CMSC 422: Introduction to Machine Learning   Spring 2020

Time and Place: Tuesdays and Thursdays, 12:30 pm – 1:45 pm, PLS 1140 (Plant Sciences Bldg.)

Instructor: James Reggia, Iribe Bldg., Rm. 4124, reggia@cs.umd.edu
Office Hours: Thursdays, 1:45 pm – 2:45 pm, or by appointment

Teaching Assistant: TA Office Hours and locations are posted online.
   Hamid Kazemi      hamidkazemi22@gmail.com
   Jingling Li       jingling@cs.umd.edu
   Runxing Lin       runxinglin008@gmail.com

Class web page: http://www.cs.umd.edu/class/spring2020/cmsc422-0101/
   Gives exam dates, homework assignments and their due dates, lecture slides, reading assignments, and links to other useful information.

Prerequisites: CMSC 320, 330 and 351, or permission of CS Dept. We assume that you have a basic knowledge of elementary probability, statistics, vector algebra, and calculus, and that you know how to program using Python.

Objectives: The primary objectives of this course are to provide a broad overview of existing methods for machine learning and an introduction to using basic machine learning software.

Midterm Exam: 12:30 pm, Thursday, March 12, 2020
Final Exam: 1:30 pm, Tuesday, May 19 (2 hours)

Content: Machine learning (ML) studies representations and algorithms that allow machines to improve their performance on a task from experience. The field has been growing very rapidly during the last decade, reflecting the successful application of increasingly powerful computational techniques and machines, and the expanded opportunities brought about by large amounts of online data. This course provides a broad overview of existing methods for machine learning and an introduction to adaptive systems in general. The enormous amount of information currently available about ML means that we will have to be selective in what we cover this semester from the topics listed below. However, by the end of the semester you should be familiar with the central concepts and basics of commonly used ML methods.

Introduction
   This provides a conceptual framework, including definitions, terminology, overview of different paradigms, history, basics of human learning, and example applications.

Supervised Learning
   Methodology: data pre-processing, measuring performance, cross validation, overfitting, comparative experimental studies, online resources
   Instance-Based Learning: k-nearest neighbors, IBn algorithms, case-based methods
   Neural Networks: linear models, perceptrons, gradient descent, logistic regression, error backpropagation, radial basis function networks, deep convolution networks (deep learning)
   Symbolic Rule Induction: search, version spaces, decision tree induction, learning association rules, sequential covering algorithms, inducing first order rules, inductive bias
   Statistical Methods: naive Bayesian classifiers, Bayesian networks, support vector machines, Bayesian Learning, etc.
   Ensemble Learning: bagging, boosting, and related techniques
Intermediate Learning

Reinforcement Learning: Q learning, temporal difference learning, applications e.g., AlphaGo

Evolutionary Computation: genetic algorithms/programs applied to machine learning, classifier and other rule-based systems, scientific discovery, creativity enhancement, Baldwin effect

Unsupervised Learning

Clustering Methods: K-means algorithm, expectation maximization, automated discovery

Neural Networks: Hebbian and competitive learning, visualization via self-organizing maps, Hopfield networks, Boltzmann machines, deep belief networks

Additional Topics as Time Permits

Semi-supervised learning, generative adversarial networks, adaptive learning rates, RPROP, BPTT, LSTM, echo state networks, imitation learning, active learning, hybrid systems, self-organizing systems in general, adaptation in artificial life systems, analytical learning (explanation-based learning), theoretical foundations such as PAC learning, VC dimension, no free lunch theorem, etc.

Workload and Grading: There will be regular reading and homework assignments. Assignments will include conducting online experiments, typically using Python, NumPy or Scikit-learn. Assignments are always to be treated as independent work. Grading will be based on homework assignments, in-class worksheets/quizzes, and class participation (collectively 15%), a small-group semester project and report (15%), a midterm exam (30%), and a final exam (40%).

Textbooks: The textbooks below provide an overview of a broad range of ML topics and principles. They will be supplemented with relevant articles (available as pdf files posted on the class website) and online information sources throughout the semester.


Disabilities: Any student eligible for and requesting reasonable academic accommodations due to a disability needs to provide the instructor with a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester.

Class Absence Policy: The campus policy governing class absences requires instructors to provide the following information. For CMSC 422, the “major scheduled grading events” are the midterm and final exams, and the semester project. A maximum of one self-signed medical excuse for late submission of other grading events will be accepted. For other campus policies, please see http://www.ugst.umd.edu/courserelatedpolicies.html

Academic Integrity: All homework assignments are to be done individually and independently; all submitted work must be your own. All students are expected to be familiar with and to uphold the campus Code of Academic Integrity (please see http://www.shc.umd.edu). Posting homework solutions online (e.g., Piazza) violates academic integrity policy. Further details of CMSC Dept. Academic Integrity policies are at http://www.cs.umd.edu/class/resources/academicIntegrity.html