MAT 226B, Winter 2018
Homework 1

(due by Tuesday, January 23, 11:59 pm)

General Instructions

- You are required to submit your homework by uploading a single pdf file to Canvas. Note that the due dates set in Canvas are hard deadlines. I will not accept any submissions outside of Canvas or after the deadline.
- If at all possible, use a text processing tool (such as LaTeX) for the preparation of your homework. If you submit scanned-in hand-written assignments, make sure that you write clearly and that you present your solutions in a well-organized fashion. If I cannot read your homework, I will not be able to grade it!
- Feel free to discuss the problems on the homework sets with other students, but you do need to submit your own write-up of the solutions and your own MATLAB codes. If there are students with solutions that were obviously copied, then each involved student (regardless of who copied from whom) will only get the fraction of the points corresponding to the number of involved students.
- Test cases for computational problems are often provided as binary Matlab files. For example, suppose the file "`LS.mat`" contains the coefficient matrix A and the right-hand side b of a system of linear equations. The Matlab command "`load('LS.mat')`" will load A and b into Matlab.
- When you are asked to print out numerical results, print real numbers in 15-digit floating-point format. You can use the Matlab command "`format long e`" to switch to that format from Matlab's default format. For example, the number 10π would be printed out as `3.141592653589793e+01` in 15-digit floating-point format
- When you are asked to write Matlab programs, include printouts of your codes in your homework.

---

1. Let $A \in \mathbb{R}^{n\times n}$ be a row-stochastic matrix, $\alpha \in \mathbb{R}$ a parameter, and

$$A_\alpha := \alpha\, A + (1-\alpha)\,\frac{1}{n}\, e\, e^T, \quad \text{where} \quad e := \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n. \tag{1}$$

Prove that the matrix A_α is row-stochastic for all parameter values $0 \le \alpha \le 1$.

2. A small company runs an internal network of 10 websites, which have no links to the outside world. The links within the internal network are as follows:

   - Website 1 has links to websites 5, 7;
   - Website 2 has links to websites 2, 6, 7, 9;
   - Website 4 has links to websites 4, 7;
   - Website 5 has links to websites 7, 9, 10;
   - Website 8 has links to websites 3, 4, 7, 8, 9;
   - Website 9 has links to websites 2, 4, 7, 8, 10;
   - Website 10 has links to websites 1, 3, 4, 7, 9, 10.

   Formulate a linear algebra problem the solution of which is the PageRank vector x of this internal network and compute x.

   According to your computed PageRank, what is the ranking of the 10 websites from most to least important?

3. Let $A \in \mathbb{R}^{n \times n}$ be a row-stochastic matrix.

   (a) Show that all eigenvalues λ of A satisfy $|\lambda| \leq 1$.

   (b) For any initial vector

   $$x^{(0)} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} \in \mathbb{R}^n \quad \text{with} \quad x^{(0)} \geq 0 \quad \text{and} \quad x^{(0)} \neq 0,$$

   we consider the iteration

   $$x^{(i+1)} = A^T x^{(i)}, \quad i = 0, 1, 2, \dots \ . \tag{2}$$

   Show that

   $$x^{(i)} \geq 0 \quad \text{and} \quad \sum_{j=1}^{n} x_j^{(i)} = \sum_{j=1}^{n} x_j^{(0)} \quad \text{for all} \quad i \geq 0.$$

   (c) We now assume that the matrix A^T is diagonalizable, i.e., there exists a nonsingular matrix $X \in \mathbb{C}^{n \times n}$ such that

   $$A^T = X \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} X^{-1}.$$

Note that the columns of the matrix X are eigenvectors corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of A^T. Furthermore, we assume that $\lambda_1 = 1$ is a dominant eigenvalue of A^T.

Show that the vectors $x^{(i)}$ generated by the iteration (2) converge and that the limit

$$x = \lim_{i \to \infty} x^{(i)}$$

satisfies $A^T x = x$, $x \geq 0$, and $x \neq 0$.

4. Let $\mathcal{G}$ be a directed graph that describes the connectivity of a set of n websites, and let $Q \in \mathbb{R}^{n \times n}$ and $A \in \mathbb{R}^{n \times n}$ be the associated matrices, as defined in class. Recall that

$$A = Q + \frac{1}{n}\, v\, e^T, \tag{3}$$

where e is the vector given in (1) and $v = \big[v_j\big]_{j=1,2,\ldots,n} \in \mathbb{R}^n$ is defined by

$$v_j = \begin{cases} 1 & \text{if } \ d_j = 0, \\ 0 & \text{if } \ d_j > 0. \end{cases}$$

Here, d_j denotes the out degree of node j of $\mathcal{G}$. In the following, we assume that the matrix A is given via the three quantities n, Q, and J_v, where J_v is a vector the entries of which are all the node indices j with $d_j = 0$.

(a) Write two Matlab functions, which use n, Q, and J_v as inputs, to compute matrix-vector products of the form

$$y = A^T x, \quad \text{where } x \in \mathbb{R}^n \text{ is a dense vector in general,}$$

and

$$y = A_\alpha^T x, \quad \text{where } x \in \mathbb{R}^n \text{ is a dense vector in general,}$$

respectively, as efficiently as possible. Here, A_α, $0 \leq \alpha \leq 1$, is the family of matrices defined in (1).

Hint: You need to exploit relations (3) and (1) in order to be able to run your functions for large graphs, such as the one given in the mat file "`www0.mat`".

Use your Matlab functions to compute the matrix-vector products

$$y = A^T e, \quad y_{0.5} = A_{0.5}^T e, \quad \text{and} \quad y_{0.85} = A_{0.85}^T e$$

for the case of $n = 10$ websites stated in Problem 2, and print out y, $y_{0.5}$, and $y_{0.85}$.

Next, run your Matlab functions on the large graph $\mathcal{G}$ with $n = 685230$ nodes and 7600595 edges given in "`www0.mat`". For the vector x_0 given in "`x0.mat`", compute the matrix-vector products

$$y = A^T x_0 \quad \text{and} \quad y_{0.85} = A_{0.85}^T x_0,$$

and print out the entries with indices 2, 222222, 300000, and 400000 of both vectors y and $y_{0.85}$.

(b) Write a Matlab program that implements the power method, as stated in the notes provided on the course website.

(c) Use your program from (b) to compute the dominant eigenvalue λ and a corresponding eigenvector x of the matrix A^T corresponding to the case of $n = 10$ websites stated in Problem 2. Run your program for the two initial vectors

$$x = e \in \mathbb{R}^{10} \quad \text{and} \quad x = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \in \mathbb{R}^{10}.$$

In both cases, use

$$\varepsilon = 10^{-15} \quad \text{and} \quad k_{\max} = 1000$$

for the convergence check and iteration limit, respectively. For both runs, print out the iteration count k, the value λ of the computed dominant eigenvalue, the computed corresponding eigenvector x, and the *relative eigenvalue residual*

$$\frac{\|A^T x - \lambda x\|_\infty}{\|x\|_\infty}, \tag{4}$$

all at termination of the algorithm. Use the computed x to rank the 10 websites.

(d) Use your program from (b) to try to compute the dominant eigenvalue λ and a corresponding eigenvector x of the matrix A^T corresponding to the large graph with $n = 685230$ nodes and 7600595 edges given in "`www0.mat`". Run your program for the two initial vectors

$$x = e \in \mathbb{R}^n \quad \text{and} \quad x = x_0, \tag{5}$$

where x_0 is the vector given in "`x0.mat`". In both cases, use

$$\varepsilon = 10^{-12} \quad \text{and} \quad k_{\max} = 10000 \tag{6}$$

for the convergence check and iteration limit, respectively. For both initial vectors, print out the iteration count k, the value λ of the computed dominant eigenvalue, and the corresponding relative eigenvalue residual (4), all at termination of the algorithm. Comment on the convergence of the power method for this case.

Rerun your program for the same initial vectors (5) and parameters (6), but now applied to the matrix $A_{0.85}^T$. For both initial vectors, print out the iteration count k, the value λ of the computed dominant eigenvalue, and the corresponding relative eigenvalue residual (4), all at termination of the algorithm. Use the computed eigenvector x to identify and list the 10 most important websites. Comment on the convergence of the power method for this case.

5. A Toeplitz matrix $T = \left[t_{k-j}\right]_{j,k=1,2,\ldots,n} \in \mathbb{R}^{n\times n}$ is determined by the $2n-1$ entries of the row vector

$$t := \left[\, t_{-(n-1)} \quad t_{-(n-2)} \quad \cdots \quad t_{-1} \quad t_0 \quad t_1 \quad t_2 \quad \cdots \quad t_{n-1} \,\right].$$

Toeplitz matrices with

$$t_{-j} = t_{n-j} \quad \text{for all} \quad j = 1, 2, \ldots, n-1$$

are called *circulant matrices*. In this case, we set $c_j := t_j$, $j = 0, 1, \ldots, n-1$. Note that a circulant matrix $C \in \mathbb{R}^{n\times n}$ is determined by the n entries of the row vector

$$c := \left[\, c_0 \quad c_1 \quad c_2 \quad \cdots \quad c_{n-1} \,\right].$$

Circulant matrices have the remarkable property that there are explicit formulas for their eigenvalues and eigenvectors. More precisely, let

$$F := \left[\, \exp\left(\tfrac{-2\pi\mathtt{i}}{n}\, jk\right) \,\right]_{j,k=0,1,\ldots,n-1} \in \mathbb{C}^{n\times n}$$

denote the $n \times n$ *Fourier matrix* and

$$\overline{F} := \left[\, \exp\left(\tfrac{2\pi\mathtt{i}}{n}\, jk\right) \,\right]_{j,k=0,1,\ldots,n-1} \in \mathbb{C}^{n\times n}$$

its complex conjugate. Here, $\mathtt{i} = \sqrt{-1}$. For any $n \times n$ circulant matrix C, the columns of $\overline{F}$ are eigenvectors of C. The corresponding eigenvalues are given by

$$\lambda_k = \sum_{j=0}^{n-1} c_j \exp\left(\frac{2\pi\mathtt{i}}{n}\, jk\right), \qquad k = 0, 1, \ldots, n-1. \tag{7}$$

Furthermore, we have the following factorization:

$$C = \frac{1}{n}\,\overline{F} \begin{bmatrix} \lambda_0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n-1} \end{bmatrix} F. \tag{8}$$

Matrix-vector products $y = Fx$ with the Fourier matrix F can be computed very efficiently via the fast Fourier transform. In Matlab, this computation is done by means of the command "`y = fft(x)`".

(a) Show that all n eigenvalues of an $n \times n$ circulant matrix C can be computed via a single call to Matlab's "`fft`" command.

(b) Use part (a) and the factorization (8) to show that each matrix-vector product $y = Cx$ with a circulant matrix C can be computed via three calls to Matlab's "`fft`" command.

Write a Matlab function that implements this approach to efficiently compute matrix-vector products with circulant matrices. Use your program to compute $y = Cx$ for

$$C = \begin{bmatrix} 2 & -1 & 0 & -3 \\ -3 & 2 & -1 & 0 \\ 0 & -3 & 2 & -1 \\ -1 & 0 & -3 & 2 \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} -1 \\ 2 \\ 1 \\ 4 \end{bmatrix}.$$

Print out your computed y and compare it with the exact y. Submit a printout of your Matlab function.

(c) Show that for any given Toeplitz matrix $T \in \mathbb{R}^{n\times n}$ there is a unique circulant matrix $C \in \mathbb{R}^{(2n-1)\times(2n-1)}$ such that T is the leading principal $n \times n$ submatrix of C, i.e.,

$$C = \begin{bmatrix} T & \star \\ \star & \star \end{bmatrix}.$$

(d) Use (c) to devise an algorithm that computes matrix-vector products $y = Tx$ with Toeplitz matrices T via your Matlab function form (b).

Write a Matlab function that implements this approach.

(e) To test your program from (d), first employ it to compute $y = Tx$ for

$$T = \begin{bmatrix} 2 & -1 & 4 & -3 \\ 1 & 2 & -1 & 4 \\ -5 & 1 & 2 & -1 \\ 6 & -5 & 1 & 2 \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} -1 \\ 2 \\ 1 \\ 4 \end{bmatrix}.$$

Print out your computed y and compare it with the exact y.

Now employ your program to compute $y = Tx$ where T is the $n \times n$ Toeplitz matrix given by the row vector t. Both t and the column vector x are provided in the mat file "`t_and_x.mat`". Here, $n = 500000$. Print out the following quantities of your computed result y:

$$y_1, \quad y_{100000}, \quad y_{200000}, \quad y_{300000}, \quad y_{400000}, \quad y_{500000}, \quad \sum_{j=1}^{n} y_j, \quad \|y\|_2.$$