CS 404: Problem Set1

Directions for Submission

E-mail your answers to me at `ajp9@cornell.edu`. The subject of your message should be "CS404 PS1," and the body of the message should contain your answers. Some mailers can format messages using HTML or RTF. Please turn this feature off and send your message using plain text. If for some reason you cannot send your message as text, you may attach your answers as a text file.

Essential Knowledge–Please give a brief answer (1-2 sentences) for each

1. Search GAMS for routines to perform principal components analysis. What is the "problem taxonomy" (Hint–once you've found routines, keep clicking "parent class" until you get to the top of the "Problem-decision tree")? Does GAMS find any non-commercial routines for PCA?

2. Describe how you would find the LAPACK routine for solving a linear least-squares problem using QR-factorization. What is the name of the double-precision routine for this problem?

3. I do most of my programming on a Silicon Graphics workstation. If I enter the command:

   ```
   cc csin.c -osintest
   ```

   I get the following error message:

   ```
   Unresolved text symbol "sin" – 1st referenced by csin.o
   ```

   The file csin.c is a simple C-program which calls the sin function. However, if I build with

   ```
   cc csin.c -osintest -lm
   ```

   the build succeeds. Why doesn't the first command work? What does the "-lm" flag do (be as specific as possible)?

Extending the PCA Code

1. The program Fpca that we built in class on Friday computes the principal components for a matrix of data. If the percent variance explained by a component is high, then we can approximate the entire data set using the component. To create the approximation, we need to extract the eigenvector we want–let's call it v, from the eigenvector matrix. We compute the component by multiplying our data matrix C by v:

   $$PC1 = C * v$$

   If C has m rows and n columns, how long must v be? How long will PC1 be? Look at the notes from Lecture01 for a quick explanation of matrix-vector multiplication.

2. Download the tar file FpcaPS.tar from the website. This file contains the Makefile and code for Fpca, with some slight modifications. In particular, I've created two arrays in main.f: v and PC1 to hold the eigenvector and principal component. The call to the LAPACK routine SSYEV saves the eigenvectors in the array Cov. After this call, I extract the last column from Cov and save it in v (the eigenvalues from SSYEV are sorted so that the last eigenvalue is the largest–it explains the most variance, so the last column in Cov is the vector corresponding to this eigenvalue). It is your job to find a BLAS routine to multiply the data matrix C by the vector v, storing the answer in PC1. You should replace the line "CALL BLASROUTINE???" with the correct call. The principal component in PC1 is then printed to the file PC1.txt. If everything works, PC1.txt should look like:

   ```
   1   3.56080
   2   24.4577
   3   21.1221
   4   41.3958
   5   50.9797
   ```

   if you use the input file data.txt (included in the tar-file). When you get your BLAS call working, copy the subroutine call and send it to me.

   Here's one hint to help you set up the call. Many BLAS routines require an integer parameter describing the "increment" of each array

(e.g. INCX). The increment is the distance between the elements in the vector. Here's how v would look with increments of 1, 2, and 3:

$$\text{INC=1: } v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \text{INC=2: } v = \begin{bmatrix} v_1 \\ ? \\ v_2 \\ ? \\ v_3 \\ ? \end{bmatrix}, \text{INC=2: } v = \begin{bmatrix} v_1 \\ ? \\ ? \\ v_2 \\ ? \\ ? \\ v_3 \\ ? \\ ? \end{bmatrix}$$

Basically, this is a long explanation for why INC should be one for all of the vectors created by Fpca. However, I'll give bonus points and/or candy to anyone who explains how you might use INC¿1.