Course Information

Lecturer:	Ronald L. Rivest	NE43–324	3–5880
TA:	Mona Singh	NE43–326	3–7953
Secretary:	Be Hubbard	NE43–322	3–6098

1 Prerequisites

This is a graduate course; undergraduates may attend only with the permission of the instructor.

Students are expected to have some familiarity with basic probability theory (as in 6.041), basic computation theory (as in 6.045/18.400), and introductory artificial intelligence (as in 6.034).

2 Lectures

Lectures will be held in Room 36–839 on Mondays and Wednesdays from 11:00 to 12:30.

3 Textbooks

The required text for this course is *An Introduction to Computational Learning Theory*, by Michael J. Kearns and Umesh V. Vazirani (MIT Press, 1994). Copies can be purchased at the MIT Coop.

We will also hand out lecture notes (see below) and copies of journal articles.

4 Computers and email

We assume that you have access to computers and email. (See the TA if this is a problem.) We may send out notices (e.g. to fix a "buggy" homework problem) over email. You will also need to have computer access to prepare the class notes (see below). There is a small likelihood (10 percent) that one of the homeworks will be a programming assignment.

5 Course notebook

Get yourself a loose-leaf notebook for use with the course. All handouts will be on standard three-hole punched paper. Handouts will be available at the beginning of class. If you miss getting a handout in lecture, you can get a copy in the file cabinet outside room NE43-311. If you take the last copy of a handout, please inform Be in NE43–322 so that more copies can be made.

6 Problem sets

Depending on the size of the class and the extent to which the department supports grading assistance (yet to be determined), problem sets will be assigned on a continuing basis, and will be graded and returned. They will often be handed out on Wednesday, and due the following Wednesday. Late homework will *not* be accepted; if you can't make a class when a problem set is due, turn it in early at the course secretary's office. We will automatically drop (omit) the grades for your three lowest problems (that's problems, not problem sets) in our grading calculation, so you may, if you wish, just not do, or not hand in, three problems of your choice. (From an educational point of view, of course, as opposed to grade-oriented point of view, it is to your advantage to do all of the assigned problems.) As noted above, one of the homework assignments may be a programming exercise, so I assume that you have access to a computer for this purpose.

You may collaborate in study groups on the solution of the homeworks. You must, however, write up solutions on your own. If you do collaborate, acknowledge your collaborators in the write-up for each problem. If you obtain a solution with help (e.g., through library work), acknowledge your source, and write up the solution on your own. Plagiarism and other antiintellectual behavior will be dealt with severely.

7 Lecture Notes

This year, as a new feature of this course, I am asking each of you, in turn, to act as *scribe* for a class and to prepare a set of lecture notes for the lecture you covered.

If the class size is small, you may have to do this more than once. If it is large, you may end up cooperating on just one class's lecture notes. We'll play it by ear.

Your notes must be prepared in Latex. The TA will provide appropriate macros so they will have a common format and style.

These notes will not only be made available to the other students in the class. They will posted on the World-Wide Web. (The whole world will see your notes, so make them good!) The TA will coordinate this posting process.

You should submit your lecture notes electronically to the TA one week after the class you covered.

This is a new procedure for me, so we'll probably have to work out some rough edges at first... More details will be available next week.

8 Term paper

Each student must write a term paper. The topic is your choice, but must relate to the field of machine learning. You may do some experimental work, some theoretical work, or survey the results in some specialized topic, for example. Originality is not mandatory, but is encouraged. Make it interesting.

A topic proposal must be turned in on *November 9th.* Give a two-page description of what you intend to do, with relevant references. This proposal will not be graded, but will be reviewed and returned to you, possibly with comments and suggestions.

The term paper is due on *December 7th.* Late papers will not be accepted without prior permission. Writing style and clarity are important in the evaluation.

9 Grading

Besides the problem sets and the term paper, there will be a mid-term exam (but no final). I expect that the grading will be roughly as follows: term paper 20 %, problem sets 30 %, lecture notes 20 %, mid-term exam 20 %, class participation 10%. This is just an estimate at this time; it may be modified somewhat, depending on how many problem sets there are in the end, the even-ness of preparing lecture notes, etc.

10 Tentative Course Contents

Here is a list of the topics covered last year. This year's class may be fairly similar. Some changes are likely, however...

1. Introduction – why learning? what is it? Models of concept learning; objects and attributes; hypotheses, representations, size, hypothesis spaces, data presentation (e.g. positive only, teacher, types of queries), criteria for learning (Mistake bounds, convergence, pac, ML, MAP).

2. The Halving and Winnow algorithms.

3. The Perceptron Convergence Theorem.

4. VCdim $\leq$ mistakes. Definition of the PAC (Probably Approximately Correct) model.
5. PAC-learning conjunctions. Hardness of learning 3-term DNF.
6. Learning 3-term DNF by CNF.
7. Occam algorithms. Learning k-DL (decision lists).
8. Learning conjunctions with few relevant variables. VC-dimension and PAC-learning.
9. More VC-dimension and PAC-learning.
10. Uniform convergence.
11. Cover's coin problem.
12. Bayesian inference.
13. Neural nets; back-propagation.
14. Examples of neural-net applications.
15. Theoretical results on training neural nets.
16. Learning from queries.
17. Hypothesis boosting.
18. More hypothesis boosting.
19. The SQ (statistical query) model of learning. Learning with noisy data.
20. Poly-size boolean formulae are inherently unpredictable.
21. Separating the PAC and mistake-bound models.
22. Learning finite automata.
23. More learning finite automata.
24. Hidden Markov models.
25. Spectral methods of learning.

11 Background Reading

Some of you may be interested in doing extra reading in areas where your background might be a little light. This section provides some relevant suggestions.

11.1 Theory of Computation

A good general *Scientific American* level book is

Algorithmics: The Spirit of Computing, by David Harel. Addison-Wesley, 1987.

For the theory of NP-complete problems, a good introduction (and a standard reference) is

Computers and Intractability: A Guide to the theory of NP-Completeness by Michael R. Garey and David S. Johnson. Freeman, 1979.

A good text on the theory of computation is

Elements of the Theory of Computation by Harry R. Lewis and Christos H. Papadimitriou. Prentice-Hall, 1981.

11.2 Probability and Statistics

A classic, very readable, introductory text is

> *An Introduction to Probability and its Applications* by William Feller. Wiley, 1968.

11.3 Artificial Intelligence

A good overview of AI, together with an extensive section on learning, can be found in the three volume set

> *The Handbook of Artificial Intelligence*, edited by Avron Barr and Edward A. Feigenbaum. Kaufmann, 1981.

In my opinion, the following encyclopedia contains a higher quality collection of articles, many of which are on learning, etc. However, its encyclopedic format isn't conducive to giving the "big picture". This two volume set is well worth browsing in.

> *Encyclopedia of Artificial Intelligence*, edited by Stuart C. Shapiro. Wiley, 1987.