Homework 1

Due September 20th, 2011

Problem 1: Norms

a) Show that $f(\mathbf{x}) = \|\boldsymbol{A}\mathbf{x}\|_p$ is a vector-norm, where $\boldsymbol{A}$ is a non-singular matrix.

b) Show that $f(\mathbf{x}) = \|\boldsymbol{A}\mathbf{x}\|_p$ is not a vector-norm if $\boldsymbol{A}$ is singular.

These norms will arise in our study of spectral graph theorem. In those cases, the matrix $\boldsymbol{A}$ is usually the diagonal matrix of degrees for each node – commonly written $\boldsymbol{D}$.

Problem 2

There are a tremendous number of matrix norms that arise. An interesting class are called the *orthgonally invariant norms.* Norms in this class satisfy:

$$\|\boldsymbol{A}\| = \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\|$$

for *square orthogonal matrices* $\boldsymbol{U}$ and $\boldsymbol{V}$. Recall that a square matrix is orthogonal when $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, i.e. $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$.

a) Show that $\|\boldsymbol{A}\|_F$ is orthogonally invariant. (Hint: use the relationship between $\|\boldsymbol{A}\|_F$ and $\operatorname{trace}(\boldsymbol{A}^T\boldsymbol{A})$.)

b) Show that $\|\boldsymbol{A}\|_2$ is orthogonally invariant. (Hint: first show that $\|\boldsymbol{U}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ using the relationshp between $\|\mathbf{x}\|$ and $\mathbf{x}^T\mathbf{x}$.)

Problem 3

In this problem, we'll work through the answer to the challenge question on the introductory survey.

Let $\boldsymbol{A}$ be the adjacency matrix of a simple, undirected graph.

a) **An upper bound on the largest eigenvalue**
Show that $\lambda_{\max}(\boldsymbol{A})$ is at most, the maximum degree of the graph. Show that this bound is tight.

b) **A lower bound on the largest eigenvalue** Show that $\lambda_{\max}(\boldsymbol{A})$ is at least, the square-root of the maximum degree of the graph. Show that this bound is tight. (Hint: try and find a lower-bound on the Rayleigh-Ritz characterization $\lambda_{\max} = \max \mathbf{x}^T\boldsymbol{A}\mathbf{x}/\mathbf{x}^T\mathbf{x}$.)

Problem 4

In this question, we'll show how to use these tools to solve a problem that arose when Amy Langville and I were studying ranking algorithms.

a) **the quiz from class** Let $\boldsymbol{A}$ be an $n \times n$ matrix of all ones:

$$\boldsymbol{A} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{bmatrix}.$$

What are the eigenvalues of $\boldsymbol{A}$? What are the eigenvectors for all non-zero eigenvalues? Given a vector $\mathbf{x}$, how can you tell if it's in the *nullspace* (i.e. it's eigenvector with eigenvalue 0) without looking at the matrix?

b) **my problem with Amy** Amy and I were studying the $n+1 \times n+1$ matrix:

$$\boldsymbol{A} = \begin{bmatrix} n & -1 & \cdots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & n \end{bmatrix}$$

that arose when we were looking at ranking problems like we saw in `http://www.cs.purdue.edu/homes/dgleich/nmcomp/lectures/lecture-1-matlab.m` What we noticed was that Krylov methods to solve

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

worked incredibly fast.
Usually this happens when $\boldsymbol{A}$ only has a few *unique* eigenvalues. Show that this is indeed the case. What are the *unique* eigenvalues of $\boldsymbol{A}$?

c) **solving the system** Once we realized that there were only a few unique eigenvalues and vectors, we wanted to determine if there was a closed form solution of:

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}.$$

There is such a form. Find it. (By closed form, I mean, given $\mathbf{b}$, there should be a simple expression for $\mathbf{x}$.)

Problem 5

In this question, you'll implement codes to convert between triplet form of a sparse matrix and compressed sparse row.

You may use any language you'd like.

a) Describe and implement a procedure to turn a set of triplet data this data into a one-index based set of arrays: `pointers,␣columns,␣and␣values` for the compressed sparse form of the matrix. Use as little additional memory as possible. (Hint: it's doable using *no* extra memory.)

```
function [pointers, columns, values] = sparse_compress(m, n, triplets)
% SPARSE_COMPRESS Convert from triplet form
%
% Given a m-by-n sparse matrix stored as triplets:
%   triplets(nzi,:) = (i,j,value)
% Output the  the compressed sparse row arrays for the sparse matrix.

% fill in the function
```

b) Describe and implement a procedure to take in the one-indexed compressed sparse row form of a matrix: `pointers,␣columns,␣and␣values` and the dimensions `m,␣n` and output the compressed sparse row arrays for the transpose of the matrix:

```
function [pointers_out, columns_out, values_out] = sparse_transpose(...
m, n, pointers, columns, values)
% SPARSE_TRANSPOSE Compute the CSR form of a matrix transpose.
%
%

% fill in the function
```

Problem 6: Make it run in Matlab/Octave/Scipy/etc.

In this problem, you'll just have to run three problems on matlab. The first one will be to use the Jacobi method to solve a linear system. The second will be to use a Krylov method to solve a linear system. The third will be to use ARPACK to compute eigenvalues on Matlab.

For this problem, you'll need to use the 'minnesota' road network. It's available on the website: http://www.cs.purdue.edu/homes/dgleich/nmcomp/matlab/minnesota.mat The file is in Matlab format. If you need another format, let me know.

a) Use the `gplot` function in Matlab to draw a picture of the Minnesota road network.

b) Check that the adjacency matrix A has only non-zero values of 1 and that it is symmetric. Fix any problems you encouter.

c) We'll do some work with this graph and the linear system described in class:

$$\boldsymbol{I} - \gamma \boldsymbol{L}$$

where $\boldsymbol{L}$ is the combinatorial Laplacian matrix.

```
% In Matlab code
L = diag(sum(A)) - A;
S = speye(n) - gamma*L;
```

For the right-hand side, label all the points above latitude line 47 with 1, and all points below latitude line 44 with -1.

```
% In Matlab code
b = zeros(n,1);
b(xy(:,2) > 47) = 1;
b(xy(:,2) < 44) = -1;
```

Write a routine to solve the linear system using the Jacobi method on the compressed sparse row arrays. You should use your code from 5a to get these arrays by calling

```
[src,dst,val] = find(S);
T = [src,dst,val];
[pointers,columns,values] = sparse_compress(size(A,1), size(A,2), T);
```

Show the convergence, in the relative residual metric:

$$\|\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\| / \|b\|$$

when `gamma␣=␣1/7` (Note that $\boldsymbol{A}$ is the matrix in the linear system, not the adjacency matrix.)

Show what happens when `gamma=1/5`

d) Try using Conjugate Gradient `pcg` and `minres` in Matlab on this same system with `gamma=1/7` and `gamma=1/5`. Show the convergence of the residuals.

e) Use the `eigs` routine to find the 18 smallest eigenvalues of the Laplacian matrix $\boldsymbol{L}$.