OVERVIEW OF CPSC 531F: TOPICS IN THE THEORY OF COMPUTATION: APPLICATIONS OF LINEAR ALGEBRA

JOEL FRIEDMAN

Disclaimer: The material may sketchy and/or contain errors, which I will elaborate upon and/or correct in class. For those not in CPSC 531F: use this material at your own risk...

Overview: The goal of this course is to give a number of applications of linear algebra to computer science, and to give concrete examples and some research problems regarding such applications. The choice of topics covered will depend on the interests of the students.

The first part of this course will review matrix theory, specifically eigenvalues and eigenvectors, to apply to matrices of large dimensions that represent graphs, Markov chains, data sets, etc. When these systems are large, it may be difficult to answer some basic questions regarding these systems, e.g., (1) does the graph contain "large clusters" or is it a "good expander," (2) does the Markov chain "mix quickly," (3) can one "reduce the dimension" of the data set, etc. At times one can approximately answer such questions when one knows the largest (or otherwise most relevant) few eigenvalues and eigenvectors of an associated matrix.

The second part of this course will give more details on a few applications. We will discuss "expanders," which are sparse graphs will good "expansion" or connectivity properties. These graph can be used to build networks with certain desirable properities, to construct error-correcting codes, to boost randomness, etc. We will also discuss examples of Markov chains, with special interest in reversible chains (whose underlying matrix is symmetric after scaling); this includes many "Monte Carlo" type chains.

We will briefly discuss other applications of linear algebra, especially those that are important and use related foundations of the matrix theory we use.

Tentative Outline of Course: The order may change, and the topics may change according to student interests. Here is the tentative outline.

Roughly 5 weeks: Examples of applications of linear algebra and needed theory. Most of these relate to symmetric matrices in some way; a few of these topics don't relate to the central theme of this course but are "too good" to omit.

(1) Symmetric matrices in graph theory: adjacency matrices.

Research supported in part by an NSERC grant.

(2) Review of linear algebra and examples: diagonalizable and nondiagonalizable matrices, similarity, biorthogonal decompositions (i.e., right and left eigenvectors).
(3) Positive semidefinite matrices in graph theory: Laplacians. Rayleigh quotients and extreme cases of "spectral clustering."
(4) Symmetric matrices in the SVD (singular value decomposition); applications to data compression, dimension reduction, and PCA (principal component analysis), "eigenfaces."
(5) Symmetric matrices in projections and the normal equations, FA (factor analysis).
(6) Markov matrices and biorthogonal decomposition (when a square matrix is not symmetric but you want a related technique); examples of slowly and quickly mixing chains, PageRank, "Monte Carlo" methods.
(7) Reversible Markov chains (which are "symmetric with respect to the appropriate inner product): theory, random walks on graphs, "Monte Carlo" methods, the "Metropolis" algorithm, simulated annealing.
(8) Brief mention of unitary matrices in quantum mechanics and computations.
(9) Possible brief mention of other common applications in computer science: linear error-correcting codes, linear and semidefinite programming, etc.

Roughly 8 weeks: Some specific examples studied in more detail. Topics may include the following.

(1) Expanders: regular graphs with one large eigenvalue and all other eigenvalues small. Applications: expansion, expander codes, randomness boosting, etc. Specific constructions of expanders, including the LPS-M Ramanujan graphs and Alon's combinations of these graphs. Open problems in the construction of expanders; the girth of regular graphs and the non-backtracking matrix.
(2) More on Markov chains and reversible chains.
(3) Relativization, block matrices, group actions, and Markov chain refinement: a way to obtain new graphs (matrices, Markov chains, etc.) from a given graph (matrix, etc.,) and to study a given graph (matrix, etc.) with symmetries.
(4) Very specific applications: (provably) detecting clusters, solving special cases of the Unique Games Conjecture, aspects of the PageRank algorithm.
(5) Other aspects of Markov chains and covering times (here eigenvalues don't give you sufficiently precise information).

Prerequisities: I assume you have had at least one term of undergraduate linear algebra. Ideally you will have had some linear algebra beyond this, either theoretical or practical; I will review many applications of matricies and give some material taught typically in a second-term course in linear algebra.

Grading: The grade will be based on problems that are assigned during class and arranged into three sets. In lieu of the last homework set, students may write short essay on an application of linear algebra of their choosing.

Grades in this course have the following meaning (this is based on recent Math Department guidelines for topics courses):

95% or higher (A+): student is strongly encouraged to pursue research in the area; the instructor will write a very strong letter of recommendation for the student.

90% or higher (A+): the student is encouraged to pursue research in the area or a related one; the instructor will write an enthusiastic letter of recommendation for the student.

85% to 89% (A): the student will be able to use some of the course material in another field of research; research in the area is not encouraged without significantly more background and study.

80% to 84% (A-): the student has fulfilled department expectations of the graduate student.

79% or below (B+ or below): the student has not fulfilled department expectations of the graduate student. For PhD students, any grade of 67below is failing. For MSc students, a C or C+ (60–67%) is technically a passing grade; any grade of 59% or below is failing.

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA.
E-mail address: jf@cs.ubc.ca
URL: http://www.cs.ubc.ca/~jf