CS 6210 Take-Home Final   Due: 12/17/15 (Thur) at Noon

You are not allowed to communicate with anybody about this exam. Aside from GVL4, properly cite all websites, papers, etc., if their content was directly use in your solution.

1 Sparse Orthogonal Matrix Set-Up

With the notation $e_i = I_3(:,i)$ here is a basis for the subspace of 3x3 symmetric matrices:

$$\begin{aligned}
S_{11} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = e_1e_1^T \\
S_{21} &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = e_2e_1^T + e_1e_2^T \\
S_{31} &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = e_3e_1^T + e_1e_3^T \\
S_{22} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = e_2e_2^T \\
S_{32} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = e_3e_2^T + e_2e_3^T \\
S_{33} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = e_3e_3^T
\end{aligned}$$

and a basis for the subspace of 3x3 skew-symmetric matrices:

$$\begin{aligned}
T_{21} &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = e_2e_1^T - e_1e_2^T \\
T_{31} &= \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = e_3e_1^T - e_1e_3^T \\
T_{32} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = e_3e_2^T - e_2e_3^T
\end{aligned}$$

The matrix $Q_{3,3} \in \mathbb{R}^{9\times 9}$ is obtained by normalizing these nine matrices so that they have unit Frobenius norm and then aggregating their vec's:

$$Q_{3,3} = [\,\text{vec}(S_{11})\ \ \text{vec}(\alpha S_{21})\ \ \text{vec}(\alpha S_{31})\ \ \text{vec}(S_{22})\ \ \text{vec}(\alpha S_{32})\ \ \text{vec}(S_{33}) \,|\, \text{vec}(\alpha T_{21})\ \ \text{vec}(\alpha T_{31})\ \ \text{vec}(\alpha T_{32})\,]$$

$$= \left[\begin{array}{cccccc|ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha & 0 & 0 & 0 & 0 & \alpha & 0 & 0 \\
0 & 0 & \alpha & 0 & 0 & 0 & 0 & \alpha & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha & 0 & 1 & 0 & 0 & -\alpha & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & \alpha \\
0 & 0 & \alpha & 0 & 0 & 0 & 0 & -\alpha & 0 \\
0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & -\alpha \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right] = [\,Q_{sym} \,|\, Q_{skew}\,] \qquad \alpha = 1/\sqrt{2}.$$

This is a sparse orthogonal matrix.

For general n, $Q_{n,n} = [Q_{sym}\ Q_{skew}] \in \mathbb{R}^{n^2\times n^2}$. The columns of the n^2-by-$n(n+1)/2$ matrix Q_{sym} are associated the S_{ij}. The mapping of the (i,j) basis matrices S_{ij} to the columns of Q_{sym} is as follows:

$$(1,1),\ (2,1), \ldots, (n,1),\ (2,2),\ (3,2), \ldots, (n,2), \ldots, (n-1,n-1),\ (n,n-1),\ (n,n).$$

The columns of the n^2-by-$n(n-1)/2$ matrix Q_{skew} are associated the T_{ij}. The mapping of the (i,j) basis matrices T_{ij} to the columns of Q_{skew} is as follows:

$$(2,1),\ (3,1), \ldots, (n,1),\ (3,2),\ (4,2), \ldots, (n,2), \ldots, (n-1,n-1),\ (n,n-1).$$

Implement the following MATLAB function:

```
    function Q = Qnn(n)
  % n is a positive integer
  % Q is the n^2-by-n^2 orthogonal matrix Q_{n,n} in sparse format.
```

The efficiency of your implementation will be a major factor in the grading. Use `sparse` intelligently. Submit `Qnn` to CMS.

2 Closest Kronecker Product to a Block Tridiagonal Matrix

Suppose A is an n_1-by-n_1 block matrix with n_2-by-n_2 blocks, e.g.,

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}.$$

Consider the problem of minimizing

$$\phi(B,C) = \| A - B \otimes C \|_F$$

where $B \in \mathbb{R}^{n_1 \times n_1}$ and $C \in \mathbb{R}^{n_2 \times n_2}$. The solution procedure is outlined in GVL4 §12.3.6. It involves computing the largest singular value and vectors of an n_1^2-by-n_2^2 matrix $\tilde{A}$ whose rows are the vec's of the blocks, e.g.,

$$\tilde{A} = \begin{bmatrix} \text{vec}(A_{11})^T \\ \text{vec}(A_{21})^T \\ \text{vec}(A_{31})^T \\ \text{vec}(A_{41})^T \\ \text{vec}(A_{12})^T \\ \text{vec}(A_{22})^T \\ \text{vec}(A_{32})^T \\ \text{vec}(A_{42})^T \\ \text{vec}(A_{13})^T \\ \text{vec}(A_{23})^T \\ \text{vec}(A_{33})^T \\ \text{vec}(A_{43})^T \\ \text{vec}(A_{14})^T \\ \text{vec}(A_{24})^T \\ \text{vec}(A_{34})^T \\ \text{vec}(A_{44})^T \end{bmatrix}.$$

In particular, if σ_1 is the largest singular value of $\tilde{A}$ and u_1, and v_1 the left and right singular vectors, then

$$B_{opt} = \sqrt{\sigma_1} \cdot \texttt{reshape}(u_1, n_1, n_1) \qquad C_{opt} = \sqrt{\sigma_1} \cdot \texttt{reshape}(v_1, n_2, n_2).$$

In this problem you are to implement this procedure for the case when A is block tridiagonal with sparse blocks:

```
  function [B,C] = NearestKP(A,n1,n2)
% A (in sparse format) is an n1-by-n1 block tridiagonal matrix with n2-by-n2 blocks.
% B (n1-by-n1) and C (n2-by-n2) minimize norm(A-kron(B,C),'fro') and are both
% in full format.
```

You may want to make use of `svds` and $\tilde{A}$'s special structure. Submit `NearestKP` to CMS.

3 A Max Trace Problem

Let $\text{tr}(A)$ denote the trace of a square matrix $A \in \mathbb{R}^{n \times n}$:

$$\text{tr}(A) \;=\; \sum_{i=1}^{n} a_{ii}$$

Here are some facts about the trace function:

1. If $F, G \in \mathbb{R}^{n \times n}$, then $\text{tr}(F+G) \;=\; \text{tr}(F) + \text{tr}(G)$.
2. If $\alpha \in \mathbb{R}$ and $F \in \mathbb{R}^{n \times n}$, then $\text{tr}(\alpha F) = \alpha\, \text{tr}(F)$.
3. If $F, G \in \mathbb{R}^{n \times r}$, then $\text{tr}(FG^T) = \text{tr}(G^T F)$.
4. If $A \in \mathbb{R}^{n \times n}$ is symmetric with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and $U \in \mathbb{R}^{n \times r}$ has orthonormal columns, then

$$\text{tr}(U^T A U) \leq \lambda_1 \;+ \cdots +\; \lambda_r$$

   and the upper bound is attained if $\text{ran}(U)$ is an invariant subspace associated with $\lambda_1, \ldots, \lambda_r$.

Implement a function `[Q,r,phiMax] = TwoMatTrace(A1,A2)` that maximizes the objective function

$$\phi(Q,r) \;=\; \text{tr}(Q_1^T A_1 Q_1) \;+\; \text{tr}(Q_2^T A_2 Q_2) \qquad\qquad Q_1 = Q(:,1{:}r),\; Q_2 = Q(:,r+1{:}n)$$

where $A_1 \in \mathbb{R}^{n \times n}$ and $A_2 \in \mathbb{R}^{n \times n}$ are given symmetric matrices, $Q \in \mathbb{R}^{n \times n}$ is orthogonal, and r is an integer that satisfies $0 \leq r \leq n$. To be clear about the "edge" cases:

$$\phi(Q,0) \;=\; \text{tr}(Q^T A_2 Q) \qquad\qquad \phi(Q,n) \;=\; \text{tr}(Q^T A_1 Q).$$

The optimum Q should be returned in `Q`, the optimum r should be returned in `r`, and `phiMax` should return the optimum value of ϕ. Submit `TwoMatTrace` to CMS.

4 Limiting Probabilities

Suppose $Q \in \mathbb{R}^{n \times n}$ has the property that its diagonal entries are negative, its off-diagonal entries are strictly positive, and its column sums are zero. It can be shown that if

$$F(t) \;=\; e^{Qt}$$

then regardless of t, all the entries in $F(t)$ are nonnegative and its column sums are one. That is, $F(t)$ is stochastic. Its entries are probabilities and the probabilities in each of its columns sum to one.

It can be shown that the diagonal entries in e^{Qt} converge:

$$\lim_{t \to \infty} \; [F(t)]_{ii} \;=\; p_i.$$

Implement a function `p = pLimits(Q)` that returns the column vector of limiting diagonal probabilities associated with e^{Qt}. You are NOT allowed to use `expm`. Submit `pLimits` to CMS.

5 A Fast Matrix-Vector Multiply

Suppose $p, q, x \in \mathbb{R}^n$ and set $A = \texttt{triu}(pq^T)$. The matrix vector product $y = Ax$ can be computed with $O(n)$ work and here is why:

$$y \;=\; Ax \quad = \begin{bmatrix} p_1q_1 & p_1q_2 & p_1q_3 & p_1q_4 \\ 0 & p_2q_2 & p_2q_3 & p_2q_4 \\ 0 & 0 & p_3q_3 & p_3q_4 \\ 0 & 0 & 0 & p_4q_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} p_1 & p_1 & p_1 & p_1 \\ 0 & p_2 & p_2 & p_2 \\ 0 & 0 & p_3 & p_3 \\ 0 & 0 & 0 & p_4 \end{bmatrix} \begin{bmatrix} q_1x_1 \\ q_2x_2 \\ q_3x_3 \\ q_4x_4 \end{bmatrix}$$

$$= \begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ 0 & 0 & 0 & p_4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_1x_1 \\ q_2x_2 \\ q_3x_3 \\ q_4x_4 \end{bmatrix}$$

$$= \begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ 0 & 0 & 0 & p_4 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} q_1x_1 \\ q_2x_2 \\ q_3x_3 \\ q_4x_4 \end{bmatrix}$$

You might want to review §12.2.1-12.2.4 in GVL4. Possibly using these ideas, implement the following function so that it performs as specified:

```
  function y = FastProd(P,Q,R,S,d,x)
% P,Q,R, and S are n-by-r, n is an integral multiple of r, and n>>r.
% d and x are column n-vectors
% y = Ax where A = tril(R*S',-1) + diag(d) + triu(P*Q',1)
```

Submit `FastProd` to CMS.