YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

| | | |
|---|---|---|
| | CptSci 445/545: | *Handout #3* |
| *V. Rokhlin* | Introduction to Data Mining | *September 11, 2013* |


# Problem Set 1

Due Date: September 30, 2013

**Use double precision computations.**

a) Write a subroutine (function) named jacobi implementing the Jacobi algorithm for the SVD of a real matrix. The singular values should be returned in decreasing order, and the order of the singular vectors should correspond to the order of singular values.

The calling sequence of the FORTRAN subroutine should be

$$jacobi(a, n, s, u, v) \tag{1}$$

The input parameters in the calling sequence are
a - the $n \times n$-matrix to be diagonalized, given as $n \times n$ array; will be destroyed by the subroutine
n - the dimensionality of the matrix

The output parameters are:
s - the spectrum of the matrix, in order of decreasing values (all should be non-negative), given as an array of length $n$
u - the left singular vectors of the matrix a, given as $n \times n$ array
v - the right singular vectors of the matrix a, given as $n \times n$ array

The calling sequence of the C function should be

$$void\ jacobi(double * a,\ int\ n,\ double * s,\ double * u,\ double * v) \tag{2}$$

The input parameters in the calling sequence are
a - the $n \times n$-matrix to be diagonalized, given as $n \times n$ array; will be destroyed by the subroutine. Please note that, unlike in the FORTRAN environment, the matrices are stored *by the row*, i.e. The elements of the matrix $a$ should be ordered as follows:
$a_{1,1},\ a_{1,2},\ \ldots,\ a_{1,n},\ a_{2,1},\ a_{2,2},\ \ldots,\ a_{2,n},\ \ldots,\ \ldots, a_{n,1},\ a_{n,2},\ \ldots,\ a_{n,n}$.
n - the dimensionality of the matrix

The output parameters are:
s - the spectrum of the matrix a, in order of decreasing values (all should be non-negative), given as an array of length $n$
u - the left singular values of the matrix a, given as $n \times n$ array
v - the right singular values of the matrix a, given as $n \times n$ array

b) Once your subroutine (function) is written and tested, apply it to the matrices of dimensionalities $10, 20, 40$, defined by the formula

$$A_{i,j} = sqrt(i * i + j * j). \tag{3}$$

In each case, your output should contain the list of all singular values of A in that are greater than 1.0E-13, as well as the coordinates of the singular vector corresponding to the largest (in the sense of absolute value) eigenvalue.

c) Apply your subroutine (function) to the matrix of dimensionality 10, defined by the formula

$$A_{i,j} = i * i + j * j. \tag{4}$$

Comment on the behavior of the algorithm: the number of sweeps and operation count as a function of $n$, convergence rate, apparent reliability, etc. If needed, conduct additional experiments.