PEST, A GENERAL PURPOSE BLUP PACKAGE FOR MULTIVARIATE PREDICTION AND ESTIMATION

Eildert Groeneveld, Milena Kovac, and Tianlin Wang
University of Illinois, Dept. of Anim. Sci., Urbana, IL, 61801 USA

ABSTRACT

PEST is a general purpose Fortran package for multivariate prediction and estimation. It has no inherent limits on number of traits, effects (fixed or random), and covariables. Interactions and polynomials are supported. It further features multiple incidence matrices in multivariate models, heterogeneous residual variances, missing values, inclusion of relationship information with and without inbreeding and genetic group models, hypothesis testing, loading of old solutions, and reading of data from memory or disk. PEST supports 5 solvers which may be combined to perform a two level block iteration. Equation systems are either set up in memory or solved by iteration on data or a combination of both allowing to choose the most efficient procedure relative to the size of the problem. As PEST operates on original codes of up to 16 characters length, no data preparation is required. PEST is driven by a parameter file.

INTRODUCTION

PEST is a software package written in FORTRAN 77 for multivariate prediction and estimation. It covers fixed, random and mixed models. As a tool for statistical evaluation it goes beyond statistical packages like SPSS, SAS and Harvey's LSML which are only univariate or have very limited scope in dealing with mixed models or cannot handle them at all. PEST can be used to rapidly implement multi- and univariate genetic evaluation in breeding programs for a variety of models including animal, sire and sire/dam models. PEST operates on original identification of animals or other class effects which can be up to 16 characters long and also allows data transformation. Thus, no data preparation is required. Because of its structure it covers the whole range of linear equation systems from small to very large, allowing to substitute processing time for memory should this become a bottle neck. PEST is an implementation of the algorithms described by (Groeneveld and Kovac, 1990).

THE USER INTERFACE

PEST is driven by a parameter file which is subdivided into eleven sections some of which are optional (for an example see figure 1). Each section name starts in column 1. Only the more important sections will be described.

The DATA section specifies the data input to PEST. The data file "input.dat" is assumed an ASCII file containing in free format the 8 columns specified. Fixed format can be specified by supplying the starting columns and their length. Class variables (effects) are identified by an estimated number of levels behind each name. Continuous variables are identified by a '0' in the column for the number of levels. No coding is required, original ASCII codes of up to 16 characters in length are accommodated by PEST. For iteration on data - as the default - data are read from memory. Specifying the keyword DISK reads them from disk instead resulting in reduced memory requirements.

Figure 1. Sample parameter file

```
DATA
     INFILE  =  input.dat
     INPUT
        animal              800
        month_of_test        15
        live_weight           0
        month_of_farrowing   51
        age_at_farrowing      0
        backfat               0
        daily_gain            0
        #_born                0
MODEL
   backfat     = live_weight(location) month_of_test location animal
   daily_gain =                        month_of_t  t location animal
   #_born      = age_at_farrowing   month_of_farrowing        animal
VE
   2.2     32.3     .01
  32.3   3200.0   12.00
    .01    12.0    5.60
VG
     VG_FOR animal
        1.2    10.4     .01
       10.4 2240.0     8.8
         .01    8.8     4.3
```

The RELATIONSHIP section specifies relationship data and its use in the model. The input file is assumed to have 3 or 4 columns: animal, sire and dam identification and, optionally, the birthdate of the animal. Together with the keyword 'inbreeding' inbreeding is taken into account (Henderson, 1975). 'GROUP' specifies that a genetic group model (Westell, et al., 1988) is to be employed. If 'DISK' is specified, data will be read from disk during iteration, otherwise from memory. Again, no coding is required.

The MODEL section in figure 1 specifies the statistical model for the evaluation: a multivariate analysis of backfat, daily gain, and number of piglets born with a different incidence matrix for each trait. Backfat has effects liveweight nested within location as a linear regression (polynomials are supported as well), class effects month of test, location and animal. Daily gain has the same suite of effects except for the covariable. Number of piglets born has covariable age at farrowing, and class effects month of farrowing and animal. There is no inherent limit on the number of traits or effects.

The VE section specifies the residual variance covariance matrix. Heterogeneous residual variances can be specified by listing the effect name that is heterogeneous and the level identification in the raw data. Missing values are automatically detected by PEST and the appropriate set of residual covariance matrices is generated. The VG section specifies the covariance matrix for the random effects. An effect that has an entry here is treated as random, otherwise it is considered fixed. The entries in figure 1 turn the above model into an animal model. Covariances for other random effects are specified similarly. There is no inherent limit on the number of random effects.

The SOLVER section allows to choose and combine the solvers appropriate to the model and memory requirement and availability. Five solvers are available: SMP, DENSE, IOC, IOD, and IOD_GS. The first three solve a system of equations with the coefficients stored in memory. IOD performs Gauss-Seidel iteration on unsorted data, in the case of relationship included it becomes Jacobi for that effect. The diagonal blocks of the coefficient matrix are half stored in memory. IOD_GS is the most memory efficient version in that it stores only one diagonal block of order of number of traits. As a default

all effects in the model are placed in IOC. Thus, omitting the solver section amounts to setting up the upper triangular part of the coefficient matrix in sparse format and solving by Gauss-Seidel iteration on coefficients. This is an efficient strategy for small to medium sized systems. The solvers DENSE and SMP are direct procedures and, thus, yield converged solutions. If memory becomes a constraint, it is useful to place a large effect in the IOD solver. Placing a large effect in IOD or IOD_GS does not only requires less memory but usually leads to a substantial increase in the rate of convergence, as the other effects are solved simultaneously in a double block iteration: the first block contains a number of effects with typically few levels like covariables, seasons, while the second block structure comes from solving blocks of order of number of traits within each level in IOD. Default stopping criteria, relaxation parameters (VanVleck and Dwyer, 1985) and maximum number of rounds can be overridden for each of the solvers.

The TRANSFORMATION section allows definition of missing values and scaling of variables. Breeding programs require repeated evaluations adding a few records to each new evaluation. This is particularly crucial in pig breeding where weekly evaluations are required. Iterative procedures allow reusing solutions from earlier rounds, thus starting much closer to the converged solutions. This may dramatically reduce the number of iterations required (Kovac and Groeneveld, 1990). This section allows dumping the current solutions along with their original identification. This is indicated by DUMP_SOL = filename under the STARTING_VALUES section. If these solutions are to be loaded in the current evaluation this has to be specified by LOAD_SOL=filename.

The optional PRINTOUT section species what should be printed and how. Powerful defaults are overruled by specifications in this section. If the keyword BASE_ZERO is specified the average breeding values of the base generation are set to zero and the breeding values for the other animals are transformed accordingly.

The HYPOTHESIS section allows specification of hypothesis. They are specified by supplying the solution number in the mixed model equations, their coefficients and the value to be tested against. Hypothesis testing is restricted to the solvers SMP and IOC.

COMBINATION OF SOLVERS AND THEIR EFFECT ON CPU AND MEMORY REQUIREMENTS

To assess the effect of distribution of effects over the available solvers genetic evaluations were performed on a medium and large size data set on a SUN 4 workstation. Data set 1 is genetic evaluation of 9296 auction boars for the traits backfat (BF) and age at test. Including all relations results in a total of 11152 animals.

Table 1: CPU and memory requirements for data set 1(30016 equations)

	SMP / IOC / DENSE			IOD		IOD_GS		IOC	IOD	IOD_GS		
	solver	relax	m it	relax		relax		# of iterations			CPU	Mem
1	SMP										1:29:39	21288
2	IOC	1.0	2000					2000			1:28:53	9001
3	IOC	1.0	1000	L	1.1	A	1.1	3861	78	78	17:53	1169
4	IOC	1.2	1000	L	1.2	A	1.2	3815	65	65	14:52	1169
5	SMP			L	1.2	A	1.2		65	65	12:07	1520
6	IOC	1.2	200	L D	1.2	A D	1.2	3549	66	66	21:02	625

L - litter; A - animal; D - data read from disk; m it - maximum number of iterations; Mem - memory KB

The model is:

bf = weight(breed) breed season herd litter animal

age = weight(breed) breed season herd litter animal

The litter effect is also quite large with 3695 levels. Thus, the complete set of MME has an order of 30016. Table 1 gives the memory and CPU requirements. Using a direct

solver with all coefficients stored in memory is costly, both, in terms of memory and CPU time. SMP required 1.5 hours and more than 20 MB to solve the problem. IOC required around half the memory but used the same time but did not quite converge to the .01 value in the allocated 2000 rounds stopping, at .03. Processing time and memory requirements drastically decrease once litter (L) and animal (A) are allocated to IOD and IOD_GS solving the other effects simultaneously by either IOC, SMP. The latter is superior in CPU time to IOC which in turn requires less memory. The minimum memory requirement is achieved by line 6, with data and pedigree info being read repeatedly from disk (D).

Data set 2 is an example of a larger problem with close to half a million equations. A genetic evaluation of auction boars for the two traits backfat and age at test has been performed. The model is the same as the previous with an added fixed effect with 9 levels. No relationship was included.

The commonly used Gauss-Seidel iteration on data (first line in table 2) required seven and a half hours to achieve a convergence of .4 . Proceeding to .01 would have taken another 250 round resulting in a total CPU time of around 30 hours. If, however, the effects with small number of levels are placed in IOC, the litter effect in IOD, and the animal effect in IOD_GS CPU time of around 1.5 hours can be achieved as shown in table 2. Again, as for data set 1, the reason lies in a much faster convergence rate if the 1774 equations placed in IOC are solved simultaneously. Memory requirements are around 15 MB. Reading from disk nearly halves this figure. The penalty for reading from disk is about seven minutes.

Table 2: CPU and memory requirements for data set 2 (472252 equations)

	IOC / DENSE			IOD		IOD_GS		IOC	IOD	IOD_GS		
	solver	relax	m it	relax		relax		# of iterations			CPU	Mem
1	DENSE			A D(.4)	1.1				83		7:30:00	11853
2	IOC	1.1	100	L	1.1	A	1.1	1350	18	18	1:19:24	15352
2	IOC	1.1	100	L D	1.1	A D	1.1	1350	18	18	1:36:53	8834

L - litter; A - animal; D - data read from disk; m it - maximum number of iterations; Mem - memory KB

COMPUTER REQUIREMENTS

PEST consists of more than 250 subroutines and more than 11000 line of source written in standard Fortran 77. The run time module apart from the data buffer requires around 500kb. The lower limit for a useful computing platform is an available address space of 1MB. At installation a data buffer has to be declared that suites the available memory on the machine. No upper limit for memory usage exists in PEST apart from operating system / hardware constraints. To date ports exist for SUN workstations and CRAYs - both under UNIX- and Macintosh. Source code copies can be obtained from the authors.

ACKNOWLEDGEMENT

Valuable discussions with Rohan L. Fernando during the development of the package and assistance with statistical problems is gratefully acknowledged.

REFERENCES

Groeneveld, E. and Kovac, M. 1990. J. Dairy Sci. 73:

Henderson, C. R. 1975. J. Dairy Sci. 58: 1917.

Kovac, M. and Groeneveld, E. 1990. J. Anim. Sci. submitted.

VanVleck, L. D. and Dwyer, D. J. 1985. J. Dairy Sci. 68: 1006-1014.

Westell, R. A., Quaas, L. R. and Van Vleck, L. D. 1988. J. Dairy Sci. 77: 1310-1318.