A Freshman Level Computational Linear Algebra Course

Laura Chowdhury

Cal State Fullerton

September 25th, 2020

Overview

- Motivation for course
- Course format
- Order of class with examples
- Lessons learned

Math 107 - Computational Linear Algebra

- Math 107 is designed to be an introduction to programming and linear algebra (primarily basic matrix computations) with applications
- Currently, math majors take Math 320 (Introduction to Mathematical Computing) or CPSC 120/121
- Math 320 is typically taken during the junior year
- The department felt that students needed to see more programming and see it earlier in their major

Math 107 - Student Learning Objectives

At the end of this course, students should have:

- An understanding of the algebraic and geometric interpretations of matrices, matrix arithmetic, vectors, and complex numbers.
- The ability to solve systems of linear equations using row-reduction techniques by hand or with the aid of a computer algebra system.
- An understanding of the basics of the eigenvalues and eigenvectors of a matrix, as well as some of their applications in various mathematical concentrations.
- Introductory-level programming skills. In particular, they should be able to model linear algebra concepts (e.g., matrix arithmetic, vectors, plotting) as well as understand the basics of loops, conditional statements, functions, and how to translate a real-world problem into a mathematical framework.
- An example base of applications of matrices and linear algebra to real-world problems.

Course Format

- Daily pre-class assignments (Ticket-in-the-door) that consist of reading and answering questions or completing a small programming task
- In class, there might be some review or expansion of the material
- Most of class is spent with class exercises with students working in groups of 2-3

Textbooks

- *When Life Is Linear: From Computer Graphics to Bracketology* by Tim Chartier
  - Introduces linear algebra concepts through applications, starting with an image being represented by a matrix
  - $34.06 on Amazon
- Main Text – Combined text of *Fundamentals of Matrix Algebra* by Gregory Hartman and *An Introduction to Matlab* by Troy Siemers
  - An additional chapter on complex numbers has been added
  - Free PDF for the students

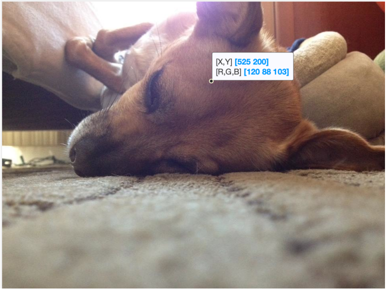
Flow for the course

- Follow the *When Life Is Linear: From Computer Graphics to Bracketology* text, introducing the necessary programming constructs and linear algebra as needed

Order of Topics

- Introduction to MATLAB
  - Compiling code, editor, vectors, matrices
- Matrix components
  - images
- Matrix arithmetic
  - Adding images - convex combination
  - Subtracting images - edge detection
- MATLAB Built-in functions
  - find function - create binary images
  - mean, max, min, size, histogram
  - plotting
- Norms (1- and 2- norms)
  - Movie ratings and recommendations
  - MNIST Handwritten number dataset

Images


[X,Y] [525 200]
[R,G,B] [120 88 103]


[X,Y] [525 200]
Index 99
[R,G,B] [0.3882 0.3882 0.3882]

Images as Matrices

D_gray

720x960 uint8

	1	2	3	4	5	6	7	8	9	10	11
1	159	160	161	162	164	165	166	167	166	170	173
2	160	160	161	162	163	165	166	167	169	172	172
3	161	161	160	160	162	163	166	167	173	174	174
4	163	162	160	160	161	162	164	166	173	175	176
5	163	162	161	160	161	161	163	164	167	171	175
6	162	162	162	162	161	161	161	161	161	165	170
7	160	161	163	163	163	161	159	158	163	164	167
8	160	161	163	165	164	161	158	156	170	168	167
9	159	163	172	179	177	170	167	169	167	172	175
10	169	170	173	177	174	167	166	169	163	164	164
11	158	157	159	163	163	162	165	171	169	167	165
12	150	148	149	154	156	156	159	165	174	173	172
13	160	157	157	159	159	154	152	154	163	168	171
14	158	155	156	161	162	158	154	154	153	160	168
15	150	147	148	155	160	159	158	159	154	159	165
16	158	152	150	155	158	157	156	156	159	159	160
17	146	150	155	155	153	151	152	153	160	159	159
18	153	152	149	146	145	148	153	158	160	159	158
19	159	154	148	145	147	151	156	158	162	160	158
20	159	155	153	153	156	158	157	155	162	160	157

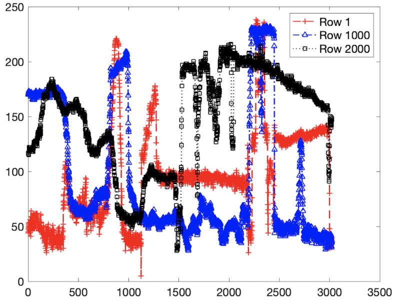
Adding Images

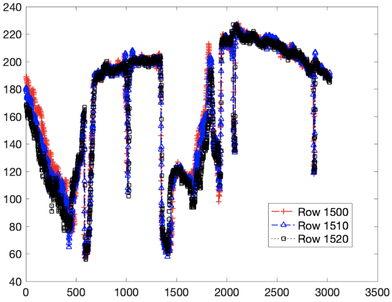
Luka

Redwoods

Luka in the Redwoods

Plotting Rows

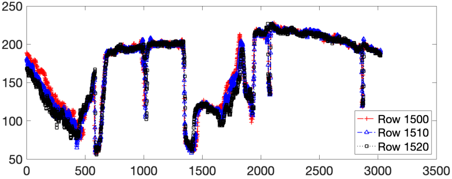


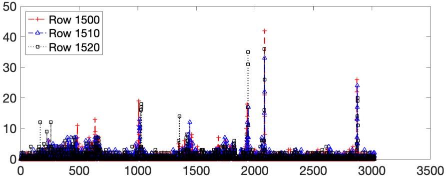


Rows far apart,

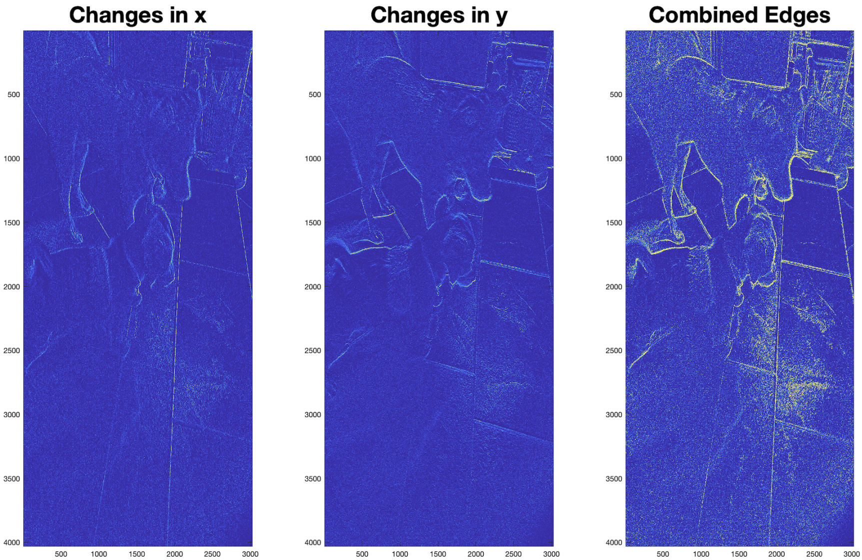
Rows close together

Subtracting Rows to See Edges





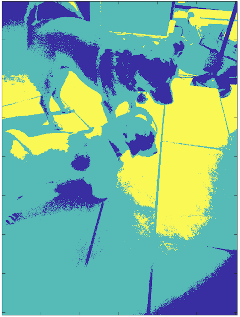
Subtracting Matrices to See Edges



Using Built-In Functions (find, mean, std)

Original,

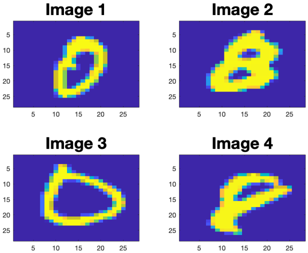
Above/Below Mean,



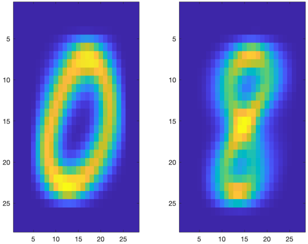
Above/Below/Within

One Standard Deviation

MNIST Handwritten Numbers



4 Digits from Database



Mean of 0 and 8

Order of Topics (continued)

- Introduction of vectors (2D and 3D)
- Complex numbers
- User-defined functions
- Conditional statements
- Loops
- Matrix multiplication
- Transformations
- Fractals

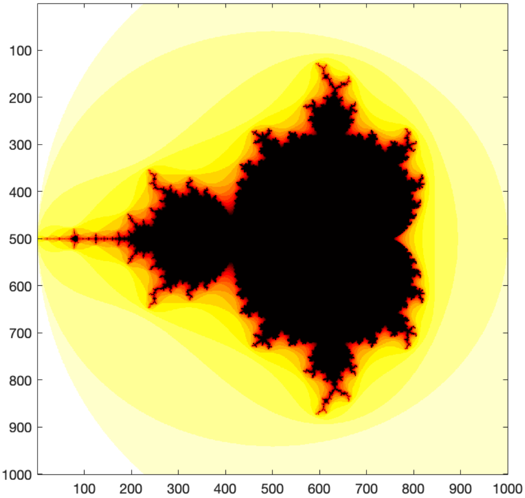
Fractals - Mandelbrot Set

```
%% Steps:
%   1.  Let x=1 and y=1, and let c = x + 1i*y (a complex number in MATLAB).
%       Let z=0 (giving z_0 = 0).  Generate the first Q=20 terms of the
%       sequence (z_n+1 = z_n^2 + c), saving each term as z, overwriting
%       the previous z at each step.

%   2.  Test to see if |z|>2 at each iteration (i.e., |z_n|>2 for n=0,1,2,..,Q).
%       If it is, stop the loop and let A = Q-n using the first n such that
%       |z_n|>2.  If it gets to Q without exceeding 2, let A=0.

%   3.  Now let x and y vary.  Let N=1000, M=1000, and let
%       x = linspace(-2,1,N); y = linspace(-1.5, 1.5, M);
%       Let A = zeros(M,N).

%   4.  Loop through y: for j=1:M
%       Loop through x: for k=1:N
%       Change x to x(k), change y to y(j), change A to A(j,k).

%   5.  Plot the results using: figure, imagesc, colormap hot, axis square
```

Fractals - Mandelbrot Set

```
% Define grid size and maximum iterations
M = 1000, N = 1000, Q = 20;
x = linspace(-2,1,N), y = linspace(-1.5, 1.5, M);
A = zeros(length(y), length(x));

for j=1:M  % Loop through rows (y values (Im))
    for k=1:N % Loop through columns (x values (Re))
        z = 0; % z_0
        n = 1; % Initialize counter to 1

        while n <= Q % Loop until point escapes or tol is reached
            c = x(k) + 1i*y(j); % Make the complex number
            z = z^2+c; % Iterate z_n+1 = z_n^2 + c

            if abs(z)>2 % If modulus > than 2, it escaped
                A(j,k) = Q-n; % Give color based off escape speed
                n = Q+1; % Terminate the while loop
            end

            n = n+1; % Update n
        end
    end
end
figure, imagesc(A), colormap hot, axis square % Plot the result
```

Fractals - Mandelbrot Set



Order of Topics

- Systems of Linear Equations
  - Gaussian elimination
  - Existence and uniqueness of solutions
- Matrix inverse and transpose
  - Lights Out Game
  - Numerical error in computing inverses
- Least-squares line
- Eigenvalues & eigenvectors and applications
  - Adjacency matrix - graph clustering
  - SVD - image compression
  - Transition matrices - Google's PageRank algorithm
  - Sports Rankings

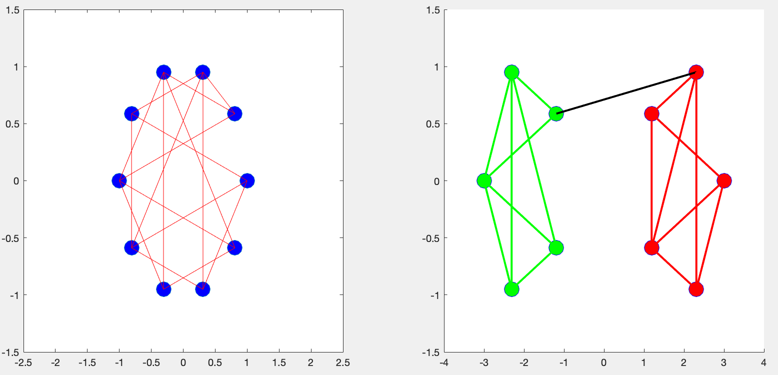
Lights Out Game

https://tinyurl.com/txqhkke

- Construct the system of linear equations to represent the game
- Solve using RREF mod 2

Adjacency Matrix - Graph Clustering

- Construct the adjacency matrix, A
- Construct the degree matrix, D
- Find the graph Laplacian, $L = D - A$
- Find the Fieldler vector - eigenvector associated with 2nd smallest eigenvalue
- Partition the nodes based on the sign of the Fiedler vector



Consider a graph where nodes are classes in the mathematics major and edges indicate a direct prerequisite relationship.

The eigenvector centrality is a measure for the importance of a node in a network. It is found by looking at the eigenvector corresponding to the largest eigenvalue of the adjacency matrix.

The nodes can then be ranked in order of importance based on the highest to lowest absolute values in the vector. Find this eigenvector for the system. Identify the two most important nodes in the system and its corresponding course. How are these related to Math 107?

Sample Homework Question

Adjacency matrix for prerequisites

Consider a graph where nodes are classes in the mathematics major and edges indicate a direct prerequisite relationship.

The eigenvector centrality is a measure for the importance of a node in a network. It is found by looking at the eigenvector corresponding to the largest eigenvalue of the adjacency matrix.

The nodes can then be ranked in order of importance based on the highest to lowest absolute values in the vector. Find this eigenvector for the system. Identify the two most important nodes in the system and its corresponding course. How are these related to Math 107?

Solution:

- **250B (Intro. to Linear Algebra & Differential Equations)**
- **307 (Linear Algebra)**

The Wiki Game - a game based off clicking links on Wikipedia pages to get from one specific page to a target page (e.g., Epic poetry to Bacteria)

https://www.thewikigame.com

PageRank Algorithm

- Create a transition matrix for a network
- Add in ability to teleport
- Find an eigenvector that gives the order of importance of websites

Sports Rankings

- Use the Wiki Game to have students compete as teams
- Massey Method: Construct a linear system whose solution gives the ranking
- Eigenvector Method: Construct a matrix whose eigenvector gives the ranking

Sample Exam Question:

Write a function that takes as input a matrix A and a tolerance p, and outputs k. Here, k is the number of columns and/or rows to include for U_k, S_k, and V_k of the SVD. This k is determined to be the smallest k such that

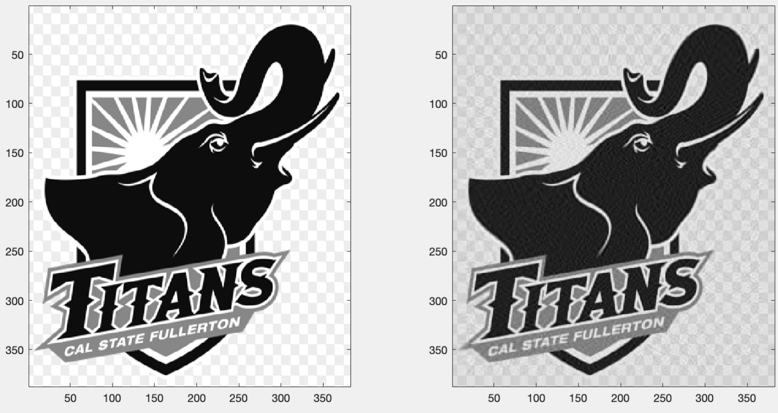
$$||A - U_k S_k V_k^T||_2 < p.$$

Use a while-loop for this function.

SVD – Image Compression (45.8%)

Original image: 387x382 (147,834 numbers to store)

Compressed image: 88 columns of U, V, and 88 singular values of S (67,760 numbers to store)



Lessons Learned

- Students will start the work the day before, no matter how frequently you tell them to start early
- Many students don't know how to download files and save them to specific folders
- Students don't know how to take notes for this type of class (mix of programming and writing by hand)
- Students don't know how to save a file with an informative title or with informative comments
- Using comments correctly (either code is commented out or comments are not commented out)
- "Plot y versus x" is not something students know
- Providing the students with named files, pre-commented text, and questions embedded drastically improved course flow and student notes

Questions?

Stay tuned in October for more of Fullerton's new lower division courses!

Laura Smith Chowdhury
lausmith@fullerton.edu