CS 111: Homework 8: Due by 11:59 pm Wednesday, March 11, 2020

Submit your paper as one PDF file, and tell GradeScope which page(s) each problem is on.

1. This problem is about another connection between graphs, matrices, and eigenvalues. This time the graph in question is undirected, with no arrows on its edges. Let G be an undirected graph with n vertices, which we take to be the integers 0 through $n-1$. An edge of G is an unordered pair of integers (i, j). We assume that G has no multiple edges (that is, edge (i, j) only occurs once) and no loops (that is, no edges (i, i)).

The *Laplacian matrix* of G is the n-by-n matrix L whose diagonal element $L[i, i]$ is the number of neighbors of vertex i (also called the *degree* of vertex i), and whose off-diagonal element $L[i, j]$ is -1 if (i, j) is an edge of G and is 0 otherwise. For example, the Laplacian matrix of the graph consisting of 4 vertices connected in a square (also called a 4-cycle) is

$$L = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}, \qquad \text{where } G = \begin{array}{ccc} 0 & \text{——} & 1 \\ | & & | \\ 3 & \text{——} & 2 \end{array}$$

The Laplacian matrix is symmetric because the graph is undirected. It's a theorem that all the eigenvalues of any symmetric real matrix are real numbers, and it's also a theorem that all the eigenvalues of the Laplacian matrix of a graph are greater than or equal to zero.

1.1. Use `linalg.eigh()` (note the h!) to find the four eigenvalues of the Laplacian matrix of the 4-cycle as above. You should find that they are all nonnegative real numbers, and that one of them is equal to zero.

1.2. In fact, zero is an eigenvalue of the Laplacian matrix of every graph. Prove this by exhibiting an n-vector $v^{(0)}$ that is an eigenvector for the eigenvalue zero for *every* n-vertex graph, and explaining why $Lv^{(0)} = 0v^{(0)}$.

1.3. Given an n-vertex graph G and any n-vector v, we can think of the n elements of the vector v as labels for the n vertices of the graph; $v[0]$ labels vertex 0, $v[1]$ labels vertex 1, and so forth. Used as labels in this way, the eigenvectors of the Laplacian matrix of a graph are in some ways analogous to the fundamental modes of vibration of a physical object.

For example, let P_n be the graph with n vertices joined in a single path, so that P_n has the $n-1$ edges $\{(0, 1), (1, 2), \ldots, (n-2, n-1)\}$. Write a python function `path(n)` that computes the n-by-n Laplacian matrix L_n of the graph P_n as a `numpy` array. Show your function, and show its output for $n = 5$ as an example. Also show the output of `linalg.eigh()` on L_5.

Now we'll see how the Laplacian eigenvectors of the graph P_n correspond to "modes of vibration." The idea is to think of the path P_n as a violin string. Use your function `path()` to compute the Laplacian matrix

L_{100} of the graph P_{100}. Then use `linalg.eigh()` to compute the eigenvalues $d_0, \ldots, d_{99}$ and eigenvectors $v^{(0)}, \ldots, v^{(99)}$ of the matrix L_{100}. Check to see whether the eigenvalues come out of `linalg.eigh()` in increasing order of size; if not, reorder both the eigenvalues and eigenvectors so that $d_0 \le d_1 \le \cdots \le d_{99}$. Don't print out L_{100} or the lists of eigenvalues and eigenvectors, but make and turn in the following three plots (all nicely labeled, of course):

- Plot the 100 elements $v_0^{(0)}, v_1^{(0)}, \ldots, v_{99}^{(0)}$ of the eigenvector $v^{(0)}$ (this is a pretty simple picture).
- Make one plot that has 4 lines on it, plotting the elements of each eigenvector $v^{(1)}$ through $v^{(4)}$.
- Plot the 100 eigenvalues d_i versus i.

1.4. How close is the temperature matrix to being a Laplacian matrix? Specifically, which (and how many) of the nonzero elements of the 2D temperature matrix with $k = 100$ would you need to change to make it into the Laplacian matrix of some graph? Can you describe in words what that graph is?

2. An important problem in classical mechanics is to determine the motion of two bodies under mutual gravitational attraction. Suppose that a body of mass m is orbiting a second body of much larger mass M, such as the earth orbiting the sun. From Newton's laws of motion and gravitation, the orbital trajectory $(x_0(t), x_1(t))$ is described by the system of second-order ODEs

$$\ddot{x}_0 = -GMx_0/r^3, \tag{1}$$

$$\ddot{x}_1 = -GMx_1/r^3, \tag{2}$$

where G is the gravitational constant and $r = (x_0^2 + x_1^2)^{1/2} = ||x||$ is the distance of the orbiting body from the center of mass of the two bodies. For this experiment, we choose units such that $GM = 1$.

2.1. Use `integrate.solve_ivp()` to solve this system of ODEs with the initial conditions

$$x_0(0) = 1 - \epsilon, \quad x_1(0) = 0, \tag{3}$$

$$\dot{x}_0(0) = 0, \quad \dot{x}_1(0) = \left(\frac{1+\epsilon}{1-\epsilon}\right)^{1/2}, \tag{4}$$

where ϵ is the eccentricity of the resulting elliptical orbit, which has period 2π. Try the values $\epsilon = 0$ (which should give a circular orbit), $\epsilon = 0.5$, and $\epsilon = 0.9$. For each case, solve the ODE for at least one orbital period and obtain output at enough intermediate points to draw a smooth plot of the orbital trajectory. Make separate plots of x_0 versus t, x_1 versus t, and x_0 versus x_1, all with well-labeled axes and clear titles. For your plot of the orbit itself, x_0 versus x_1, use `plt.gca().axis('equal')` to make sure the scale is the same on both axes, so that a circle will look like a circle.

Experiment with different error tolerances (use `help(integrate.solve_ivp)` to find out how to set error tolerances) to see how they affect (i) the amount of time required for the solution and (ii) how close the orbit comes to being closed. If you trace the orbit through several periods, does the orbit tend to wander or remain steady? In addition to your plots, turn in an explanation in English of what experiments you did, what you observed, and what your conclusions were.

2.2. Physics tells us that the orbit described by this ODE (like any closed physical system) should conserve both energy and angular momentum. Check your numerical solutions from part (a) to see how well the following quantities remain constant throughout time.

Conservation of energy:

$$\frac{\dot{x}_0^2 + \dot{x}_1^2}{2} - \frac{1}{r}$$

Conservation of angular momentum:

$$x_0\dot{x}_1 - x_1\dot{x}_0$$

Present your results using whatever numbers or graphs you think are most informative, and describe your conclusions in English.