Review for Final Exam

CS 111: Introduction to Computational Science

Spring 2019 Lecture #17

Ziad Matni, Ph.D.

When both your parents are software developers! ☺

Final Exam

- In this classroom on **Wednesday, June 12th at 12:00 PM**
- Arrive 10-15 minutes early
- Pre-arranged seating

- Material to study: Everything!
- Allowed in the exam:
  1 sheet of paper (8.5"x11") both sides ok

Topics: Basic Linear Algebra Concepts

You have to know how to do the matrix operations in math AND in Python code

- Matrix multiplication
- Matrix rank, transpose
- Generating random matrices
- Solving **x** in **Ax** = **b**

Topics: Interesting Matrices

- Identity M
- Diagonal M
- Permutation M
- Invertible Ms
- SPD Ms
- Orthonormal Ms
- Sparse vs. Dense Ms
- Diagonally dominant Ms

Topic: Matrix Factorization

- Gaussian Elimination
- LU factorization
  - Using pivots vs. not using pivots
- Cholesky factorization
- QR factorization
- Eigenvalue factorization

Topic: EigenVs, Stability

- Eigenvalues & Eigenvectors
  - How to calculate both
  - How to tell if a vector is an eigenvector

- Matrix stability
  - Forward vs backward error
  - The condition number, how to calculate it, and what it means

The Temperature Problem

- Recognize it as an example of solving a PDE
- Know how to set it up and how to find a solution

Specific $Ax = b$ Solvers/Methods

- Jacobi Method
- Conjugate Gradient Method
- Least Squares Method

IEEE Floating Point Protocol

- What it is (16b, 32b, 64b versions)
- How do we find it specifically (32b and 64b CPUs)
- Decimal ←→ Binary ←→ Hexadecimal conversions
- Machine epsilon

PageRank

- Network Analysis
  - What is it for? Where is it used?
- Nodal Analysis: Centrality Measures
  - In/Out-degree vs. Eigenvector cent., betweenness cent.
- What is PageRank for?
- Adjacency matrix
- Link matrix
- Markov matrix
- Iterative Power method to find the PageRank solution
- Process starting with a graph and ending with a ranked node vector solution

ODEs

- What's an ODE? What's a PDE?
- Solving ODEs by hand
- System of ODEs
- Using **solve_ivp()** (Runge-Kutta, Radau Methods)
- Using **ode1()**, **ode2()** (Euler Methods)
- Non-Linear ODEs (Lotka-Volterra Eqs)
- Stiffness Problem in ODEs

Re: Python Code
What do you Need to Know…

- How to code for graphs?
  - Yes! You need to know how to set these up!

- What import libraries go with what functions?
  - Yes!

- The full definition code of functions we've used?
  e.g. `LUFactor()`, `Lsolve()`, `Usolve()`, `Jsolve()`, etc…
  - No! But know how they work and how to use them correctly (functionally & also syntactically)

Example Question

Let x , p , A , and I be defined by the following **numpy** statements:

- `x = np.array([12, 34, 56])`
- `p = np.array([1, 2, 0])`
- `A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`
- `I = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])`

What are:

a) `x[p]`

b) `A @ p`

c) `x[::-1]`

d) `A @ I`

e) `A * I`

Answers:

a) [34, 56, 12]

b) $[5,\ 14,\ 23]^T$

c) [56, 34, 12]

d) The matrix A

e) A diag. matrix with **1, 5, 9** as diagonal values

Example Question

If we can do a Cholesky factorization on the matrix **A**, calculate it.

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{4} & \mathbf{3} \end{bmatrix}$$

- First of all, check if A is SPD:
  - $\lambda 1 + \lambda 2 = 4$ and $\lambda 1 . \lambda 2 = 3 - 8 = -5$
  - So, $\lambda 1 = 5$ and $\lambda 2 = -1$

- A is NOT SPD, so we cannot do a Cholesky factorization!

Example Question

If we can do a Cholesky factorization on the matrix **A**, calculate it.

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{4} & \mathbf{3} \end{bmatrix}$$

- First of all, check if A is SPD
- A is NOT SPD, so we cannot do a Cholesky factorization!

Example Question

When doing a QR factorization on the matrix **A**, we got **Q** as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix}$$

Calculate the matrix **R**. Is it in any special form that you expected?

- $R = Q^T A = \begin{bmatrix} -1 & -2 \\ 0 & -1 \end{bmatrix}$

It is in an upper-triangle form, which is expected.

Example Question

a) Given 3 nodes in a network that are all **fully** connected to one another, write the adjacency matrix **E**.

b) What does the following code do?

c) What does the matrix **M** represent?

$$E = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

```
n = 3
outdegree = np.sum(E,0)
A = E / outdegree
print(A)
S = np.ones((n,n)) / n
m = 0.15
print(S)
print(m * S)
M = (1 - m) * A + m * S
print(M)
```

It prints matrices A, S, m*S, and M as:

$$A = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 \end{bmatrix}$$

$$m*S = \begin{bmatrix} 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \end{bmatrix} \quad M = \begin{bmatrix} 0.05 & 0.475 & 0.475 \\ 0.475 & 0.05 & 0.475 \\ 0.475 & 0.475 & 0.05 \end{bmatrix}$$

M is the Markov Matrix (a.k.a. the PageRank Matrix) for the network described here. It shows the probability of transition from one node to another.

Example Question

Express the following ODE problem in the standard matrix form of **X' = A X**

$$2x'' = x - x'$$

- Let $y = x'$
- So, $y' = x'' = 0.5(x - x')$

- Our system of 2 equations are:
  $x' = y$
  $y' = 0.5x - 0.5y$

- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

BEST OF LUCK ON ALL YOUR FINAL EXAMS!

GO TO SECTION TODAY!

STUDY HARD!

(THEN REST HARD!)

</LECTURE>