High Performance Computing Project 2013-2014

Solving a linear mixed model using ScaLAPACK

Introduction

We have compiled a dataset containing the scores of a math test for a large number of students, sampled from different schools. We want to investigate whether the IQ of the students is a good predictor for their math test score. Of course one could perform a simple regression analysis, but we suspect that, next to the effect of IQ, there is also an effect of the school. Unfortunately, including the school effect in a linear regression analysis is difficult and statistically unsound (the residuals are not independent). Instead, we will use a linear mixed model for the analysis. Mixed models are a class of statistical models that can be used to relate an outcome variable to both fixed (the IQ) and random effects (the schools).

The statistical model contains an intercept, an IQ-effect and a school-effect, and is written like this:

$$y = Xb + Zu + e$$

n = the total number of students
s = the number of schools
y = the math score result for each student (*n* x 1)

$$\begin{bmatrix} 220 \\ 183 \\ 242 \\ \dots \end{bmatrix}$$

X = the regressor matrix (*n* x 2) which relates the intercept and the IQ to the math score:

$$\begin{bmatrix} 1 & 112 \\ 1 & 96 \\ 1 & 137 \\ & \dots \end{bmatrix}$$

b = the regression coefficients that need to be estimated (2 x 1). It contains the intercept and the IQ-effect.

$$\begin{bmatrix} intercept \\ IQ - effect \end{bmatrix}$$

Z = the regressor matrix (*n* x *s*) which relates the students and the schools. A non-zero value indicates the fraction of the time a student (row) spent his/her time on a school (column). For the sake of simplicity, non-zero values are either 1 or 0.5.

$$\begin{bmatrix} 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & \dots \\ 0.5 & 0.5 & 0 & \dots \\ & \dots & & \end{bmatrix}$$

u = the school effect (s x 1). This effect is a random effect, assumed to be drawn from a Normal distribution $N(0, \sigma_s^2)$, with σ_s known in advance.

$$\begin{bmatrix} -5.8 \\ -64.2 \\ -25.2 \\ 40.6 \\ \dots \end{bmatrix}$$

e = error term per student (n x 1), assumed to be drawn from a Normal distribution $N(0, \sigma_e^2)$, with σ_e known in advance.

We can use this model (assuming IQ is normally distributed with mean 100 and standard deviation 15, $\sigma_e = 10.0$, $\sigma_s = 30.0, s = 10, n = 50, b' = [100.0\ , 1.0]$) to simulate a dataset (the Y vector). An example dataset is shown in Figure 1 (left). Using the following equation, we can solve the linear mixed model, i.e. estimate the model's intercept, the IQ-effect and the different school effects. In other words, we need to determine the *b* and the *u* vectors:

$$\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + \frac{\sigma_e^2}{\sigma_s^2} I \end{bmatrix} \begin{bmatrix} b \\ u \end{bmatrix} = \begin{bmatrix} X^T y \\ Z^T y \end{bmatrix}$$

Solving this model is trivial when the number of schools is limited. Indeed, the number of schools is determining the size of the left-most matrix in the mixed model equations. In our simple example, where there are only 10 schools, this results in a 12x12 matrix that is easily inverted. The result is listed in Result 1. As can be seen from the right part of Figure 1, when we use the coefficients obtained after solving the mixed model to predict the test scores (i.e., calculating $\hat{y}$), we obtain a very good fit.

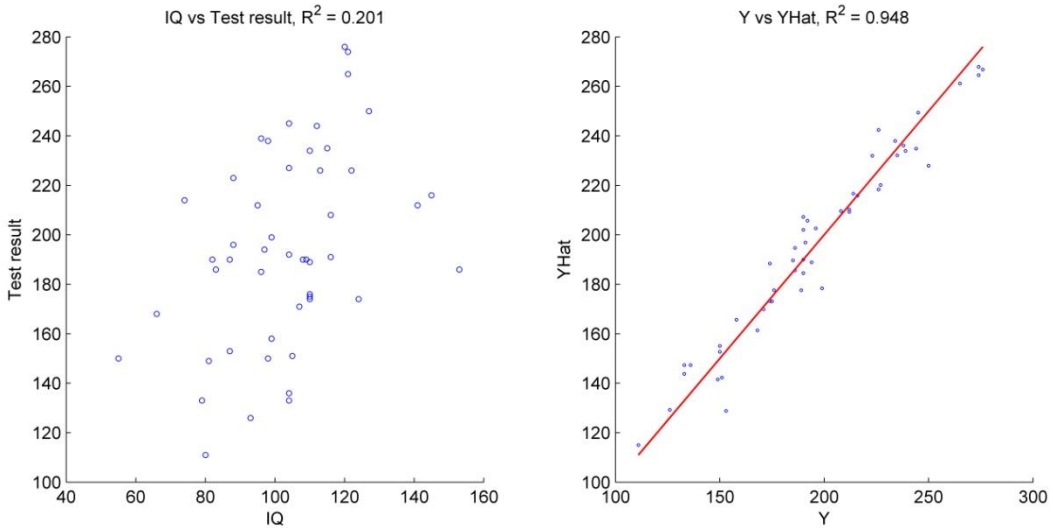

Figure 1. Left: a very small dataset illustrating the apparently weak relationship between IQ and test score. Right: the relation between the test score and the predicted test score after fitting a linear mixed model.

Result 1. The original model used for data simulation and the corresponding solution of the mixed model.

```
real model: y = Xb+Zu+e
----------------------------------------------------------------
            b' = [100.000,1.000]
            u' = [-5.824,-64.151,-25.188,40.638,-32.165,28.829,3.721,43.101,-58.827,-5.9]
            sigmaU = 30.0
```

```
            sigmaE = 10.0

solution: y = Xb+Zu+e
---------------------------------------------------------------------
            b' = [81.159,1.091]
            u' = [8.223,-53.410,-23.538,48.002,-28.040,25.476,13.900,54.765,-47.269,1.9]
            sigmaU = 30.0
            sigmaE = 10.0
```

The problem to solve

We have generated a very large dataset, with *s*=10,000, and *n*=1,000,000. All data files can be downloaded from http://bioinformatics.intec.ugent.be/scalapack. The following files are available:

- X_mmSchools.txt: IQ for each student
- Y_mmSchools.txt: test score for each student
- Z_mmSchools.txt: school-ID(s) for each student. When multiple IDs are given, assume an equal fraction for each school (adding up to 1).

Each file starts with the number of lines in the file.

For testing purposes, we also added a small dataset

- sX_mmSchools.txt: IQ for each student
- sY_mmSchools.txt: test score for each student
- sZ_mmSchools.txt: school-ID(s) for each student. When multiple IDs are given, assume an equal fraction for each school (adding up to 1).
- small_model.txt: the model parameters used to generate the data, and the corresponding solution found by solving the mixed model equations in Matlab

When trying to solve the large problem on a single computer, this problem will very likely not fit in memory, as this would require approximately (in double precision) 10,000x10,000x8 bytes = 800MB, just for storing the left-most matrix of the mixed model equations. Even more problematic is the calculation of Z^TZ, because Z is a 1,000,000 x 10,000 matrix. Therefore, we will try to solve the mixed model equations using a parallel computing approach. The final code should be run in Delcatty (UGent HPC).

When evaluating your solution, you can check the following:

1. The B-vector should be very close to [100, 1], as we used these parameters to generate the simulated test scores.
2. Once you have determined the B and u vectors, you can calculate the predicted y values ($\hat{y}$). $\hat{y}$ should correlate strongly with the original y values, analogous to the right part of Figure 1.

The parallel solution

To overcome the memory limitations of a single machine, we will make use of PBLAS and ScaLAPACK, libraries that implement parallel, distributed-memory versions of BLAS and LAPACK respectively.

Your solution will most likely consist of the following parts (many variants are possible)

1. Read all data
2. Set up a process grid
3. Distribute, using a block cyclic distribution, the data across the process grid
4. Calculate the Z^TZ matrix
5. Construct the mixed model equations
6. Solve the mixed model equations
7. Evaluate your solution (check the values of the B vector, and calculate the correlation between the predicted y values ($\hat{y}$) and y)
8. Assess the speed-up as a function of the number of processes used (on Decatty).

We refer to the documentation of the API at http://www.netlib.org/scalapack/. Many of the examples and the source files are given in FORTRAN, but the conversion to C or C++ should be straightforward. This said, getting ScaLAPACK to work on your machine can be problematic. Some tips:

1. First, try the small dataset, with the solution provided.
2. Use a Linux machine: ScaLAPACK for Windows is not an option, unless you use e.g. the Intel Math Kernel Library (MKL), which is not freely available. If you don't have a Linux machine, you could e.g. set up a virtual CentOS, Ubuntu or Fedora.
3. Once you have Linux up and running, go look for an MPI implementation (e.g. openMPI)
4. Next, install the ScaLAPACK libraries that were built using your MPI implementation. Normally your package manager will recognize the ScaLAPACK library, if not, have a look at
    a. http://pkgs.org/centos-6/puias-computational-i386/scalapack-openmpi-1.7.5-10.puias6.i686.rpm.html
    b. http://albertskblog.blogspot.be/2013/04/how-to-install-lapack-and-scalapack-on.html
5. Next to installing ScaLAPACK, you will also need the BLAS and LAPACK routines.
6. When using C or C++ you have to declare all ScaLAPACK routines you use in a header file, that is included before the routines are called e.g. like this:

    ```
    extern "C" void pdgemm_(char* TRANSA, char* TRANSB, int *M, int *N, int *K,
    double *ALPHA, double *A, int *IA, int *JA, int *DESCA,double * B, int *IB, int
    *JB, int *DESCB,double *BETA, double *C, int *IC, int *JC, int *DESCC);
    ```

    Note that all arguments to FORTRAN routines are passed by reference (not by value)!
7. A very good (and unfortunately probably the only) example of calling ScaLAPACK from C/C++ can be found on http://andyspiros.wordpress.com/2011/07/08/an-example-of-blacs-with-c/. Note that on this website, the ScaLAPACK routines are called using a C wrapper (hence the 'C' prefix, and the arguments passed by value rather than by reference. We suggest to directly calling the FORTRAN routines).
8. Linking with ScaLAPACK can be difficult. This is an example link line (also see the example on Minerva):

    ```
    /usr/lib64/openmpi/bin/mpic++  //notice that we use the mpic++ compiler script
    [files to link]
    -L/usr/lib64/openmpi/lib      //maybe you should add other library paths
    -lscalapack
    -lmpiblacsCinit
    ```

```
-lmpiblacs
-llapack
-lblas
```

Project modalities

Hand in the source code and the report (in .pdf) using the Minerva drop box. Do not zip or otherwise pack these two files. The deadline is **Friday, May 16th 2014 at 12:00 (noon) CET**. This deadline is firm and non-negotiable. Contact Lieven Verbeke (lieven.verbeke@intec.ugent.be) for questions about the project.