Linear Algebra

Due at 11:59pm ET on 6 April 2023

What you need to get

- `YOU_a2d-q1q2q3.ipynb`: Jupyter notebook for Q1, Q2, and Q3

If you want to typeset your solutions to Q4, Q5 and Q6 in LaTeX, then you're welcome to use:

- **https://www.overleaf.com/read/vztpkwrpmtcn**: (optional) Overleaf template

What to do

Task 1: Google Page Rank

The objective of this task is to develop Python code that computes the Page Rank of a set of web pages based on the network adjacency graph. The adjacency graph is represented by a matrix G, where

$$G_{ij} = \begin{cases} 1 & \text{if } \exists \text{ a link from } j \text{ to } i \\ 0 & \text{otherwise} \end{cases}$$

1. [10 marks] Complete the Python function

   ```
   y = SparseMatMult(G, x)
   ```

   which multiplies the sparse matrix `G` by the vector `x`. The matrix is stored using the *dictionary-of-keys* method. You can get the key-value pairs using the built-in call `G.nonzero()`. See the function's documentation for full details.

2. [15 marks] Complete the Python function

   ```
   p, iters = PageRank(G, alpha)
   ```

   that finds the steady-state solution (eigenvector for $\lambda = 1$) of the Page Rank problem using the iterative method discussed in class. The output `p` is an $R \times 1$ vector (where R is the number of nodes) containing the node scores, and `iters` is the number of iterations the method took to converge. Your method should use your implementation of `SparseMatMult` from question 1, and must **not** form a full $R \times R$ matrix at any time (even as an intermediate while evaluating an expression). To input the initial adjacency matrix, you can use `dok_matrix` from the `scipy.sparse` module (see class demonstration `Randy_demo`). Your iterations should start with a uniform distribution in `p`, and terminate once the solution is found to within a tolerance of 10^{-8} (ie. none of the elements in `p` changes by more than the tolerance).

Task 2: Illegal Trading Network

Here is the next forensic problem in the criminal case.

3. [15 marks] The employee of First National Bank that wrote the code for `CalculateNet` is named Dennis Reader. After interrogating Dennis Reader, evidence of an underground crime ring was uncovered. The police plea-bargained with Dennis, and he agreed to inform them about the illegal trading of contraband occurring in a black market. The police want to arrest the most influential criminal in the trading network and replace him or her with an undercover police officer. This officer will then use the criminal's accounts to infiltrate the trading network, and find out more about the identities of the other criminals.

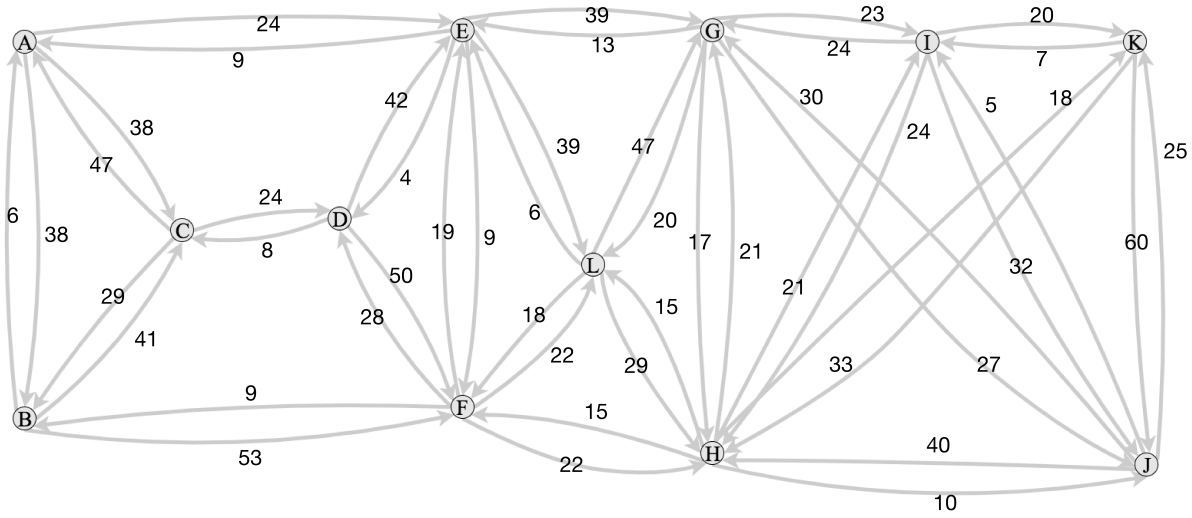
The information Dennis has provided is shown in the diagram below, where each member of the network is represented by a letter. For each member, the arrows pointing away from the node show the percentage of their sales that go along that channel. To find the most influential criminal in the ring, do the following:

(a) Add lines of Python code to the notebook to create a sparse matrix representing the network. You should use the `scipy.sparse` module.

(b) Add lines to the notebook that run your `PageRank` function (from question 2) on the network with $\alpha = 1$, creating a bar plot (or stem plot) of the node scores to determine the most influential 'node'. As with any plot, make sure it is labelled appropriately.

(c) Write a brief note to the police indicating which node in the trading network is the most influential.

Note: You may assume that the connection matrix has only one eigenvalue of 1. The Markov matrix is not a *positive* matrix, but it is *aperiodic*, which is a different but related property that gives us the same uniqueness guarantee on the largest eigenvalue.



Task 3: LU Factorization

4. [20 marks] Consider the system of equations:

$$\begin{aligned} -24x_0 + 12x_1 + 36x_2 - 12x_3 &= 36 \\ -12x_0 + 30x_1 - 30x_2 - 18x_3 &= -18 \\ -12x_0 - 2x_1 + 40x_2 + 22x_3 &= 18 \\ 6x_0 - 15x_1 + 3x_2 + 33x_3 &= -39 \end{aligned}$$

(a) Write the coefficient (system) matrix A and the right-hand-side vector b so that $Ax = b$.

(b) By manual calculation (showing your work), compute the LU factorization (with row pivoting) of the system matrix in part (a). That is, find a permutation matrix P, a unit-diagonal, lower-triangular matrix L, and an upper-triangular matrix U such that $PA = LU$.

(c) Solve the system manually using the LU factorization above. Show your work.

5. [15 marks] Suppose you are given an $N \times N$ matrix A and its LU factorization $LU = PA$ where L is lower-triangular and unit-diagonal, U is upper-triangular, and P is a permutation matrix. How

can you use that factorization to solve the system

$$xA = b$$

in $\mathcal{O}(N^2)$ flops, where x and b are both $1 \times N$ row vectors? You may refer to algorithms discussed in lectures, but be specific about which algorithm you would use, and when. This is not a programming question – you do not need to write Python code.

6. [25 marks] Consider the linear system of equations

$$\begin{bmatrix} 4 & -4.2 & 0 \\ -2 & 2.099 & 7 \\ 4 & -1.7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4.2 \\ 4.901 \\ 5.7 \end{bmatrix} . \tag{1}$$

   (a) Evaluate an accurate solution to (1). You may use Python's `numpy.linalg.solve` function. You do not have to show how you got the solution, but write the solution in your answer.

   (b) Simulating 5-significant-digit rounding by hand, solve (1) using Gaussian elimination **without row pivoting**, followed by back substitution. Show your work.

   (c) Simulating 5-significant-digit rounding by hand, solve (1) using Gaussian elimination **with row pivoting**, followed by back substitution. Show your work.

   (d) Which method, (b) or (c), is more accurate?

What to submit

You must submit a series of PDF documents to Crowdmark. Each coding question should be in a PDF export of a jupyter notebook (sometimes several questions in a single notebook). When a proof or manual calculation is requested, you can typeset your solution using LaTeX or Word, or handwrite them; these solutions should also be submitted as PDFs. **Photographs or scans of handwritten solutions should be legible, otherwise, the TAs might deduct marks.**

Submit the following PDF documents:

1. A **single** PDF of the notebook containing solutions for all of Q1, Q2, and Q3
2. A PDF for Q4
3. A PDF for Q5
4. A PDF for Q6

Finally, upload your Python notebooks on Learn dropbox.