Reading:* R. Storn & K. Price, "Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," International Computer Science Institute, Technical Report TR-95-012, Machien 1995. 33–57.
- Week 14 (11/27/19 – 12/1/19) — No class on Wednesday or Friday (Thanksgiving Holiday)
  - *Topic:* Introduction to Swarms
  - *Reading:* R. Poli, J. Kennedy, & T. Blackwell, "Particle Swarm Optimization: An Overview," *Swarm Intelligence* (2007) 1:33–57.
- Week 15 (12/2/19 – 12/6/19)
  - *Topic:* Ant Colony Optimization and Particle Swarm Optimization
  - *Reading:* M. Dorigo, G. DiCaro, and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," *Artificial Life* 5 (1999) 137–172.
- Finals Week (12/11/19, 8:00 am)
  - Discussion of results of Project #4 and course experiences
  - Final iClicker exercise