CSCI 4100/6100 RPI
Machine and Computational Learning Fall 2011

Introduction to the theory and applications of *Learning From Data*. The (adaptive) outline is:

1. Foundations of Learning From Data

(i) *The Learning Problem*: Is learning from data possible?
(ii) *Training Versus Testing*: How the learning process will perform on unseen data.
(iii) *The Linear Model*: A simple, yet fundamental, learning model for you to play with.
(iv) *Overfitting*: Why test performance can be much worse than training performance.
(v) *Three Learning Principles*: Occam's razor and falsifiability; Overfitting; Data Snooping.

2. Learning Techniques (we will cover a few)

(i) *Similarity Based Methods*
(ii) *Neural Networks*
(iii) *Support Vector Machines*
(iv) *Aggregation Methods*: methods for combining models.
(v) *Learning Aides*: methods for improving the performance.

3. Other Learning Paradigms

(i) *Reinforcement Learning*

Text: *Learning From Data* (currently in draft form) by Abu-Mostafa, Magdon-Ismail, Lin.
book.caltech.edu (login and password are required and will be assigned to you).

Learning Outcomes. Students entering this course should have a solid mathematical foundation in analysis, probability and linear algebra, in addition to a sound understanding of algorithms, data-structures and programming. This course will build from here to develop an understanding of the limits to which information can be learned from data, and how. In doing so, the student will gain an understanding of basic tradeoffs in learning a model from data, and the general pitfalls one may encounter. The student should leave the course with the ability to implement, as well as critique basic models of learning from data. The student should be able to formulate a learning problem precisely, in terms of inputs and outputs; the student should be able to select a learning model and algorithm; run it on the data; interpret the results and provide some measures of how effective the learning was. The student should know how to use the basic mathematical techniques used in learning from data, which would allow him/her to read and critique recent published literature in the field.

Prerequisites

Mathematics: Calculus at the level of MATH-1020 is a minimum, multivariate is much preferred. Familiarity with probability theory is necessary, preferably at the level of MATP 4600. Familiarity with linear algebra and matrices is also required. Students who have failed to have the mathematical preparation for this course in the past have had serious difficulty, with adverse effects on their grade.

Computing Skills: Ability to program and develop algorithms in some programming language, at a proficiency equivalent to CSCI-2300. Students who have never done a serious programming project may have difficulty in this course. We will *not* be offering "debugging help" in this course.

Policies

Grade

Homeworks will be handed out roughly every week, and will closely follow lectures. Each homework will be worth 1000 points. There will be no final but the last homework is usually handed out on the last day of class, and substitutes for the final. Your worst homework will be counted 50% less than the other homeworks. So, for example, if there are 7 homeworks and your scores are 400,500,600,700,800,900,1000, then your total score is $4700/6500 \approx 72.3\%$ (If all homeworks were counted equally then your score would be 70%).

Your final grade will be based on your homeworks, using separate curves for the 4100 level and the 6100 level courses. Some of the requirements on the homeworks for the 4100 level may be different to the requirements for the 6100 level. Thus, you should read the problem sets *carefully.*

Collaboration and Academic Dishonesty

Discussion is allowed. Copying (from **anywhere** other than the class text) is not. You should write and understand all solutions yourself. Do not destroy any of your code. In the event of strange results, we may ask you to submit your code.

You are expected to treat your work with pride and respect the work of others. In the event that you find someone else's work to be of interest and relevance to any work you will hand in to this course, to the extent that you use their results or techniques, you should: (i) indicate how this third party work was used to solve your tasks, and (ii) achnowledge the original of authors of the work in a bibliography section.

Plagiarizing someone else's work is a **serious issue**. In cases of academic dishonesty, the minimum penalty will be an automatic grade of F, in addition to other institute mandated protocols.

Late Assignments

Assignments are generally due one week after being handed out. Late assignments are penalized 20% per day (except in institute established illness or emergency). Late assignments should be turned into the TA.

Related Reading

LEARNING
Bishop, *Neural Networks for Pattern Recognition*
Mitchell, *Machine Learning*
Sutton&Barto, *Reinforcement Learning*
Vapnik, *Statistical Learning Theory*
Duda & Hart, *Pattern Classification and Scene Analysis*
Haykin, *Neural Networks – A Comprehensive Foundation*
Nilsson, *Mathematical Foundations of Learning Machines*
Ripley, *Pattern Recognition and Neural Networks*

GENERAL MATHEMATICS
Pitman, *Probability*
Degroot, *Probability and Statistics*
Khuri, *Advanced Calculus with Applications in Statistics*
Lang, *Undergraduate Algebra*
Horn and Johnson, *Matrix Analysis*
Golub & Van Loan, *Matrix Computations*