Project 1
Due on Monday, Feb 21

An alien community consists of Venusians $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and pundits $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$. Each pundit p_k has an opinion $y_k \in \mathbb{R}$ which is not influenced by the opinion of anyone else. Each Venusian v_j has an opinion $x_j \in \mathbb{R}$ which is a weighted average of the opinions of some trusted neighbors and pundits:

$$\left(\sum_{j=1}^{n} w_{ij} + \sum_{k=1}^{m} w'_{ik} \right) x_i = \sum_{j=1}^{n} w_{ij} x_j + \sum_{k=1}^{m} w'_{ik} y_k.$$

We write this as a matrix equation

$$Ax = By$$

where

$$A_{ij} = \begin{cases} \sum_{k=1}^{n} w_{ik} + \sum_{k=1}^{m} w'_{ik}, & i = j \\ -w_{ij}, & i \neq j \end{cases}$$

and

$$B_{ik} = w'_{ik}.$$

The matrix entry $(A^{-1}B)_{ik}$ represents the influence of pundit p_k on the opinion of Venusian v_i. We call the kth column of $A^{-1}B$ the *influence vector* of pundit p_k. We assume all the weights are non-negative; this implies that all the elements of A^{-1} (and therefore all elements of each influence vector) are also non-negative. We say the *total influence* of pundit p_k is the sum of the components of his influence vector.

As CEO of Martian Media Moguls, the pundits come to you to evaluate their influence and to tell them how to have more impact. Specifically, they would like to know:

1. What is the total influence for each pundit?

2. Given pundit opinions y, what is the mean Venusian opinion μ?

$$\mu = \frac{1}{n} \sum_{j=1}^{n} x_j.$$

3. Suppose we only know an approximation $\hat{y}_i$ of each pundit's opinion, and that approximation has a relative error bounded by δ; that is, for every i,
$$|\hat{y}_i - y_i|/|y_i| < \delta.$$
What is the maximum possible approximation error in the opinion vector ($\|\hat{x} - x\|_1$)? What is the maximum possible relative error in the mean opinion μ?

4. What is the 1-norm condition number of A?[1]

5. Suppose one weight w'_{ik} changes. What is the effect on the total influence of each pundit?

We want *efficient* solutions to each of these problems. Assume that n is fairly large (say $n = 10^4$), but that the topology of A is nice enough that it is practical to solve linear systems with A or A^T using MATLAB's sparse direct solver. It turns out that if we can compute the total influence vector (which can be done with one linear solve), then 2–3 can be done with no additional linear solves, and 4 can be done with one additional linear solve, and 5 can be done with two additional linear solves.

Your solution should consist of five MATLAB function files:

```
function z  = influence(L,U,P,Q, B)
function mu = mean_opinion(z, y, n)
function [abserr_1norm, relerr_mu] = sensitivity_bound(z, y, delta, n)
function [condA] = laplace_cond1(A, L,U,P,Q)
function znew = update_influence(z, L,U,P,Q, B, i,k, new_wp_ik)
```

Here, A and B are MATLAB sparse matrices and L, U, P, and Q are the factors returned from running MATLAB's `lu` on a sparse matrix A. Try `help lu` in MATLAB to see more.

I recommend that you write naive versions of each code (not worrying about efficiency and using MATLAB intrinsics like `sum`, `mean`, and `cond` where appropriate) to check the correctness of your function for small networks. You are not *required* to submit test routines with your code; but if you compute the wrong answer, we will not even consider partial credit unless there is a plausible-looking test code present.

[1] The pundits have also taken an introductory scientific computing course[2], and they know that you should be wary if your matrices are ill-conditioned.

[2] I did mention that these are space aliens, right?

Notes

1. We will assume that the matrix A is nonsingular. In this context, nonsingularity is equivalent to saying that there is a path from each Venusian to some pundit.

2. If done cleverly, each of these functions can be written in one or two lines of MATLAB.

3. The fact that every element of A^{-1} is positive turns out to be incredibly useful in this context. For example, I do not know how to compute the 1-norm of A^{-1} with only one linear solve for almost any other type of matrix (though one can *estimate* the condition number with a small number of linear solves using `condest` in MATLAB).

4. As we will discuss in class, sparse direct solvers run into scalability for certain types of networks — and social networks are often problematic. One could bypass this issue by using an iterative solver, but that is beyond the scope of the current discussion.

5. For the last problem (computing the effect of a change in one of the weights), keep in mind that you will be affecting both A and B. However, the change to A only affects the ith diagonal element, and so the modified matrix has the form $A + \rho e_i e_i^T$, where ρ is the change in the weight and e_i is the ith standard basis vector (i.e. the vector of all zeros except for a one in component i). It may be useful to look up the *Sherman-Morrison* formula for a rank-one update to an inverse:
$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

6. If you think these ideas sound interesting outside the context of space aliens, a good place to start reading is Section 8.3 of *Social and Economic Networks* by Matt Jackson.