Machine Learning

CSCI 4830, Section 007

Location: Tuesday and Thursday, 09:30-10:45, ECCR 137
Instructor: Professor Michael Mozer
Office: ECOT 741
Office Hours: Tuesday and Thursday, 11:00-12:30
Phone: (303) 492-4103
E-Mail: mozer@colorado.edu
Course URL: http://www.cs.colorado.edu/~mozer/courses/ugrad-ml

Course Description

The goal of *machine learning* research is to build computer systems that learn from experience and that adapt to their environments. Machine learning systems do not have to be programmed by humans to solve a problem; instead, they essentially *program themselves* based on examples of how they should behave, or based on trial-and-error experience trying to solve the problem. Machine learning systems require *learning algorithms* that specify how the system should change its behavior as a result of experience. Researchers in machine learning develop new learning algorithms, and try to understand which algorithms should be applied in which circumstances.

Machine learning is an exciting interdisciplinary field. Its historical roots are in theoretical computer science, statistics, pattern recognition, and even neuroscience. In the past 15 years, many of these approaches have converged and led to rapid theoretical advances as well as real-world applications.

This course will focus on the methods that have proven valuable and successful in practical applications. The course will also contrast the various methods, with the aim of explaining the situations in which each is most appropriate.

Course Goals

When you have completed this course, you will have an appreciation for the most popular and useful machine learning methods, and should be able to apply these methods to solve learning problems of moderate complexity. You should also be able to read current research papers in machine learning and understand the issues raised by the research.

A major component of the course work will consist of implementing machine learning algorithms. Although the ability to code algorithms from a spec is critical for any programming job, my experience is that students have had little opportunity to do so in first and second year computer science courses. Thus, you will have to learn how to design and organize a programming project from scratch, and to debug complex numerical algorithms.

Prerequisites

You should have junior standing, have completed data structures (CSCI 2270) and Calculus 2, and you should understand basic probability. Knowledge of statistics and linear algebra will be a big help. You should be able to get through the class without this background, but plan on spending extra time to catch up if you don't have the background.

Text

The required text is called *Machine Learning* by Tom Mitchell. Although it is the best and most thorough text in the field, it overlaps only by about 50% with the material I think is important to cover. Consequently, many of the class lectures will be on material not contained in the book. I will hand out supplementary readings for some of the topics not covered in the text. Reading assignments from the text are listed in the syllabus; other reading assignments will be given out as the class dates approach.

Handouts

Course handouts will be brought to class *once*. If you miss a class in which a handout is distributed, you may obtain a copy from the "Course work Handout File" located outside the Computer Science Department main office, room ECOT 717, during regular business hours. I will also try to make postscript copies of the handouts available by ftp from the class web site (see URL at the top of this handout).

Students with Disabilities

If you have specific physical, psychiatric, or learning disabilities and require accommodations, please let me know early in the semester so that your learning needs may be appropriately met. You will need to provide documentation of your disability to the Disability Services Office in Willard 322 (phone 303-492-8671).

Course Work

I expect this to be an intense yet rewarding course. If you do not have a strong interest in the field, I suggest that you take some less challenging course, such as Operating Systems or Basket Weaving. If you have no background in statistics, and if you are bad at math, expect to put extra time into the course.

The course work will consist of two components: homework assignments and a final exam. Grades will be assigned on the following basis:

- 80% on homework assignments
- 15% on final exam
- 5% on class participation

I believe that you can best learn about machine learning by implementing machine learning algorithms and experimenting with them first hand. I could provide you a simulator (an implementation of a machine learning algorithm, wrapped in a graphical user interface), but to really understand the algorithms, I believe that you have to know them well enough to program them and debug them yourselves. Because of this belief, I am focusing on the homework assignments for the course grade. However, an exam is a necessary evil to ensure that you are motivated to attend class and do the assigned readings.

One or more of the homework assignments may involve a competition among participants in the class. I will give everyone the same data for training a machine learning system, and we will determine whose system performs the best on test data.

I expect students to attend all classes. Attendance is important because about half of the material that I will cover is not included in the text. To emphasize the importance of attendance, 5% of your grade will be based on attendance and participation in class.

Final grades will be based on a class curve. I will give each homework and test a numerical score (0-100), and will try to give you an idea of where you lie along the course curve as the course progresses.

Exams

The final exam will cover the entire semester's work, focusing on concepts that we have discussed in class but which have not been emphasized in the homework assignments. The exam will be closed book, but you will be allowed a "cheat sheet"—one 8.5"X11" paper with whatever notes you wish to bring to the exam. The exam will cover material in the text as well as material presented in lectures which is not referred to in the text.

Homework Assignments

The homework assignments will generally involve programming to implement machine learning algorithms from scratch, and experimentation to test your algorithm on some data. Programming can be done in any language (e.g., Java, C++, perl, lisp, matlab). You will be asked to summarize your work in brief (2 page) write ups.

Because the homework assignments comprise the bulk of your grade, *collaboration on the assignments in not allowed*. Each student is responsible for his or her own work. Discussion of homework assignments and programs should be limited to clarification of the assignment hand out itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.

The tentative schedule of assignments is listed in the syllabus. All assignments will count roughly equally toward the final grade.

The tentative schedule of assignment due dates is given in the class-by-class schedule below. This schedule is subject to change as the course progresses and adapts to your interests.

Lecture Date	*Topics*	*Assignments in/out*
Aug 28	Introduction I • Examples of machine learning systems • Learning paradigms • Classification versus regression	HW1 assigned Ch. 1
Aug 30	Introduction II	HW1 due
Sep 4	Introduction III	
Sep 6	Probability theory • Conditional probabilities • Bayes theorem Naive Bayes classifier	Ch 6.2-6.3; 6.9; 8.1-8.2 HW 2 assigned
Sep 11	K-Nearest Neighbor Classifier Overfitting • Model complexity • Bias-variance trade off	
Sep 13	Evaluation • Cross validation • Hypothesis testing	Ch. 5
Sep 18	Decision trees I	Ch. 3
Sep 20	Decision trees II	HW2 due; HW3 assigned
Sep 25	Simple neural networks I • Perceptron algorithm	Ch. 4.1-4.4
Sep 27	Simple neural networks II • LMS algorithm	
Oct 2	Multilayer neural networks • Back propagation	Ch. 4.5-4.7
Oct 4	FALL BREAK	
Oct 9	Neural networks — tricks and variations I	Ch. 4.8 HW3 due; HW4 assigned
Oct 11	Neural networks — tricks and variations II	Ch 6.5; 8.4; 12.3-12.4
Oct 16	*review and catch up day*	
Oct 18	Recurrent neural networks • Application: music composition • Application: time series prediction	
Oct 23	Ensemble techniques I • Mixture of experts • Bagging	
Oct 25	Ensemble techniques II • Boosting	HW4 due; HW5 assigned
Oct 30	Application: data mining in the real world	
Nov 1	Open; discussion of HW5	

Lecture Date	*Topics*	*Assignments in/out*
Nov 6	Unsupervised learning I • K means clustering • topographic maps • mixture models / density estimation	Ch. 6.12 reading (to be announced)
Nov 8	Unsupervised learning II • self-supervised back propagation • generative models	
Nov 13	Reinforcement learning I	Ch. 13
Nov 15	Reinforcement learning II	HW 5 due HW6 assigned
Nov 20	Genetic algorithms	Ch. 9
Nov 22	THANKSGIVING VACATION	
Nov 27	Application: adaptive house	
Nov 29	Hidden Markov models • forward-backward algorithm • application: speech recognition Graphical Models	Ch. 6.11 reading (to be announced)
Dec 4	NIPS CONFERENCE	
Dec 6	NIPS CONFERENCE	
Dec 11	Advanced topics I • Gaussian processes • Support vector machines • Computational learning theory • Models of human learning and adaptation	Ch. 7 (?) HW 6 due
Dec 13	Advanced topics II	