CS 404: Problem Set2 Key

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?

   The problem taxonomy for PCA is:

   ↪ L: Statistics and probability
   ↪ L13: Covariance structure models
   ↪ L13b: Principal components analysis

   There are no non-commercial PCA routines available through GAMS.

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?

   The easiest way to find LAPACK routines is through the LAPACK search engine (Go to Netlib, find LAPACK, and click on the link at the top). Click on "Driver Routines" on the left. This brings up a list of the 9 catagories of driver routines in LAPACK. Click on the one for "Linear Least-Squares." You will then get a Java dialog with several radio-button menus and buttons. Select "Real, Double" for the precision and "QR or LQ" for the operation type. This will tell you that the routine you want is called DGELS.

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)?

The first command fails because the sin function is defined in a library that is not automatically included by cc. To "-lm" command in the second command tells the compiler to find a library called "libm.a" in one of the system library directories and link to its routines if necessary.

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.

If C is m-by-n, then v must be length n. PC1 will then be length m.

2. 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 correct call is:

```
CALL SGEMV('n', m, n, 1.0, C, NMAX, v, 1, 0.0, PC, 1)
```

To make this call work, you must tell BLAS the length of a column in C, which is NMAX. Also, you need to make sure that the scalars α and β are specifed as real numbers not integers–they should be 1.0 and 0.0, rather than 1 and 0.