Original article

A reparameterization to improve numerical optimization in multivariate REML (co)variance component estimation

E Groeneveld

*Institute of Animal Husbandry and Animal Behaviour,
Federal Agricultural Research Center,
Höltystr 10, 31535 Neustadt, Germany*

Summary – Multivariate restricted maximum likelihood (REML) (co)variance component estimation using numerical optimization on the basis of Downhill–Simplex (DS) or quasi-Newton (QN) procedures suffers from the problem of undefined 'covariance matrices' as are produced by the optimizers. So far, this problem has been dealt with by assigning 'bad' function values. For this procedure to work, it is implied that the information this 'bad' function value conveys is sufficient to avoid going in the same direction in the following optimization step. To a limited degree DS can cope with this situation. On the other hand QN usually breaks down if this situation occurs too frequently. This contribution analyzes the problem and proposes a reparameterization of the covariance matrices to solve it. As a result, faster converging QN optimizers can be used, as they no longer suffer from lack of robustness. Four real data sets were analyzed using a multivariate model estimating between 17 and 30 (co)variance components simultaneously. Optimizing on the Cholesky factor instead of on the (co)variance components themselves reduced the computing time by a factor of 2.5 to more than 250, when comparing the robust modified DS optimizer operating on the original covariance matrices to a QN optimizer using reparameterized covariance matrices.

multivariate REML / optimization / quasi-Newton / Downhill–Simplex / reparameterization

Résumé – **Un reparamétrage pour améliorer l'optimisation numérique dans une estimation REML multivariate de composantes de variance-covariance.** *L'estimation du maximum de vraisemblance restreinte (REML) des composantes de variance-covariance à l'aide des procédures numériques d'optimisation Simplex-Descendant (SD) ou quasi-Newton (QN) se heurte à la difficulté résultant de la production de matrices de covariances non définies. Jusqu'à présent, cette difficulté a été résolue en attribuant une vraisemblance arbitrairement mauvaise à de telles matrices, de façon à éviter de revenir dans cette même direction dans les étapes suivantes d'optimisation. Dans une certaine mesure, la procédure SD est capable de faire face à cette situation, mais la procédure QN ne*

converge plus lorsque cette situation se reproduit trop souvent. Cette note propose un reparamétrage pour résoudre le problème. Il devient ainsi possible d'utiliser la procédure QN dont la convergence est rapide et la robustesse assurée. Quatre fichiers de données ont été analysés pour estimer simultanément de 17 à 30 composantes de (co)-variance. L'optimisation du facteur de Cholesky au lieu des composantes elles-mêmes réduit le temps de calcul d'un facteur compris entre 2,5 et plus de 500, quand on compare la procédure QN avec reparamétrage des matrices de covariance à la procédure SD modifiée appliquée aux matrices de covariance d'origine.

REML multivariate / optimisation / quasi-Newton / Simplex-Descendant / reparamétrage

THE PROBLEM

In restricted maximum likelihood (REML) (Patterson and Thompson, 1971), maximization of the likelihood is done using either the EM algorithm (Dempster *et al*, 1977) or procedures that do not require explicit derivatives. A problem specific to the latter class of optimizers is addressed in this paper. Graser *et al* (1987) proposed a sampling technique that spanned the complete parameter space for a single trait analysis. Meyer (1989) used a Dowhnill–Simplex (DS) and quasi-Newton (QN) technique, Kovac (1992) modified the DS procedure and also expanded Powell's method of conjugate gradients.

All these authors had to deal with problems arising from parameters and posed by optimizers that lie outside the parameter space. Despite this problem, a host of REML estimates have been reported with a varying number of simultaneous traits (Groeneveld, 1991; Meyer, 1991; Tixier-Boichard *et al*, 1992; Ducos *et al*, 1993; Spilke *et al*, 1993; Mielenz *et al*, 1994).

Consider a mixed linear model with one random effect and no missing values as specified in equations [1–5]. With a residual and an additive genetic effect the log likelihood in equation [6] must be maximized. The computational procedure requires setting up and solving the mixed-model equations (MME).

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{1}$$

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{e} \\ \mathbf{u} \end{bmatrix} \sim N \left(\begin{bmatrix} \mathbf{X}\boldsymbol{\beta} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{V} & \mathbf{R} & \mathbf{Z}\mathbf{G}_\mathrm{A} \\ \mathbf{R} & \mathbf{R} & 0 \\ \mathbf{G}_\mathrm{A}\mathbf{Z}' & 0 & \mathbf{G}_\mathrm{A} \end{bmatrix} \right) \tag{2}$$

$$\mathbf{V} \ = \mathbf{R} + \mathbf{Z}\mathbf{G}_\mathrm{A}\mathbf{Z}' \tag{3}$$

$$\mathbf{R} \ = \mathbf{I}_\mathrm{e} \otimes \mathbf{R}_0 \tag{4}$$

$$\mathbf{G}_\mathrm{A} = \mathbf{A} \otimes \mathbf{G}_0 \tag{5}$$

where:
y   – vector of observations
X   – incidence matrix for all fixed effects

Z – incidence matrix for the animal effect
β – vector of unknown parameters for fixed effects
u – vector of unknown parameters for the animal effect
e – vector of residuals
A – relationship matrix of order number of animals and their known ancestors
$\mathbf{R}_0$ – residual (co)variance among traits
$\mathbf{G}_0$ – covariance matrix for additive genetic effects among traits
⊗ – Kronecker product

This requires a set of covariance matrices for both the residual and the additive genetic components $\mathbf{R}_0$, $\mathbf{G}_0$. Their inverses are used in setting up the MME. After setting up the MME the system of equations may be solved by Cholesky factorization and backward substitution.

$$LV = \mathbf{y}'\mathbf{R}^{-1}\mathbf{y} - \mathbf{y}'\mathbf{R}^{-1}\mathbf{W}\mathbf{b}^0 + \log|\mathbf{C}^*| + n_\mathrm{a} \log|\mathbf{G}_0| + n \log|\mathbf{R}_0| \qquad [6]$$

where

LV – value proportional to the logarithm of the likelihood function
$\mathbf{b}^0$ – solution vector of the MME
$\mathbf{C}^*$ – inverse of the coefficient matrix of the MME
$\mathbf{W}$ – $(\mathbf{X}|\mathbf{Z})$
n_a – number of animals
n – number of observations

In DS, a complete vertex is computed before the optimization begins. The DS procedure used here follows Kovac (1992), and is, thus, very different from the original DS as proposed by Nelder and Mead (1965). Initially, it performs frequent restarts, terminating the iteration at increasing accuracy. This procedure alleviated the well-known problem of the DS to get stuck at suboptimal points. In QN, gradients are required. They may either be supplied in their analytical form or approximated using finite differences (Schnabel *et al*, 1982).

REML is an iterative procedure and so the $\mathbf{R}_0$ and $\mathbf{G}_0$ matrices have to be valid for each round when the MME are set up. Thus, initially, valid covariance matrices have to be provided to the algorithm. For a 2-trait model with 1 additive genetic component the residual and additive genetic covariance matrices $\mathbf{R}_0$ and $\mathbf{G}_0$ of dimension 2 have to be estimated amounting to an optimization in a 6-dimensional parameter space.

However, in the following iteration, new sets of (co)variance are provided by the optimizers. For both DS and QN, the covariance matrices are an unstructured vector of parameters, in this case a vector of 6 values. The constraints of covariance matrices, *ie* that eigenvalues have to be positive (equation [7]), are therefore unknown to the optimizers. Given this background it is not surprising that the outcome of an optimization step, *ie* a new set of (co)variances, is not guaranteed to meet the requirements of covariance matrices. In practical terms, this means that the determinant of the coefficient matrix generated on the basis of these covariance matrices will become less than zero, thus aborting the whole process, because the log of a negative number cannot be taken.

The danger of obtaining undefined 'parameters' from the optimizers obviously increases with the number of traits involved (Ducos *et al*, 1993; Spilke and

Groeneveld, 1994). Furthermore, when the true correlation between the traits is high, the 'covariance matrices' proposed by the optimizers are more likely to lie outside the parameter space, and the true covariance matrices to be located close to the edge of the parameter space.

An obvious (at least in the context of DS) solution to the treatment of undefined covariance matrices is to assign a 'bad' likelihood value, should negative eigenvalues occur. This will tell the optimizer (DS) to avoid this direction in subsequent optimization steps. While this procedure may work reasonably well with sampling-based optimization algorithms, it produces major problems with QN, which requires the function to be continuous and differentiable. If the condition occurs during the proces of approximating the gradients by finite differencing, assigning a 'bad' likelihood will result in a nonsensical gradient. When used during the following optimization step obviously likewise nonsensical directions are chosen. In short, assigning 'bad' likelihood values when undefined covariance matrices occur will often result in aborting the QN optimization step.

We can thus observe that the optimizers that have super linear convergence properties (and are thus much faster than sampling-based procedures like DS (Dennis and Torczon, 1991)) fail increasingly as the dimensionality of the problem increases. Thus, the problem of undefined parameters arising during optimization leads to the paradoxical situation that an efficient class of optimizers can only be used with confidence on small problems with 1 or 2 traits, where computing time is not important and efficiency of optimization not an issue, whereas inefficient sampling-based optimizers, which are at best only linearly converging, must be used on larger problems.

A SOLUTION

Part of the problem can be solved by performing a constrained optimization. This is relatively easy for the variances in which the constraint is only that positive numbers be chosen. However, no technique seems to be available to impose a set of constraints such that no negative eigenvalues λ occur for a subset of the dimensionality of the optimization space as given in equation [7].

$$\lambda_i(\mathbf{R}_0, \mathbf{G}_0) \geqslant 0 \forall i \qquad [7]$$

Instead, we propose to perform the optimization on the Cholesky factor of the covariance matrices. Let:

$$\mathbf{R}_0 = \mathbf{C}_r' \mathbf{C}_r \qquad [8]$$

and

$$\mathbf{G}_0 = \mathbf{C}_g' \mathbf{C}_g \qquad [9]$$

Thus, instead of optimizing on the $\mathbf{R}_0$, its Cholesky factor $\mathbf{C}_r$ is used applying the same operation to all covariance matrices in the model. Operationally, this implies the following steps:

1) User supplies initial covariance matrices.
2) The Cholesky factorization is performed on all covariance matrices.
3) The optimizer is called with the Cholesky factors.

4) The function SMME is called by the optimizer. This function sets up and solves MME and computes the likelihood value passing the Cholesky factor as parameters.

5) SMME computes the original $\mathbf{R}_0$ and $\mathbf{G}_0$ from the factors, sets up and solves the MME and computes the likelihood value.

6) Control refers back to the optimizer (*Step 4*) to have it decide on the next step based on the last *LV* and the current set of factors.

This process results in a matrix that always has the properties of a covariance matrix, irrespective of the values that the optimizers may come up with. This is because:

$$\mathbf{x}'\mathbf{R}_0\mathbf{x} = \parallel \mathbf{C}_r\mathbf{x} \parallel^2 \geqslant 0 \forall \mathbf{x} \qquad [10]$$

As a result, undefined 'covariance' matrices cannot occur.

A special case arises when certain covariances are not estimable. This may happen with residual covariances when measurements on different traits are mutually exclusive, a situation frequently occurring in joint analyses of data from 2 test environments. During optimization, the non-defined component is skipped, thereby reducing its dimensionality. The current implementation of the reparameterization computes the Cholesky factor on the basis of the complete covariance matrix with zeros inserted for the undefined parameters. Although the values of the Cholesky factor depend on the zero inserted for undefined convariances, the same optimum has always been reached for optimization on the reparameterized and on the original scale.

RESULTS

Four mutivariate runs are given to assess the effect of optimizing on the Cholesky factors. The timings listed refer to a Hewlett Packard 7100 computer system with the 99 MHz processor.

Run 1 was an analysis of a selection experiment in chickens with 5 traits, 2 fixed effects, year and barn, and the animal component. As no traits were missing a canonical transformation was performed, which reduced the number of numerical operations dramatically. Thirty components must be estimated simultaneously. The data set was kindly supplied by C Hagger (Swiss Federal Institute of Technology).

Run 2 was an analysis of 3 meat quality traits on around 2 000 pigs. Not all records were complete. There were 6 class effects and 1 covariable in the model, which was identical for all traits. Random components were common litter and animal, resulting in 18 components to be estimated (Dietl *et al*, 1993).

Run 3 was an analysis of 4 048 station test records from swine comprising the 4 traits daily gain, feed conversion efficiency, valuable cuts, and a meat quality parameter. There were 2 covariables, 4 fixed class effects, 1 random litter component and the random correlated animal effect. The covariable weight at the end of test was not defined for the trait daily gain. In all, 30 components were to be estimated.

Run 4 analyzed 2 traits from field test in pigs and a third from a test station measured on different animals. Three class effects comprised common litter and animal as random components and herd–year–season as a fixed effect and a covariable weight for 1 trait only. Because daily gain was measured either in

the field or in test stations, but not in both environments, the corresponding residual covariance component was not estimable. This results in 17 variances and (co)variances to be estimated. All 4 models included the full relationship among animals. A summary description of the runs is given table I.

Table II shows results from DS and QN using this procedure. The number of function evaluations declines substantially for both.

Table I. Data structure.

Run	No of effects	No of traits	No of components	No of equations	Different models	Missing values
1	3	5	30	49 100	N	N
2	7	3	18	12 720	N	Y
3	8	4	30	33 612	Y	N
4	4	3	17	48 084	Y	Y

Table II. Number of function evaluations.

Run	DS	Original scale		Chol factor			Performance	
		Illegal points	QN	Illegal points	DS	QN	Speedup^2	CPU_{QN}
1	536 860	8 557	226^1	30	26 749	2 125	252.6	2:58:52
2	8 524	3 084	207^1	20	4 986	932	9.1	21:29
3	21 739	5 332	1 148^1	113	10 376	1 387	15.7	2:22:06
4	3 658	13	1 065	96	3 845	1 466	2.5	6:30:22

^{1} Optimization aborted; ^{2} speedup is relative to DS on original scale.

The general picture shows a much smaller number of function evaluations for the QN optimizer compared with DS. This is to be expected as QN optimizers have a super linear rate of convergence, *ie* the number of iterations decreases for a fixed increase in accuracy as convergence is approached. However, QN optimizers aborted in 3 out of the 4 models when optimization was done on the original (co)variance matrices attesting to the problems of undefined parameters outlined above.

With optimization on the components of the (co)variance matrices, only the modified DS succeeded in locating the optimum (DS column in table II). The costs, however, were large, particularly for run 1 (at 3.7 s per function evaluation). This converged at the number of rounds indicated with an accuracy of 10^{-6} on the distance between the worst and best parameter set in the vertex, but had still not quite reached the optimum. Interestingly, apart from the 8 557 illegal points that were produced by the DS optimizer, a loss of rank of the coefficient system occurred 23 120 times. This condition is encountered during factorization when a zero pivot is detected. In this case, factorization is aborted and, again, a 'bad' function value assigned to the current set of parameters. Losses of rank

are partly due to badly conditioned covariance matrices, that just about pass the test for positive eigenvalues, but still do not render the coefficient matrix positive definite. It is thus an effect of limited accuracy on digital computers for these 2 tests. Obviously, this phenomenon also produced a large amount of directional misinformation, wasting a substantial amount of CPU time.

Runs 1–3 had to cope with a large number of undefined parameters nearly 1 undefined for the 3 or 4 that were within the parameter space, resulting in a substantial amount of conflicting directional information. The situation was better in run 4, where DS only left the parameter space 13 times.

QN aborted in the first 3 runs after a varying number of function evaluations because of a discontinuous function surface introduced by the 'bad' function values given to undefined points. Only run 4 was completed successfully, despite the occurrence of 96 illegal points.

With optimization on the Cholesky factor of the (co)variance matrices the picture changes drastically. The extent to which the DS optimizer benefited was dependent on the number of illegal points prior to reparameterization. Accordingly, run 1 finished in less than a 20th of the time converging to the best point, while run 2 and 3 finished twice as fast. Only run 4 did not benefit, which was to be expected. In fact, the reparameterization resulted in more function evaluations. Whether this is a chance result or an indication of a more general pattern, cannot be conclusively decided at this point. However, experience from a large number of other runs has shown a substantial variability in the number of function evaluation till convergence for seemingly identical models in terms of number of parameters. It is therefore assumed that the observed slow down is more likely a chance result than a general phenomenon.

QN found the solution in all runs. Depending on the number of illegal points encountered before, the speed up was substantial. This was computed as the ratio of the number of function evaluations of QN with Cholesky factor *versus* DS on the original scale. If QN and DS can only be used on the basis of the original covariance matrices, only DS will reliably give results. Thus, this is considered as the reference point.

Run 1 was particularly impressive: with DS operating on the original covariance matrices, optimization was basically impracticable as computing time was at around 3 weeks or more of CPU time prohibitively high (and the best point was still not quite reached). QN, on the other hand, solved the problem in less than 3 h. The other 3 runs showed a speed-up factor of between 2.5 and 15.7.

CONCLUSIONS

Multivariate REML estimates for general statistical models suffer from high computational demands and the rather low dimensionality in terms of number of traits that they could handle. In most cases, only bivariate analyses could be done with general models producing suboptimal covariance matrices of higher order (Spilke and Groeneveld, 1994). The indicated improvement in speed by optimizing on the Cholesky factor of the covariance matrices will have a 2-fold effect: firstly, it will speed up convergence for any class of optimizer; and secondly, it will lead to a shift from sampling-based optimizers (that were previously considered robust) to

much more efficient QN algorithms, which will considerably extend the scope of multivariate REML (co)variance component estimation. Most importantly, it will allow users to increase the dimensionality of the models that can be handled, thus helping close the gap between the number of traits used in genetic evaluation and (co)variance component estimation.

This analysis was done with the VCE program, a multivariate, multimodel REML (co)variance component estimation package (Groeneveld, 1994), which is available for research purposes on the anonymous ftp server under the Internet number 192.103.38.1.

ACKNOWLEDGMENT

REFERENCES

Dempster APN, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data *via* the EM algorithm. *J R Stat Soc Ser B* 1-38

Dennis JE, Torczon V (1991) Direct search methods on parallel machines. *SIAM J Sci Comput* 1, 448

Dietl G, Groeneveld E, Fiedler I (1993) Genetic parameters of muscle structure traits in pigs. *In: 44th Annual Meeting of the European Association for Animal Production*, Aarhus, Denmark

Ducos A, Bidanel J, Ducrocq V, Boichard D, Groeneveld E (1993) Multivariate restricted maximum likelihood estimation of genetic parameters for growth, carcass and meat quality traits in French Large White and French Landrace Pigs. *Genet Sel Evol* 25, 475

Graser HU, Smith SP, Tier B (1987) A derivative-free approach for estimating variance components in animal models by restricted maximum likelihood. *J Anim Sci* 64, 1362

Groeneveld E (1991) Simultaneous REML estimation of 60 covariance components in an animal model with missing values using a Downhill–Simplex algorithm. *In: 42nd Annual Meeting of the European Association for Animal Production*, Berlin, Germany

Groeneveld E (1994) REML VCE – a multivariate multimodel restricted maximum likelihood (co)variance component estimation package. *In: Proceedings of an EC Symposium on Application of Mixed Linear Models in the Prediction of Genetic Merit in Pigs* (E Groeneveld, ed) (in press)

Kovac M (1992) Derivative-free methods in covariance component estimation. Ph D thesis, University of Illinois at Urbana-Champaign, USA

Lindstrom MJ, Bates DM (1988) Newton-Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. *J Amer Stat Assoc* 83, 1014

Meyer K (1989) Restricted maximum likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. *Genet Sel Evol* 21, 317

Meyer K (1991) Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. *Genet Sel Evol* 23, 67

Mielenz N, Groeneveld E, Müller J, Spilke J (1994) Simultaneous estimation of covariances with REML and Henderson 3 in a selected chicken population. *Br Poult Sci* (in print)

Nelder JA, Mead R (1965) A simplex method for function minimization. *Computer J* 7, 308

Patterson HD, Thompson R (1971) Recovery of inter-block information when block sizes are unequal. *Biometrika* 58, 545

Schnabel R, Koontz J, Weiss B (1982) A Modular System of Algorithms for Unconstrained Minimization. Technical Report CU-CS-240-82, Comp Sci Dept, Univ Colorado, Boulder, CC, USA

Spilke J, Groeneveld E (1994) Comparison of four multivariate REML (co)variance component estimation packages. *In: 5th World Congress on Genetics Applied to Livestock Production, Guelph,* vol 22, 11-14

Spilke J, Groeneveld E, Mielenz N (1993) Monte-Carlo simulation in variance-covariance-component estimation in mixed linear multiple trait models. *Arch Anim Breed* 6, 679

Tixier-Boichard M., Boichard D, Bordas A, Groeneveld E (1992) Genetic parameters of residual feed intake in adult males and females from a layer Rhode Island Red Population. *In: 19th World's Poultry Congress,* 209-210, Amsterdam, The Netherlands