Course Syllabus

COMS W3251 - Computational Linear Algebra

Course Description

This course is an introduction to linear algebra with an emphasis on computational applications. The study of linear equations, linear functions, and their representations pervades numerous fields of study. Students will learn and practice fundamental ideas of linear algebra and simultaneously be exposed to and work with real-world applications of these ideas. As the complementary course to discrete mathematics (COMS 3203), this course will continue emphasizing a rigorous approach to mathematics, which will serve as a foundation for future courses like computer graphics, machine learning, and robotics. The learning and usage of Python and libraries such as NumPy is an essential component of the course, as is the development of basic skills of computational programming.

Course Objectives

- Learn how to operate with vectors and matrices and use them to solve systems of linear equations;
- Understand properties of vector spaces and related concepts such as basis, dimension, orthogonality;
- Analyze linear transformations, both from an algebraic and geometric perspective;
- Learn how to diagonalize matrices by finding their eigenvalues and eigenvectors;
- Study different types of decompositions, such as Jordan decomposition, SVD and QR;
- Apply these concepts to different branches of mathematics and computer science, such as dynamical systems, Markov chains, least square regression and principal component analysis;

- Demonstrate proficiency in Python for elementary programming tasks;
- Demonstrate proficiency in NumPy and related libraries for scientific computational tasks involving real data.

Prerequisites

- Calculus I; Calculus II or Calculus III or equivalent strongly recommended
- Programming experience: COMS 1004 or COMS 1007 or equivalent.

Textbooks

The main textbooks of this class will be Lay, Lay and MacDonald, Fifth edition (LLM), along with an online textbook by Boyd and Vandeberghe (BV), serving as the primary reference for the computational aspects. Another reference for a more theoretical perspective is Lax, Second Edition (LAX). All readings will be posted on the course schedule and will be complemented by lecture notes of the instructor on the topic.

Course Staff

Instructor:

Francesco Preta <fp2428@columbia.edu>
Office hours: Thursdays 11:30 AM-12:30 on Zoom

https://nyu.zoom.us/j/4513541210

Meeting ID: 451 354 1210

Instructional Assistants (office hours):

Ryan Chen <ryan.chen@columbia.edu>
Office hours: Mondays 9-10 AM on Zoom (link TBA)

In terms of contacting the course staff, your first venue should always be Piazza for non-urgent questions; you can also choose to ask a private question to the TA there if necessary. Attendance is optional but encouraged.

Lectures

We will not explicitly take attendance for lectures, but of course we believe that you will get the most out of this class if you attend and keep up regularly.

Time: Tuesdays & Thursdays 1:00pm - 4:10pm
Zoom room:

Join Zoom Meeting

https://columbiauniversity.zoom.us/j/95699065258?pwd=Tkd2SzVkdFJWKzVTcS9RMFhlQm9MQT09

Meeting ID: 956 9906 5258

Password: 169006

Grading

Homework	30%
Labs	25%
Exams	40%
Participation (in class and on Piazza)	5%

Homework

Written homeworks are assigned weekly and consist of several problems intended for problem-solving practice with concepts.

These are to be completed individually and will be graded for correctness. Supplementary problems may be included as well; these are recommended for exam studying but not required to be turned in. The required problems are to be submitted via Canvas; LaTeX typesetting is highly recommended.

Labs
Throughout the course there will be several lab assignments that will take up a couple weeks at a time. Each lab revolves around one or two applications of concepts from lecture. These will involve both programming (using Python in Jupyter Notebook) and analysis of the problems at hand.

Late Policy
Late homeworks are not accepted. Labs may be turned in up to 24 hours late for a 50% deduction.

Exams
There are two exams: one midterm and one final. They will be "take-home" exams to be returned in 24 hours, including all the topics covered up until that point. The midterm will be released on Thursday July 23th at 5:00 PM, while the final will be released on Thursday August 13th at 5:00 PM. The midterm is nominally worth 15% of the overall grade; the final is nominally worth 25%.

Regrading
Regrading for both homework and exams will be considered on a case by case basis, all handled through Canvas. For both cases, all regrade requests should be made within one week of receiving feedback. Regrades must be justified; extraneous requests, for example those in clear contradiction to posted solutions, can result in a lower grade.

Participation and Piazza
Your class participation will be evaluated on a combination of class participation and general participation on Piazza. In evaluating participation we will be understanding of the time

differences, but we encourage nonetheless to participate to discussions on Piazza. The ideal scenario is to have had enough interaction with either the TA or the instructor by the end of the session so that we know who you are.

Please sign up to Piazza here.

Academic Conduct and Integrity

Any form of academic misconduct will result in a homework or exam grade of zero and be reported to Student Conduct and Community Standards (SCCS). In addition to these policy, the CS department's academic honesty policy applies to this course.

Collaboration: We encourage you to discuss and brainstorm both ideas and your assignments together. However, all homework assignments must be written up and coded individually. In your submissions, you must note all collaborators that worked with you throughout the process. Failure to do so will automatically be considered an academic violation.

Online Material: As with written problems, you are not permitted to copy any part of other people's work without attribution. This applies to code produced by other students and to material found on the internet. Sometimes online sources (e.g., Stack Overflow) can be useful as a reference. If you have to use code snippets found online you must attribute your source in a comment (complete link). You are not allowed to copy non-trivial code fragments from these sources.

Non-trivial code is defined as

- Any code you do not fully understand.
- Code longer than three lines or complete class or method definitions that directly relate to the homework problem.

Example: You may copy a one-liner that opens and reads from a text file, if you attribute your source, but it is not okay to copy an entire method for inserting into a balanced search tree.

Inclusive Class Climate: We strive to maintain a class climate that fosters a sense of belonging for *all* students. Keep in mind that you will inevitably interact with students and staff of all backgrounds. If you see or feel any incident that makes you or someone else uncomfortable, report it right away. Any intentional discriminatory or harassing behaviors will be dealt with accordingly with SCCS. Sometimes incidents may also be unintentional--we are all human with our own personal biases after all. These may also be brought up for general discussion if appropriate, especially when such questions are relevant for our course.