Probability, Statistics, and Data Analysis for Computer Science

10-751, Fall 2002

Center for Automated Learning and Discovery
Computer Science Department
Carnegie Mellon University

Lectures:	Tues/Thurs 10:30-11:50, NSH 1305
Instructor:	John Lafferty Wean 4111, 268-6791 `lafferty@cs.cmu.edu`
TA:	Guy Lebanon NSH 4522, 268-3951 `lebanon@cs.cmu.edu`
Administrative assistant:	Diane Stidle Wean 4609, 268-1299 `diane@cs.cmu.edu`
WWW:	`http://www.cs.cmu.edu/~10-751`

1 Motivation for the Course

Ideas and tools based on probability, statistics and data analysis are becoming increasingly important in computer science research. In areas such as artificial intelligence and theory, probabilistic methods and ideas based on randomization are central, and in other areas such as networks and systems they are becoming increasingly useful. Research that involves manipulating large data sets is also becoming common, and in all areas of computer science statistical techniques are important in the design and analysis of experimental work.

This course gives an introduction to probability and data analysis from a computer science perspective, including many of the fundamental concepts and techniques that are most relevant to current research areas. The course will put an emphasis on *computing*. Research in CS is fast-paced, and researchers often need to be proficient at manipulating data to draw insights and probe research questions. The course will make use of three general scientific computing environments to help develop the necessary skills: Matlab, R, and the GNU Scientific Libary (more on these below). Topics will include the rudiments of probability and random variables, estimation, special distributions and sampling, Markov processes, hypothesis testing, graphics and visualization, matrix methods and optimization techniques.

2 Who Should (and Shouldn't) Take This Course?

CMU has a large number of students doing research in "machine learning"—a fairly broad umbrella term for work that often involves heavy use of statistical techniques and that has applications from robotics to information retrieval. This course is not intended for such students. The Statistics Department and the Center for Automated Learning and Discovery have a number of excellent course offerings for intermediate and advanced material related to machine learning. If this is your main line of research, you will want a much more complete study than will be provided by this course. In particular, for students requiring a mathematically oriented and in-depth course in probability and mathematical statistics at an intermediate level, Larry Wasserman's courses 36-725 and 36-726 are the ones to take.

This course is intended for students who anticipate a significant component of probability and data analysis in their work, but whose focus is on computer systems or networks, for example, rather than machine learning, and who do not want to take a multi-course, multi-year sequence in probability, statistics, machine learning, statistical graphics, and numerical methods. The present course will attempt to distill many of the most important topics into a one-semester course, while at the same time giving students the necessary skills for applying the concepts in their own research. At the end of the course they should also know how and where to look to learn more. The course is intended for graduate students in the School of Computer Science, but undergraduates are welcome too.

There is a lot of material to pack into one semester, and the pace will be fairly fast. But it is an experimental "service" course; I will be open to suggestions for topics that should be added, deleted, or given greater emphasis.

3 Topics and Outline

The course will be conceptually divided into three parts. The first part will cover the essentials of probability, the second will cover designing and analyzing experiments, including elements of statistical graphics, and the third part will cover matrix methods and optimization. A more detailed outline of the topics follows.

Part I: Elements of Probability

- Sample spaces, events, conditioning
- Random variables
- Basic inequalities
- Special distributions
- Expectations and moments
- Laws of large numbers
- Point estimators and maximum likelihood
- Combinatorial probability
- Markov processes
- The importance of being random; different views of probability

Part II: Designing, Analyzing, and Presenting Experiments

- Standard error and confidence intervals
- Hypothesis testing, p-values
- Sign and rank tests
- Linear and non-parametric regression
- Cross-validation, bootstrap
- Random variate generation
- Synthetic data and controlled experiments
- Graphics and visualization

Part III: Optimization Techniques and Matrix Methods

- Newton's method
- Line search and gradient ascent
- Conjugate gradient
- Linear and quadratic programming
- Eigenvalues and eigenvectors
- Sparse matrix methods
- Matrix decomposition algorithms

4 Reference Texts and Online Course Notes

There is no textbook that covers all of the material of the course, but there are some excellent books that cover basic probability and statistics from a traditional perspective. We will use DeGroot's text for the basic reference:

> Morris H. DeGroot and Mark J. Schervish, *Probability and Statistics*, Third Edition, Addison-Wesley, 2002.

DeGroot was one of the founders of the statistics department at CMU, and a legendary person (...perhaps the CMU Statistics counterpart of people like Newell, Habermann and Simon in Computer Science). This is an excellent book that you won't regret having on your shelf.

A more "modern" and computer science oriented version of an intermediate probability and statistics text is in the works by Larry Wasserman. We will use parts of this as well; see Larry's home page `http://www.stat.cmu.edu/~larry` for pointers to the notes.

There are some other books that might be useful:

> William Feller, *An Introduction to Probability Theory and Its Applications*, Volume I, Third Edition, Wiley, 1950.

Feller's text is a classic that is full of useful information and insights. It is strong on combinatorial aspects of probability.

For references on the topics having to matrix analysis and optimization:

> William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C*, Second Edition, Cambridge University Press, 1992.
>
> Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, Third Edition, The Johns Hopkins University Press, 1996.

Numerical Recipes is kind of the Cliff Notes of numerical computing, but might be useful for the quick read on a technique. Golub and Van Loan is more in-depth, but also not exactly what we need for this course. I'll look for more suitable references.

5 Computing

We will use three computing environments in the course: Matlab, R and the Gnu Scientific Library. Matlab is a general purpose environment for scientific computing that is particularly strong on matrix methods; it is also good for basic graphics. R was developed for statistical computing and is also very good for the kinds of graphics that arise in statistics. One of these might suffice, but the course will introduce both of them so that students are aware of the alternatives for their research.

Both Matlab and R are high-level environments, and although they can be called from external programs, for getting down and dirty with a large system it can be convenient to have numerical libraries that you can call directly, with a cleanly laid out API. The new Gnu Scientific Library can be very useful for such applications.

For documentation and a tutorial on Matlab see

`http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml`

`http://www-2.cs.cmu.edu/~dst/Tutorials/Matlab/`

The entry point to documentation, downloads and information on R is here:

`http://www.R-project.org`

You can run Splus under Andrew, but it is recommended that you simply download R, which is free. I can look into getting facilities to install an SCS-maintained version of R.

And here is the GSL home page on RedHat (which distributes it):

`http://sources.redhat.com/gsl/`

As a side note, Matlab is commercial software, but R and GSL are public domain, under Gnu licenses. R is a public and improved version of the commercial Splus software. You can now buy books with the documentation for R and GSL, but before doing so you might want to consider how quickly these two projects are evolving.

6 Course Requirements

Grading for the course will be based on written assignments and exams. Written problems will be assigned every 1-2 weeks, and will involve conceptual problems as well as computing. There will be three exams in the course, two in class and one take home. The first two exams will cover parts I and II, respectively, and the third will be cumulative. This work will count towards a grade as follows:

Written assignments:	30%
Exam 1 (in class):	20%
Exam 2 (in class):	20%
Exam 3 ("take home"):	30%

Typically one week will be given for completing a written assignment. No makeup exams will be given—the dates will be posted early in the semester, so please make sure that you are in class on those dates.

All written work must be your own. You may discuss problems with others, but if you receive significant help from someone, or from some other source, it should be fully acknowledged on the write up that you pass in. If there is any question on policy for the course, do not hesitate to contact me.

7 Office Hours

To set up a meeting with John Lafferty, send email to `lafferty@cs.cmu.edu` or call 8-6791. Guy Lebanon will hold regular office hours, time and place TBA.