Swiss Federal Institute of Technology Zurich

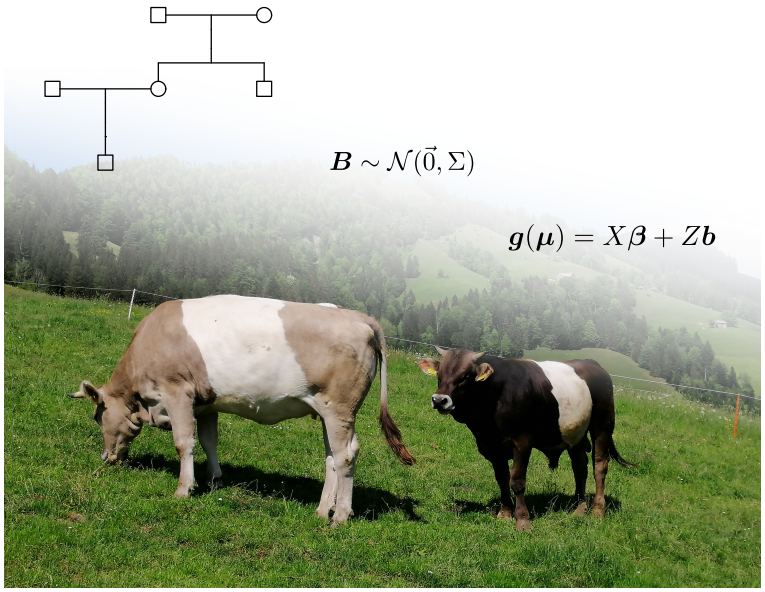
Seminar for Statistics

Department of Mathematics

Master Thesis Spring Semester 2020

Reto Zihlmann

Generalized Linear Mixed Effects Models in Genetic Evaluations



Submission Date: August 31th 2020

Co-Adviser Dr. Peter von Rohr
Adviser: Dr. Lukas Meier

Abstract

Health and reproductive traits are increasingly important in cattle breeding programms all around the world. In contrast to productivity traits, health and reproductive traits are often measured on a nominal or ordinal scale which makes classical breeding value estimation via linear mixed effects models (LMMs) inappropriate. Despite extensive litherature, application of generalized linear mixed effects models (GLMMs) and threshold models in practical breeding value estimation remains challenging due to limited availability of software implementation for this specific purpose. In this study we present available software packages, show their weaknesses and implement improvements. The implementations were tested on simulated data sets and compared with respect to computation time and accuracy of the estimated breeding values. The best implementations were applied to real-world data sets of some major Swiss cattle populations. Traits of interest were multiple birth, early-life calf survival and carcass confirmation scores. GLMMs and threshold models clearly improved the prediction of breeding values compared to LMMs when applied to simulated binary and ordinal traits. Bayesian implementations performed relatively slow for small data sets but returned trustworthy standard errors of the estimated breeding value by accounting for the uncertainty of variance component estimation. The improvements also came at a higher computational cost, however, the cost was largely reduced by assuming known variance components. A similar strategy was successfully applied to the much larger real world data sets by separately estimating variance components and animal breeding values. This study shows that GLMMs and threshold models can and should be applied for non-normal traits in order to improve the properties of estimated breeding value and obtain unbiased heritability estimates which allow for well-informed constructions of selection indices.

Contents

List of Figures

List of Tables

Notation

Throughout this thesis, R packages and objects are written in gray boxes with typewriter font. Functions are followed by parenthesis, for example, `lmer()`. Random variables are denoted as upper-case letter (e.g. G) and the corresponding lower-case letter denotes a particular value of the random variable (e.g. g). Vectors are indicated by bold letters (e.g. $\boldsymbol{b}$), matrices by uppercase letters (e.g. A). Special symbols include I_n which represents the identity matrix of dimension n, $\vec{0}$ which is a vector of appropriate dimensions with all elements equal to zero and $\otimes$ which represents the Kronecker product. Results are followed by a proof of the result, which ends with an open square ($\square$). A filled square ($\blacksquare$) denotes the end of an example.

The following list includes all abbreviations

BLUP	Best linear unbiased prediction
EDM	Exponential distribution model
GLM	Generalized linear model
GLMM	Generalized linear mixed effects model
IBD	Identical by descent
IBS	Identical by state
INLA	Integrated nested Laplace approximation
LMM	Linear mixed effects model
MCMC	Markov chain Monte Carlo
NLMM	Nonlinear mixed effects model
PIRLS	Penalized iteratively reweighted least squares
PQL	Penalized quasi-likelihood approximation

Chapter 1

Introduction

In the second half of the 20th century, livestock breeding adopted modern statistical approaches in order to select animals for breeding based on phenotypic observations. In a series of papers, Henderson developed the theoretical foundations of mixed models in animal breeding and provided a methodology to estimate best linear unbiased predictions (BLUPs) of the random effects (summarized in Henderson 1982). He also provided efficient methods to include the pedigree derived correlation between related animals into the model. The methodology was quickly implemented in breeding programs all over the world and lead to a tremendous increase in response to selection (Hill 2014). With increasing availability of genomic data, Meuwissen, Hayes, and Goddard (2001) presented an adapted mixed model methodology which directly makes use of genetic information. Selecting animals based on genetic breeding values shortens the generation interval and is expected to at least double the rate of genetic gain in the dairy industry (Hayes et al. 2009). Both methodologies, pedigree based and genomic based, primarily target the evaluation of quantitative traits with a normal distribution conditional on the predictor variables.

Traditionally, livestock breeding strongly focused on quantitative productivity traits as for example milk yield in dairy cattle or daily liveweight gains in beef cattle. However, other traits, often referred to as secondary or nonproduction traits, are also of great economic importance and have gained attention in recent years. Specifically, health and reproductive fitness directly affect dairy production profitability and have an effect on animal welfare at the same time. Years of breeding on productivity traits led to the emergence of production diseases such as health and fertility problems (Miglior et al. 2017). Breeding focus has gradually shifted and nowadays includes traits that are associated with robustness and sustainability (O'Neill, Swain, and Kadarmideen 2010). Important secondary traits in dairy cattle include mastitis resistance, resistance to lameness, fertility, calving ease and longevity (Kadarmideen, Thompson, and Simm 2000, Neuenschwander et al. 2005). These traits are often discrete or measured on a nominal or ordinal scale. For example, clinical mastitis can be recorded as a binary variable, measuring the presence or absence of the disease or as a count variable, measuring the number of mastitis cases in a lactation period (Vazquez, Gianola, et al. 2009).

Binary or count variables are clearly not conditionally normal and the classical methods to estimate breeding values do not apply. Fortunately, generalized linear mixed effects models (GLMMs) provide a well-researched generalization of mixed models which can be used to

extend the classical methods to allow for a wide variety of trait distributions, including the binomial distribution for binary data or the Poisson distribution for count data. Threshold models further extend the scope to ordinal response variables by assuming a continuous underlying variable called liability and thresholds which separate the liability into regions of different trait expression.

A variety of studies already applied GLMMs and threshold models in genetic evaluations. A review on the early development of GLMMs in cattle breeding can be found in Tempelman (1998). GLMMs were used to model binary data such as health traits (Yin et al. 2014) and count data such as mastitis cases (Rodrigues-Motta et al. 2007, Vazquez, Perez-Cabal, et al. 2012). First theoretical considerations of the threshold model are presented in Gianola and Foulley (1983). Threshold models have successfully been applied to categorical data such as fertility (Kadarmideen, Thompson, and Simm 2000), multiple birth (Van Tassell, Van Vleck, and Gregory 1998) or number of services to conception (Chang et al. 2006).

Despite above mentioned examples, in practice, breeding organizations often model non-normal response variables with simple linear mixed effects models (LMMs). The theoretical foundation for the genetic evaluation of non-normal traits are well understood but the practical application of theses models is still hampered by the lack of software implementations which are easy to use, computationally efficient and specifically designed to include the complex correlation structure between animals. On the other hand, software solutions for LMMs are readily available, familiar and well understood. Several studies showed that evaluating binary data with LMMs might still result in a reasonable ranking of animals (Negussie, Strandén, and Mäntysaari 2008, Vazquez, Perez-Cabal, et al. 2012). Still, using LMMs on non-normal data violates the basic assumptions of normally distributed residual errors, such that inferential results based on LMM are invalid. Predictions will likely not persevere the support of the response especially for discrete response variables. A bounded support often also implies a conditional variance which depends on the conditional mean and a complex relationship between the linear predictor and the conditional mean. Both are not natively supported by LMMs.

Qualitas, the company responsible for breeding value estimation of the major Swiss cattle breeds, faces the same challenges in the process of estimating breeding values for two binary and one categorical traits:

- *Multiple birth* poses a health risk for calf and dam. Qualitas investigates the possibility to decrease multiple birth rate in the Swiss Braunvieh population which is mainly used for milk production.
- *Early-life calf survival* is an important health trait and covers a life period which is currently neglected in most Swiss cattle breeding programs. An increase in early-life calf survival would have positive economic and animal welfare consequences.
- *Carcass confirmation* describes the visual classification of carcasses into seven quality classes. The trait has a major influence on the value of animals for slaughter and is an important selection criterion in beef cattle.

Qualitas modeled the traits with classical LMMs in order to obtain breeding values for all animals. The fitted models are characterized by relatively low explanatory power and large standard errors of the predicted breeding values. The bad performance of classical LMMs motivated Qualitas to evaluate the same traits with more appropriate models and

possibly include GLMMs and threshold models in their routine breeding value estimation.

The aim of this study is to assess GLMMs and threshold models for genetic evaluations of traits with non-normal conditional distribution. To this end we start with a detailed theoretical description of the models and describe their estimation from a likelihood and a Bayesian perspective (Chapter 2). Current implementations of GLMMs and threshold models for genetic evaluations in R (R Core Team 2020) were analyzed and improved (Chapter 3). Different implementations were tested on simulated data sets (Chapter 4) and finally applied on the Qualitas data sets (Chapter 5). The thesis concludes with a discussion of the results and implications for future breeding value estimation at Qualitas (Chapter 6 and 7).

Chapter 2

Theory

2.1 Quantitative Genetics

2.1.1 Decomposing the Phenotype

Animals show variation of morphological, physiological and behavioral properties between species and individuals. Those properties often referred to as traits mainly determine whether a certain species or a certain individual is considered for production in an agricultural setting. The collection of all traits is called *phenotype*. The phenotype is to some degree pre-determined by genetic factors which are summarized by the term *genotype*. The remaining variability in the phenotype comes from environmental factors. Measuring a single trait in metric units results in a value called phenotypic value P. Following the influencing factors of the phenotype, we can decompose the phenotypic value into a genotypic value G and environmental deviation E. This is usually summarized by the identity

$$P = G + E.$$

The genotypic value is the average phenotypic value of an animal with a certain genotype if it would have been observed in different environments. The deviation from that average value is caused by the environmental deviation. The equation above, as it is stated in many textbooks (e.g. Falconer and Mackay 1996), suggests a pure additive effect of the genotype and the environment on the phenotype. This assumption does usually not hold in practice where we observe *genotype-by-environment* interaction.

Further decomposing G requires a closer look on how genetic information is inherited. Unlike crops, livestock is usually diploid meaning that there are always pairs of homologous chromosomes containing corresponding loci. Each locus has therefore two possibly different forms called alleles. Different alleles may have different effects on the phenotypic value.

A common statistical approach to estimate the effect of one quantity on another is via linear regression. We define the allele content N_i as the number of copies of a particular allele i in the genome of an animal. In case of diploid organisms $N_i \in \{0, 1, 2\}$. The regression of the penotypic value on the allele content can be written as

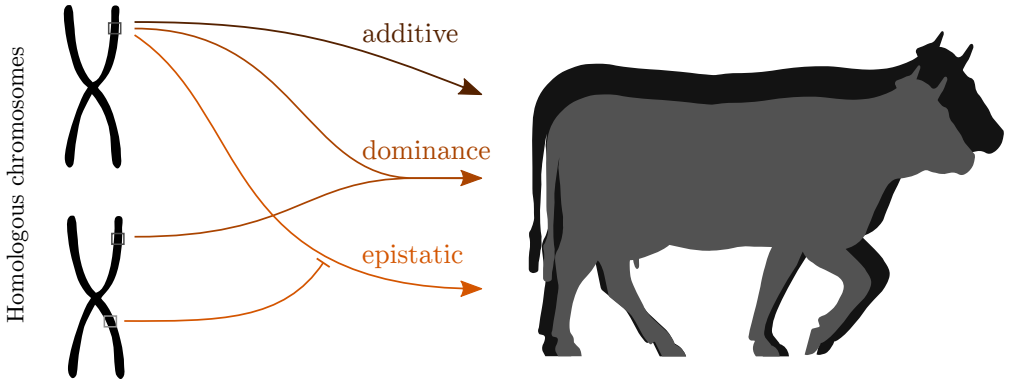


Figure 2.1: Additive, dominance and epistatic effect of alleles.

$$G = \mu_G + \sum_{i=1}^{n} \alpha_i N_i + \delta \tag{2.1}$$

were μ_G is the mean genotypic value in the population, n is the number of alleles, α_i is the slope of predictor N_i and δ is the residual error. The slope α_i is commonly referred to as *additive genetic effect* of allele i. In the case of random mating, it can be shown that the additive genetic effect of allele i is equal to the difference in the average genotypic value between animals containing allele i and the population (e.g. Lynch and Walsh 1998). Formally,

$$\alpha_i = \bar{G}_i - \mu_G$$

where $\bar{G}_i$ is the mean genotypic value of all animals containing allele i. The additive effect of allele i can therefore be seen as the average effect of replacing a random allele by allele i. The effects depends on allele frequencies, the population structure, the environment and many more factors. As an example, assume a population where the allele of interest is already fixed. Replacing an allele by the same allele has no effect on the phenotype. Therefore the additive effect would be zero even though the allele might have a strong causal relationship with the phenotype. The additive effect might be largely different in another population with different allele frequencies.

Because each locus has two alleles the effect of replacing the allele on only one chromosome will likely depend on the allele on the second chromosome. This kind of interaction within one locus is called dominance. Additionally the effect might be influenced by other loci. The interaction between different loci is called epistasis (Figure 2.1). Dominance and epistasis entirely explain the remaining error term δ in Equation (2.1). Therefore, we can further decompose the phenotypic value as

$$\begin{aligned} P &= G + E \\ &= \mu_G + B + D + I + E \end{aligned} \tag{2.2}$$

where $B = \sum_{i=1}^{n} \alpha_i$ is the sum of each alleles additive effects, D is the dominance deviation, I is the epistatic deviation and E is the environmental deviation. In a similar way we can also decompose the variance of the phenotype. As already shown by Fisher (1918), there is no covariance between B and D. From today's perspective this can easily be verified by viewing the interaction deviations as residual error of regression (2.1). Further assuming random mating, independent segregation of loci and no covariance between genotypic and environmental variance, we can decompose the phenotypic variance as

$$\sigma_P^2 = \sigma_B^2 + \sigma_D^2 + \sigma_I^2 + \sigma_E^2$$

2.1.2 Breeding Value

Only one of the two alleles in diploid species is passed to the gamete which determines the offspring's genotype. In randomly mating populations, it is combined randomly with a second allele at the same locus and further alleles at different loci. All interaction effects as dominance deviation and epistatic deviation will therefore change during recombination. Only the additive effect of each allele will be conserved when passed from parents to offspring. For a breeder with the goal to improve the phenotype over several generations, it is therefore only B, the sum of all additive effects, which determines the potential of an animal as parent. For this reason B is called the breeding value.

In many textbooks (e.g. Willam and Simianer 2017) the breeding value is defined as twice the expected deviation of its offspring's mean phenotypic value from the population mean. What could be the reason for defining it based on related animals rather than directly based on the sum of allele effects? To answer this question we have to consider that estimating components of decomposition (2.2) is impossible having only observed a single animal in a single environment. Only by using phenotypic information of related animals, we can reliably distinguish between additive effects and deviations due to interaction within or between loci as well as the environment. A crucial information for estimating the additive effects is the expected phenotypic covariance due to the similar additive effects of related animals.

2.1.3 Covariance between Related Animals

Assume that for each animal the alleles are sampled from the pool of alleles in the population. Some animals will by chance sample the same allele at one locus. We call the allele of those animals to be *identical by state* (IBS). Animals with alleles IBS also experience the same effect of these alleles on the phenotype. Offsprings will not sample randomly from the pool of alleles in the population but from their parents. Some of the alleles which are identical between parent and offspring will not only be IBS but also *identical by descent* (IBD). IBD alleles are identical not because by chance two animals sampled the same allele from the population but because they were directly passed from parent to progeny. The correlation in phenotypic value between relatives is directly related to alleles which are IBD.

In the previous section we have seen that only the additive genetic effect is passed from parents to their offsprings. By combining all remaining terms we rewrite Equation (2.2) as

$$P = \mu_G + B + e$$

where $e = D + I + E$ containing all effects which are not passed from parents to their offsprings. It follows that the correlation between the phenotypic value of a parent p and an offspring o is

$$\begin{aligned}\mathrm{Cov}\left(P_p, P_o\right) &= \mathrm{Cov}\left(B_p + e_p, B_o + e_o\right) \\ &= \mathrm{Cov}\left(B_p, B_o\right) + \underbrace{\mathrm{Cov}\left(B_p, e_o\right) + \mathrm{Cov}\left(e_p, B_o\right) + \mathrm{Cov}\left(e_p, e_o\right)}_{=0}.\end{aligned}$$

Assuming random mating, independent segregation of loci and no covariance between genotypic and environmental variance all covariances containing an e term become zero. Usually, non of these assumption is fulfilled in an animal breeding setting. Still, it is common to make these assumptions in order to get a parsimonious model with not too many parameters.

The breeding value B was defined as deviation from population mean μ_G. The expected value of B is zero and we can further simplify

$$\begin{aligned}\mathrm{Cov}\left(B_p, B_o\right) &= \mathbf{E}\left[B_p B_o\right] - \underbrace{\mathbf{E}\left[B_p\right]\mathbf{E}\left[B_o\right]}_{=0} \\ &= \mathbf{E}\left[B_p B_o\right].\end{aligned}$$

Writing out the breeding values as sums of additive effects of single loci

$$\begin{aligned}\mathbf{E}\left[B_p B_o\right] &= \mathbf{E}\left[(\alpha_{p11} + \alpha_{p12} + \alpha_{p21} + \alpha_{p22} + \dots)(\alpha_{o11} + \alpha_{o12} + \alpha_{o21} + \alpha_{o22} + \dots)\right] \\ &= \underbrace{\mathbf{E}\left[\alpha_{p11}\alpha_{o11}\right] + \mathbf{E}\left[\alpha_{p11}\alpha_{o12}\right] + \mathbf{E}\left[\alpha_{p12}\alpha_{o11}\right] + \mathbf{E}\left[\alpha_{p12}\alpha_{o12}\right]}_{\text{Terms within first loci}} \\ &\quad + \underbrace{\mathbf{E}\left[\alpha_{p21}\alpha_{o21}\right] + \dots}_{\text{within second loci}} + \underbrace{\dots}_{\text{within further loci}} \\ &\quad + \underbrace{\mathbf{E}\left[\alpha_{p11}\alpha_{o21}\right] + \dots}_{\text{across loci}}\end{aligned} \tag{2.3}$$

where α_{ijk} is the additive effect of allele at loci j on chromosome k in animal i. All cross terms where $j \neq j'$ cancel as we assume independent segregation of loci and the expected value of α is zero, therefore

$$\mathbf{E}\left[\alpha_{pjk}\alpha_{oj'k}\right] = \mathbf{E}\left[\alpha_{pjk}\right]\mathbf{E}\left[\alpha_{oj'k}\right] = 0. \tag{2.4}$$

We define the *coefficient of coancestry* Θ_{po} as the probability of an allele to be IBD in animal p and o. For terms where $j = j'$ we can write

$$\mathbf{E}\left[\alpha_{pjk}\alpha_{oj'k'}\right] = \begin{cases} \mathbf{E}\left[\alpha_{pjk}^2\right] & \text{with probability } \Theta_{po} \\ \mathbf{E}\left[\alpha_{pjk}\right]\mathbf{E}\left[\alpha_{oj'k'}\right] = 0 & \text{with probability } 1-\Theta_{po}. \end{cases} \tag{2.5}$$

Based on our assumptions, the variance of the breeding value can be separated into the contributions of the individual loci

$$\begin{aligned} \sigma_b^2 = \mathrm{Var}\left(B\right) &= \mathrm{Var}\left(\alpha_{i11} + \alpha_{i12} + \alpha_{i21} + \ldots\right) \\ &= \mathrm{Var}\left(\alpha_{i11}\right) + \mathrm{Var}\left(\alpha_{i12}\right) + \mathrm{Var}\left(\alpha_{i21}\right) + \ldots \\ &= \mathbf{E}\left[\alpha_{i11}^2\right] + \mathbf{E}\left[\alpha_{i12}^2\right] + \mathbf{E}\left[\alpha_{i21}^2\right] + \ldots \\ &= 2\left(\mathbf{E}\left[\alpha_{i1k}^2\right] + \mathbf{E}\left[\alpha_{i2k}^2\right] + \ldots\right) \end{aligned} \tag{2.6}$$

Inserting (2.4) and (2.5) into (2.3) and using (2.6) leads to

$$\begin{aligned} \mathrm{Cov}\left(B_p, B_o\right) = \mathbf{E}\left[B_p B_o\right] &= 4\Theta_{po}\mathbf{E}\left[\alpha_{i1k}^2\right] + 4\Theta_{po}\mathbf{E}\left[\alpha_{i2k}^2\right] + \ldots \\ &= 4\Theta_{po}\left(\mathbf{E}\left[\alpha_{i1k}^2\right] + \mathbf{E}\left[\alpha_{i2k}^2\right] + \ldots\right) \\ &= 4\Theta_{po}\frac{\sigma_b^2}{2} \\ &= 2\Theta_{po}\sigma_b^2 \end{aligned} \tag{2.7}$$

which is the covariance between any two related animals. The coefficient of coancestry Θ can be directly calculated from the pedigree for any pair of animals. The calculation becomes increasingly complex for larger pedigrees with inbreeding. A detailed derivation can be found in Lynch and Walsh (1998) Chapter 7 or Falconer and Mackay (1996) Chapter 5. The basic idea of the calculation includes the following steps

1. Represent the pedigree by a graph
2. Follow the paths which connect the animals via a single common ancestor, as well as paths which connect those common ancestors
3. Calculate the probability for inheriting a certain allele along each path
4. Sum up all probabilities

A second possibility is to compute recursively all coefficients in the pedigree starting from the oldest animals. The coefficient between two animals can be directly calculated from the coefficients of their parents.

2.2 Basic Model

2.2.1 Breeding Values in a Linear Model

Our goal is to set up a model which allows predicting B. Following Equation (2.2), the phenotypic value can be decomposed into different sources. The environmental variance σ_E^2

can make up for a large proportion of the total phenotypic variance σ_P^2 of some traits. Part of σ_E^2 can be explained by adding environmental explanatory variables to the model and estimating their fixed effects. All remaining random terms except for B can be combined to the residual error

$$\begin{aligned} P &= \mu_G + B + D + I + E \\ &= \underbrace{\mu_G + E_{\text{fix}}}_{\text{fixed effects}} + B + \underbrace{D + I + E_{\text{rand}}}_{\text{random residuals}}. \end{aligned}$$

Turning to vector matrix notation, we can write

$$\boldsymbol{Y} = X\boldsymbol{\beta} + Z\boldsymbol{B} + \boldsymbol{\varepsilon} \tag{2.8}$$

where $\boldsymbol{Y}$ is the vector of random phenotypic values, $X\boldsymbol{\beta}$ contains the model matrix and vector of fixed effects, Z is the model matrix relating the breeding value of an animal to the observations of the same animal, $\boldsymbol{B}$ is the random vector of breeding values and $\boldsymbol{\varepsilon}$ is the vector of random errors. Note that Equation (2.8) contains two random terms on the right hand side. Let's first have a closer look at the vector of breeding values. We have seen that breeding values are defined as deviation from the population mean, therefore $\mathbf{E}[\boldsymbol{B}] = \vec{0}$. Using Equation (2.7), the variance covariance matrix is given by

$$\operatorname{Var}(\boldsymbol{B}) = 2 \begin{pmatrix} \Theta_{11} & \dots & \Theta_{1n} \\ \vdots & \ddots & \vdots \\ \Theta_{n1} & \dots & \Theta_{nn} \end{pmatrix} \sigma_b^2 = A\sigma_b^2.$$

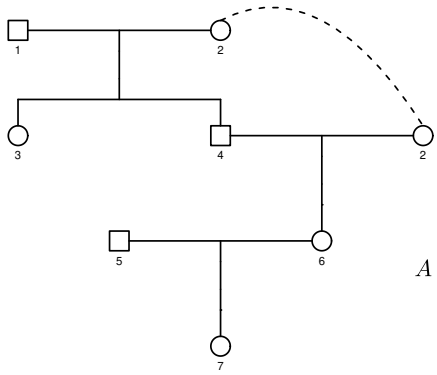
The matrix A contains twice the coefficient of coancestry between each animal. Its use in the numerator of the formula to calculate classical relationships led to the name *additive numerator relationship matrix*. Figure 2.2 shows a small example of a pedigree with the corresponding additive numerator relationship matrix. The breeding value is the sum of many small allele effects, therefore it might be justified to assume it follows a normal distribution. The second random term in Equation (2.8) is the vector of residual errors which is also assumed to be the result of many small effects. Because it contains deviations from the population mean and all components are assumed independent even between related animals we assume

$$\boldsymbol{\varepsilon} \sim \mathcal{N}_{\text{n}}\left(\vec{0}, I_n\sigma^2\right). \tag{2.9}$$

Using all this information we can set up the final model as

$$\begin{aligned} (\boldsymbol{Y} \mid \boldsymbol{B} = \boldsymbol{b}) &\sim \mathcal{N}_{\text{n}}\left(X\boldsymbol{\beta} + Z\boldsymbol{b}, I_n\sigma^2\right) \\ \boldsymbol{B} &\sim \mathcal{N}_{\text{q}}\left(\vec{0}, A\sigma_b^2\right) \end{aligned} \tag{2.10}$$

which is a linear mixed effect model with an unusually complex correlation structure in the random effect vector $\boldsymbol{B}$.



□ sire ○ dam

$$A = \begin{pmatrix} 1.000 & & & & & & \\ 0.000 & 1.000 & & & & & \\ 0.500 & 0.500 & 1.000 & & & & \\ 0.500 & 0.500 & 0.500 & 1.000 & & & \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & & \\ 0.250 & 0.750 & 0.500 & 0.750 & 0.000 & 1.250 & \\ 0.125 & 0.375 & 0.250 & 0.375 & 0.500 & 0.625 & 1.000 \end{pmatrix}$$

Figure 2.2: Example of pedigree with corresponding additive numerator relationship matrix A. The graph on the left is a pedigree chart where parents (top) are connected to their offspring (bottom). The node symbol represents the sex of each animal. The dashed line connects symbols which represent the same animal. The matrix on the right shows the lower triangle of the symmetric additive numerator relationship matrix A. Animal 2 is mother and grandmother for animal 6, which is therefore highly inbreed (usually such strong inbreeding is prevented in practice). The probability for IBD allels between parent and progeny of inbreed animals is > 0.25, leading to offdiagonal elements which are > 0.5. Also the probability of sampling IBD alleles from the same animal are > 0.5 leading to diagonal elments which are > 1.

2.2.2 Animal Model and Sire Model

In the way we derived the model in Chapter 2.2.1, $\boldsymbol{B}$ would contain the breeding values of all animals in the pedigree. In some breeding populations this might be several million animals which makes the model difficult to fit in reasonable time. One way of simplifying the model is to associate the random effect with a parent instead of the animal. The number of sires in a cattle breeding pedigree is usually much smaller compared to number of animals. The dimension of $\boldsymbol{B}$ can be drastically reduced if the random effect is associated with the sire. However, this simplification comes at a cost. First, only breeding values of the sires can be predicted. Dams can therefore no longer be selected based on a breeding value. Another disadvantage is that the variance covariance matrix of $(\boldsymbol{Y} \mid \boldsymbol{B} = \boldsymbol{b})$ will no longer be truly diagonal, because we only account for half of the additive effect. With a diagonal variance covariance matrix, full-sibs are assumed to have the same correlation as halve-sibs from the same sire, an assumption which is obviously not true. Still, assuming a diagonal variance covariance matrix might result reasonable predictions of $\boldsymbol{B}$.

The model where each animal has a breeding value is commonly referred to as *animal model* (Henderson 1984). In contrary, the model were the random effects are associated with the sires is referred to as *sire model*. In the following we show a small numerical example of an animal and a sire model.

Example 2.1. We observed a trait y in 3 animals (Table 2.1). The predictor Lact is assumed to have a fixed effect on the response variable. The pedigree is the same as in Figure 2.2.

Our goal is to set up all model components for the animal model and the sire model. The

Table 2.1: Example data set

Animal	Sire	Herd	Lact	y
3	1	1	1	5.3
3	1	1	2	5.1
6	4	1	1	3.4
6	4	1	2	3.1
7	5	2	1	7.2
7	5	2	2	7.0

$\boldsymbol{y}$ vector and the fixed terms are the same in both models

$$\boldsymbol{y} = \begin{pmatrix} 5.3 \\ 5.1 \\ 3.4 \\ 3.1 \\ 7.2 \\ 7.0 \end{pmatrix} \qquad X\boldsymbol{\beta} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_G \\ \beta_1 \\ \beta_2 \end{pmatrix}$$

with the side constrain $\beta_1 = -\beta_2$. The random component is different between the two models. For the animal model we have

$$Z\boldsymbol{b} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix}.$$

The A matrix is the same as in Figure 2.2. Including random effect b_1, b_2 and b_4 into the model may seem unnecessary because they are not connected to any observation. Including them still makes sense in this specific case. It appears that b_1, b_2 and b_4 can be predicted because they are correlated with random effects for which we have observations via the additive numerator relationship matrix. This makes it possible to predict breeding values for animals without any observations. However, only breeding values of animals which are related to animals with observations should be included in vector $\boldsymbol{b}$.

For the sire model the same components would be

$$Z\boldsymbol{b} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} \qquad A = \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In real world data sets, the dimension of A is reduced drastically by using a sire model, which in turn speeds up the estimation of the model.

■

2.2.3 General Form

Breeding values were the only random term besides the residual term in the previous chapters. In practice there are usually additional environmental factors which can be considered as random. Therefore, we have to extend the random part which should no longer be limited to breeding values. Additionally, the effect of alleles often interacts with environmental variables. For example, it is well known that some alleles which are associated with higher milk productivity in optimal environments, have a negative effect on milk yield in unfavorable environments (O'Neill, Swain, and Kadarmideen 2010). Such interaction between a predictor variable and a random factor is usually called random regression.

Model (2.10) can easily be extended to allow for additional random factors and random regression. A more general correlation structure in the random effects can be introduced by rewriting the model as

$$\begin{aligned} (\boldsymbol{Y} \mid \boldsymbol{B} = \boldsymbol{b}) &\sim \mathcal{N}_{\text{n}}\left(X\boldsymbol{\beta} + Z\boldsymbol{b}, I_n\sigma^2\right) \\ \boldsymbol{B} &\sim \mathcal{N}_{\text{q}}\left(\vec{0}, \Sigma\right). \end{aligned} \tag{2.11}$$

The extensions to additional random factors and random regression will only affect the structure of three model components: Z, $\boldsymbol{B}$ and Σ. However, before we define the structure of these components we need to introduce some notation. We use the same notation as in Bates, Mächler, et al. (2015). New variables and subscripts are explained in Table 2.2 and 2.3. Using R mixed model formula notation, we allow for models of the form

`y ~ FEexpr + (REexp1 | factor1) + (REexpr2 | factor2) + ...`.

On the right hand side of the modeling equation we have three terms, `FEexpr` which contains all information to build the fixed effect model matrix X, as well as two random effect terms. The random effect terms include an expression and a grouping factor separated by a vertical bar. The expressions contain the information to build the raw random effects model matrix X_i whereas the factor contains the information for the indicator matrix J_i. Random effects of different terms are assumed to be independent. For detailed explanation about R mixed model formula notation see Bates, Mächler, et al. (2015).

We define the order of random effects in the following way

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_k \end{pmatrix} \qquad \text{with} \qquad \boldsymbol{B}_i = \begin{pmatrix} \begin{pmatrix} B_{11} \\ \vdots \\ B_{1p_i} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} B_{l_i1} \\ \vdots \\ B_{l_ip_i} \end{pmatrix} \end{pmatrix}.$$

The Z matrix can be calculated in the same way as described in Bates, Mächler, et al.

Table 2.2: Dimensions of linear mixed models (from Bates, Mächler, et al. 2015). The subscript $i = 1, \ldots, k$ denotes a specific random effects term.

Symbol	Description
n	Length of the response vector, $\boldsymbol{Y}$.
p	Number of columns of the fixed effects model matrix, X.
$q = \sum_i^k q_i$	Number of columns of the random effects model matrix, Z.
p_i	Number of columns of the raw model matrix, X_i.
l_i	Number of levels of the grouping factor indices.
$q_i = p_i l_i$	Number of columns of the term-wise model matrix, Z_i.
k	Number of random effects terms.
$m_i = \binom{p_i+1}{2}$	Number of covariance parameters for term i.
$m = \sum_i^k m_i$	Total number of covariance parameters.

Table 2.3: Symbols used to describe the structure of the random effects model (adapted from Bates, Mächler, et al. 2015). The subscript $i = 1, \ldots, k$ moves along random effect terms, $j = 1, \ldots, n$ along observations, $g = 1, \ldots, p_i$ along columns of raw model matrix and $h = 1, \ldots, l_i$ along levels of the grouping factor.

Symbol	Size	Description
$\boldsymbol{B}_i$	$p_i l_i \times 1$	Vector of random effects of ith term
B_{hg}	1×1	random effect of hth level and gth column of raw model matrix
X_i	$n \times p_i$	Raw random effects model matrix.
J_i	$n \times l_i$	Indicator matrix of grouping factor indices.
$\boldsymbol{X}_{ij}$	$p_i \times 1$	Column vector containing jth row of X_i.
$\boldsymbol{J}_{ij}$	$l_i \times 1$	Column vector containing jth row of J_i.
$\boldsymbol{i}_i$	n	Vector of grouping factor indices.
Z_i	$n \times q_i$	Term-wise random effects model matrix.
$\boldsymbol{\theta}$	m	Covariance parameters.
T_i	$p_i \times p_i$	Lower triangular template matrix.
Λ_i	$q_i \times q_i$	Term-wise relative covariance factor.
Σ_i	$p_i \times p_i$	Term-wise covariance matrix.
A_i	$l_i \times l_i$	Term-wise numerator relationship matrix.
L_{A_i}	$l_i \times l_i$	Term-wise Cholesky factor.

(2015)

$$Z^\intercal = \begin{pmatrix} \boldsymbol{J}_{11} \otimes \boldsymbol{X}_{11} & \boldsymbol{J}_{12} \otimes \boldsymbol{X}_{12} & \dots & \boldsymbol{J}_{1n} \otimes \boldsymbol{X}_{1n} \\ \boldsymbol{J}_{21} \otimes \boldsymbol{X}_{21} & \boldsymbol{J}_{22} \otimes \boldsymbol{X}_{22} & \dots & \boldsymbol{J}_{2n} \otimes \boldsymbol{X}_{2n} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}. \tag{2.12}$$

For Σ we need some adaptations to allow for correlation across animals or sires. The general structure can be expressed with

$$\Sigma = \begin{pmatrix} A_1 \otimes \Sigma_1 & & \\ & \ddots & \\ & & A_k \otimes \Sigma_k \end{pmatrix} = \operatorname{diag}\left(\{A_i \otimes \Sigma_i\}_{i=1}^{k} \right).$$

Example 2.2. We want to set up a model of the form `y ~ (1|Herd) + (1+Lact|Sire)` using the same data set as in Table 2.1. The fixed effects can be encoded with $X = \vec{1}$ and $\boldsymbol{\beta} = \mu_B$. The vector of random effects is defined by

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{\text{herd}} \\ \boldsymbol{B}_{\text{sire}} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \\ \begin{pmatrix} s_{1i} \\ s_{1l} \\ s_{2i} \\ s_{2l} \\ s_{3i} \\ s_{3l} \end{pmatrix} \end{pmatrix}$$

Using Equation (2.12) we can calculate matrix Z

$$\boldsymbol{i}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix} \qquad J_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \qquad X_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\boldsymbol{i}_2 = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \end{pmatrix} \qquad J_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad X_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

The variance covariance matrix of the random effects is defined by

$$\Sigma_{\text{herd}} = \sigma_h^2 \qquad A_{\text{herd}} = I_2 \qquad \Sigma_{\text{sire}} = \begin{pmatrix} \sigma_i^2 & \sigma_{i,l} \\ \sigma_{i,l} & \sigma_l^2 \end{pmatrix} \qquad A_{\text{sire}} = \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\Sigma = \left(\begin{array}{cc|cccc} \sigma_h^2 & 0 & \multicolumn{4}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ 0 & \sigma_h^2 & & & & \\ \hline & & \sigma_i^2 & \sigma_{i,l} & 0.5\sigma_i^2 & 0.5\sigma_{i,l} & \multirow{2}{*}{$\mathbf{0}$} \\ & & \sigma_{i,l} & \sigma_l^2 & 0.5\sigma_{i,l} & 0.5\sigma_l^2 & \\ \multicolumn{2}{c|}{\multirow{3}{*}{$\mathbf{0}$}} & 0.5\sigma_i^2 & 0.5\sigma_{i,l} & \sigma_i^2 & \sigma_{i,l} & \multirow{2}{*}{$\mathbf{0}$} \\ & & 0.5\sigma_{i,l} & 0.5\sigma_l^2 & \sigma_{i,l} & \sigma_l^2 & \\ & & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} & \sigma_i^2 \quad \sigma_{i,l} \\ & & & & & & \sigma_{i,l} \quad \sigma_l^2 \end{array}\right).$$

■

2.2.4 Transformation to Independant Random Effects

As seen in Example 2.2 the variance covariance matrix of the random effects can become quite complex. Most software implementations for predicting random effects are not able to handle such models by default. However as already shown by Harville and Callanan (1989), a simple transformation can make random effects of different animals independent.

For example take Model (2.10) and define the left Cholesky factor L_A as

$$A = L_A L_A^\intercal \tag{2.13}$$

and the transformed random effects $\boldsymbol{B}^*$ as

$$\boldsymbol{B}^* = L_A^{-1}\boldsymbol{B}. \tag{2.14}$$

It can easily be verified that the individual random effects of $\boldsymbol{B}^*$ are mutually independent

$$\begin{aligned} \operatorname{Var}(\boldsymbol{B}^*) &= \operatorname{Var}\left(L_A^{-1}\boldsymbol{B}\right) \\ &= L_A^{-1}\operatorname{Var}(\boldsymbol{B})\,(L_A^{-1})^\intercal \\ &= L_A^{-1} A \sigma_b^2 (L_A^{-1})^\intercal \\ &= L_A^{-1} L_A L_A^\intercal (L_A^{-1})^\intercal \sigma_b^2 \\ &= I_q \sigma_b^2. \end{aligned}$$

In the following we present an extension of this method to the general Model (2.11).

Result 2.1. Assume a random vector $\boldsymbol{B}_i \sim \mathcal{N}\left(\vec{0}, A_i \otimes \Sigma_i\right)$ with $A_i = L_{A_i} L_{A_i}^\intercal$. The distribution of the transformed random vector

$$\boldsymbol{B}_i^* = (L_{A_i} \otimes I_{p_i})^{-1}\boldsymbol{B}_i$$

will be $\boldsymbol{B}_i^* \sim \mathcal{N}\left(\vec{0}, I_{l_i} \otimes \Sigma_i\right)$.

Proof. Because $\boldsymbol{B}_i^*$ is a linear transformation of a multivariate Gaussian, it will also be multivariate Gaussian. The expected value and variance covariance matrix of $\boldsymbol{B}_i^*$ are

$$\mathbf{E}\left[\boldsymbol{B}_i^*\right] = \mathbf{E}\left[(L_{A_i} \otimes I_{p_i})^{-1}\boldsymbol{B}_i\right] = 0$$

and

$$\begin{aligned}
\operatorname{Var}\left((L_{A_i} \otimes I_{p_i})^{-1}\boldsymbol{B}_i\right) &= (L_{A_i} \otimes I_{p_i})^{-1} \operatorname{Var}\left(\boldsymbol{B}_i\right)\left((L_{A_i} \otimes I_{p_i})^{-1}\right)^{\intercal} \\
&= (L_{A_i}^{-1} \otimes I_{p_i})(A_i \otimes \Sigma_i)\left((L_{A_i}^{-1})^{\intercal} \otimes I_{p_i}\right) \\
&= \left((L_{A_i}^{-1}A_i) \otimes (I_{p_i}\Sigma_i)\right)\left((L_{A_i}^{-1})^{\intercal} \otimes I_{p_i}\right) \\
&= \left(L_{A_i}^{-1}A_i(L_{A_i}^{-1})^{\intercal}\right) \otimes (I_{p_i}\Sigma_i I_{p_i}) \\
&= I_{l_i} \otimes \Sigma_i
\end{aligned}$$

using $(G \otimes H)(J \otimes K) = (GJ) \otimes (HK)$.

□

Applying Result 2.1 we are going to define the *transformed general model* as

$$\begin{aligned}
(\boldsymbol{Y} \mid \boldsymbol{B}^* = \boldsymbol{b}^*) &\sim \mathcal{N}_{\mathrm{n}}\left(X\boldsymbol{\beta} + Z^*\boldsymbol{b}^*, I_n\sigma^2\right) \\
\boldsymbol{B}^* &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, \Sigma^*\right).
\end{aligned} \tag{2.15}$$

The transformed elements are defined as

$$\boldsymbol{B}^* = \begin{pmatrix} \boldsymbol{B}_1^* \\ \vdots \\ \boldsymbol{B}_k^* \end{pmatrix} \qquad \text{with} \qquad \boldsymbol{B}_i^* = (L_{A_i} \otimes I_{p_i})^{-1}\boldsymbol{B}_i$$

and

$$Z^* = \begin{pmatrix} Z_1^* & \cdots & Z_k^* \end{pmatrix} \qquad \text{with} \qquad Z_i^* = Z_i(L_{A_i} \otimes I_{p_i})$$

Alternatively, using a more compact notation

$$\begin{aligned}
\boldsymbol{B}^* &= T_A^{-1}\boldsymbol{B} \\
Z^* &= ZT_A.
\end{aligned} \qquad \text{with} \qquad T_A = \operatorname{diag}\left(\{L_{A_i} \otimes I_{p_i}\}_{i=1}^k\right). \tag{2.16}$$

Note that $Z^*\boldsymbol{B}^* = Z\boldsymbol{B}$, therefore Model (2.11) and (2.15) are equivalent. However, the random effects in the transformed model have the desired property of being uncorrelated between animals

$$\Sigma^* = \begin{pmatrix} I_{l_1} \otimes \Sigma_1 & & \\ & \ddots & \\ & & I_{l_k} \otimes \Sigma_k \end{pmatrix} = \operatorname{diag}\left(\{I_{l_i} \otimes \Sigma_i\}_{i=1}^{k}\right).$$

Therefore, the transformed model can be estimated with common software implementations for predicting random effects.

It remains to explain how to calculate L_{A_i}. We have seen that L_A can be obtained via a Cholesky decomposition of the additive numerator relationship matrix. Calculating the additive numerator relationship matrix for a large pedigree comes at a huge computational cost. Luckily, Vazquez, Bates, et al. (2010) present a method to calculate L_A without having to construct A explicitly. The method is based on a second decomposition

$$A = TDT^\intercal \tag{2.17}$$

where T is a lower triangular matrix and D is a diagonal matrix. Both, T and D can be directly calculated from the pedigree without having to calculate A (Mrode and Thompson 2005, Sargolzaei and Iwaisaki 2005). Using (2.13) and (2.17) we can calculate L_A as

$$L_A = TD^{1/2} \tag{2.18}$$

Example 2.3. Continuing from Example 2.2 the transformed random effects associated with the sire term are given by

$$\boldsymbol{B}^*_{\text{sire}} = (L_{A_{\text{sire}}} \otimes I_2)^{-1}\boldsymbol{B}_{\text{sire}} \qquad \text{with} \qquad L_{A_{\text{sire}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The transformed random effects are independent across animals

$$\operatorname{Var}(\boldsymbol{B}^*) = \left(\begin{array}{cc|cccc} \sigma_h^2 & 0 & \multicolumn{4}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ 0 & \sigma_h^2 & & & & \\ \hline & & \sigma_i^2 & \sigma_{i,l} & \mathbf{0} & \mathbf{0} \\ & & \sigma_{i,l} & \sigma_l^2 & & \\ & \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} & \sigma_i^2 & \sigma_{i,l} & \multicolumn{1}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ & & \multicolumn{2}{c}{} & \sigma_{i,l} & \sigma_l^2 & \multicolumn{1}{c}{} \\ & & \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{0}} & \sigma_i^2 \quad \sigma_{i,l} \\ & & \multicolumn{2}{c}{} & \multicolumn{2}{c}{} & \sigma_{i,l} \quad \sigma_l^2 \end{array}\right).$$

■

2.2.5 Identifiability

The animal model as formulated in Example 2.1 might raise questions with regard to the identifiability of the model. For real world data sets, it might well be the case that we have only one observation per animal. Random effects models with only one observation per random intercept are usually not identifiable because the random intercept and the residual error are indistinguishable. It seems necessary to derive general conditions on

the pedigree and therefore the additive numerator relationship matrix, in order to ensure identifiability of the model.

Result 2.2. The random intercept of an animal model is not identifiable with respect to the residual error if and only if the additive numerator relationship matrix A is diagonal i.e. all animals are unrelated.

Proof. Assume a model where the random intercept per animal is the only random term next to the residual error and each animal has exactly one observation

$$\begin{aligned}(\boldsymbol{Y} \mid \boldsymbol{B} = \boldsymbol{b}) &\sim \mathcal{N}_{\mathrm{n}}\left(\boldsymbol{b}, I_n\sigma^2\right) \\ \boldsymbol{B} &\sim \mathcal{N}_{\mathrm{n}}\left(\vec{0}, A\sigma_b^2\right)\end{aligned}. \tag{2.19}$$

The transformed model matrix for the random term is given by $Z^* = L_A$. We use Theorem 4.1 of Wang (2013): A linear mixed effects model is not identifiable if and only if $\forall(\Sigma_u, \Sigma_\varepsilon)$, $\exists(\tilde{\Sigma}_u, \tilde{\Sigma}_\varepsilon) \neq (\Sigma_u, \Sigma_\varepsilon)$, such that the following three equations hold

$$Z^{*\intercal}\Sigma_\varepsilon Z^* \neq Z^{*\intercal}\tilde{\Sigma}_\varepsilon Z^* \tag{2.20}$$

$$\Sigma_\varepsilon - \tilde{\Sigma}_\varepsilon = \underbrace{Z^*(Z^{*\intercal}Z^*)^{-1}Z^{*\intercal}}_{H_Z}(\Sigma_\varepsilon - \tilde{\Sigma}_\varepsilon) \tag{2.21}$$

$$\tilde{\Sigma}_{\boldsymbol{b}} - \Sigma_{\boldsymbol{b}} = (Z^{*\intercal}Z^*)^{-1}Z^{*\intercal}(\Sigma_\varepsilon - \tilde{\Sigma}_\varepsilon)Z^*(Z^{*\intercal}Z^*)^{-1}. \tag{2.22}$$

Applied to model (2.19) we have

$$\Sigma_\varepsilon = I_n\sigma^2 \qquad \tilde{\Sigma}_\varepsilon = I_n\tilde{\sigma}^2$$
$$\Sigma_{\boldsymbol{b}} = I_n\sigma_b^2 \qquad \tilde{\Sigma}_{\boldsymbol{b}} = I_n\tilde{\sigma}_b^2.$$

From Equation (2.20) it directly follows that $\sigma^2 \neq \tilde{\sigma}^2$. Equation (2.21) always holds as can be verified by simplifying H_Z

$$\begin{aligned}H_Z &= Z^*(Z^{*\intercal}Z^*)^{-1}Z^{*\intercal} \\ &= L_A(L_A^\intercal L_A)^{-1}L_A^\intercal \\ &= L_A L_A^{-1}(L_A^\intercal)^{-1}L_A^\intercal \\ &= I_n\end{aligned}$$

Finally, simplifying Equation (2.22) results in

$$\begin{aligned}\tilde{\Sigma}_{\boldsymbol{b}} - \Sigma_{\boldsymbol{b}} &= (Z^{*\intercal}Z^*)^{-1}Z^{*\intercal}(\Sigma_\varepsilon - \tilde{\Sigma}_\varepsilon)Z^*(Z^{*\intercal}Z^*)^{-1} \\ \Rightarrow \quad I_n(\tilde{\sigma}_{\boldsymbol{b}}^2 - \sigma_{\boldsymbol{b}}^2) &= (L_A^\intercal L_A)^{-1}L_A^\intercal I_n(\sigma_\varepsilon^2 - \tilde{\sigma}_\varepsilon^2)L_A(L_A^\intercal L_A)^{-1} \\ \Rightarrow \quad I_n\frac{\tilde{\sigma}_{\boldsymbol{b}}^2 - \sigma_{\boldsymbol{b}}^2}{\sigma_\varepsilon^2 - \tilde{\sigma}_\varepsilon^2} &= \underbrace{(L_A^\intercal L_A)^{-1}L_A^\intercal L_A}_{I_n}(L_A^\intercal L_A)^{-1} \\ &= (L_A^\intercal L_A)^{-1}.\end{aligned} \tag{2.23}$$

Define $\theta = \frac{\tilde{\sigma}_b^2 - \sigma_b^2}{\sigma_\varepsilon^2 - \tilde{\sigma}_\varepsilon^2}$ and solve Equation (2.23) for L_A results

$$L_A = (L_A^\intercal)^{-1}\theta^{-1}. \tag{2.24}$$

Insert Equation (2.24) into $A = L_A L_A^\intercal$ results in

$$A = (L_A^\intercal)^{-1} L_A^\intercal \theta^{-1} = I_n \theta^{-1}$$

i.e. Equation (2.22) is only fulfilled if matrix A is diagonal.

□

2.3 Model Extension: GLMM and Threshold

Using LMMs was based on the assumption that many alleles have a small effect on the phenotypic value. Under this assumption the phenotypic value will necessarily be a continuous variable which follows asymptotically a normal distribution. This is what we observe for many traits which are relevant to breeding such as milk yield. However, some traits are fundamentally different as they might be discrete or measured on a nominal or ordinal scale. Using an LMM in such a case does not well reflect the true model and may lead to predictions outside the support of $\boldsymbol{Y}$. Therefore, a better model seems necessary.

In the best case scenario we would know the entire physiological process involved in the expression of $\boldsymbol{Y}$ and could build up a mechanistic model to estimate the effects of the alleles on $\boldsymbol{Y}$. Unfortunately, we usually do not know the physiological process and even if we would, the estimation might be way too complex to get reasonable estimates with limited data. Therefore, we are looking for a simple modification of Model (2.15) allowing for $\boldsymbol{Y}$'s which are continuous or discrete and on a nominal, ordinal, ratio or interval scale.

2.3.1 Generalized Linear Mixed Effects Model

A well known strategy is to model the continuous parameters of a distribution which draws values with the desired properties. This strategy is formalized as generalized linear model (GLM). It allows modeling response variables from an entire class of distributions. GLM can be extended to the generalized linear mixed effects model (GLMM) allowing besides the fixed effects also random effects in the linear predictor.

A generalized version of Model (2.15) would be

$$\begin{aligned} (\boldsymbol{Y} \mid \boldsymbol{B}^* = \boldsymbol{b}^*) &\sim \mathrm{EDM}_n(\boldsymbol{\mu}, I_n\phi) \\ \boldsymbol{g}(\boldsymbol{\mu}) &= X\boldsymbol{\beta} + Z^*\boldsymbol{b}^* \\ \boldsymbol{B}^* &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, \Sigma^*\right). \end{aligned} \tag{2.25}$$

where EDM is any distribution from the exponential distribution family, $\boldsymbol{\mu}$ is the conditional mean, ϕ is the common scale parameter and $\boldsymbol{g}(\cdot)$ is the vectorized link function. Vectorized means that the same univariate link function $g(\cdot)$ is applied to each element of $\boldsymbol{\mu}$. Distribution and link function are chosen according to the nature of the response variable $\boldsymbol{Y}$. The choice of distribution affects weather the response is continuous or discrete

and determines the relationship between $\boldsymbol{\mu}$ and the variance of $\boldsymbol{Y}$. The link function can be any monotonic, differentiable function which links the conditional mean to the linear predictor $\eta = X\boldsymbol{\beta} + Z^*\boldsymbol{b}^*$. It is chosen in a way such that it maps the unbounded linear predictor $\eta \in [-\infty, \infty]$ to the appropriate range of μ. A common choice is the canonical link which is defined via the canonical parameter of the EDM family (Dunn and Smyth 2018). The LMM (2.15) can be seen as a special case of Model (2.25) with the normal distribution and the identity link.

2.3.2 Threshold Model

The generalization as stated in (2.25) allows us to model a large variety of response variables on different scales. However, the exponential distribution family shows limitations when it comes to ordinal response variables. Ordinal variables are categorical with a natural order of the levels but with nonmetric distances between them. In animal breeding they may occur if a metric trait is difficult to measure and therefore has to be visually classified.

Threshold models build on linear models and offer an simple approach to evaluate ordinal data (Bürkner and Vuorre 2019). Depending on the area of research they are also know by the name ordinal regression model, proportional odds linear regression model, cumulative model or graded response model. In animal breeding, threshold model is the most commonly used term probably due to the early publication by Gianola and Foulley (1983). Therefore, we are going to stick with this name. In its basic form, the threshold model assumes an underlying latent (i.e. not observable) continuous variable for each observations which is usually called liability. The observed ordinal response originates from the categorization of the continuous liability by breaking it up at specific latent thresholds. For an ordinal variable with $k+1$ categories, there are a total of k thresholds necessary which are collected in the vector $\boldsymbol{\tau}$. The classification is completely deterministic given the liability. Liabilities of all observations are collected in the vector $\boldsymbol{l}$. Threshold models may differ in the way how $\boldsymbol{l}$ is modeled from the predictor variables. One of the simplest way is to assume an LMM with $\boldsymbol{l}$ as response vector resulting in the following model

$$(Y_i \mid l_i, \boldsymbol{\tau}) = \begin{cases} 1 & \text{for} \quad -\infty \leq l_i \leq \tau_1 \\ 2 & \text{for} \quad \tau_1 < l_i \leq \tau_2 \\ \vdots & \vdots \\ k+1 & \text{for} \quad \tau_k < l_i \leq \infty \end{cases} \tag{2.26}$$

$$\begin{aligned} (\boldsymbol{l} \mid \boldsymbol{B}^* = \boldsymbol{b}^*) &\sim \mathcal{N}_{\text{n}}\left(X\boldsymbol{\beta} + Z^*\boldsymbol{b}^*, I_n\right) \\ \boldsymbol{B}^* &\sim \mathcal{N}_{\text{q}}\left(\vec{0}, \Sigma^*\right). \end{aligned}$$

The liability has a normal distribution with the mean given by the linear predictor (Figure 2.3). The variance is set equal to one because it would be unidentifiable as long as we allow for arbitrary scaling of the threshold values.

2.4 Maximum Likelihood Estimation

In the previous chapters we have introduced LMMs, GLMMs and threshold models. All of them are suitable to model the effect of many alleles on a trait of interest. In the following

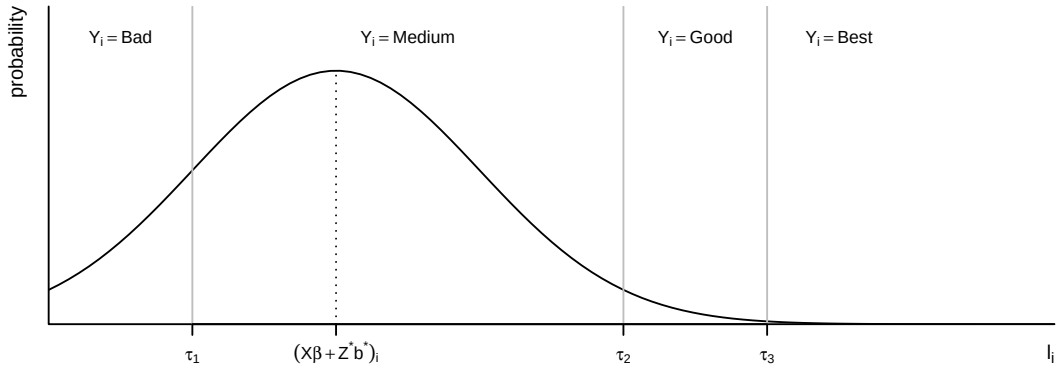


Figure 2.3: Distribution of the liability for observation i in a threshold model. The liability is normally distributed with the mean given by the linear predictor. Thresholds split the liability into regions of different categories. Observation i in this example would have a large probability to express level "Medium".

we are going to show how the unknown model parameters in LMMs and GLMMs can be estimated using the maximum likelihood framework. Threshold models will only be covered in the section about Bayesian estimation (Chapter 2.5).

2.4.1 Linear Mixed Effects Model

Model (2.10), (2.11) and (2.15) are all LMMs with varying complexity of the random effect's variance covariance matrix. All of them can be estimated using the same procedure, however, we will focus on Model (2.15) because it is more general than Model (2.10) and has a simpler variance covariance matrix than Model (2.11). Model components X, Z^*, $\boldsymbol{y}$ and T_A are obtained from a data set and the associated pedigree. The data set contains the response variable $\boldsymbol{y}$ along with the animals id and factors which are assumed to have a fixed effect on $\boldsymbol{y}$. The information in the pedigree can be used to set up T_A as defined in Equation (2.16). $\boldsymbol{\beta}$ and Σ^* are the unknown parameters which need to be estimated. Additionally, we are interested in predicting the mode of the random vector $\boldsymbol{B}$ conditional on the observed $\boldsymbol{y}$.

In a first step we again transform the random effects similarly to what was done in Chapter 2.2.4 but now the goal is to make them completely independent and not only independent between animals. The transformed variance covariance matrix is decomposed into two Cholesky factors and the residual variance

$$\Sigma^* = \Lambda_{\boldsymbol{\theta}} \Lambda_{\boldsymbol{\theta}}^\intercal \sigma^2 \tag{2.27}$$

where $\Lambda_{\boldsymbol{\theta}}$ is a sparse lower triangular block diagonal matrix. It consists of k main blocks and the ith main block consists of l_i subblocks, each of which is a copy of a $p_i \times p_i$ lower triangular template matrix T_i. The template matrix can be seen as the Cholesky decomposition of the term-wise covariance matrix while factoring out the residual variance

$$\Sigma_i = T_i T_i^\intercal \sigma^2.$$

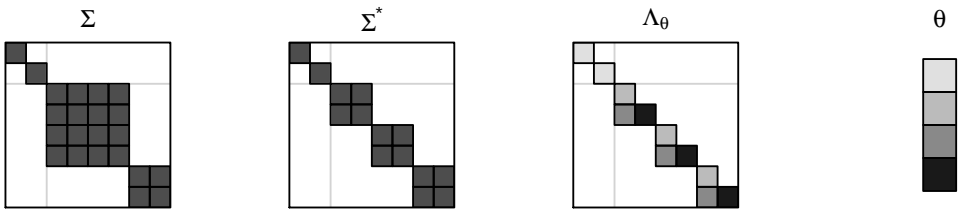


Figure 2.4: Matrices and vectors which determine variance covariance structure of random effects in Example 2.2. The non-zero elements are shown as darkened squares. In $\Lambda_{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ the same gray color represent the same number. Σ is the symmetric variance covariance matrix of $\boldsymbol{B}$. The random effects of related animals are correlated. Σ^* is the symmetric variance covariance matrix of the transformed random effects $\boldsymbol{B}^*$. Random effects within one animal might be correlated but are now independent between related animals. $\Lambda_{\boldsymbol{\theta}}$ is proportional to the lower triangular Cholesky factor of Σ^*. The unique elements of $\Lambda_{\boldsymbol{\theta}}$ are column-wise combined to vector $\boldsymbol{\theta}$ which typically is low dimensional.

The modeling assumptions can lead to many zero elements in $\Lambda_{\boldsymbol{\theta}}$ which do not need to be estimated. Optimization only needs to be done over the value of all possibly non-zero elements. All these elements are column-wise combined in the vector $\boldsymbol{\theta}$ which completely determines $\Lambda_{\boldsymbol{\theta}}$ given a specific model. The dimension of θ is typically small which makes numerical optimization possible (Figure 2.4).

The spherical random effects U are defined by

$$\boldsymbol{B}^* = \Lambda_\theta \boldsymbol{U}. \tag{2.28}$$

It may seem more intuitive to write Equation (2.28) as $\boldsymbol{U} = \Lambda_\theta^{-1}\boldsymbol{B}^*$, however, Λ_θ just as Σ^* could be singular, in which case only Equation (2.28) is valid. It can easily be verified that $\boldsymbol{U} \sim \mathcal{N}_{\text{q}}\left(\vec{0}, I_q\sigma^2\right)$ as shown in Bates (2020). Using the transformation in Equation (2.28) we can define the *independent model* as

$$\begin{aligned}(\boldsymbol{Y} \mid \boldsymbol{U} = \boldsymbol{u}) &\sim \mathcal{N}_{\text{n}}\left(X\boldsymbol{\beta} + Z^*\Lambda_\theta \boldsymbol{u}, I_n\sigma^2\right) \\ \boldsymbol{U} &\sim \mathcal{N}_{\text{q}}\left(\vec{0}, I_q\sigma^2\right).\end{aligned} \tag{2.29}$$

The careful reader will have noticed that the above procedure could also be directly applied to the untransformed Model (2.11). Instead of sequentially calculating the transformed Model (2.15) and then the independent Model (2.29) we could calculate it directly from (2.11) in one step. The reason for doing it in the outlined two step approach lies in computational efficiency. Σ can be a huge matrix consisting of term-wise variance covariance matrices Σ_i and term-wise numerator relationship matrices A_i. Setting up Σ and calculating the Cholesky decomposition of it would come at high computational cost. With the two step approach, we make use of the fast calculation of L_{A_i} (Equation 2.18) and avoid the calculation of A_i. Setting up Λ_θ as decomposition of the simplified Σ^* can be done

without even explicitly evaluating Σ^*, something which would be impossible in a one step approach.

The likelihood of the parameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and σ given the observed data $\boldsymbol{y}$ is the marginal density of $\boldsymbol{Y}$ evaluated at $\boldsymbol{y}$ and seen as a function of the parameters. In order to get the marginal density, we integrate the joint density of $\boldsymbol{Y}$ and $\boldsymbol{U}$ with respect to $\boldsymbol{u}$

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma \mid \boldsymbol{y}) &= f_{\boldsymbol{Y}}(\boldsymbol{y}) \\ &= \int_{\mathbb{R}^q} f_{\boldsymbol{Y},\boldsymbol{U}}(\boldsymbol{y}, \boldsymbol{u}) \, \mathrm{d}\boldsymbol{u} \\ &= \int_{\mathbb{R}^q} f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) f_{\boldsymbol{U}}(\boldsymbol{u}) \, \mathrm{d}\boldsymbol{u}. \end{aligned} \tag{2.30}$$

In LMMs, $f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u})$ and $f_{\boldsymbol{U}}(\boldsymbol{u})$ are densities of multivariate normal distributions

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\|\boldsymbol{y} - X\boldsymbol{\beta} - Z^*\Lambda_\theta \boldsymbol{u}\|^2}{2\sigma^2}\right) \\ f_{\boldsymbol{U}}(\boldsymbol{u}) &= \frac{1}{(2\pi\sigma^2)^{q/2}} \exp\left(-\frac{\|\boldsymbol{u}\|^2}{2\sigma^2}\right). \end{aligned}$$

leading to the likelihood function

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma \mid \boldsymbol{y}) = \int_{\mathbb{R}^q} \frac{1}{(2\pi\sigma^2)^{(n+q)/2}} \exp\left(-\frac{\|\boldsymbol{y} - X\boldsymbol{\beta} - Z^*\Lambda_\theta \boldsymbol{u}\|^2 + \|\boldsymbol{u}\|^2}{2\sigma^2}\right) \mathrm{d}\boldsymbol{u}. \tag{2.31}$$

The integral can be solved analytically as shown in Bates (2020). Still, numerical optimization with respect to $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and σ would be expensive because the dimension of $\boldsymbol{\beta}$ can be large. Luckily, we can directly (i.e. non-iteratively) determine the conditional estimates $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ and $\hat{\sigma}^2_{\boldsymbol{\theta}}$ which will maximize the likelihood given a value of $\boldsymbol{\theta}$. Therefore, $\boldsymbol{\beta}$ and σ can be profiled out of the likelihood and numerical optimization only needs to be done with respect to $\boldsymbol{\theta}$.

In order to maximize the likelihood, the conditional estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ together with the conditional mode $\tilde{\boldsymbol{u}}$ will have to satisfy

$$\begin{pmatrix} \tilde{\boldsymbol{u}} \\ \hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} \end{pmatrix} = \arg\min_{\boldsymbol{u},\boldsymbol{\beta}} \|\boldsymbol{y} - X\boldsymbol{\beta} - Z^*\Lambda_\theta \boldsymbol{u}\|^2 + \|\boldsymbol{u}\|^2. \tag{2.32}$$

Equation (2.32) can be identified as penalized least squares problem which can be rewritten using the "pseudo-data" approach

$$\begin{pmatrix} \tilde{\boldsymbol{u}} \\ \hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} \end{pmatrix} = \arg\min_{\boldsymbol{u},\boldsymbol{\beta}} \left\| \underbrace{\begin{pmatrix} \boldsymbol{y} \\ \vec{0} \end{pmatrix}}_{\boldsymbol{y}^*} - \underbrace{\begin{pmatrix} Z^*\Lambda_{\boldsymbol{\theta}} & X \\ I_q & \vec{0} \end{pmatrix}}_{X^*} \underbrace{\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\beta} \end{pmatrix}}_{\boldsymbol{\beta}^*} \right\|^2.$$

The penalty is added as additional dimensions to the residuals. The contribution of the additional dimensions to the total euclidean length of the residuals vector is exactly the same as the $\|\boldsymbol{u}\|^2$ penalty term in Equation (2.32). The normal equation of the above least squares problem is given by

$$\underbrace{\begin{pmatrix} \Lambda_{\boldsymbol{\theta}}^\intercal Z^{*\intercal} Z^* \Lambda_{\boldsymbol{\theta}} + I_q & \Lambda_{\boldsymbol{\theta}}^\intercal Z^{*\intercal} X \\ X^\intercal Z^* \Lambda_{\boldsymbol{\theta}} & X^\intercal X \end{pmatrix}}_{X^{*\intercal} X^*} \underbrace{\begin{pmatrix} \tilde{\boldsymbol{u}} \\ \hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} \end{pmatrix}}_{\boldsymbol{\beta}^*} = \underbrace{\begin{pmatrix} \Lambda_{\boldsymbol{\theta}}^\intercal Z^{*\intercal} \boldsymbol{y} \\ X^\intercal \boldsymbol{y} \end{pmatrix}}_{X^{*\intercal} \boldsymbol{y}^*}. \tag{2.33}$$

Solving for $\tilde{\boldsymbol{u}}$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ involves the Cholesky decomposition of the system matrix $(X^{*\intercal} X^*)$. Equation (2.33) has obvious similarities with the Henderson equation (Henderson 1982) which is well known in the animal breeding community. However, Bates, Mächler, et al. (2015) notes two key differences which favor the use of Equation (2.33)

1. The order of $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{u}}$ is reversed. In this way some of the components necessary to evaluate the likelihood are a side product of solving the normal equation. Also, it allows to exploit the sparsity of $\Lambda_{\boldsymbol{\theta}}^\intercal Z^{*\intercal} Z^* \Lambda_{\boldsymbol{\theta}} + I_q$ in the Cholesky factorization. This might be less relevant in our examples as the transformed matrix Z^* is usually no longer sparse in contrast to the untransformed Z.
2. The Henderson equation in its classical formulation contains Σ^{*-1}. However, Σ^* might become singular during the numerical optimization and even the final estimate of Σ^* might be singular. Equation (2.33) is more stable because it does not contain any inverse of a possibly singular matrix.

Equation (2.33) needs to be solved many times in the course of the numerical optimization with respect to $\boldsymbol{\theta}$. A fast algorithm to solve Equation (2.33) is the key component in each implementation to fit LMMs under a likelihood framework.

2.4.2 Generalized Linear Mixed Effects Model

The maximum likelihood estimation for the parameters $\boldsymbol{\beta}$ and Σ^* in Model 2.25 follows a similar structure as in Chapter 2.4.1. The random effects are again transformed to be mutually independent. However, because σ^2 is only meaningful for the normal distribution, we redefine Λ_θ as

$$\Sigma^* = \Lambda_{\boldsymbol{\theta}} \Lambda_{\boldsymbol{\theta}}^\intercal.$$

The *independant generalized model* would therefore be defined as

$$\begin{aligned} (\boldsymbol{Y} \mid \boldsymbol{U} = \boldsymbol{u}) &\sim \mathrm{EDM}_n(\boldsymbol{\mu}, I_n \phi) \\ \boldsymbol{g}(\boldsymbol{\mu}) &= X\boldsymbol{\beta} + Z^* \Lambda_\theta \boldsymbol{u} \\ \boldsymbol{U} &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, I_q\right). \end{aligned} \tag{2.34}$$

The likelihood of the parameters can again be written as

$$L(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \boldsymbol{y}) = \int_{\mathbb{R}^q} f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) f_{\boldsymbol{U}}(\boldsymbol{u}) \, \mathrm{d}\boldsymbol{u}. \tag{2.35}$$

The main difference to Equation (2.30) concerns $f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u})$ which is no longer necessarily a Gaussian. This seemingly small change has two serious implications which makes the estimation process much more involved. First, unlike the LMM case, there is no guarantee for a closed form solution to the integral in the GLMM case. Even worse, the integral might be very high dimensional and contain terms in the integrand which are close to zero, which makes numerical integration difficult (Jiang 2007). Second, there is no longer a direct (non-iterative) solution for the conditional estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ and the conditional mode $\tilde{\boldsymbol{u}}$.

As we will see below, the first problem can be tackled by using Laplace approximation. The integrand is approximated with a second-order Taylor series expansion around $\tilde{\boldsymbol{u}}$. However, because we need to know $\tilde{\boldsymbol{u}}$ it makes sense to start with the second problem, finding the conditional estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ and the conditional mode $\tilde{\boldsymbol{u}}$ given $\boldsymbol{\theta}$.

Conditional Modes

In contrast to LMMs, the maximum likelihood estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ can no longer be exactly determined from the integrand $f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) f_{\boldsymbol{U}}(\boldsymbol{u})$ as a function of $\boldsymbol{\theta}$. The final estimate needs to be found simultaneously with $\boldsymbol{\theta}$ by numerical optimization. The $\boldsymbol{\beta}$ which maximizes the integrand, we call it conditional mode $\tilde{\boldsymbol{\beta}}$, still provides a good starting value for the numerical optimization. The two conditional modes can be written as

$$\begin{aligned}\begin{pmatrix}\tilde{\boldsymbol{u}}\\ \tilde{\boldsymbol{\beta}}\end{pmatrix} &= \arg\max_{\boldsymbol{u},\boldsymbol{\beta}} f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) f_{\boldsymbol{U}}(\boldsymbol{u}) \\ &= \arg\min_{\boldsymbol{u},\boldsymbol{\beta}} -2\log\Big(f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u})\Big) - 2\log\left(f_{\boldsymbol{U}}(\boldsymbol{u})\right)\end{aligned}$$

The first density $f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u})$ can be any density of the exponential distribution family. The parameter which minimize its deviance also minimize a weighted least squares criterion (Bradley 1973, Charnes, Frome, and Yu 1976). The contribution of the second density can be added as penalty similarly to Equation (2.32) leading to

$$\begin{pmatrix}\tilde{\boldsymbol{u}}\\ \tilde{\boldsymbol{\beta}}\end{pmatrix} = \arg\min_{\boldsymbol{u},\boldsymbol{\beta}} \left\| W^{1/2}\left(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}\right)\right\|^2 + \|\boldsymbol{u}\|^2 \tag{2.36}$$

where W is a diagonal weights matrix which ith diagonal element is proportional to the variance of $(Y_i \mid \boldsymbol{U} = \boldsymbol{u})$. Note that in general for the exponential distribution family, the variance may depend on the location μ_i, meaning that W_{ii} also depends on $\boldsymbol{u}$ and $\boldsymbol{\beta}$.

The conditional modes which satisfy (2.36) can be found with a penalized iteratively reweighted least squares (PIRLS) algorithm as it is presented in Bates (2018). We are going to motivate the PIRLS algorithm in three steps. In the first step the Gauss-Newton Method is presented to find parameter estimates of a nonlinear least squares problem. In the following two steps, we add the weights matrix and penalty to end up with the final equation which is solved iteratively in the course of optimization.

Gauss-Newton Method Assume the least squares problem $\min_{\boldsymbol{u},\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}\|^2$ where $\boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}$ is a nonlinear function of $\boldsymbol{u}$ and $\boldsymbol{\beta}$. The first order Taylor series approximation of $\boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}$ around an initial guess $\boldsymbol{u}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ is

$$\boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}} \approx \boldsymbol{\mu}_{\boldsymbol{u}^{(0)},\boldsymbol{\beta}^{(0)}} + S^{(0)}\boldsymbol{\delta}^{(0)}$$

where $S^{(0)}$ and $\boldsymbol{\delta}^{(0)}$ are defined as

$$S^{(0)} = \begin{pmatrix} U^{(0)} & V^{(0)} \end{pmatrix} = \begin{pmatrix} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{u}^{(0)}} & \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}^{(0)}} \end{pmatrix}$$
$$\boldsymbol{\delta}^{(0)} = \begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{u}}^{(0)} \\ \boldsymbol{\delta}_{\boldsymbol{\beta}}^{(0)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u} - \boldsymbol{u}^{(0)} \\ \boldsymbol{\beta} - \boldsymbol{\beta}^{(0)} \end{pmatrix}.$$

The vector of residuals can therefore be written as

$$\underbrace{\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}}_{\boldsymbol{r}} \approx \underbrace{\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{u}^{(0)},\boldsymbol{\beta}^{(0)}}}_{\boldsymbol{r}^{(0)}} - S^{(0)}\boldsymbol{\delta}^{(0)}.$$

Instead of solving the nonlinear least squares problem $\min_{\boldsymbol{u},\boldsymbol{\beta}} \|\boldsymbol{r}\|^2$, we can use the local linear approximation of $\boldsymbol{r}$ and solve the linear least squares problem $\min_{\boldsymbol{\delta}_{\boldsymbol{u}}^{(0)},\boldsymbol{\delta}_{\boldsymbol{\beta}}^{(0)}} \left\|\boldsymbol{r}^{(0)} - S^{(0)}\boldsymbol{\delta}^{(0)}\right\|^2$ with normal equation

$$\underbrace{\begin{pmatrix} U^{(0)\intercal}U^{(0)} & U^{(0)\intercal}V^{(0)} \\ V^{(0)\intercal}U^{(0)} & V^{(0)\intercal}V^{(0)} \end{pmatrix}}_{S^{(0)\intercal}S^{(0)}} \underbrace{\begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{u}}^{(0)} \\ \boldsymbol{\delta}_{\boldsymbol{\beta}}^{(0)} \end{pmatrix}}_{\boldsymbol{\delta}^{(0)}} = \underbrace{\begin{pmatrix} U^{(0)\intercal}\boldsymbol{r}^{(0)} \\ V^{(0)\intercal}\boldsymbol{r}^{(0)} \end{pmatrix}}_{S^{(0)\intercal}\boldsymbol{r}^{(0)}}.$$

Solving for $\boldsymbol{\delta}^{(0)}$ gives the direction in which $\boldsymbol{u}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ needs to be updated in order to minimize the original criterion. The updated vectors $\boldsymbol{u}^{(1)}$ and $\boldsymbol{\beta}^{(1)}$ are then used to calculate the updated $\boldsymbol{\mu}_{\boldsymbol{u}^{(1)},\boldsymbol{\beta}^{(1)}}$, $S^{(1)}$ and $\boldsymbol{r}^{(1)}$ and solve for $\boldsymbol{\delta}^{(1)}$ in the updated normal equation. The process is repeated until a convergence criterion is fulfilled (Bates and Watts 1988). A geometric interpretation of the method is shown in Figure 2.5.

Add Weight Matrix A weighted nonlinear least squares criterion $\left\|W^{1/2}[\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{u},\boldsymbol{\beta}}]\right\|^2$ can be approximated locally by the weighted linear least squares criterion $\left\|W^{1/2}[\boldsymbol{r}^{(i)} - S^{(i)}\boldsymbol{\delta}^{(i)}]\right\|^2$ in the same way as outlined above. The normal equation for the weighted least squares problem is given by

$$\underbrace{\begin{pmatrix} U^{(i)\intercal}WU^{(i)} & U^{(i)\intercal}WV^{(i)} \\ V^{(i)\intercal}WU^{(i)} & V^{(i)\intercal}WV^{(i)} \end{pmatrix}}_{S^{(0)\intercal}WS^{(0)}} \underbrace{\begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{u}}^{(i)} \\ \boldsymbol{\delta}_{\boldsymbol{\beta}}^{(i)} \end{pmatrix}}_{\boldsymbol{\delta}^{(0)}} = \underbrace{\begin{pmatrix} U^{(i)\intercal}W\boldsymbol{r}^{(i)} \\ V^{(i)\intercal}W\boldsymbol{r}^{(i)} \end{pmatrix}}_{S^{(0)\intercal}W\boldsymbol{r}^{(0)}}$$

Add Penalty The penalty term $\|\boldsymbol{u}\|^2$ in Equation (2.36) can be added as "pseudo-data" to the weighted least squares criterion leading to

$$\min_{\boldsymbol{\delta}_{\boldsymbol{u}}^{(0)},\boldsymbol{\delta}_{\boldsymbol{\beta}}^{(0)}} \left\| \begin{pmatrix} W^{1/2} & 0 \\ 0 & I_q \end{pmatrix} \left[\begin{pmatrix} \boldsymbol{r}^{(i)} \\ -\boldsymbol{u}^{(i)} \end{pmatrix} - \begin{pmatrix} U^{(i)} & V^{(i)} \\ I_q & \vec{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{u}}^{(i)} \\ \boldsymbol{\delta}_{\boldsymbol{\beta}}^{(i)} \end{pmatrix} \right] \right\|^2$$

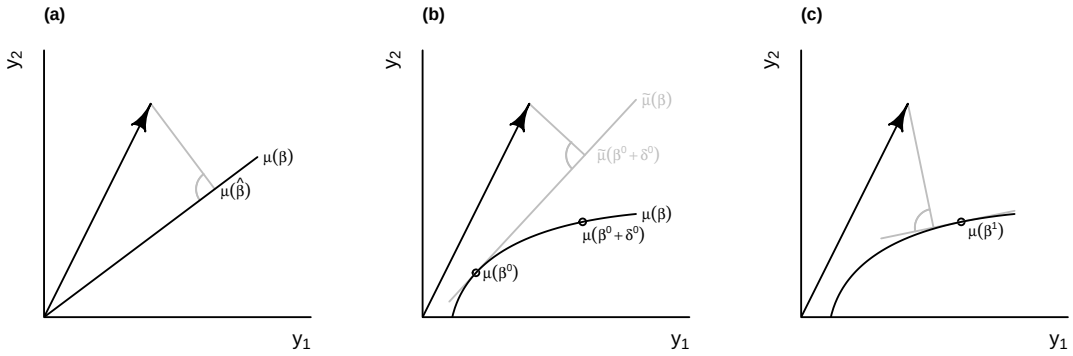


Figure 2.5: Geometric interpretation of Gauss-Newton method. The response vector $\boldsymbol{y}$ is shown as arrow in the response space. The solid black line represents the expectation $\boldsymbol{\mu}$ as a function of parameter $\boldsymbol{\beta}$. The goal of least squares estimation is to find $\hat{\boldsymbol{\beta}}$ which defines the closest orthogonal projection of the response vector on the expectation space. **(a)** In case of linear regression, the expectation space is a linear subspace spanned by $X\boldsymbol{\beta}$. There is an explicit solution for $\hat{\boldsymbol{\beta}}$. **(b)** In case of nonlinear regression, the expectation space is a nonlinear subspace and $\hat{\boldsymbol{\beta}}$ needs to be determined iteratively. $\mu(\boldsymbol{\beta})$ is linearly approximated around the initial guess $\boldsymbol{\beta}^0$. The response vector is orthogonality projected on the linear approximation $\tilde{\mu}$ to determine $\boldsymbol{\delta}^0$. **(c)** $\boldsymbol{\beta}^0$ is updated to $\boldsymbol{\beta}^1$ based on $\boldsymbol{\delta}^0$. The new estimate of $\boldsymbol{\beta}$ is already closer to the final estimate. The procedure is repeated until convergence.

with corresponding normal equation

$$\begin{pmatrix} U^{(i)\intercal} W U^{(i)} + I_q & U^{(i)\intercal} W V^{(i)} \\ V^{(i)\intercal} W U^{(i)} & V^{(i)\intercal} W V^{(i)} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_{\boldsymbol{u}}^{(i)} \\ \boldsymbol{\delta}_{\boldsymbol{\beta}}^{(i)} \end{pmatrix} = \begin{pmatrix} U^{(i)\intercal} W \boldsymbol{r}^{(i)} - \boldsymbol{u} \\ V^{(i)\intercal} W \boldsymbol{r}^{(i)} \end{pmatrix}. \tag{2.37}$$

Equation (2.37) can be seen as the Henderson equation for GLMMs. It has to be solved in each iteration in the course of finding conditional mode $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{\beta}}$ which in turn needs to be determined many times during the numerical optimization with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Fast methods to solve the normal equation are therefore even more important than in the LMM case.

Laplace Approximation

The integral in Equation (2.35) can be approximated with Laplace approximation using the conditional mode $\tilde{\boldsymbol{u}}$. First we define

$$g(\boldsymbol{u}) = \log\left(f_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y} \mid \boldsymbol{u}) f_{\boldsymbol{U}}(\boldsymbol{u})\right).$$

Like the integrand, the function $g(\boldsymbol{u})$ will have a global maximum at $\tilde{\boldsymbol{u}}$. A second order Taylor series expansion of $g(\boldsymbol{u})$ around $\tilde{\boldsymbol{u}}$ is given by

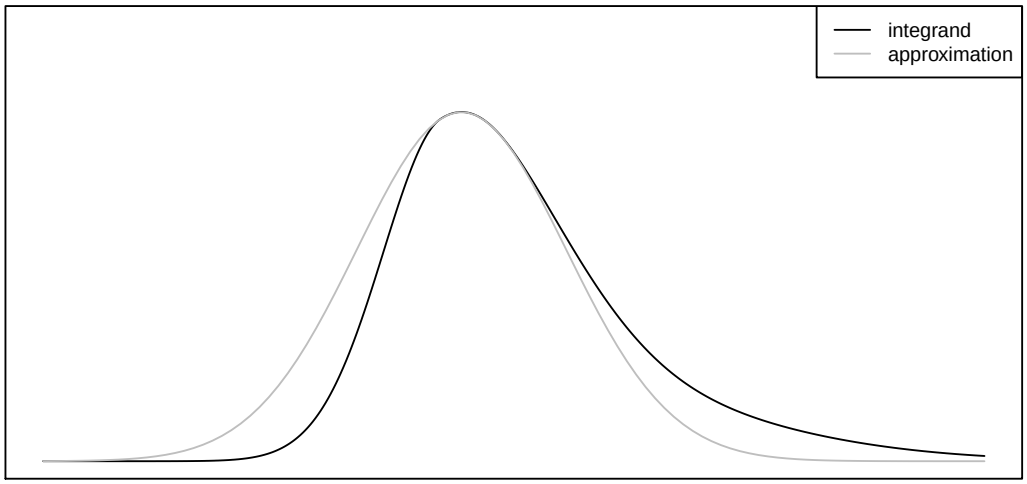


Figure 2.6: Laplace approximation. The integrand is approximated by an unnormalized Gaussian density with the same mode and second derivative at the mode.

$$g(\boldsymbol{u}) \approx g(\tilde{\boldsymbol{u}}) + \frac{1}{2} g''(\tilde{\boldsymbol{u}})(\boldsymbol{u} - \tilde{\boldsymbol{u}})^2$$

where the first order derivative $g'(\tilde{\boldsymbol{u}})$ is zero because $\tilde{\boldsymbol{u}}$ is a global maximum. Using the Taylor series approximation we can solve the Integral

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \boldsymbol{y}) = \int_{\mathbb{R}^q} \exp\left(g(\boldsymbol{u})\right) \mathrm{d}\boldsymbol{u} &\approx \int_{\mathbb{R}^q} \exp\left(g(\tilde{\boldsymbol{u}}) + \frac{1}{2} g''(\tilde{\boldsymbol{u}})(\boldsymbol{u} - \tilde{\boldsymbol{u}})^2\right) \mathrm{d}\boldsymbol{u} \\ &= \exp\left(g(\tilde{\boldsymbol{u}})\right) \int_{\mathbb{R}^q} \exp\left(-\frac{1}{2} \frac{(\boldsymbol{u} - \tilde{\boldsymbol{u}})^2}{(-g''(\tilde{\boldsymbol{u}}))^{-1}}\right) \mathrm{d}\boldsymbol{u} \\ &= \exp\left(g(\tilde{\boldsymbol{u}})\right) \sqrt{2\pi(-g''(\tilde{\boldsymbol{u}}))^{-1}}. \end{aligned}$$

The last step was done by identifying a Gaussian integral (Figure 2.6) with mean $\tilde{\boldsymbol{u}}$ and variance $(-g''(\tilde{\boldsymbol{u}}))^{-1}$ which satisfies

$$\sqrt{\frac{1}{2\pi(-g''(\tilde{\boldsymbol{u}}))^{-1}}} \int_{\mathbb{R}^q} \exp\left(-\frac{1}{2} \frac{(\boldsymbol{u} - \tilde{\boldsymbol{u}})^2}{(-g''(\tilde{\boldsymbol{u}}))^{-1}}\right) \mathrm{d}\boldsymbol{u} = 1.$$

2.5 Bayesian Estimation

In the Bayesian framework, we separate between unknown (unobserved) and known (observed) quantities. All unknown quantities are assumed to be random variables with a prior distribution, expressing our prior believe about the quantity. Bayesian inference for GLMMs has several advantages over likelihood inference (Zhao et al. 2006, Bürkner 2018, Gabry and Goodrich 2020a).

- Likelihood estimation for GLMMs usually involves Laplace approximation. Studies showed that the approximation can be inaccurate and lead to asymptotically biased estimators (e.g. Lin and Breslow 1996).
- Uncertainty estimates are considered to be more accurate. Accounting for the uncertainty related to the estimation of variance components is difficult in likelihood estimation (McCulloch and Searle 2000).
- Prior knowledge about parameters can be incorporated into the model by specifying informative prior distributions.
- The output is a posterior distribution which allows to make probability statements for every quantity of interest.
- Extensions to more complex models as generalized nonlinear mixed models or generalized additive mixed models are typically easier accomplished in the Bayesian framework.

The downside of Bayesian estimation is that it usually comes at a higher computational cost.

2.5.1 Generalized Linear Mixed Effects Model

In the likelihood framework, estimation of the LMM was fundamentally different compared to the GLMM. LMMs allowed to analytically integrate out $\boldsymbol{b}$ whereas in GLMMs this step had to be done using an approximation. In the Bayesian framework there is no need to integrate $\boldsymbol{b}$ out of the likelihood and the estimation procedures are the same (Gabry and Goodrich 2020a). Therefore, we are going to explain the procedure for GLMMs with the side note that LMMs are a special case of GLMMs where the response variable follows a normal distribution and the conditional mean is related the linear predictor by the identity link.

The unknown quantities are $\boldsymbol{b}$, $\boldsymbol{\theta}$, ϕ and $\boldsymbol{\beta}$. According to Bayes' Theorem, we can write their joint distribution conditional on the data as

$$\underbrace{P(\boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta} \mid \boldsymbol{y})}_{\text{posterior}} \propto \underbrace{P(\boldsymbol{y} \mid \boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta})}_{\text{likelihood}} \underbrace{P(\boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta})}_{\text{prior}}. \tag{2.38}$$

Note that $\boldsymbol{\theta}$ only acts on $\boldsymbol{y}$ through $\boldsymbol{b}$ (Figure 2.7), therefore

$$P(\boldsymbol{y} \mid \boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta}) = P(\boldsymbol{y} \mid \boldsymbol{b}, \phi, \boldsymbol{\beta}). \tag{2.39}$$

Assuming independence of the random effects parameters $\boldsymbol{b}$ and $\boldsymbol{\theta}$ from the remaining parameters $\boldsymbol{\beta}$ and ϕ, we can write

$$P(\boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta}) = P(\boldsymbol{b}, \boldsymbol{\theta})\, P(\phi, \boldsymbol{\beta}) \tag{2.40}$$

and from the definition of the conditional probability if follows

$$P(\boldsymbol{b}, \boldsymbol{\theta}) = P(\boldsymbol{b} \mid \boldsymbol{\theta})\, P(\boldsymbol{\theta}). \tag{2.41}$$

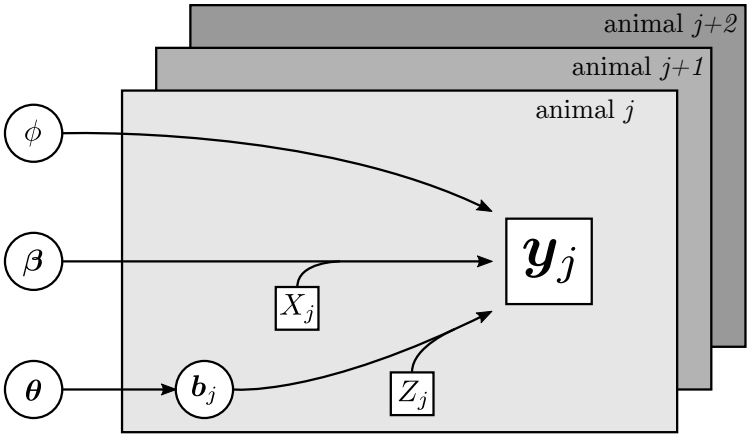


Figure 2.7: Graph representing the hierarchical character of a GLMM. Unknown quantities are shown in circles whereas known quantities are shown in squares. The arrows show how the quantities act on each other.

Inserting (2.39), (2.40) and (2.41) into (2.38) results in the 2-stage hierarchical Bayes model

$$P(\boldsymbol{b}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta} \mid \boldsymbol{y}) \propto P(\boldsymbol{y} \mid \boldsymbol{b}, \phi, \boldsymbol{\beta})\, P(\boldsymbol{b} \mid \boldsymbol{\theta})\, P(\boldsymbol{\theta})\, P(\phi, \boldsymbol{\beta})$$

with sampling distributions

$$\begin{aligned}(\boldsymbol{y} \mid \boldsymbol{b}, \phi, \boldsymbol{\beta}) &\sim \mathrm{EDM}_n(g^{-1}(X\boldsymbol{\beta} + Z\boldsymbol{b}), I_n\phi) \\ (\boldsymbol{b} \mid \boldsymbol{\theta}) &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, \Sigma_{\boldsymbol{\theta}}\right).\end{aligned}$$

Taking samples from the second sampling distribution will be difficult. Current random number generators cannot draw high dimensional vectors (order usually $> 10^6$) with such complicated correlation structure (Fouilloux and Laloë 2001). Similarly to what was done in the likelihood framework we can apply transformation (2.16) and (2.28) using Cholesky decomposition $\Sigma^* = \Lambda_{\boldsymbol{\theta}}\Lambda_{\boldsymbol{\theta}}^{\intercal}$. The resulting model is given by

$$P(\boldsymbol{u}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta} \mid \boldsymbol{y}) \propto P(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}, \phi, \boldsymbol{\beta})\, P(\boldsymbol{u})\, P(\boldsymbol{\theta})\, P(\phi, \boldsymbol{\beta}) \tag{2.42}$$

with sampling distributions

$$\begin{aligned}(\boldsymbol{y} \mid \boldsymbol{u}, \phi, \boldsymbol{\beta}) &\sim \mathrm{EDM}_n\left(g^{-1}(X\boldsymbol{\beta} + Z^*\Lambda_\theta \boldsymbol{u}), I_n\phi\right) \\ \boldsymbol{u} &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, I_q\right).\end{aligned}$$

The model is completed by assigning a prior distribution to parameter $\boldsymbol{\theta}$, ϕ and $\boldsymbol{\beta}$. Choosing the prior distribution is part of the modeling assumption and has to be done for each

data set separately according to prior knowledge. It is difficult to give general recommendations which hold in all possible situations. Still, there are some guidelines in literature on how to specify priors for GLMMs.

For the population-level ('fixed') effects $\boldsymbol{\beta}$ it is common to use so called uninformative priors like a Gaussian around zero with very large variance or even flat priors with uniform probability over all real numbers. However, such uninformative priors tend to put too much probability mass on very large parameter values. Generally, unless the data is very strong, it is more wisely to use mildly informative priors which provide moderate regularization and help to stabilize computation (Gabry and Goodrich 2020b).

The prior choice for the variance components $\boldsymbol{\theta}$ and ϕ is quite complex and has been subject to extensive research over the last few years. Variance components are special because the parameter space consists of only non-negative real numbers. Common prior choices for a single variance parameter include uniform (Gelman 2006), half Student-t (Gelman 2006) and inverse Gamma (Zhao et al. 2006). For random factors associated with multiple predictor variables, we do not only want to estimate the variance of the individual effects but also the covariance between them. Inverse Wishart distributions are a popular prior for covariance matrices in GLMMs because they are conditionally conjugate and therefore have good properties for the Gibbs sampler (Zhao et al. 2006). However, as shown in Natarajan and Kass (2000), the inverse Wishart prior can lead to poor estimates of the variance covariance matrix. An alternative is to separate the covariance matrix into a vector of standard deviations and a correlation matrix. Half Cauchy or half Student-t priors are applied on each element of the standard deviation vector whereas the correlation matrix can be modeled by an LKJ-correlation prior (Lewandowski, Kurowicka, and Joe 2009). The half Student-t prior with 3 degrees of freedom has commonly the better convergence property than the half Cauchy prior for elements of the standard deviation vector (Bürkner 2017).

In animal breeding literature, common prior choices for $\boldsymbol{\beta}$ are bounded uniform distributions and an inverse Wishart prior for the variance components (Chang et al. 2006, Heringstad et al. 2006).

The careful specification of prior distributions according to prior knowledge is especially important when the sample sizes are small. In this case, the posterior is not dominated by the sampling distributions and prior terms in Equation (2.42) may have a strong influence. In general, informative priors should be used if additional information is available.

2.5.2 Threshold Model

Model (2.26) can be transformed in the same way as done above (Equation 2.28) resulting in a model for the liability which is given by

$$
\begin{aligned}
(\boldsymbol{l} \mid \boldsymbol{U} = \boldsymbol{u}) &\sim \mathcal{N}_{\mathrm{n}} \left(X\boldsymbol{\beta} + Z^{*}\Lambda_{\theta}\boldsymbol{u}, I_n \right) \\
\boldsymbol{U} &\sim \mathcal{N}_{\mathrm{q}} \left(\vec{0}, \Sigma \right).
\end{aligned}
$$

Further, let's define μ_i as the expected liability of the ith observation

$$\begin{aligned}\mu_i &= \mathbf{E}\left[l_i \mid \boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\beta}\right] \\ &= (X\boldsymbol{\beta} + Z^*\Lambda_\theta \boldsymbol{u})_i.\end{aligned}$$

The classification is completely determined by the realized liability, therefore we can write

$$\begin{aligned}P(Y_i = y_i \mid \mu_i) &= P(\tau_{y_i-1} < l \leq \tau_{y_i} \mid \mu_i) \\ &= P(l \leq \tau_{y_i} \mid \mu_i) - P(l \leq \tau_{y_i-1} \mid \mu_i) \\ &= \Phi(\tau_{y_i} - \mu_i) - \Phi(\tau_{y_i-1} - \mu_i) \end{aligned} \tag{2.43}$$

with Φ denoting the cumulative distribution function of the standard normal distribution. Equation (2.43) seen as a function of the parameters represents the likelihood for one observation. Using the likelihood, we can formulate the posterior distribution of the parameters as

$$P(\boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\tau} \mid \boldsymbol{y}) \propto P(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\tau})\, P(\boldsymbol{u})\, P(\boldsymbol{\theta})\, P(\boldsymbol{\beta})\, P(\boldsymbol{\tau}) \tag{2.44}$$

with sampling distributions

$$\begin{aligned}P(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\tau}) &= \prod_{i=1}^{n} \Phi(\tau_{y_i} - \mu_i) - \Phi(\tau_{y_i-1} - \mu_i) \\ \boldsymbol{u} &\sim \mathcal{N}_{\mathrm{q}}\left(\vec{0}, I_q\right).\end{aligned}$$

Various prior distributions of $\boldsymbol{\tau}$ are suggested in literature including the Student-t and the uniform distribution (Van Tassell, Van Vleck, and Gregory 1998, Heringstad et al. 2006, Bürkner and Vuorre 2019).

2.5.3 Estimating the posterior distribution

The posterior distribution in Equation (2.42) and (2.44) cannot be derived analytically for most combinations of sampling and prior distributions. The difficulty lies in the evaluation of the normalization constant in Equation (2.42). It involves a high dimensional integral which in practice might be impossible to solve. Markov chain Monte Carlo (MCMC) methods allow us to sample from the unnormalized posterior distribution and use the finite sample to approximate the true posterior distribution. The idea is to create a Markov chain of observations which moves stochastically through the parameter space and single steps are chosen such that the final chain is more likely to stay at regions of high probability. Metropolis-Hastings is the most prominent algorithm to get a Markov chain with the desired property. Given an initial state, the algorithm first proposes a random step in the parameter space. The step is always taken if the unnormalized posterior probability at the proposed state is higher than at the initial state. However, if the unnormalized posterior probability is lower, the step will only be taken with probability proportional to the ratio of the unnormalized posterior probabilities. It can be shown that the distribution of the samples obtained by the Metropolis-Hastings algorithm will asymptotically be

the posterior distribution (Hastings 1970). Therefore, we can approximate the posterior distribution without having to calculate the normalization constant.

Despite the great success of the Metropolis-Hastings algorithm, it is not always the computationally most efficient method for obtaining a sample from an unnormalized distribution. Especially in a very high dimensional parameter space, most of the proposals will be rejected and a relatively low number of steps are taken compared to the number of calculations which need to be done. A better approach would not just randomly propose steps in any direction of the parameter space but instead make an informed suggestion based on the geometrical properties of the unnormalized probability function. This is the main idea behind Hamiltonian Monte Carlo. It is based on the fact that for high dimensional distributions, the vast majority of the probability mass is located at an intermediate region, called typical set, in between the median region and the outer regions where the probability is close to zero. Hamiltonian Monte Carlo algorithms allow to wander within the typical set without unnecessarily proposing and rejecting steps which would lead outside of the typical set. A nice visual introduction to Hamiltonian Monte Carlo can be found in Betancourt (2018).

Chapter 3

Implementation

3.1 Likelihood Approach

3.1.1 Existing Packages

The most commonly used packages in R for fitting mixed models in the likelihood framework are `nlme` (Pinheiro, Bates, and R-core 2020) and `lme4` (Bates, Maechler, et al. 2020). The package `nlme` was primarily designed for fitting LMMs and nonlinear mixed effects models (NLMMs). It is relatively slow, lacks support for GLMMs and is no longer actively developed, which makes it a bad starting point for our implementation. `lme4` on the other hand features sparse matrix representation and is partially implemented in C++ which makes it much faster for large data sets. The package is primarily used for fitting LMMs and GLMMs but also has some limited capabilities for NLMMs. It is well documented and still actively maintained and developed. For this reason we have chosen `lme4` as the base structure for estimating breeding values in the likelihood framework.

The fitting functions of the package are structured in four modules (Bates, Mächler, et al. 2015).

1. The *formula module* creates a list of objects which are necessary for model fitting. This includes the model matrices X and Z as well as the left Cholesky factor $\Lambda_{\boldsymbol{\theta}}$.
2. The *objective function module* creates a function which takes $\boldsymbol{\theta}$ as an input and returns the deviance. In LMMs, Equation (2.33) is solved for $\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{u}}$ and then, in a second step, the function evaluates the deviance by solving the integral in Equation (2.31) using $\tilde{\boldsymbol{u}}$.
3. The *numerical optimization module* uses a gradient free algorithm to minimize the deviance function with respect to $\boldsymbol{\theta}$. Commonly used optimizers which are well supported by `lme4` include BOBYQA and Nelder-Mead. However, the user can also define his own optimizer.
4. The *output module* stores the results in a useful object of class `MerMod`. There are several methods available for this class to extract information from the model and do inference.

The structure is slightly changed for GLMMs . As mentioned in Chapter 2.4.2, numerical optimization in GLMMs needs to be done with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. A preceding simplified optimization can be useful to find good starting values. To this end, the deviance function

is first minimized with respect to $\boldsymbol{\theta}$ alone, using the conditional mode $\tilde{\boldsymbol{\beta}}$ as estimate for $\boldsymbol{\beta}$. The resulting estimate $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\beta}}$ evaluated at $\tilde{\boldsymbol{\theta}}$ are then used as starting value for the final numerical optimization with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

The time-critical step of solving Equation (2.33) in LMMs and (2.37) in GLMMs make use of the CHOLMOD library of C functions (Chen et al. 2008). In this way, the spares structure of Z and $\Lambda_{\boldsymbol{\theta}}$ can be exploited in the Cholesky decomposition of the system matrix.

`lme4` can fit a wide variety of models with nested, partially- or fully crossed random effects. However, it is relatively restrictive with Σ, the covariance matrix of the random effects. By default there is no support for defining a specific correlation structure in the random effects as we observe it between the breeding values of related animals. The package `pedigreemm` (Bates and Vazquez 2014) was released in 2009 to address this feature. It is built on top of `lme4` with the goal of making `lme4`'s fast computational machinery available for genetic evaluations (Vazquez, Bates, et al. 2010).

`pedigreemm` contains classes and functions to define a pedigree and calculate the corresponding left Cholesky factor L_A. At the core of the package is the function `pedigreemm()`. It allows fitting LMMs and GLMMs with correlated random effects based on a specified pedigree. Internally, the function performs the following steps

1. All arguments given to the function `pedigreemm()` except for the `pedigree` argument are passed to the corresponding fitting function of `lme4`.
2. The model is fitted without considering the correlation structure of the random effects.
3. The left Cholesky factor L_A is calculated from the pedigree.
4. Model matrix Z is extracted from the output of step 2 and transformed to Z^* in a similar but less general way as it is done in Equation (2.16).
5. The transformed model has uncorrelated random effects between animals and is passed to `lme4` for estimation. The starting values of the optimizer are chosen according to the fitted model in step 2.
6. All model components are collected in a list and returned as an object of class `lmerpedigreemm`.

The output can be inspected with the same functions as used on fitted objects in `lme4`. A method of the generic function `ranef()` automatically transforms the predicted random effects $\tilde{b}^*$ back to their original scale. Unfortunately, the package has some serious flaws which needs to be addressed.

Bug in back transformation of random effects

The back transformation of the random effects contains a small bug with serious consequences. The transformation is done with the right Cholesky factor $L_A^\intercal$ instead of the left Cholesky factor L_A. The resulting random effects are wrong for all animals with older related animals in the pedigree. We have performed some simulations based on the pedigrees included in the package. The predicted random effects had consistently a larger mean squared error (MSE) than predicted random effects resulting from the correct

```
pedigree <- list(grouping_factor_1 = pedigree_1,
                 ...,
                 grouping_factor_N = pedigree_N)
for (i in 1:N) {
  LA <- relfactor(pedigree_i)
  index <- index of columns associated with grouping factor i
  Z[,index] <- Z[,index] %*% LA
}
```

Code 3.1: Pseudocode of Z matrix transformation in `pedigreemm` package

transformation. The increase in MSE highly depended on the signal to noise ratio of the simulation and the average relationship of the animals.

The bug was already present in the first version of the package and surprisingly has been undetected ever since. The package has already more than 38'000 downloads from CRAN and has been cited by 79 publications (status as of 2020-06-29), many of which using it to predict breeding values and relying on the wrong results. The corresponding author was contacted but the bug has not been resolved yet.

Animal model with single observations

The animal model assumes an individual random effect for each animal. For some traits, there is only one observation per animal. The model would be unidentifiable without considering the correlation between breeding values. However, an animal model with single observations should be identifiable as long as not all animals are unrelated as we have seen in Chapter 2.2.5.

`lme4` by default performs some checks in order to detect ill-specified models. It will automatically throw an error if the number of levels in one grouping factor is not smaller than the number of observations. The automatic check is not suppressed by `pedigreemm`. Therefore, it is impossible to fit animal models which actually would be identifiable. This issue has repeatedly been raised in an R mailing list (R Mailing List 2010, R Mailing List 2012, R Mailing List 2014) but the package authors have not addressed the issue so far.

No support for random regression models

The package implements the transformation of the model matrix Z in a very simple way. Users can associate a grouping factor with a pedigree in the argument `pedigree`. Each association is saved as one element of a list (Code 3.1). The function `pedigreemm` loops trough all elements, calculates L_A of the pedigree and overwrites the columns of matrix Z which corresponds to the grouping factor by the transformed version. The simple implementation does not allow for more than one term in the random effect expression (i.e. $p_i > 1$). Also, the pedigree grouping factor can only be used in one term, otherwise the function will throw an error.

Slow for large data sets

Data sets in animal breeding can be huge. `lme4` does have a focus on performance but the methods generally rely on the sparsity of matrix Z and $\Lambda_{\boldsymbol{\theta}}$. The transformed matrix

Z^* is no longer sparse in general leading to drastic increase in running time.

3.1.2 Generalization and improvement

Our goal was to implement an improved version of the `pedigreemm()` function which is able to fit LMMs and GLMMs with correlated random effects based on a pedigree. The function should focus on speed, allow for random regression models and fix the critical bug of the `pedigreemm` package. The structure was chosen to follow as closely as possible the current structure of the `lme4` fitting functions, allowing for an easier inclusion of future `lme4` updates. Therefore, we decided to write two separate functions, `cowfit_lmer()` for fitting LMMs and `cowfit_glmer()` for fitting GLMMs.

Avoid model fitting before transformation

The `pedigreemm()` function started with fitting the model before transforming it. In many cases this can make sense. However, the untransformed model might be unidentifiable leading to estimates for $\boldsymbol{\theta}$ which are way off and represent bad starting points for the subsequent optimization of the transformed model. The `cowfit` functions avoid fitting the untransformed model and instead only calculates the important model components (Z, $\Lambda_{\boldsymbol{\theta}}$, ...) which can be achieved quickly with the formula module functions of `lme4`. The check for the number of levels in each grouping factor is suppressed, allowing to fit animal models with single observations.

Model transformation

The transformation of the Z matrix is based on Equation (2.16). The function `get_TAt()` (Code 3.2) is used to calculate the transformation factor $T_A^\intercal$ from the output of the formula module function and the argument `pedigree`. Note that we directly calculate the transpose of the transformation factor in Equation (2.16). $T_A^\intercal$ appears to be more useful due to some computational shortcuts in `lme4`. The transpose of the model matrix Z shows a regular column patterns which is preferred by the CHOLMOD library for storage reasons (Bates, Mächler, et al. 2015). It appears that all calculations in `lme4` can be done with $Z^\intercal$ whereas Z never has to be evaluated. For this reason we will have to transform $Z^\intercal$ which can be done using

$$Z^{*\intercal} = T_A^\intercal Z^\intercal.$$

$T_A^\intercal$ is additionally stored in the final output of the fitting function.

Optimization with predefined variance components

Speed is one of the major issues when using `pedigreemm()` to fit a full data set of an animal population. The runtime quickly increases for more complex models with larger number of animals. Profiling the functions shows that the most computationally demanding part is by far the numerical optimization with respect to $\boldsymbol{\theta}$ (and $\boldsymbol{\beta}$ in case of GLMM).

From the breeders point of view, the variance components, which are determined by $\boldsymbol{\theta}$, can be regarded as nuisance parameters and are not of primary interest. The main focus is on the breeding values or more specifically the ranking of the breeding values which usually is not too much affected by assuming slightly wrong variance components. In breeding

```
1  get_TAt <- function(lmod, pedigree) {
2    pnms <- names(pedigree)        # Names of pedigree factors
3    fl <- lmod$reTrms$flist        # Factor list
4    asgn <- attr(fl, "assign")     # Which factor corresponds to which term
5    TAt_list <- vector("list", length = length(asgn))  # output list
6    for(i in seq_along(asgn)){
7      fac_name <- names(fl)[asgn[i]]
8      p_i <- length(lmod$reTrms$cnms[[i]])                    # dim of Sigma_i
9      l_i <- length(levels(lmod$reTrms$flist[[fac_name]]))    # dim of L_Ai
10     on_list <- fac_name %in% pnms
11     if (on_list) {
12       Zt_i <- lmod$reTrms$Ztlist[[i]]                       # get Zt matrix
13       fac_levels <- rownames(Zt_i)[seq(1, length(rownames(Zt_i)), p_i)]
14       Lt_Ai <- relfactor(pedigree[[fac_name]], fac_levels)
15       TAt_list[[i]] <- kronecker(Lt_Ai, diag(p_i))
16     } else {
17       TAt_list[[i]] <- diag(p_i*l_i)
18     }
19   }
20   TAt <- Matrix::bdiag(TAt_list)
21   as(TAt, "dtCMatrix")
22 }
```

Code 3.2: The function to calculate the transformation factor $T_A^\intercal$. The argument `lmod` contains the output of the formula module function in `lme4` whereas the second argument `pedigree` is a named list similar as in Code 3.1. `lme4` changes the order of random terms according to the number of levels for efficiency reasons. `asgn` keeps track of which grouping factor corresponds to which of the reordered terms (line 4). The function loops through all terms (line 6–18) and checks for each term whether it has a grouping factor which is associated with a pedigree (line 10). If this is the case, $L_{A_i}^\intercal$ is calculated with the function `relfactor()` and the term-wise transformation factor is obtained from $L_{A_i} \otimes I_{p_i}$ (line 15). An identity matrix of the correct dimensions $p_i l_i$ is constructed if the grouping factor is not associated with a pedigree. Finally, all term-wise transformation factors are combined to one block diagonal matrix and returned as an object of class `dtCMatrix`.

programs of Qualitas, variance components of most traits are estimated only once every four year and afterwards assumed to stay constant for regular breeding value estimations. This highlights the importance of having a fast method for estimating breeding values given the variance components.

Unfortunately, `lme4` does not support specifying the variance components by default and we need to come up with a solution by ourselves. The variance components should be passed as a vector to the fitting function. First, we need to define the ordering of the variance components in the vector. We stick to the same rules as used by `lme4` in the method `as.data.frame()` for objects of class `VarCorr.merMod` which we describe in the following.

First we have a look at the ordering of the variance components within one term. The random term i of form `(REexp_i | factor_i)` with p_i columns in the raw model matrix X_i will add

$$m_i = \binom{p_i + 1}{2}$$

variance components to the model. We require the variance components to be specified in the following order

1. All variance terms in the order as they are specified in `REexp_i`.
2. All covariance terms between column $j = 1, \ldots, p_i$ and all subsequent columns in the order as they are specified in `REexp_i`.

Between the different terms, the ordering is defined by the number of levels in the grouping factor in descending order. The residual variance is added as the last element. The last element should be set to 1 for GLMMs without common scale parameter ϕ.

Example 3.1. Assume an LMM of the form `y ~ (a + b + c | fac1) + (d + e | fac2)` where factor 2 has more levels than factor 1. The term-wise covariance matrices will have the form

$$\Sigma_1 = \begin{pmatrix} \sigma_a^2 & \sigma_{ab}^2 & \sigma_{ac}^2 \\ \sigma_{ab}^2 & \sigma_b^2 & \sigma_{bc}^2 \\ \sigma_{ac}^2 & \sigma_{bc}^2 & \sigma_c^2 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} \sigma_d^2 & \sigma_{de}^2 \\ \sigma_{de}^2 & \sigma_e^2 \end{pmatrix}$$

The ordered variance component vector is given by

$$\begin{pmatrix} \sigma_d^2 & \sigma_e^2 & \sigma_{de}^2 & \sigma_a^2 & \sigma_b^2 & \sigma_c^2 & \sigma_{ab}^2 & \sigma_{ac}^2 & \sigma_{bc}^2 & \sigma^2 \end{pmatrix}^\intercal$$

■

Because the ordering is quite complex and error-prone, we provide the helper function `cowfit_var_comp()` which prints the ordering for a specific model formula.

The correctly ordered variance component vector is passed to the fitting function via the argument `var_comp`. The fitting function will check the variance components and throw an error if the vector has the wrong length or if the resulting variance covariance matrix Σ would not be positive semidefinite. The variance component vector is transformed to the vector $\boldsymbol{\theta}$ by the function `var_to_theta()` (Code 3.3). $\boldsymbol{\theta}$ can finally be used to evaluate the deviance function. However, this last step has major differences between LMMs and GLMMs.

In LMMs, `lme4` profiles σ^2 out of the deviance function. The variance components are not directly contained in $\boldsymbol{\theta}$ but on a scale relative to σ^2. Therefore, a predefined $\boldsymbol{\theta}$ will only fix the ratio between variance components and σ^2 but not the absolute value of the variance components. The genetic variance components might be well known for a certain trait in a specific population and should not change too much over time because they are only determined by the available alleles and their frequency. However, σ^2 depends on the data quality which might be highly specific for each data set. The fitted variance components are not exactly equal to the predefined if the predefined σ^2 does not exactly match the σ^2 of the data set.

There is the possibility to numerically determine $\boldsymbol{\theta}$ which minimize the MSE between predefined and obtained variance component. The procedure is inspired by a post of

```
1  var_to_theta <- function(var_comp, cnms){
2    sigma_order <- function(nc){
3      M <- matrix(0,nc,nc)
4      diag(M) <- 1:nc
5      M[lower.tri(M)] <- (nc+1):(nc*(nc+1)/2)
6      M[lower.tri(M, TRUE)]
7    }
8    nc <- lengths(cnms)
9    ncseq <- seq_along(nc)
10   lt <- split(var_comp[1:(length(var_comp)-1)],
11               rep.int(ncseq, (nc * (nc + 1))/2))
12   out <- vector("list", length = length(nc))
13   for(i in seq_along(nc)){
14     rowIndices <- rep(1:nc[i], 1:nc[i])
15     colIndices <- sequence(1:nc[i])
16     template <- sparseMatrix(rowIndices, colIndices, x = 1)
17     template@x <- as.double(lt[[i]][sigma_order(nc[i])])      # fill in sigma
18     Sigma_i <- Matrix::forceSymmetric(template, uplo = "L")  # make symmetric
19     chol_sigma <- tryCatch({chol(Sigma_i)},
20                            error = function(cond){
21         stop("var_comp does not lead to positive semi-definite matrix.")})
22     Lambdat <- t(chol_sigma/sqrt(var_comp[length(var_comp)]))
23     out[[i]] <- Lambdat@x   # extract non-zero elements
24   }
25   unlist(out)               # combine all theta_i to theta
26 }
```

Code 3.3: The function to transform the variance component vector to vector $\boldsymbol{\theta}$. Argument `cnms` is a list where the ith entry contains the column names of matrix X_i and argument `var_comp` contains the correctly ordered variance components. The variance components are split into a list with k entries containing the variance components of each term (line 8–11). The function loops through all terms. A lower-triangular template matrix of Σ_i with the proper dimensions is created (line 16). The `sigma_order()` function reorders the variance components of a single term, such that they can be column-wise filled into the lower triangular template of matrix Σ_i. The filled template is expanded to the symmetric matrix Σ_i (line 18). Λ_i is calculated according to the Equation $\Sigma_i = \Lambda_i \Lambda_i^\intercal \sigma^2$ and the elements of Λ_i are column wise combined to $\boldsymbol{\theta}_i$. Finally, $\boldsymbol{\theta}$ is created by combining all $\boldsymbol{\theta}_i$.

Ben Bolker on Stack Overflow (Stack Overflow 2016) and implemented in our function `Bolker_exact_var_comp()`. Our implementation is much faster compared to the one suggested on Stack Overflow because we do not optimize with respect to the possibly higher dimensional $\boldsymbol{\theta}$ but only with respect to the one dimensional scaling factor (Code 3.4). Still, the algorithm relies again on numerical optimization and is much slower than the direct approach where only the ratio between variance components and σ^2 is conserved. The user can choose with the argument `exact_var_comp` whether the final variance components should exactly match the predefined ones or if only the ratio of variance components and σ^2 is fixed (Figure 3.1).

In GLMMs, not all distributions have a scale parameter and it is not profiled out of the deviance function by default. Therefore, the variance components are determined by $\boldsymbol{\theta}$ on an absolute scale and there is no need to numerically determine the scaling in order to get exact variance components in the final model. Still, the optimization with prespecified variance components in GLMMs is not trivial because, in contrast to LMMs, numerical

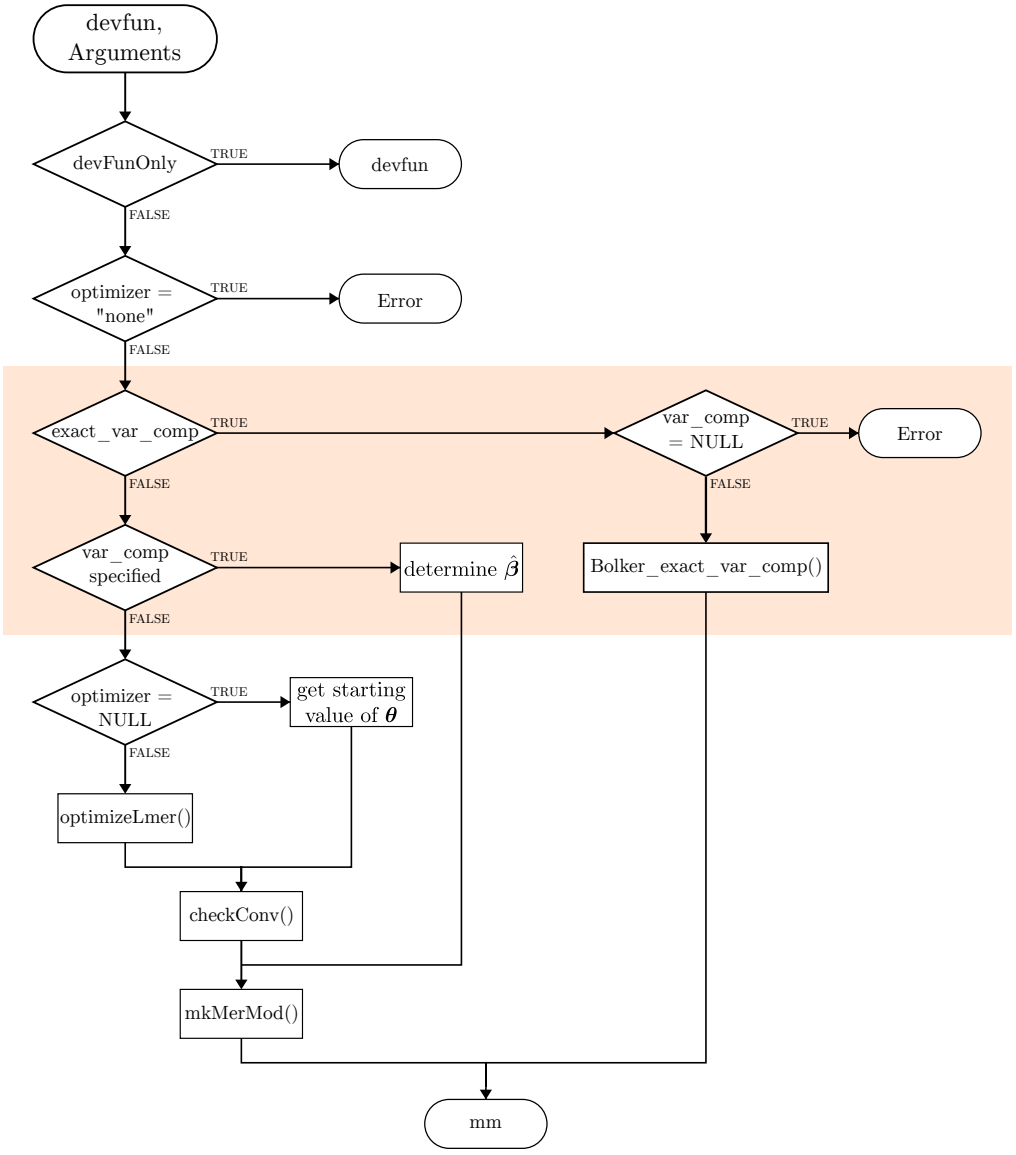


Figure 3.1: Numerical optimization module of `cowfit_lmer()`. The colored part is added to the usual `lmer()` procedure to allow for prespecified variance components.

```r
Bolker_exact_var_comp <- function(devfun, var_comp, lmod, mcout) {
  manual_theta <- var_to_theta(var_comp = var_comp, cnms = lmod$reTrms$cnms)
  buildMM <- function(theta){
    ff <- devfun(theta)
    opt <- list(par=theta, fval = ff, conv = 0)
    mkMerMod(environment(devfun), opt, lmod$reTrms, fr = lmod$fr, mc = mcout)
  }
  objfun <- function(x, target = var_comp[-length(var_comp)]){
    scaled_theta <- manual_theta*x
    mm <- buildMM(scaled_theta)
    myvcov <- as.data.frame(VarCorr(mm))$vcov
    return(sum((myvcov[-length(myvcov)] - target)^2))
  }
  opt <- optim(fn=objfun, par = 1, method = "L-BFGS-B", lower = 0)
  buildMM(manual_theta*opt$par)
}
```

Code 3.4: Numerical optimization to get exact variance components in LMMs. The function `buildMM()` takes the vector $\boldsymbol{\theta}$ and returns the fitted model. The function `objfun()` scales $\boldsymbol{\theta}$ by the scaling factor `x`, fits the model and returns the sum of the squared deviation between predefined and obtained variance components. `optim` uses numerical optimization to minimize this criterion with respect to `x`. Finally, the fitted model at the best scaled $\boldsymbol{\theta}$ is returned.

optimization needs to be done with respect to $\boldsymbol{\beta}$ (Figure 3.2).

Predict random effects

Random effects cannot be estimated because they are random. However, we can calculate the mode $\tilde{\boldsymbol{b}}^*$ of the conditional distribution $\boldsymbol{B}^* \mid \boldsymbol{Y} = \boldsymbol{y}$ which represents the most likely realization of $\boldsymbol{B}^*$ given $\boldsymbol{y}$. The calculation of $\tilde{\boldsymbol{b}}^*$ and separation into components of different terms is well implemented in `lme4` by the function `ranef()`. It only remains to back-transform them to the original random effects vector $\boldsymbol{b}$. `pedigreemm` does this separately for each term which becomes increasingly complex with random regression models. An easier solution is to back-transform $\tilde{\boldsymbol{b}}^*$ before it is separated into components of different terms. The method of `ranef()` for objects of class `lme4cowfit` achieves this by only adding a single line of code to the original `ranef()` function.

3.2 Bayesian Approach

3.2.1 Existing Packages

Hamiltonian Monte Carlo sampling is implemented in the programming language Stan (Carpenter et al. 2017). The package `rstan` (Guo, Gabry, and Goodrich 2020) provides an R interface to Stan allowing to fit Stan models directly from R. Still, it requires the user to be fluent in the Stan language and implies much more typing to fit a GLMM compared to the simple `lme4` syntax. Luckily, there are packages which translates R formula syntax to arguments required by `rstan`, notably `rstanarm` (Gabry and Goodrich 2020c) and `brms` (Bürkner 2020). As an additional benefit, these packages guarantee an efficient parameterization of the Stan model and provide weakly informative default priors which show good performance in most practical applications.

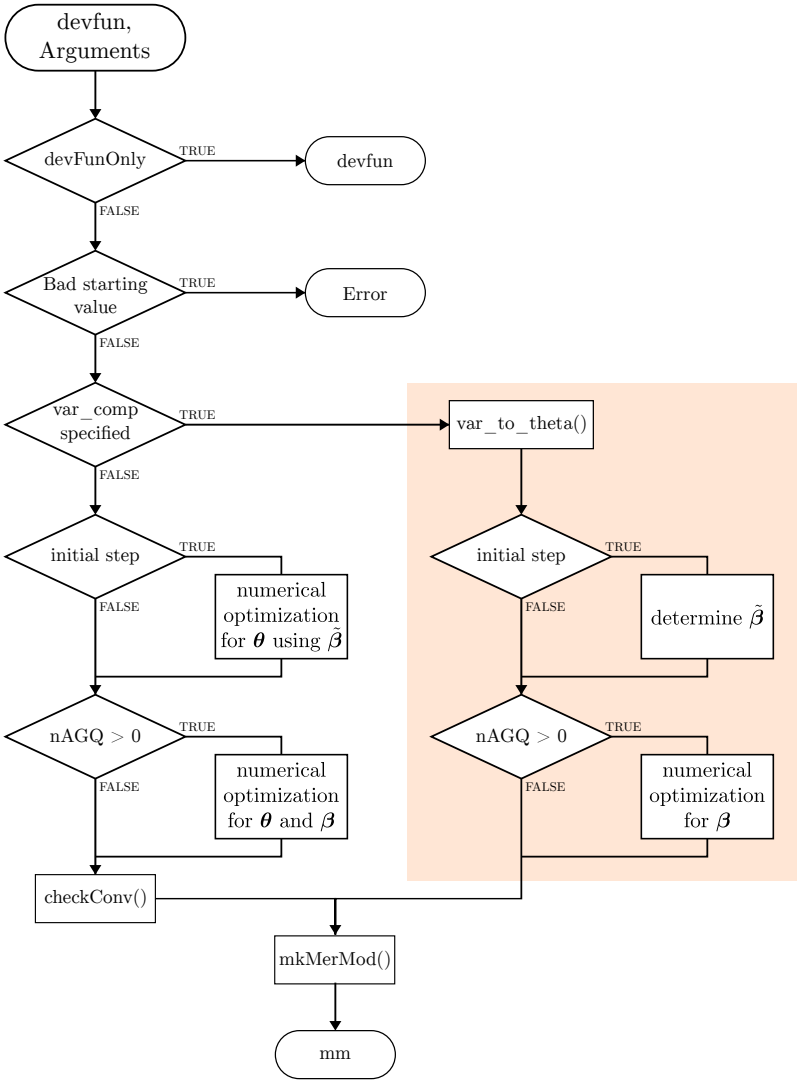


Figure 3.2: Numerical optimization module of `cowfit_glmer()`. The colored part is added to the usual `glmer()` procedure to allow for prespecified variance components. The structure with prespecified variance components mirrors the original structure. The initial step is a simplified optimization to find good starting values. In the case of prespecified variance components the initial step only includes the PIRLS algorithm in order to find the conditional mode $\tilde{\boldsymbol{\beta}}$. In the second stage, $\tilde{\boldsymbol{\beta}}$ is used as a starting value for the numerical optimization of the deviance function at a fixed $\boldsymbol{\theta}$. The second stage can be suppressed with setting argument `nAGQ = 0`.

The main difference between the packages concerns the compilation of the Stan script. `brms` writes the Stan script and compiles it in runtime which makes it very flexible but, due to the additional compilation time, relatively slow for small data sets. `rstanarm` on the other hand has precompiled Stan scripts which makes it impossible to implement models which are not natively supported.

An implementation of GLMMs with pedigree correlations structure of the random effect turned out to be very difficult in `rstanarm`. The function code allows to transform the random effect model matrix Z, however, the precompiled Stan script does not always make use of the Z matrix directly. Instead, under some conditions it uses some computational shortcuts which do not reliably work with a transformed Z matrix. Therefore, our implementation of Bayesian GLMMs is based on `brms`.

Parameterization in `brms` differs from the one used in Chapter 2.5.1 (which closely followed the parameterization used by `lme4`). Previously, all variance component parameters were combined to vector $\boldsymbol{\theta}$. There is no vector $\boldsymbol{\theta}$ in `brms`. Instead, the parameters of term i are separated between vector $\boldsymbol{\sigma}_i$ and Cholesky factor C_i. Together, they define the term-wise variance covariance matrix Σ_i. $\boldsymbol{\sigma}_i$ contains the root of the diagonal elements of Σ_i and C_i is the left Cholesky factor of the corresponding correlation matrix of Σ_i such that

$$\begin{aligned}\Sigma_i &= \operatorname{diag}(\boldsymbol{\sigma}_i)\,\Sigma_{\text{cor},i}\,\operatorname{diag}(\boldsymbol{\sigma}_i)\\ &= \operatorname{diag}(\boldsymbol{\sigma}_i)\,C_i C_i^{\intercal}\,\operatorname{diag}(\boldsymbol{\sigma}_i)\,.\end{aligned}$$

The random effects of different terms are stored in separate variables, therefore the linear predictor becomes

$$\boldsymbol{\eta} = X\boldsymbol{\beta} + Z_1\boldsymbol{b}_1 + \cdots + Z_k\boldsymbol{b}_k.$$

`brms` has built-in support for correlated random effects. The variance covariance matrix of the random effects can be assigned to a specific grouping factor using the function `gr()` in the model formula

`y ~ FEexpr + (REexp1 | gr(factor1, cov = VarCovMatrix1)) + ...`.

The covariance is taken into account by transforming a vector of initially independent random effects similarly to the transformation approach described in Chapter 2.2.4. The notation differs due to the different parameterization but the resulting random effects are equivalent. If we apply the transformation as it is defined in Equation (2.16) and (2.28) with $\Sigma_i^* = \Lambda_{\boldsymbol{\theta}i}\Lambda_{\boldsymbol{\theta}i}^{\intercal}$, the linear predictor becomes

$$\boldsymbol{\eta} = X\boldsymbol{\beta} + Z_1^*\Lambda_{\boldsymbol{\theta}1}\boldsymbol{u}_1 + \cdots + Z_k^*\Lambda_{\boldsymbol{\theta}k}\boldsymbol{u}_k.$$

The contribution of the ith random term can now be rewritten as

$$
\begin{aligned}
Z_i^* \Lambda_{\boldsymbol{\theta} i} \boldsymbol{u}_i &= Z_i \left(L_{A_i} \otimes I_{p_i} \right) \left(I_{l_i} \otimes T_i \right) \boldsymbol{u}_i \\
&= Z_i \left(L_{A_i} \otimes T_i \right) \boldsymbol{u}_i \\
&= Z_i \left(L_{A_i} \otimes \operatorname{diag} \left(\boldsymbol{\sigma}_i \right) C_i \right) \boldsymbol{u}_i
\end{aligned} \tag{3.1}
$$

which represents the transformation of the ith term as it is done in `brms`.

At the core of the `brms` package is the fitting function `brm()`. It contains two important internal functions namely `.make_stancode()` which writes the Stan script and `.make_standata()` which prepares a list of all input data to be used in the Stan script. `brms` in its current form already allows to fit GLMMs for genetic evaluations, thanks to the implemented transformation to independent random effects. Additionally, it has built-in support for threshold models which can be used by specifying the argument `family = cumulative("probit")`. Further information about the threshold model implementation in `brms` can be found in Bürkner and Vuorre (2019). Still, compared to the likelihood implementation described in Chapter 3.1.2, there are a few adaptations which could be useful

1. The syntax to specify the variance covariance matrix is different to the syntax as we know it from the package `pedigreemm`. It could be easier for the end user if the functions for likelihood and Bayesian estimation both use the same syntax.
2. The correlation between random effects can only be specified by explicitly calculating the variance covariance matrix. However, as we have seen in Chapter 2.2.4 the calculation of the additive numerator relationship matrix for large pedigrees comes at huge computational cost whereas the calculation of its Cholesky factor is much simpler. Only the Cholesky factor is necessary to apply the random effect transformation in Equation (3.1). Therefore, it should be possible to avoid the calculation of the additive numerator relationship matrix and directly pass the Cholesky factor to an adapted version of `brm()`
3. Currently, there is no option to prespecify the variance components in `brms`. Prespecifying the variance components would be useful for faster estimation of the random effects.

3.2.2 Implement improvements

In order to use a similar syntax in the Bayesian implementation, we created the wrapper function `cowfit_brm()`. The function takes similar arguments as `cowfit_glmer()` and transforms them into the corresponding `brms()` syntax.

The second improvement avoids the calculation of the additive numerator relationship matrix. The wrapper function directly passes the Cholesky factors to `brm()`. Two internal functions of `brms` needs to be changed in order to make sure the Cholesky factors are not used as variance covariance matrices. The function `validate_recov_matrix()` would test the input for symmetry and positive definiteness. In a later step, `.make_standata()` applies a Cholesky decomposition to the input and saves it in a list containing all data used by the Stan script. Our adapted versions of the internal functions avoid the tests and Cholesky decomposition. Instead, the input is directly used as Cholesky factor in the Stan script.

Two changes are necessary to implement predefined variance components in `brms`. The variance components needs to be transformed to the `brms` parameterization and the internal functions `.make_stancode()` and `.make_standata()` needs to be adapted. The first change is implemented in the function `var_comp_to_stan_format()`. As input, it takes the variance component in the same format as outlined in Chapter 3.1.2. The term-wise variance covariance matrices are constructed and decomposed into $\boldsymbol{\sigma}_i$ and C_i. These elements are added to the data list constructed by an adapted version of `.make_standata()`. Finally, `.make_stancode()` is adapted such that $\boldsymbol{\sigma}_i$ and C_i are no longer added as parameters but instead as data (Code 3.5).

3.3 Package

The new functions are included in the package `cowfit`. Installing the package guarantees that all necessary dependencies are also installed. Additionally, the exported functions come with a help file including useful examples on how to apply them. A vignette about how to get started is also provided. The package is still in development and not everything might be perfectly tested. It can be obtained from GitHub (Zihlmann 2020).

```
1  data {
2    int<lower=1> N;          // number of observations
3    vector[N] Y;             // response variable
4    int<lower=1> N_1;        // number of grouping levels
5    int<lower=1> M_1;        // number of coefficients per level
6    int<lower=1> J_1[N];     // grouping indicator per observation
7    matrix[N_1, N_1] Lcov_1; // cholesky factor of known covariance matrix
8    vector[N] Z_1_1;         // group-level predictor values
9    vector[N] Z_1_2;
10 }
11 parameters {
12   real Intercept;           // temporary intercept for centered predictors
13   real<lower=0> sigma;      // residual SD
14   vector<lower=0>[M_1] sd_1;  // group-level standard deviations
15   matrix[M_1, N_1] z_1;     // standardized group-level effects
16   cholesky_factor_corr[M_1] L_1;  // cholesky factor of correlation matrix
17 }
18 transformed parameters {
19   matrix[N_1, M_1] r_1;      // actual group-level effects
20   // using vectors speeds up indexing in loops
21   vector[N_1] r_1_1;
22   vector[N_1] r_1_2;
23   // compute actual group-level effects
24   r_1 = as_matrix(kronecker(Lcov_1, diag_pre_multiply(sd_1, L_1)) *
25                     to_vector(z_1), N_1, M_1);
26   r_1_1 = r_1[, 1];
27   r_1_2 = r_1[, 2];
28 }
29 model {
30   // initialize linear predictor term
31   vector[N] mu = Intercept + rep_vector(0, N);
32   for (n in 1:N) {
33     // add more terms to the linear predictor
34     mu[n] += r_1_1[J_1[n]] * Z_1_1[n] + r_1_2[J_1[n]] * Z_1_2[n];
35   }
36 }
```

Code 3.5: Parts of Stan code for GLMM written by `brms` for a model of the form `y` $\sim$ `(x1|factor)`. The object names correspond to the following variables: `Lcov_1` $= L_{A_1}$, `sd_1` $= \boldsymbol{\sigma}_1$, `L_1` $= C_1$, `z_1` $= u_1$. For efficiency reasons, relating the random factor to the observation is not done with a Z matrix but instead with indicator vector `J_1` and predictor vectors `Z_1_1` and `Z_1_2`. Line 24 and 25 correspond to the transformation of the random effects as it was shown in Equation (3.1). In order to allow for predefining the variance components, line 14 and 16 need to be moved to the data sector (line 1–10).

Chapter 4

Benchmarking

In this chapter we compare different implementations and their options. First, we focus on using the improved implementations outlined in Chapter 3 and test which functions and argument settings perform best for a simulated data set with respect to computation time and correct ranking of the animals. In a second stage, we compare the improved implementations with alternatives which might be considered for this estimation problem.

4.1 Simulation and Estimation

We simulated a variety of comparably small animal models. Two pedigrees which are contained in the package `pedigreemm` were used. The smaller (`pedSires`) was a 3-generation pedigree containing a total of 352 animals. The bigger (`pedCowsR`) contained five generations and a total of 6'547 Holstein cattle. A random breeding value was sampled for each animal in the pedigree. To this end, we first sampled independent transformed breeding values ($\boldsymbol{B}^*$) from a normal distribution and transformed them back to the dependent scale using the left Cholesky factor of the additive numerator relationship matrix (Equation 2.14). Variance components and in some models additional predictors were specified and finally the response variables were simulated according to LMMs, GLMMs and threshold models.

All models were fitted on a CPU with a 3.5GHz Intel© Xeon© E5-1650 v3 processor. For Bayesian estimation, we used four independent Markov chains each running on a separate core.

4.2 Specific Evaluations

4.2.1 Consider Correlation between Animals

The first goal was to compare estimation methods which take into account the complex correlation structure between animals against estimation methods which falsely assume independence between animals. To this end, simple LMMs were simulated containing only the random animal effects and the residual error. Three data sets were created for each pedigree containing different number of observations per animal. The signal-to-noise ratio was kept constant except for the small pedigree with three observations per animal where it was varied over several orders of magnitude. The simulated data sets were fitted with

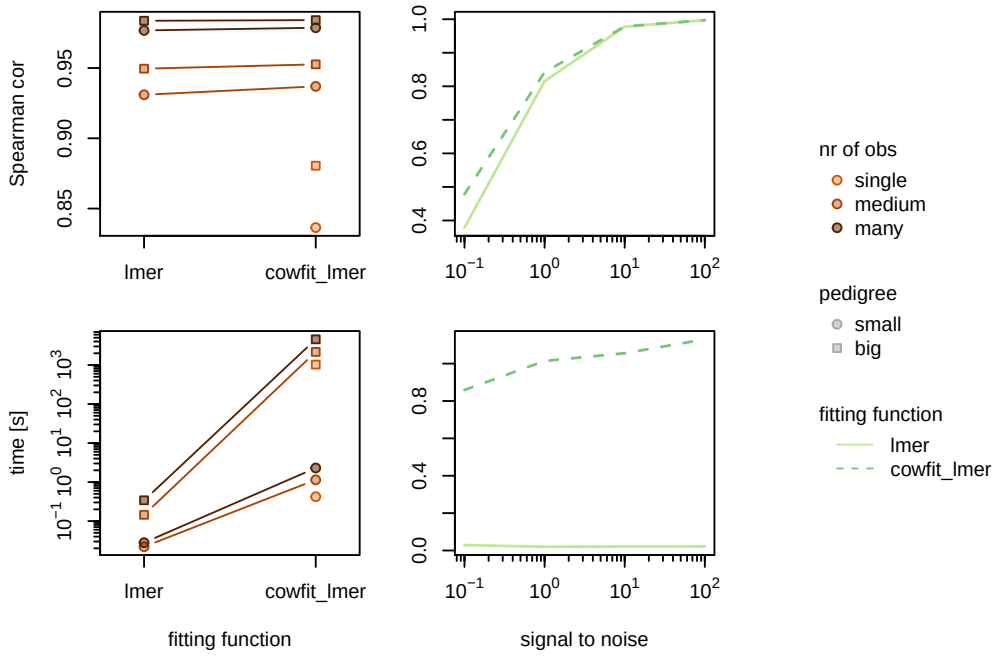


Figure 4.1: Effect of taking the pedigree into account on Spearman correlation and computation time. Simulated data sets of different sizes and with varying signal-to-noise ratio were fitted with `lmer()` and `cowfit_lmer()`. Note that `lmer()` cannot fit models with single observation per animal.

the functions `lmer()` and `cowfit_lmer()`.

Unsurprisingly, the ranking of the animals consistently improved when the correlation structure was taken into account (Figure 4.1). The effect was stronger at low signal-to-noise ratios where the additional information from related animals were able to clearly improve the estimation of breeding values. It is expected to be even stronger if we observe higher correlations between animals in a pedigree for example due to inbreeding. Still, the overall effect was small in relation to the massive increase in computation time. Therefore, it might be justified to assume independence between animals at least for model selection and account for the correlation only in the evaluation of the final model.

4.2.2 Likelihood vs Bayesian Framework

We compared the performance of GLMM estimation in the likelihood and the Bayesian framework. The linear predictor consisted of only the random breeding value and was transformed to a probability via the inverse logit function. A varying number of observations were drawn from a Bernoulli distribution for each animal. The signal-to-noise ratio was again only varied in the small pedigree with three observations per animal. All data sets were fitted with `cowfit_glmer()` and `cowfit_brm()`.

The results indicate similar ranking performance of the likelihood and the Bayesian implementation (Figure 4.2). The likelihood implementation had slightly better Spearman correlations for data sets with single observations whereas for data sets with multiple observations it was vice versa. A major difference between the two implementations arises

with regard to the computation time. The likelihood implementation was much faster for small data sets. The main reason was the compilation of the Stan script which took about half a minute at each execution of `cowfit_brm()`, independent of the data set size. For large data sets the computation time became more equal and for very large data sets the Bayesian implementation might actually be faster (see Chapter 5).

4.2.3 Given Variance Components

Allowing to fit models with given variance components was one of the major changes from existing implementations. A small simulation was performed to verify the expected decrease in computation time. The same data sets as in Chapter 4.2.1 and 4.2.2 were used and now estimated with given variance components. The fitting was done with functions `cowfit_lmer()`, `cowfit_glmer()` and `cowfit_brm()`.

The results showed a considerable drop in computation time by providing the variance components (Figure 4.3). The computation time was 4-8 times lower in the likelihood setting. In the Bayesian setting the drop was not as pronounced, especially with larger models. Spearman correlations very slightly increased in about a quarter of the fitted models and stayed constant in the remaining ones.

4.2.4 Sensitivity on Variance Components

In the previous chapter we assumed the given variance components to exactly match the "true" underlying ones. These would likely be unknown in a real world setting and the variance components of previous evaluations or from literature might be slightly wrong for the specific data set. For this reason, we evaluated the sensitivity of the estimated animal ranking on wrong variance component specification. Three LMMs with increasingly complex model structure were considered. The first only included the random animal effect, in contrast to the second and third model which also included a second random effect. The levels of the second term were either largely independent of the animal levels or strongly correlated. The models were fitted by `cowfit_lmer` with given variance components. The true animal variance component was scaled over several orders of magnitude. Additionally the model with correlated random terms was fitted assuming wrong residual error variance. The goal was to check whether the argument `exact_var_comp` improved the estimation of $\boldsymbol{\theta}$ and therefore the ranking of the animals.

Overall, the effect of the wrong variance component specification strongly depended on the degree of dependence between the two random effects (Figure 4.4). Only the ranking of the model with strongly correlated levels was negatively affected by slight misspecifications of the variance components. This observation can be easily explained by an example: let's assume we strongly underestimate the animal variance component and there is a largely correlated predictor (e.g. herd). Most of the variance which should be explained by animal will be assigned to herd and therefore the ranking of the animals will become worse. On the other hand, if there is no correlated predictor, the variance which should be explained by animal cannot be assigned to any other predictor. The ranking of the random animal effects will stay conserved, even if we underestimated the variance component by orders of magnitude. However, the absolute values of the random animal effects as well as estimates of heritability which are calculated from the variance components will be greatly underestimated. Therefore, special care in variance component specification has to be taken in models with strong correlation between the animal or sire levels and at least one additional predictor. All models with the same predictors had similar computation time independent

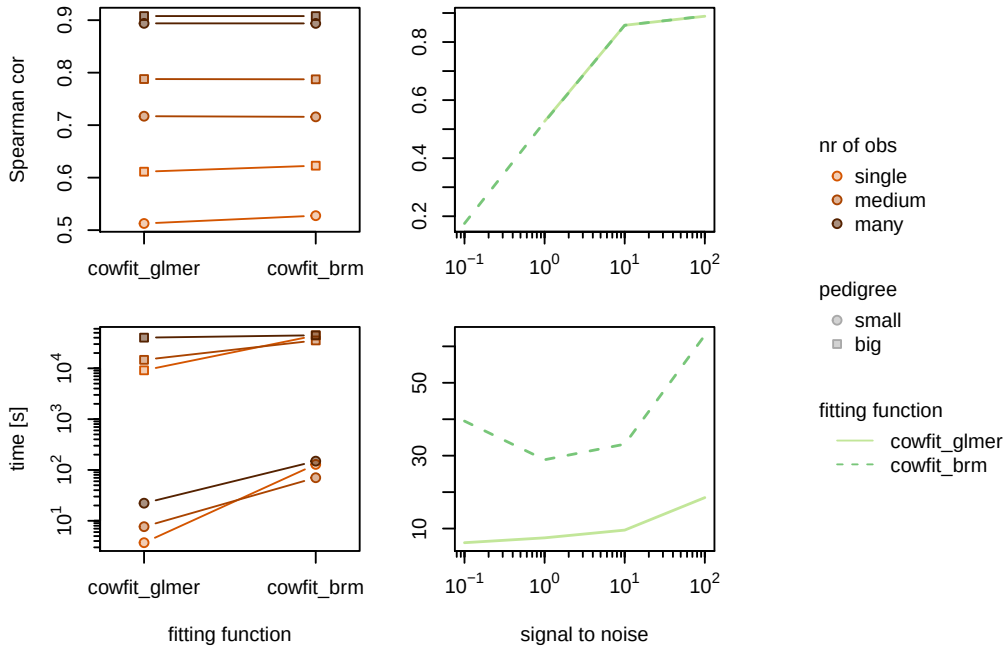


Figure 4.2: Comparison of likelihood and Bayesian implementation with respect to Spearman correlation and computation time.

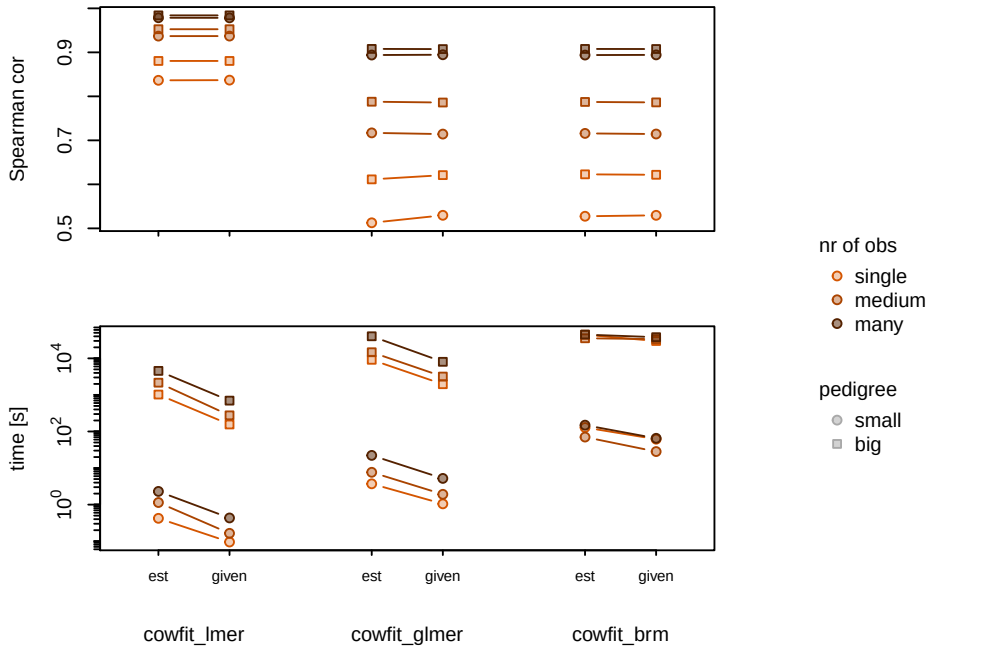


Figure 4.3: Effect of given variance components on Spearman correlation and computation time. The variance components were either estimated from the data (est) or given to the fitting function (given).

of the variance components which were used. The argument `exact_var_comp` clearly improved the ranking of animals especially if the residual error was largely underestimated. However, it came at large computational cost, as was expected due to the numerical optimization with respect to $\boldsymbol{\theta}$.

4.2.5 Numerical Optimization for Fixed Effects

`lme4::glmer()` allows to suppress the numerical optimization with respect to the fixed effect parameters by setting the argument `nAGQ = 0`. This greatly speeds up the estimation but also leads to less accurate estimations of the fixed effect parameters. In animal breeding we are not so much interested in the fixed effect parameters but mainly in the prediction of the random effects. Also, computation time is a big issue due to the large data sets. For this reason, the `nAGQ` argument was also transferred into the `cowfit_glmer()` estimation procedure (see Figure 3.2). We tested the effect of the argument `nAGQ` on the estimation performance of simulated GLMMs containing a random animal term and a fixed factor with five levels. Only simulated data sets based on the small pedigree were considered but with varying numbers of observations per animal.

Avoiding the numerical optimization with respect to the fixed effect parameters decreased the computation time considerably (Figure 4.5). It is a great option in cases where there is no correlation between fixed and random predictors. The resulting estimate of $\boldsymbol{\beta}$ was clearly biased and not as accurate as it would have been with numerical optimization.

4.2.6 Evaluate binary data with LMMs

As mentioned in Chapter 1, non-normal response variables are often modeled with LMMs despite several disadvantages of this practice. Our goal was to evaluate the effect on the ranking of the animals, the breeding value estimates and the breeding value standard deviation estimates. To this end, we simulated 100 GLMMs with Bernoulli distribution, logit link and only the random animal effect in the linear predictor. For each animal 20 observations were sampled. We assumed independent animals and fitted the models with `lmer()`, `glmer()` and `brm()`. The fitted model was used to predict breeding values which were compared against the true ones. In LMMs, the true breeding values were first transformed by the inverse logit function before comparing them with the predicted breeding values such that they would be on the same absolute scale. For each breeding value we calculated the *standardized estimation bias* and the *standardized sd bias* which were defined as

$$\text{standardized est bias} = \frac{\hat{b}_i - b_i}{\text{sd}(\hat{b}_i)}$$
$$\text{standardized sd bias} = \frac{\hat{\text{sd}}(\hat{b}_i) - \text{sd}(\hat{b}_i)}{\text{sd}(\hat{b}_i)}.$$

The results show a comparable ranking over all three estimation methods (Figure 4.6), indicating that LMMs might actually be a simple alternative for some genetic evaluations which primarily target the ranking of the animals. However, the absolute value of the estimated breeding values were strongly biased in the LMM evaluation. Only the Bayesian implementation obtained good estimates of the breeding value accuracy. LMM estimations of breeding value accuracy were strongly biased and not trustworthy.

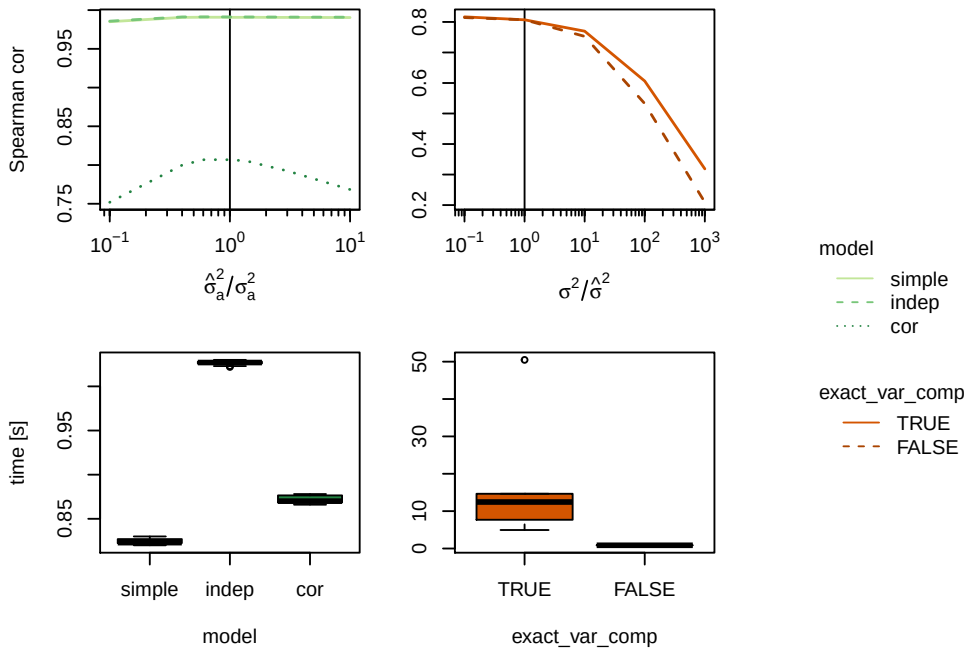


Figure 4.4: Sensitivity of Spearman correlation and computation time on wrong variance component specification. Left: three models of different complexity and the effect of wrong animal variance component specification. The simple model only includes one random term. The independent model (indep) contains two random terms which are largely independent. The correlated model (cor) has two random terms which are strongly correlated. Right: effect of wrong residual error variance specification on the correlated model with different settings for argument `exact_var_comp`.

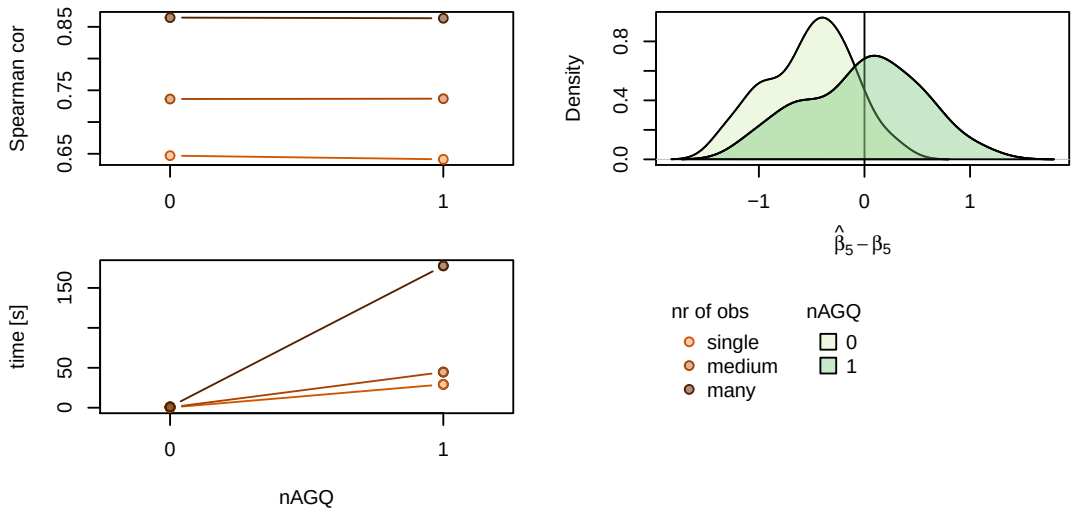


Figure 4.5: Avoiding numerical optimization with respect to $\boldsymbol{\beta}$. The right plots show the effect on Spearman correlation and computation time. The density plot shows the distribution of the fifth element in the $\boldsymbol{\beta}$ vector.

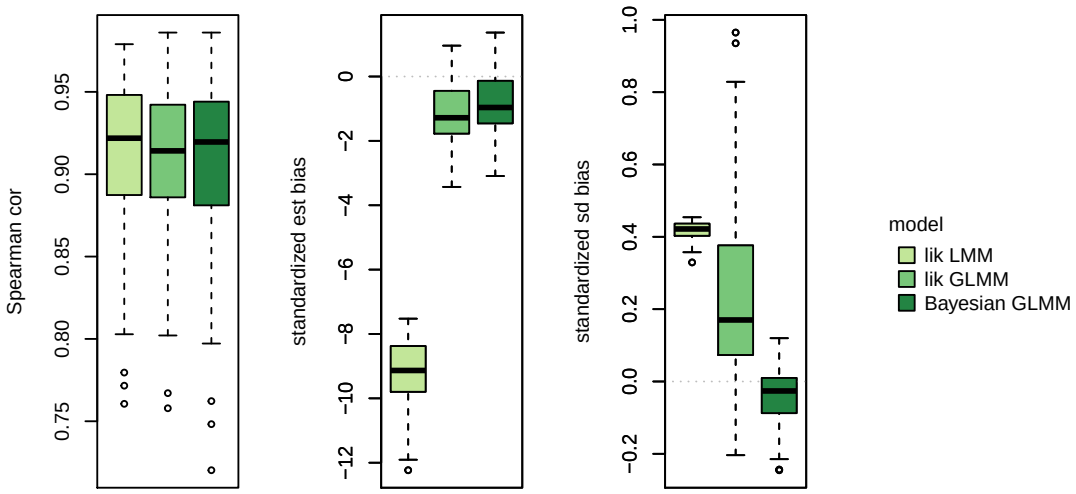


Figure 4.6: Evaluating binary data with LMMs. Binary data which were simulated based on a GLMM were evaluated as LMM and GLMM using likelihood (lik) or Bayesian estimation methods.

4.2.7 Evaluate ordinal data with LMMs

In the same way as in Chapter 4.2.6 we also evaluated the effect of falsely using metric models for ordinal data. A threshold model with one fixed effect term and a random animal effect was set up and 100 data sets were simulated from it. The threshold values were randomly drawn from a uniform distribution in each simulation cycle. The data sets were analyzed with `cowfit_lmer()` and `cowfit_brm()` by treating the response variable either as metric using an LMM or as ordinal using a threshold model.

The threshold model improved the animal ranking in 97% of all simulated data sets. However, on average the Spearman correlation only increased by 0.02 from 0.67 to 0.69. The better ranking came also at greater computational cost. Within the likelihood framework, the LMMs were fitted in less then a second due to the small size of the data set. The Bayesian estimation of the LMMs took on average 37 seconds but most of it can be attributed to the compilation of the Stan script. Estimating the threshold model increased the computation time on average by 60% compared to the Bayesian LMM.

4.2.8 Alternative Implementations

Our implementations are built on the packages `pedigreemm`, which uses maximum likelihood or restricted maximum likelihood estimation, as well as `brms`, which is based on Hamiltonian Monte Carlo. A frequently used alternative for GLMM estimation which we did not consider are approximations like the penalized quasi-likelihood approximation (PQL) and the integrated nested Laplace approximation (INLA). Packages like `spaMM` (Rousset, Ferdy, and Courtiol 2020) and `AnimalINLA` (Holand et al. 2013) implement these methods and promise a fast and accurate prediction of random effects. We also wanted to test alternative MCMC implementations which use Metropolis-Hastings updates and Gibbs sampling instead of Hamiltonian Monte Carlo. A widely used implementation is the package `MCMCglmm` (J. Hadfield 2019).

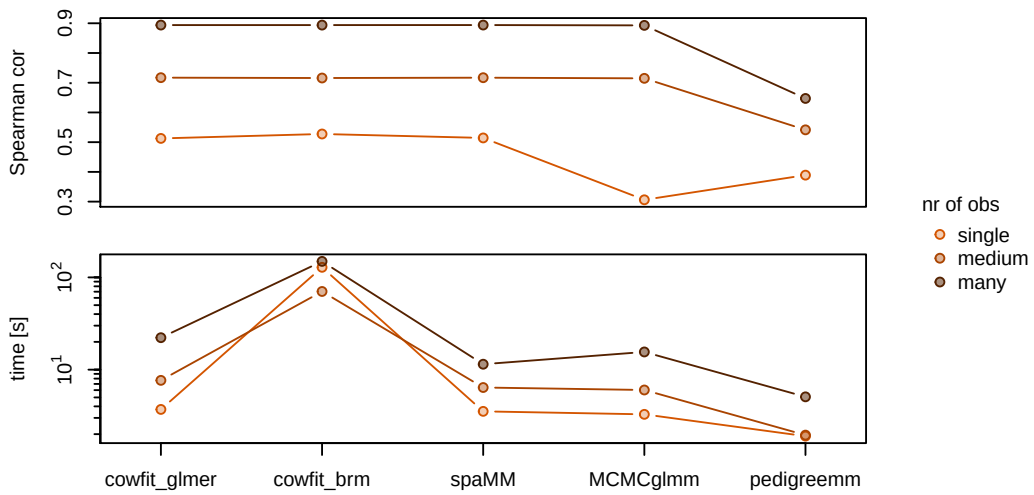


Figure 4.7: Comparison of GLMM implementations for genetic evaluations.

The simulated data of Chapter 4.2.2 based on the small pedigree were fitted with packages `pedigreemm`, `spaMM` using PQL and `MCMCglmm`. Especially `spaMM` showed promising results for the relatively small data sets (Figure 4.7). Trials on larger data sets did not converge in reasonable time. `MCMCglmm` was a much faster Bayesian alternative but the predicted ranking was not as precise in the case of single observations. The original `pedigreemm` implementation had much lower correlations in all models due to the bug explained in Chapter 3.1.1.

Chapter 5

Swiss Cattle Data

5.1 Multiple Birth

5.1.1 Theory

Multiple birth is a relatively rare phenomenon in cattle and its frequency strongly depends on the cattle breed. The frequency is usually below 1% in beef cattle whereas in dairy herds we observe a higher frequency of 4–5% (Komisarek and Dorynek 2002). Mechanisms behind multiple birth include multiple ovulation and spontaneous single embryo division. The first mechanism comprises more than 90% of all double births in cattle (Cady and Van Vleck 1978). Multiple ovulation can be considered as a trait of the dam and therefore, multiple birth can be largely associated with genetic predisposition of the dam and not with the genetic predisposition of the zygote (Johansson, Lindhé, and Pirchner 1974). The trait is strongly affected by age and parity of the dam, with a higher rate of multiple birth in older dams and dams at higher parity. It might also be affected by seasonality with more multiple births during spring and autumn (Karlsen et al. 2000, Gregory, Echternkamp, et al. 1990).

Multiple birth in combination with intensive management usually has a positive effect on productivity in beef cattle. In dairy cattle it is the complete opposite and animals with low risk for multiple birth are preferred. The negative effects of multiple births include freemartinism, dystocia, premature calving and retained placenta. Freemartinism describes the phenomenon that 82–92% of heifers in a mixed sex multiple birth event are not fertile (Zhang et al. 1994, Komisarek and Dorynek 2002). The cardiovascular system of cattle siblings usually becomes connected at an early stage of pregnancy. The resulting exchange of hormones between siblings of different sex leads to infertile females and reduced fertility in males. Heifers from mixed sex multiple birth events can therefore not be used for milk production. Dystocia refers to difficult births usually due to an abnormal presentation of one or both foetuses at parturition. Together with premature calving and retained placenta it poses a health risk to calves and dam and increases calf mortality. Breeding on lower multiple birth rate in dairy cattle is impaired by the unfavorable correlation between twin birth frequency and milk production (Komisarek and Dorynek 2002). Additional difficulties include the low heritability, sex-limited trait expression and the long generation interval prolonging the time span until descendants show phenotypic observations for progeny testing. Nevertheless, there are successful examples of increasing the multiple birth rate over several selection cycles in beef cattle (Morris and Wheeler

2002, Gregory, Bennett, et al. 1997).

5.1.2 Exploratory Data Analysis

The multiple birth data set of Qualitas contains 19'104 birth events with 705 twins and one triplet (Figure 5.1). It is a subsample of a larger data set which was collected by the Swiss Braunvieh breeding association and includes Braunvieh cattle from Switzerland between 1990–2019. Possibly important covariates were also collected namely dam id, parity of the dam, birth season, number of inseminations to achieve pregnancy and sexing method which describes the procedure used to separate seamen by sex. The variable parity is right censored at the 5th parity. The data set contains 8'637 dams which are direct descendants from 1'534 sires. A detailed pedigree including all dams and their sires is available. The average sire is associated with 12.5 birth events, whereby 571 is the largest numbers of birth events associated with one sire. 78% of all sires are never associated with a multiple birth event (Figure 5.3).

Possible dependencies between the covariates and the multiple birth trait were visually investigated (Figure 5.2). Only parity was found to be strongly associated with multiple birth. Additionally, there is a strong dependency between parity and the number of inseminations (Figure 5.3), meaning that cows at higher parity need significantly more inseminations to become pregnant.

5.1.3 Modeling

It is a difficult task to find a good fitting model at reasonable computational cost, given the size of data set and pedigree. We applied some heuristic methods to save computation time. First, we started fitting a sire model which contains a much lower number of random effects compared to the animal model. The model selection was entirely based on the wrong assumption of independent sires, leading to a sparse Z matrix and faster estimation of the model parameters. A variety of hierarchical models were fitted by adding and removing covariates. The models were compared based on a partial F-test. The finally selected model was fitted by additionally considering the correlation structure in the random effects due to the relationship between the sires.

The best fitting sire models were also fitted as animal models and compared based on a partial F-test in order to confirm that the same model would be selected in the animal model framework. Again, we tried to estimate the selected model including the correlation structure of the random effects. We failed to directly fit the animal model in reasonable time and decided to separately estimate the variance components with two different strategies

1. Directly adopt the variance components of the fitted sire model to use them in the animal models. Each sire inherits only half of its additive genetic effect to the offspring. Therefore, the variance component of the animal model σ_a^2 can be estimated from the sire model using

   $$\sigma_{\mathrm{a}}^2 = 4\sigma_{\mathrm{s}}^2$$

   (see for example Kriese, Bertrand, and Benyshek 1991).

2. Estimate the variance components from a smaller subset of all data. The data set was reduced by removing observations which lead to the largest reduction in the number of random factor levels.

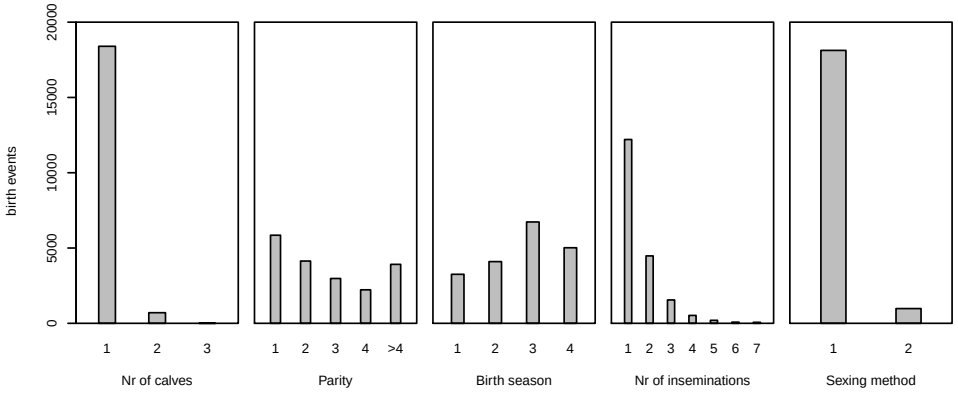


Figure 5.1: The multiple birth data set. There are a total of 19'104 birth events of which 3.7% are multiple births. Each birth season contains 3 months where birth season 1 corresponds to Dec–Feb, season 2 corresponds to Mar–May and so on. Semen was separated according to the Beltsville sperm sexing technology (method 1) or according to the Bovitel^{TM} procedure (method 2).

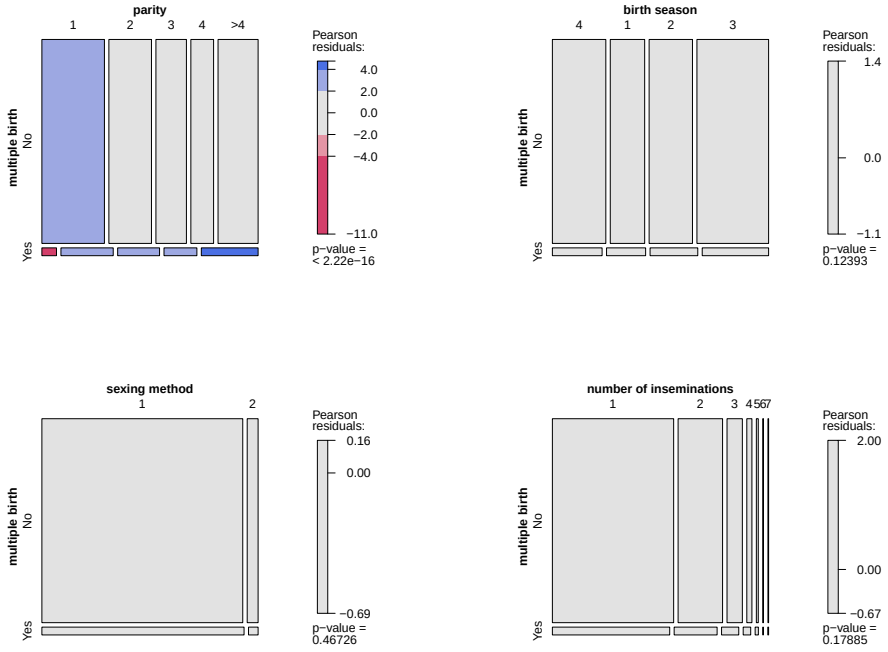


Figure 5.2: Mosaic plot of multiple birth and possibly relevant covariates. The area of each tile is proportional to the number of observations within that category. The p-value results from a Pearson chi-squared test with the null hypothesis (H_0) assuming independence between multiple birth and the covariate. A large positive Pearson residual (blue) indicate a larger number of observations then would be expected under H_0. We observe a strong deviation from the expected frequencies under H_0 between multiple birth and parity.

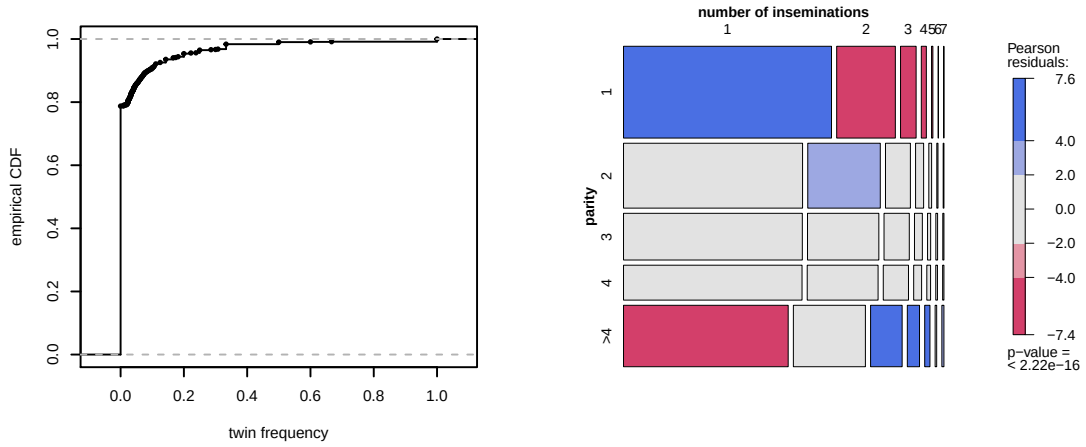


Figure 5.3: Left: empirical cumulative distribution function of twin frequency for the sire population. 78% of all sires have a twin frequency of zero and 97% of all sires are associated with multiple birth in less than 1/3 of all associated birth events. Right: mosaic plot of parity and number of inseminations, which are strongly related.

Subsequently, the full animal model was fitted using the separately estimated variance components.

5.1.4 Results

The model selection process resulted in the following model

`mult_birth` ∼ `as.factor(parity) + (1|hy) + (1|sire)`.

Parameter values of covariates birth season, sexing method and number of insemination were not significantly different from zero when added separately to the model. Including parity as a factor accounted for the non-linear increase in multiple birth with higher parity and was strongly significant. The herd year effect `hy` was added as random effect because it has many levels and can be considered as an additional error term.

The sire model with unknown variance components took roughly 9:30 h to fit with the likelihood estimation and 34 minutes using the Bayesian estimation running on four separate cores. The parameter estimates associated with parity were almost identical between the two estimation methods (Table 5.1). An average cow in the first lactation had a fitted probability of 0.007 for multiple birth. The probability increased to 0.038 in the second lactation and was relatively constant thereafter.

For heritability estimation we were using the formula for logistic regression described in Vazquez, Gianola, et al. (2009)

$$h^2 = \frac{4\sigma_s^2}{\sigma_s^2 + \frac{\pi^2}{3}}$$

which resulted in a heritability between 0.13–0.14. This estimate was clearly above the estimates found in other studies which ranged between 0.01–0.10 (Cady and Van Vleck 1978, Gregory, Bennett, et al. 1997, Karlsen et al. 2000, Maijala and Osva 1990, Ron,

Table 5.1: Estimates and standard error of multiple birth model parameters. The sire model was estimated in the likelihood (lik) and Bayesian framework. The animal model was estimated either on a subset of the data or on the full data set using variance components from the sire model or from the subset model. For each model the parameter estimates (est) and the standard error (se) is reported if available.

parameter	sire				animal					
	lik		Bayes		subset		σ_a sire		σ_a subset	
	est	se	est	se	est	se	est	se	est	se
σ_a	-	-	-	-	0.65	-	-	-	-	-
σ_s	0.33	-	0.35	0.08	-	-	-	-	-	-
σ_{hy}	0.39	-	0.38	0.09	0.39	-	-	-	-	-
intercept	-4.88	0.16	-4.89	0.17	-4.52	0.56	-5.11	0.18	-5.09	0.19
parity 2	1.67	0.15	1.68	0.16	0.93	0.59	1.70	0.16	1.69	0.16
parity 3	1.80	0.16	1.80	0.17	0.65	0.60	1.82	0.16	1.82	0.16
parity 4	1.85	0.17	1.86	0.17	0.75	0.59	1.88	0.17	1.88	0.17
parity 5	1.84	0.15	1.84	0.16	1.01	0.54	1.87	0.16	1.87	0.16

Ezra, and Weller 1990, Syrstad 1984, Van Vleck and Gregory 1996, Ghavi Hossein-Zadeh et al. 2009).

The breeding values of the sires ranged between -0.51–0.54 and the 95% equal-taild credible intervals were comparably large (Figure 5.4). Descendants of the upper 10% of sires had an odds ratio of multiple birth which was 1.58 times higher than the odds ratio of the lower 10%. For example, in the second parity this corresponds to a probability of 0.05 in the upper 10% versus a 0.03 in the lower 10%. The effect was relatively small as we would expect from the low heritability.

Transforming the sire model variance component to the animal model variance component resulted in a variance estimate of $\sigma_a^2 = 0.44$. The variance component estimate from the subset was slightly lower ($\sigma_a^2 = 0.42$). Fitting the full model with given variance components took about 15 h with Bayesian estimation using four cores. The variance component estimates from the two different estimation strategies were so close such that there was not a notable improvement of using one over the other. The ranking of the estimated breeding values from both models were almost identical (Spearman correlation > 0.99).

5.2 Calf Mortality

5.2.1 Theory

The calf mortality rate of European countries range between 4–7% (including stillbirths, Svensson, Linder, and Olsson 2006). A slight increase in calf mortality was noted in resent years (Santman-Berends, Buddiger, et al. 2014). However, Santman-Berends, Schukken, and van Schaik (2019) found that small differences in the definition of the trait have a large effect on possible trends. There is no trend observable over the last few years in the Swiss Braunvieh population (Berweger 2020). Still, due to the economic importance of the trait, it would be desirable to include it in the overall selection criterion (Østerås et al.

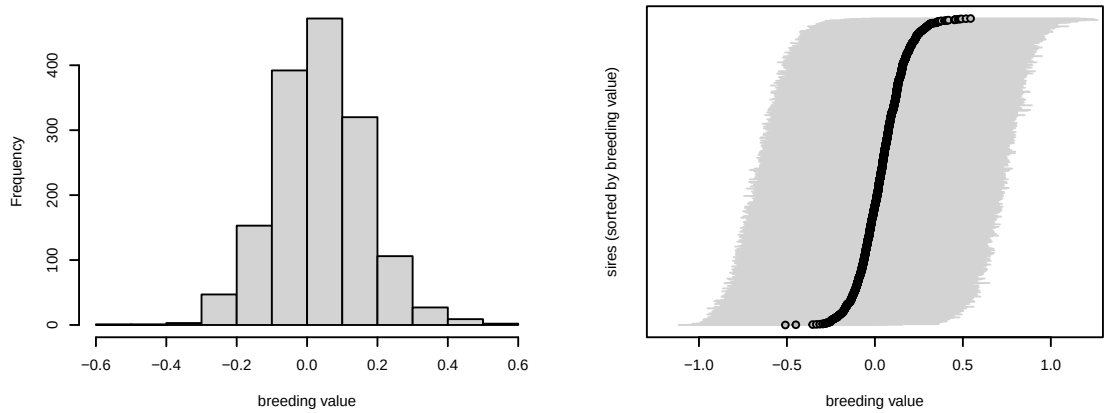


Figure 5.4: Distribution of multiple birth breeding values. The left plot shows a histogram of the sire breeding values. The right plot includes the 95% equal-taild credible intervals (gray line) to each sire breeding value.

2007).

Literature mentions a few important factors influencing calf mortality rate like sex, herd size, birth season, multiple birth and housing (Del Río et al. 2007, Gulliksen et al. 2009). Male calves, calves in large herds and calves born during the cold season are all associated with a higher mortality rate. Twins and triplets are at risk because of birth complications and the generally lower birth weight. Calves housed in a group pen rather than individually were also found to have an increased mortality rate probably because of the exposure to higher levels of infectious agents. In the first 30 days of life, calves are especially sensitive to respiratory diseases and diarrhea. These diseases play a minor role at later stages. Therefore, it can make sense to separate calf mortality into different traits according to the rearing period (Carlen et al. 2016).

The current Swiss Braunvieh breeding program already includes calf survival during first hours and cow longevity. However, survival rate in the period between the already considered traits have not been included so far. Calf mortality is mainly influenced by management and has therefore a low heritability. Still, Carlen et al. (2016) showed that it is possible to achieve breeding progress in the trait. The study observed a general increased progress in health traits but also a slight reduction in yield progress as a result of selection on lower calf mortality rate.

5.2.2 Exploratory Data Analysis

Qualitas has calf mortality data of the Swiss Braunvieh breed from the time period between 1990–2019. The subsampled data set includes 103'066 calves. Mortality was divided into two age groups, namely 3–30 days and 31–458 days. Mortality rate was 2.8% and 3.5% in the first and second age group, respectively. We only focus on the first age group in this analysis. Available covariates are calf id, parity, sex, herd and year. Male calves and calves of the first parity had a higher mortality rate (Figure 5.5). The calf id links each calf to a pedigree including 181'198 related animals.

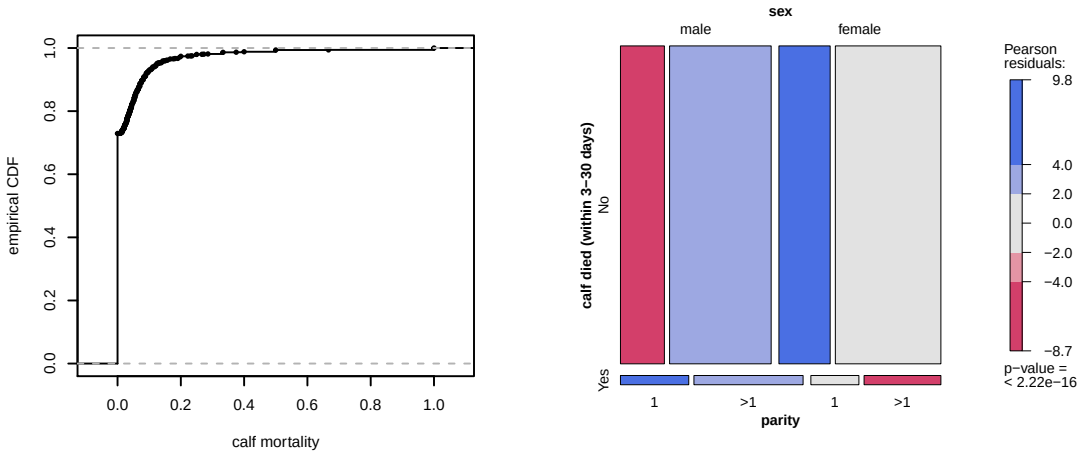


Figure 5.5: Left: empirical cumulative distribution function of calf mortality rate for the sire population. 73% of all sires are never associated with calf mortality and 92% in less than 10% of all offspring. Right: mosaic plot of calf mortality, sex and parity. Male calves have a high risk especially in the first parity.

5.2.3 Modeling

Model selection and variance component estimation was based on the same approach as described in Chapter 5.1.3.

5.2.4 Results

The final sire model included fixed predictors sex and parity as well as the random factors sire and a combination of herd and year (rhby)

`P1 ~ sex + parity + (1|rhby) + (1|sire)`.

Fitting the sire model with unknown variance components took 2:40 h in the Bayesian framework. The same model in the likelihood framework did not reach convergence after more than 27 h and was stopped. Parity and sex were strongly significant as already expected from the exploratory data analysis. The fitted odds ratio for calf mortality was increased for male calves and calves of the first parity by a factor of 1.54 and 1.38, respectively. Male calves of the first parity had a fitted mortality of 0.04 compared to a mortality of 0.02 in female calves of higher parities.

The estimated variance component for the random sire and herd year effect was 0.09 and 0.66, respectively. The resulting heritability was 0.10. Again, this was higher than the 1–3% which are reported in literature (Carlen et al. 2016). All breeding values are shown in Figure 5.6. The breeding value of the best sire (id 46'332) was -0.37 which corresponds to a fitted mortality of only 0.013 in female calves of second or higher parity. The observed mortality rate of the 288 calves which are associated to this sire was 0.014, which was very low compared to the overall mortality rate.

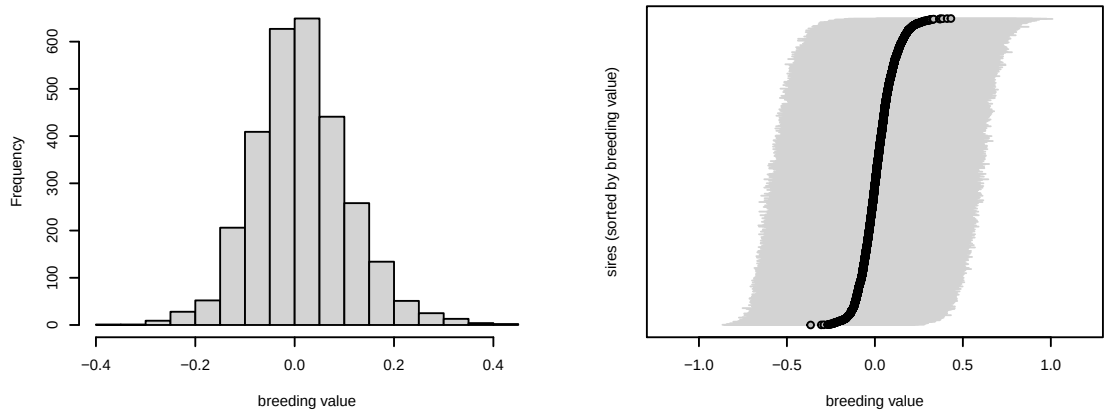


Figure 5.6: Distribution of calf mortality breeding values. The left plot shows a histogram of the sire breeding values. The right plot includes the 95% equal-taild credible intervals (gray line) to each sire breeding value.

5.3 Carcass Conformation

5.3.1 Theory

The value of an animal for slaughter is largely determined by carcass traits. Important carcass traits include carcass weight (which is not the same as live weight), carcass fatness and carcass conformation. The last one describes the proportion between valuable parts and less valuable parts of the carcass. It is difficult to measure carcass conformation on a metric scale which is why the trait is often visually scored. Many countries including Switzerland have their own scoring system which makes it difficult to compare results among countries.

Carcass conformation is a highly heritable trait and therefore an interesting breeding target. Heritability estimates found in literature span over a wide range somewhere between 0.1–0.44 (Hickey et al. 2007, Varona, Moreno, and Altarriba 2009, Kause et al. 2015). In practice, the estimation of breeding values is often achieved by assuming a metric scale and using LMMs. Estimation with threshold models would be favorable as they account for the ordinal nature of the scores (Gianola and Foulley 1983). Several studies obtained a higher expected selection response with the threshold model, especially if they allowed for slaughterhouse specific thresholds (Varona, Moreno, and Altarriba 2009, Tarrés et al. 2011). Possibly important predictors include sex, parity, age, season, year, herd, multiple birth and slaughterhouse. Age might have to be squared before including it in the linear predictor in order to account for its nonlinear effect. The model parameters have been shown to be highly breed specific (Kause et al. 2015).

Selection on carcass conformation also affects other carcass traits. Generally, carcass conformation is positively correlated with carcass weight. Correlation with carcass fatness is again highly breed specific with for example a negative correlation coefficient in breed Limousin and a positive correlation coefficient in breed Hereford (Kause et al. 2015).

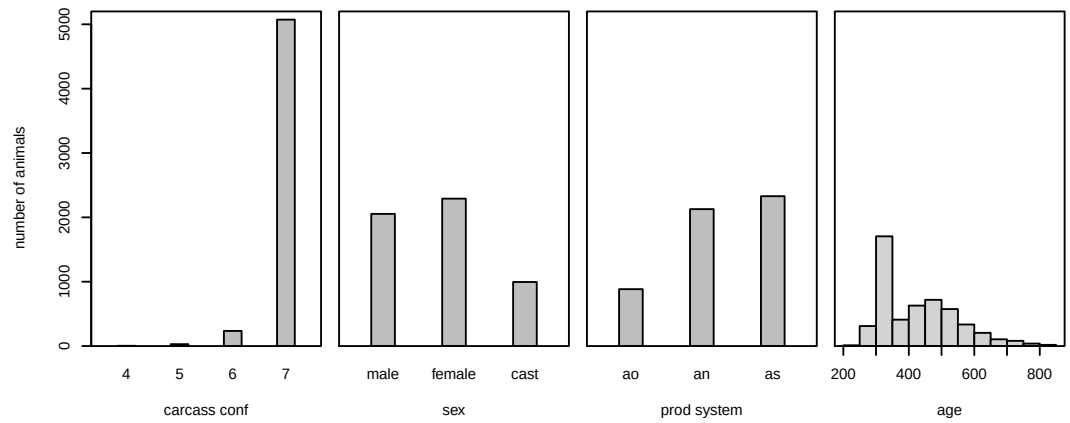


Figure 5.7: Carcass conformation data set. 95% of all animals have the highest conformation score. The majority are male animals where cast stands for castrated male animals. There are three different production system with abbriviation "ao" standing for conventional beef cattle production, "an" for suckler cow husbandry which are culled at an age of 10 months and "as" for suckler cow husbandry which are culled arround an age of 13–20 months. The last panel shows a histogram of the age distribution at culling. The peak arround 300 days can be mostely attributed to animals of the "an" production system.

5.3.2 Exploratory Data Analysis

Qualitas has a data set containing 5'341 animals of beef cattle breed Limousin which are born between 2012–2019. For each animal the carcass conformation score was determined after slaughter. The overwhelming majority of animals achieved the highest carcass conformation score, which is typical for animals of breed Limousin (Figure 5.7, Kause et al. 2015). The data set also contains possible predictors including sex, age, season, year, herd, production system, slaughterhouse and id of the trained technician which performed the visual scoring. Each observation is linked to a pedigree containing a total of 13'903 animals. All phenotyped animals are direct descendants to 495 sires. The number of phenotyped offspring per sire ranges between 1–250 with an average of 10.8.

Pairwise analysis of the variables revealed a strong relationship between carcass conformation and the predictors sex and production system (Figure 5.8). Female animals and animals from the production system "an" were largely over-represented in the group of animals with a carcass conformation score below seven.

5.3.3 Modeling

Fitting separate thresholds for each slaughterhouse was unrealistic as the data set was relatively small and some slaughterhouses had only very few observations. For this reason, we included slaughterhouse as a random factor together with id of the technician, sire and a combination of herd and year. The full model additionally included sex, production system and a polynomial age term of second degree as fixed effects.

In order to get a more parsimonious model, we applied step-wise backward elimination using the function `lmerTest::step()` (Kuznetsova, Bruun Brockhoff, and Haubo Bojesen Christensen 2020). Elimination was based on p-values with threshold 0.1 for random

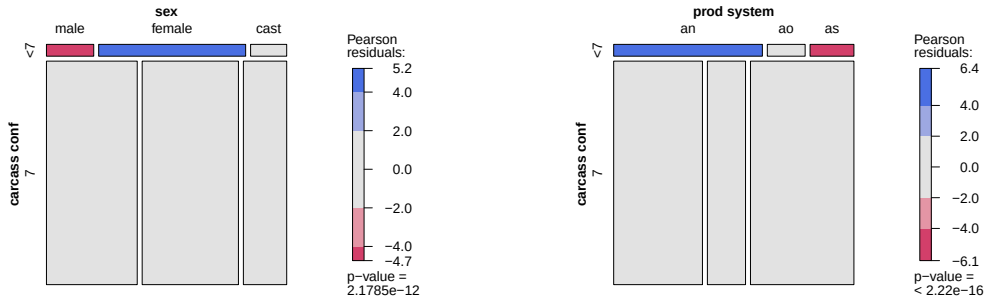


Figure 5.8: Mosaic plot of carcass conformation score and possibly relevant covariates. There is a strong dependency between carcass conformation and the predictors sex and production system.

effects and 0.05 for fixed effects and assuming independence between sires as well as a conditional normal distribution of the carcass confirmation scores. The reduced linear predictor was used in a threshold sire model, a threshold animal model and a sire GLMM using a binary response which indicated whether the carcass conformation score was seven or not. All three models accounted for the correlation structure given by the pedigree and were fitted with `cowfit_brm()`.

5.3.4 Results

Only the quadratic age term was dropped during the model selection procedure leading to the linear predictor

$$\texttt{cc} \sim \texttt{sex} + \texttt{prodSys} + (1|\texttt{slaughterhouse}) + (1|\texttt{idTech}) + (1|\texttt{herdYear}) + (1|\texttt{sire}).$$

Fitting the models took between 9–52 minutes. With default settings the Markov chain of the threshold models contained several divergent transitions. After manually adjusting the target average proposal acceptance probability, which corresponds to the `adapt_delta` parameter in `brms`, we obtained better results in the sire model with decent looking trace plots. The animal model still contained many divergent transitions and was therefore not trustworthy. As expected from the exploratory data analysis, female animals and animals from the production system "an" had the lowest fitted liability (Table 5.2). Using the variance component estimates, we calculated the heritability of the threshold sire model

Table 5.2: Estimates (est) and standard error (se) of carcass conformation threshold sire model parameters. The reference level is given by a male animal of production system "an".

parameter	est	se	parameter	est	se	parameter	est	se
$\sigma_{\text{slaughterhouse}}$	0.52	0.29	τ_4	-3.35	0.30	β_{female}	-0.38	0.08
$\sigma_{\text{herd:year}}$	0.16	0.07	τ_5	-2.62	0.26	β_{cast}	-0.08	0.11
σ_{sire}	0.36	0.06	τ_6	-1.64	0.26	β_{ao}	0.32	0.14
						β_{as}	0.52	0.12

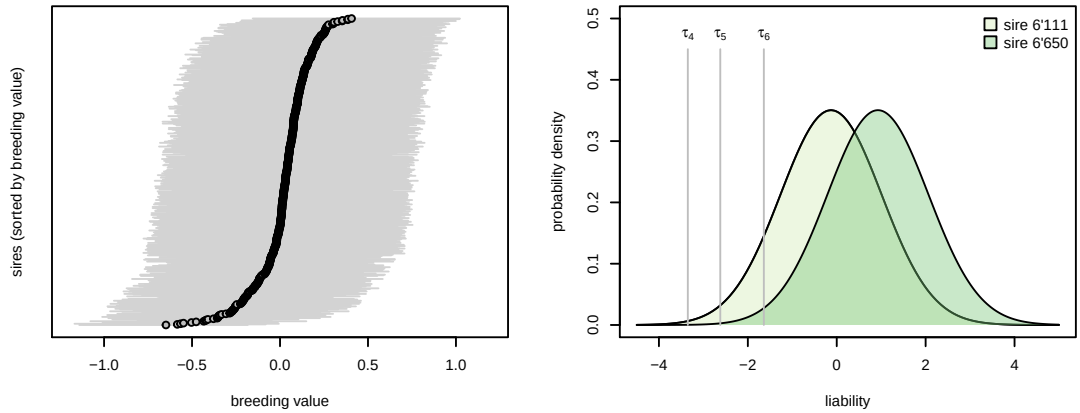


Figure 5.9: Carcass conformation breeding values. Left: sorted breeding values with 95% equal-taild credible intervals. Right: liability distribution of male offspring from sire 6'111 (lowest breeding value) and sire 6'650 (highest breeding value) assuming production system "as". Animals above threshold τ_6 have a carcass conformation score of seven. The probability of falling below the threshold was 0.09 and 0.01 for offspring of sire 6'111 and sire 6'650, respectively.

with the formula

$$h^2 = \frac{\sigma^2_{\text{sire}}}{\sigma^2_{\text{sire}} + \sigma^2_{\text{slaughterhouse}} + \sigma^2_{\text{herd:year}} + \sigma^2_e}$$

which resulted in a heritability estimate of 0.09. This rather low value might be attributed to the highly unbalanced data set with most observations having the highest carcass conformation score.

The sire breeding values ranged between -0.64–0.41 and their distribution was slightly left skewed (Figure 5.9). The low variance at the upper end of breeding values can again be explained by the fact that a large fraction of sires had no progenies with carcass conformation score below seven. Evaluating the same data set with the binary response GLMM resulted in almost the same ranking of the sires (Spearman correlation > 0.98) at a lower computational cost, showing that in this specific unbalanced data set the response variable was already close to binary. Analyzing the breeding values over time revealed no trend in the last few years. The top ten sires with the highest breeding value are summarized in Table 5.3. As it is common in animal breeding, the breeding values are reported together with the accuracy r^2 which was calculated with the formula

$$r^2_i = 1 - \frac{\hat{\text{se}}(\hat{b}_i)^2}{\hat{\sigma}^2_s}.$$

Table 5.3: Top sires for each trait (bv = breeding value, r^2 = accuracy)

rank	multiple birth			calf mortality			carcass conf		
	id	bv	r^2	id	bv	r^2	id	bv	r^2
1	6966	-0.51	0.33	46'332	-0.37	0.32	6'650	0.41	0.32
2	5177	-0.45	0.33	114'799	-0.30	0.23	6'801	0.39	0.35
3	9619	-0.35	0.40	105'513	-0.29	0.10	6'822	0.39	0.36
4	9607	-0.33	0.40	106'495	-0.27	0.30	3'459	0.36	0.26
5	1'2452	-0.31	0.00	101'271	-0.27	0.10	7'709	0.35	0.30
6	3467	-0.30	0.17	93'582	-0.26	0.17	7'400	0.33	0.31
7	1'4712	-0.30	0.03	74'795	-0.26	0.11	7'070	0.31	0.38
8	1'2719	-0.28	0.04	100'818	-0.26	0.07	9'637	0.31	0.30
9	1'3082	-0.28	0.09	53'858	-0.25	0.22	7'664	0.30	0.32
10	1'0979	-0.28	0.16	72'353	-0.25	0.32	9'235	0.30	0.22

Chapter 6

Discussion

In this thesis, we investigated the use of GLMMs and threshold models for the genetic evaluation of non-normal response variables. The models, which both are widely used in other areas than animal breeding, clearly showed some advantages over LMMs in the benchmarking and were successfully applied on real world cattle data.

6.1 Insights from Benchmarking

Only implementations which accounted for the complex relationship between animals made use of all available information in order to predict breeding values. This led to a general improvement of the animal ranking but also showed the enormous increase in computational cost associated with the correlation structure. The ranking of the animals was still relatively good under the simplified independent animal assumption and also led to similar absolute breeding values as were obtained under the model accounting for the correlation. This finding was used as a justification to assume an independent model for the computationally expensive model selection process. Despite having similar breeding values, the simplification might have undesired effects on model selection in more complex setting. Better model selection strategies which account for the correlation between animals should be further investigated. A promising option might be lasso regression which performs regularization and model selection simultaneously. Model selection becomes even more important for genomic models which contain a large amount of predictors (Haws et al. 2015). Lasso already showed good performance in the prediction of genomic breeding values (Ogutu, Schulz-Streeck, and Piepho 2012).

The comparison of likelihood and Bayesian estimation clearly revealed the advantages and disadvantages of both methods. The main advantage of likelihood estimation was the fast computation when applied to small data sets. Bayesian estimation was relatively slow for small data sets but comparably fast when applied to the much larger Qualitas data sets. An additional benefit of the Bayesian method was the improved estimation of the breeding value accuracy. The Bayesian method takes into account the uncertainty with respect to the variance component estimation, something which is, up to now, not possible in the likelihood framework (McCulloch and Searle 2000). Therefore, estimated credible intervals and standard errors of breeding values obtained from Bayesian methods are far more trustworthy than those obtained from likelihood estimation. Finally, the Bayesian implementations were highly versatile allowing to fit a wide variety of models with the same

function including GLMMs and threshold models. Overall, in our opinion the advantages of the Bayesian implementations clearly outweigh the additional computation time for smaller models. Especially for the estimation of GLMMs, where the likelihood estimation heavily relies on the iterative PIRLS algorithms, Bayesian estimation methods have proven to be not that much slower while offering all the above mentioned benefits. Still, likelihood implementations remained useful in the case where a large number of models needed to be fitted on a relatively small data set. This situation was given for example in model selection based on a subset of the data.

Most packages, especially those which were not primarily designed for genetic evaluations, had little to no support for prespecifying variance components. Our implementations show that it is relatively easy to allow for such a feature. In most cases, prespecified variance components led to a large decrease in computation time. The animal ranking was quite robust even if the variance component was wrongly estimated by a factor of 10, something which could usually be prevented by comparing variance component estimates with those found in literature. For this reason, we strongly recommend a wider implementation of the possibility to prespecify variance components and note this as a key feature for any breeding value estimation software.

Even though the ranking of animals was clearly improved with the threshold model, evaluating binary and ordinal variables with LMMs still lead to a relatively good ranking of the animals. Therefore, we can confirm the findings of previous studies which suggested a good ranking performance of LMMs for binary and ordinal data (Negussie, Strandén, and Mäntysaari 2008, Vazquez, Perez-Cabal, et al. 2012). Besides the ranking performance, however, LMMs showed clear deficiencies with respect to estimating the absolute breeding values as well as providing reliable accuracy estimates of the breeding values. The ranking of animals is of major importance when the goal is to select based on only one trait. However, modern breeding programs include a large variety of selection criteria which need to be weighted according to their economic importance. For each trait the response to selection has to be taken into account for the decision whether it should be included in the aggregate genotype. The response to selection is largely influenced by the heritability. As it is shown in Golan, Lander, and Rosset (2014), estimates of heritability from LMMs applied to binary data are biased and underestimate the true heritability. The bias increases with the size of the data set and with increasing imbalance between the two outcomes in the data. Both factors are highly relevant with respect to the Qualitas data sets. Therefore, using LMMs for evaluation of binary or ordinal data may lead to an underestimation of the potential response to selection and following from this a underrepresentation in the aggregate genotype. At least a subset of the data should be fitted with a realistic model which takes into account the binary or ordinal nature of the response variable. From the resulting variance component estimates, a realistic heritability can be calculated leading to better decision making in the breeding program.

Analyzing potential alternative implementations showed some promising candidates. The package `spaMM` performed well for small data sets but completely failed to converge for larger ones. We did not perform a detailed analysis of the problem and it could be worth to further investigate the possibilities of this package. `pedigreemm` was our starting point for the likelihood implementation. The package had due to the bug in the random effect transformation major deficiencies in predicting accurate breeding values which was reflected in the bad ranking performance. However, the fitting of GLMMs was clearly faster compared to `cowfit_glmer()`. In contrast to `pedigreemm`, our implementation

avoids the initial model fitting without considering the animal correlation structure, as this caused problems in the evaluation of LMMs with single observations per animal. However, as we have seen, the initial model fitting can significantly speed up the subsequent fitting of the complete model. As a further improvement of `cowfit_glmer()`, it might be good to avoid model fitting for LMMs with single observations but allow it for all other models. One of the most promising implementations was `MCMCglmm`. The package is specifically designed for analyzing phylogenetic models and implements the concept of the reduced animal model in order to speed up the MCMC sampling for categorical traits (for details see Quaas and Pollak 1980 and J. D. Hadfield 2015). `brms` and Stan might be generally advantageous for complex models but for this specific task, `MCMCglmm` seems to have an edge due to the above mentioned computational shortcuts. It is also worth pointing out that both, `spaMM` and `MCMCglmm` allow for random regression, prespecified variance components and animal models with single observations per animal. A more detailed comparison of all implementations would be desirable for a better understanding of their specific advantages and disadvantages.

Overall, the benchmarking provided valuable insights into the different models and a better understanding of the major fitting functions. Still, it is important to keep in mind that the simulations which were used are simplified and only provid an artificial representation of real world data sets. Not all conclusions from these data sets must necessarily apply to more complex models.

6.2 Qualitas Data

Applying common model selection strategies on a full animal model with unknown variance components has proven to be difficult due to the size of the data sets. As already mentioned in Chapter 6.1, fast model selection remains challenging and might be a reason why most studies in animal breeding do not report anything about how they came up with their final set of predictors. Model selection should always be performed with the main purpose of the model in mind, which in our case is to predict accurate breeding values. Therefore, the parsimony of the model for easier interpretation of model parameters is not so much of importance. Still, parsimonious models are desirable in order to decrease the computational complexity for estimating large data sets. For the same reason, we applied relatively low p-value thresholds in our model selection procedure. The heuristic approach of estimating animal variance components with simple sire models or subsets of the data was applicable for all traits and resulted in comparable variance component estimates from both methods. The procedure can be regarded as common practice in animal breeding (Calus, Schrooten, and Veerkamp 2014, Gilmour and Thompson 2003). Estimated variance components led to estimates of heritability which were not always in range of the heritability found in literature. The discrepancy might be explained by differences between the models in use as well as the large imbalance in the distribution of the response variables in our data set.

All three traits had a strongly imbalanced response distribution. Neither GLMMs nor threshold models formally require a balanced response distribution. However, the distribution can be so imbalanced such that the information content is too low to accurately estimate the model parameters, which in turn leads to a higher variability in the estimates (Salas-Eljatib et al. 2018). Less problems are expected with increasing amount of data. Therefore, special care has to be taken if the data is subsetted for variance component estimation.

Despite the highly unbalanced distribution of multiple birth, our model resulted in a high heritability estimate compared to literature. However, compared to other traits, the heritability is still low which negatively affects the expected response to selection. The positive correlation of the trait with milk yield further hampers the possibility to achieve large breeding progress simultaneously in both traits. The economic importance of the trait might be decreasing due to the widespread use of sexed semen resulting in a lower rate of mixed sex twins and therefore also a lower prevalence of freemartinism. Consequentially, adding the trait to the aggregate genotype should be carefully evaluated and possible trade-offs have to be taken into account.

The calf mortality data set was by far the largest one and caused some problems with the animal model which we were not able to resolve yet. The problems are caused by the need to subset the Cholesky factor such that only animals with observations are included. The subsetting is implemented in `pedigreemm` and requires the calculation of the numerator relationship matrix as an intermediate step. The calculation currently returns an error due to the size of the pedigree. Future improvements of `cowfit` should implement the possibility to estimate breeding values of all animals in the pedigree, thereby avoiding the need to subset the Cholesky factor.

The carcass conformation data set was relatively small and highly unbalanced which limited the possibilities to explore threshold models with higher complexity. A follow up project could include data of different breeds. Based on the results of Kause et al. (2015), different breeds may have different model parameters. Therefore, the additional observations would either have to be fitted with a separate model or the predictor breed could possibly be included as random regression factor allowing deviations from the global parameter values depending on the breed. Slaughterhouse specific thresholds should be further investigated if a reasonable number of observations from each slaughterhouse is available .

6.3 Further research

All genetic evaluations presented in this thesis are pedigree based and do not make use of genomic data. Estimation of genomic breeding values has gained huge popularity in recent years and might replace the pedigree approach completely within the next decades. A elegant way to include genomic data into our current implementation would be to estimate the additive numerator relationship matrix not via the pedigree but instead using genomic marker data.

Chapter 7

Conclusion

The widespread practice in animal breeding of evaluating non-normal data with LMMs violates basic model assumptions and may result in biased breeding value predictions and biased heritability estimates. While the animal ranking performance is still relatively good, the consequences of biased heritability estimates are far worse and may include an underrepresentation in the aggregate genotype. GLMMs and threshold models offer a promising solution for the genetic evaluation of non-normal response variables. Both models are highly flexible and capable of modeling the vast majority of traits which are relevant in animal breeding. After several decades of research and application in various areas, they are well documented and implemented in most statistical software solutions. Several packages in R are capable of including the important correlation structure between animals into the model. Small data sets can rapidly be fitted using likelihood implementations whereas Bayesian implementations are particularly flexible and better capable of estimating the uncertainty in the random effect prediction. Implementations were successfully tested on simulated and real world data sets. For this reason we highly recommend to include GLMMs and threshold models into the routine breeding value estimation process.

Acknowledgement

Many people contributed to the completion of this thesis. A special thanks goes to my supervisors, Lukas Meier (ETH) and Peter von Rohr (Qualitas) for their constant support in all subjects. Although there was rarely a meeting due to the Coronavirus pandemic, we managed to maintain a regular exchange of ideas and they helped me whenever I got stuck. Further, the autor would like to acknowledge the helpful Stan and Stack Overflow community as well as the Qualitas team for sharing their data and providing valuable feedback. Finally a big thank you to Larissa and the members of WG Rübis Stübis for proofreading and the moral support.

Bibliography

Bates, Douglas (2018). "Computational Methods for Mixed Models". en. Unpubished (http://cran.fhcrc.org/web/packages/lme4/vignettes/Theory.pdf).

– (2020). *lme4: Mixed-effects modeling with R*. Unpublished. URL: https://stat.ethz.ch/~maechler/MEMo-pages/lMMwR.pdf (visited on 04/23/2020).

Bates, Douglas, Martin Mächler, et al. (2015). "Fitting Linear Mixed-Effects Models Using Lme4". en. In: *Journal of Statistical Software* 67.1. ISSN: 1548-7660. DOI: 10.18637/jss.v067.i01.

Bates, Douglas, Martin Maechler, et al. (2020). *lme4: Linear Mixed-Effects Models using 'Eigen' and S4*. R package version 1.1-23. URL: https://CRAN.R-project.org/package=lme4.

Bates, Douglas and Ana Ines Vazquez (2014). *pedigreemm: Pedigree-based mixed-effects models*. R package version 0.3-3. URL: https://CRAN.R-project.org/package=pedigreemm.

Bates, Douglas and Donald G. Watts (1988). *Nonlinear Regression Analysis and Its Applications*. en. Wiley Series in Probability and Mathematical Statistics. New York: Wiley. ISBN: 978-0-471-81643-0.

Berweger, Madeleine (2020). *Zuchtwertschätyung Aufzuchtverluste. Internal Report*. Tech. rep. Qualitas AG.

Betancourt, Michael (July 2018). "A Conceptual Introduction to Hamiltonian Monte Carlo". In: *arXiv:1701.02434 [stat]*. arXiv: 1701.02434 [stat].

Bradley, Edwin L. (Mar. 1973). "The Equivalence of Maximum Likelihood and Weighted Least Squares Estimates in the Exponential Family". In: *Journal of the American Statistical Association* 68.341, pp. 199–200. ISSN: 0162-1459. DOI: 10.1080/01621459.1973.10481364.

Bürkner, Paul-Christian (Aug. 2017). "Brms: An R Package for Bayesian Multilevel Models Using Stan". en. In: *Journal of Statistical Software* 80.1, pp. 1–28. ISSN: 1548-7660. DOI: 10.18637/jss.v080.i01.

– (2018). "Advanced Bayesian Multilevel Modeling with the R Package Brms". en. In: *The R Journal* 10.1, p. 395. ISSN: 2073-4859. DOI: 10.32614/RJ-2018-017.

– (2020). *brms: Bayesian Regression Models using 'Stan'*. R package version 2.13.3. URL: https://CRAN.R-project.org/package=brms.

Bürkner, Paul-Christian and Matti Vuorre (2019). "Ordinal Regression Models in Psychology: A Tutorial". en. In: *Advances in Methods and Practices in Psychological Science* 2.1, pp. 77–101. ISSN: 2515-2459, 2515-2467. DOI: 10.1177/2515245918823199.

Cady, R. A. and L. D. Van Vleck (Apr. 1978). "Factors Affecting Twinning and Effects of Twinning in Holstein Dairy Cattle". en. In: *Journal of Animal Science* 46.4, pp. 950–956. ISSN: 0021-8812. DOI: 10.2527/jas1978.464950x.

Calus, Mario PL, Chris Schrooten, and Roel F. Veerkamp (Sept. 2014). "Genomic Prediction of Breeding Values Using Previously Estimated SNP Variances". In: *Genetics Selection Evolution* 46.1, p. 52. ISSN: 1297-9686. DOI: 10.1186/s12711-014-0052-x.

Carlen, Emma et al. (Dec. 2016). "Youngstock Survival in Nordic Cattle Genetic Evaluation". en. In: *Interbull Bulletin* 50. ISSN: 2001-340X.

Carpenter, Bob et al. (Jan. 2017). "Stan: A Probabilistic Programming Language". en. In: *Journal of Statistical Software* 76.1, pp. 1–32. ISSN: 1548-7660. DOI: 10.18637/jss.v076.i01.

Chang, Y.M. et al. (Feb. 2006). "Bivariate Analysis of Number of Services to Conception and Days Open in Norwegian Red Using a Censored Threshold-Linear Model". en. In: *Journal of Dairy Science* 89.2, pp. 772–778. ISSN: 00220302. DOI: 10.3168/jds.S0022-0302(06)72138-5.

Charnes, A., E. L. Frome, and P. L. Yu (Mar. 1976). "The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family". In: *Journal of the American Statistical Association* 71.353, pp. 169–171. ISSN: 0162-1459. DOI: 10.1080/01621459.1976.10481508.

Chen, Yanqing et al. (Oct. 2008). "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate". In: *ACM Transactions on Mathematical Software* 35.3, 22:1–22:14. ISSN: 0098-3500. DOI: 10.1145/1391989.1391995.

Del Río, N. Silva et al. (Mar. 2007). "An Observational Analysis of Twin Births, Calf Sex Ratio, and Calf Mortality in Holstein Dairy Cattle". en. In: *Journal of Dairy Science* 90.3, pp. 1255–1264. ISSN: 0022-0302. DOI: 10.3168/jds.S0022-0302(07)71614-4.

Dunn, Peter K. and Gordon K. Smyth (2018). *Generalized Linear Models with Examples in R*. Springer Texts in Statistics. New York, NY: Springer. ISBN: 978-1-4419-0118-7.

Falconer, Douglas S. and Trudy F. C. Mackay (1996). *Introduction to Quantitative Genetics*. en. 4. ed. Harlow: Pearson, Prentice Hall. ISBN: 978-0-582-24302-6.

Fisher, R. A. (1918). "The Correlation between Relatives on the Supposition of Mendelian Inheritance." en. In: *Transactions of the Royal Society of Edinburgh* 52.2, pp. 399–433. ISSN: 0080-4568. DOI: 10.1017/S0080456800012163.

Fouilloux, Marie-Noëlle and Denis Laloë (Sept. 2001). "A Sampling Method for Estimating the Accuracy of Predicted Breeding Values in Genetic Evaluation". In: *Genetics Selection Evolution* 33.5, p. 473. ISSN: 1297-9686. DOI: 10.1186/1297-9686-33-5-473.

Gabry, Jonah and Ben Goodrich (2020a). *Estimating Generalized (Non-)Linear Models with Group-Specific Terms with rstanarm*. URL: https://cran.r-project.org/web/packages/rstanarm/vignettes/glmer.html (visited on 07/09/2020).

– (2020b). *Prior Distributions for rstanarm Models*. URL: https://cran.r-project.org/web/packages/rstanarm/vignettes/priors.html (visited on 07/23/2020).

– (2020c). *rstanarm: Bayesian Applied Regression Modeling via Stan*. R package version 2.19.3. URL: https://CRAN.R-project.org/package=rstanarm.

Gelman, Andrew (Sept. 2006). "Prior Distributions for Variance Parameters in Hierarchical Models (Comment on Article by Browne and Draper)". EN. In: *Bayesian Analysis* 1.3, pp. 515–534. ISSN: 1936-0975. DOI: 10.1214/06-BA117A.

Ghavi Hossein-Zadeh, N. et al. (July 2009). "Estimation of Variance Components and Genetic Trends for Twinning Rate in Holstein Dairy Cattle of Iran". en. In: *Journal of Dairy Science* 92.7, pp. 3411–3421. ISSN: 0022-0302. DOI: 10.3168/jds.2008-1631.

Gianola, D. and J. L. Foulley (Apr. 1983). "Sire Evaluation for Ordered Categorical Data with a Threshold Model". In: *Génétique, sélection, évolution* 15.2, p. 201. ISSN: 1297-9686. DOI: 10.1186/1297-9686-15-2-201.

Gilmour, A. R. and Robin Thompson (2003). "Options for Estimating Variance Components in Large Mixed Models". In: *Proc Adv Anim Breed Gen* 15, pp. 206–209.

Golan, David, Eric S. Lander, and Saharon Rosset (Dec. 2014). "Measuring Missing Heritability: Inferring the Contribution of Common Variants". en. In: *Proceedings of the National Academy of Sciences* 111.49, E5272–E5281. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1419064111.

Gregory, K. E., G. L. Bennett, et al. (May 1997). "Genetic and Environmental Parameters for Ovulation Rate, Twinning Rate, and Weight Traits in a Cattle Population Selected for Twinning". en. In: *Journal of Animal Science* 75.5, pp. 1213–1222. ISSN: 0021-8812. DOI: 10.2527/1997.7551213x.

Gregory, K. E., S. E. Echternkamp, et al. (July 1990). "Twinning in Cattle: I. Foundation Animals and Genetic and Environmental Effects on Twinning Rate". en. In: *Journal of Animal Science* 68.7, pp. 1867–1876. ISSN: 0021-8812. DOI: 10.2527/1990.6871867x.

Gulliksen, S. M. et al. (June 2009). "Calf Mortality in Norwegian Dairy Herds". en. In: *Journal of Dairy Science* 92.6, pp. 2782–2795. ISSN: 0022-0302. DOI: 10.3168/jds.2008-1807.

Guo, Jiqiang, Jonah Gabry, and Ben Goodrich (2020). *rstan: R Interface to Stan*. R package version 2.19.3. URL: https://CRAN.R-project.org/package=rstan.

Hadfield, Jarrod (2019). *MCMCglmm: MCMC Generalised Linear Mixed Models*. R package version 2.29. URL: https://CRAN.R-project.org/package=MCMCglmm.

Hadfield, Jarrod D. (June 2015). "Increasing the Efficiency of MCMC for Hierarchical Phylogenetic Models of Categorical Traits Using Reduced Mixed Models". en. In: *Methods in Ecology and Evolution* 6.6, pp. 706–714. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12354.

Harville, D. A. and T. P. Callanan (1989). "Computational Aspects of Likelihood-Based Inference for Variance Components". In: *Advances in Statistical Methods for Genetic Imporvement of Livestock*. Ed. by D. Gianola and K. Hammond. Berlin, Germany: Springer-Verlag, pp. 136–176.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1, pp. 97–109. ISSN: 0006-3444. DOI: 10.2307/2334940.

Haws, David C. et al. (June 2015). "Variable-Selection Emerges on Top in Empirical Comparison of Whole-Genome Complex-Trait Prediction Methods". en. In: *PLOS ONE* 10.10, e0138903. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0138903.

Hayes, B. J. et al. (Feb. 2009). "Invited Review: Genomic Selection in Dairy Cattle: Progress and Challenges". en. In: *Journal of Dairy Science* 92.2, pp. 433–443. ISSN: 0022-0302. DOI: 10.3168/jds.2008-1646.

Henderson, Charles R. (1982). "Analysis of Covariance in the Mixed Model: Higher-Level, Nonhomogeneous, and Random Regressions". In: *Biometrics* 38.3, pp. 623–640. ISSN: 0006-341X. DOI: 10.2307/2530044.

– (1984). *Applications of Linear Models in Animal Breeding*. Vol. 462. University of Guelph Guelph.

Heringstad, B. et al. (Oct. 2006). "Genetic Analysis of Number of Mastitis Cases and Number of Services to Conception Using a Censored Threshold Model". en. In: *Journal of Dairy Science* 89.10, pp. 4042–4048. ISSN: 00220302. DOI: 10.3168/jds.S0022-0302(06)72447-X.

Hickey, J. M. et al. (Feb. 2007). "Genetic Parameters for EUROP Carcass Traits within Different Groups of Cattle in Ireland". en. In: *Journal of Animal Science* 85.2, pp. 314–321. ISSN: 0021-8812. DOI: 10.2527/jas.2006-263.

Hill, William G. (Jan. 2014). "Applications of Population Genetics to Animal Breeding, from Wright, Fisher and Lush to Genomic Prediction". In: *Genetics* 196.1, pp. 1–16. ISSN: 0016-6731. DOI: 10.1534/genetics.112.147850.

Holand, Anna Marie et al. (Aug. 2013). "Animal Models and Integrated Nested Laplace Approximations". en. In: *G3: Genes, Genomes, Genetics* 3.8, pp. 1241–1251. ISSN: 2160-1836. DOI: 10.1534/g3.113.006700.

Jiang, Jiming (2007). *Linear and Generalized Linear Mixed Models and Their Applications*. Springer Series in Statistics. New York ; London: Springer. ISBN: 978-0-387-47941-5.

Johansson, Ivar, Bengt Lindhé, and Franz Pirchner (1974). "Causes of Variation in the Frequency of Monozygous and Dizygous Twinning in Various Breeds of Cattle". en. In: *Hereditas* 78.2, pp. 201–234. ISSN: 1601-5223. DOI: 10.1111/j.1601-5223.1974.tb01443.x.

Kadarmideen, H. N., Robin Thompson, and G. Simm (Dec. 2000). "Linear and Threshold Model Genetic Parameters for Disease, Fertility and Milk Production in Dairy Cattle". en. In: *Animal Science* 71.3, pp. 411–419. ISSN: 1357-7298, 1748-748X. DOI: 10.1017/S1357729800055338.

Karlsen, A. et al. (Jan. 2000). "Twinning Rate in Norwegian Cattle: Frequency, (Co)Variance Components, and Genetic Trends". en. In: *Journal of Animal Science* 78.1, pp. 15–20. ISSN: 0021-8812. DOI: 10.2527/2000.78115x.

Kause, A. et al. (Jan. 2015). "Genetic Parameters for Carcass Weight, Conformation and Fat in Five Beef Cattle Breeds". en. In: *animal* 9.1, pp. 35–42. ISSN: 1751-7311, 1751-732X. DOI: 10.1017/S1751731114001992.

Komisarek, Jolanta and Zbigniew Dorynek (2002). "Genetic Aspects of Twinning in Cattle". eng. In: *Journal of Applied Genetics* 43.1, pp. 55–68. ISSN: 1234-1983.

Kriese, L. A., J. K. Bertrand, and L. L. Benyshek (Feb. 1991). "Age Adjustment Factors, Heritabilities and Genetic Correlations for Scrotal Circumference and Related Growth Traits in Hereford and Brangus Bulls". en. In: *Journal of Animal Science* 69.2, pp. 478–489. ISSN: 0021-8812. DOI: 10.2527/1991.692478x.

Kuznetsova, Alexandra, Per Bruun Brockhoff, and Rune Haubo Bojesen Christensen (2020). *lmerTest: Tests in Linear Mixed Effects Models*. R package version 3.1-2. URL: https://CRAN.R-project.org/package=lmerTest.

Lewandowski, Daniel, Dorota Kurowicka, and Harry Joe (Oct. 2009). "Generating Random Correlation Matrices Based on Vines and Extended Onion Method". en. In: *Journal of Multivariate Analysis* 100.9, pp. 1989–2001. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2009.04.008.

Lin, Xihong and Norman E. Breslow (Sept. 1996). "Bias Correction in Generalized Linear Mixed Models with Multiple Components of Dispersion". In: *Journal of the American Statistical Association* 91.435, pp. 1007–1016. ISSN: 0162-1459. DOI: 10.1080/01621459.1996.10476971.

Lynch, Michael and Bruce Walsh (1998). *Genetics and Analysis of Quantitative Traits*. Sunderland, Mass: Sinauer Assoc. ISBN: 978-0-87893-481-2.

Maijala, K. and Anu Osva (1990). "Genetic Correlations of Twinning Frequency with Other Economic Traits in Dairy Cattle". en. In: *Journal of Animal Breeding and Genetics* 107.1-6, pp. 7–15. ISSN: 1439-0388. DOI: 10.1111/j.1439-0388.1990.tb00003.x.

McCulloch, Charles E. and Shayle R. Searle (Dec. 2000). *Generalized, Linear, and Mixed Models*. en. First. Wiley Series in Probability and Statistics. Wiley. ISBN: 978-0-471-19364-7. DOI: 10.1002/0471722073.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard (Apr. 2001). "Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps". en. In: *Genetics* 157.4, pp. 1819–1829. ISSN: 0016-6731, 1943-2631.

Miglior, Filippo et al. (Dec. 2017). "A 100-Year Review: Identification and Genetic Selection of Economically Important Traits in Dairy Cattle". en. In: *Journal of Dairy Science* 100.12, pp. 10251–10271. ISSN: 0022-0302. DOI: 10.3168/jds.2017-12968.

Morris, C. A. and M. Wheeler (Jan. 2002). "Genetic Variation in Twin Calving Incidence in Herds with a High Phenotypic Mean". In: *New Zealand Journal of Agricultural Research* 45.1, pp. 17–25. ISSN: 0028-8233. DOI: 10.1080/00288233.2002.9513489.

Mrode, R. A. and Robin Thompson (2005). *Linear Models for the Prediction of Animal Breeding Values*. en. 2nd ed. Wallingford, UK ; Cambridge, MA: CABI Pub. ISBN: 978-0-85199-000-2.

Natarajan, Ranjini and Robert E. Kass (2000). "Reference Bayesian Methods for Generalized Linear Mixed Models". en. In: *Journal of the American Statistical Association* 95.449, pp. 227–237. ISSN: 0162-1459. DOI: 10.1080/01621459.2000.10473916.

Negussie, Enyew, Ismo Strandén, and Esa A. Mäntysaari (Aug. 2008). "Genetic Analysis of Liability to Clinical Mastitis, with Somatic Cell Score and Production Traits Using Bivariate Threshold–Linear and Linear–Linear Models". en. In: *Livestock Science* 117.1, pp. 52–59. ISSN: 1871-1413. DOI: 10.1016/j.livsci.2007.11.009.

Neuenschwander, T. et al. (Apr. 2005). "Genetics of Parity-Dependant Production Increase and Its Relationship with Health, Fertility, Longevity, and Conformation in Swiss Holsteins". en. In: *Journal of Dairy Science* 88.4, pp. 1540–1551. ISSN: 0022-0302. DOI: 10.3168/jds.S0022-0302(05)72823-X.

O'Neill, Christopher J., David L. Swain, and H. N. Kadarmideen (Sept. 2010). "Evolutionary Process of Bos Taurus Cattle in Favourable versus Unfavourable Environments and Its Implications for Genetic Selection: Evolutionary Process of B. Taurus". en. In: *Evolutionary Applications* 3.5-6, pp. 422–433. ISSN: 17524571. DOI: 10.1111/j.1752-4571.2010.00151.x.

Ogutu, Joseph O., Torben Schulz-Streeck, and Hans-Peter Piepho (May 2012). "Genomic Selection Using Regularized Linear Regression Models: Ridge Regression, Lasso, Elastic Net and Their Extensions". In: *BMC Proceedings* 6.2, S10. ISSN: 1753-6561. DOI: 10.1186/1753-6561-6-S2-S10.

Østerås, Olav et al. (Dec. 2007). "Perinatal Death in Production Animals in the Nordic Countries – Incidence and Costs". In: *Acta Veterinaria Scandinavica* 49.1, S14. ISSN: 1751-0147. DOI: 10.1186/1751-0147-49-S1-S14.

Pinheiro, José, Douglas Bates, and R-core (2020). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-148. URL: https://CRAN.R-project.org/package=nlme.

Quaas, R. L. and E. J. Pollak (Dec. 1980). "Mixed Model Methodology for Farm and Ranch Beef Cattle Testing Programs". en. In: *Journal of Animal Science* 51.6, pp. 1277–1287. ISSN: 0021-8812. DOI: 10.2527/jas1981.5161277x.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

R Mailing List (2010). *[R-sig-ME] pedigreemm with one observation per individual*. URL: https://stat.ethz.ch/pipermail/r-sig-mixed-models/2010q1/003340.html (visited on 06/29/2020).

– (2012). *[R-sig-ME] pedigreemm and heritability of binary data*. URL: https://stat.ethz.ch/pipermail/r-sig-mixed-models/2012q2/018369.html (visited on 06/29/2020).

R Mailing List (2014). *[R-sig-ME] pedigreemm number of levels per grouping factor*. URL: https://stat.ethz.ch/pipermail/r-sig-mixed-models/2014q1/021609.html (visited on 07/04/2020).

Rodrigues-Motta, M. et al. (Nov. 2007). "A Zero-Inflated Poisson Model for Genetic Analysis of the Number of Mastitis Cases in Norwegian Red Cows". en. In: *Journal of Dairy Science* 90.11, pp. 5306–5315. ISSN: 00220302. DOI: 10.3168/jds.2006-898.

Ron, M., E. Ezra, and JI Weller (Oct. 1990). "Genetic Analysis of Twinning Rate in Israeli Holstein Cattle". In: *Genetics Selection Evolution* 22.3, p. 349. ISSN: 1297-9686. DOI: 10.1186/1297-9686-22-3-349.

Rousset, François, Jean-Baptiste Ferdy, and Alexandre Courtiol (2020). *spaMM: Mixed-Effect Models, Particularly Spatial Models*. R package version 3.4.1. URL: https://CRAN.R-project.org/package=spaMM.

Salas-Eljatib, Christian et al. (Feb. 2018). "A Study on the Effects of Unbalanced Data When Fitting Logistic Regression Models in Ecology". en. In: *Ecological Indicators* 85, pp. 502–508. ISSN: 1470-160X. DOI: 10.1016/j.ecolind.2017.10.030.

Santman-Berends, I. M. G. A., M. Buddiger, et al. (2014). "A Multidisciplinary Approach to Determine Factors Associated with Calf Rearing Practices and Calf Mortality in Dairy Herds". en. In: *Preventive Veterinary Medicine*. Special Issue: SVEPM 2014 - Supporting Decision Making on Animal Health through Advanced and Multidisciplinary Methodologies, 2014 Society of Veterinary Epidemiology and Preventive Medicine Conference 117.2, pp. 375–387. ISSN: 0167-5877. DOI: 10.1016/j.prevetmed.2014.07.011.

Santman-Berends, I. M. G. A., Y.H. Schukken, and G. van Schaik (2019). "Quantifying Calf Mortality on Dairy Farms: Challenges and Solutions". en. In: *Journal of Dairy Science* 102.7, pp. 6404–6417. ISSN: 00220302. DOI: 10.3168/jds.2019-16381.

Sargolzaei, Mehdi and Hiroaki Iwaisaki (2005). "Comparison of Four Direct Algorithms for Computing Inbreeding Coefficients". en. In: *Animal Science Journal* 76.5, pp. 401–406. ISSN: 1740-0929. DOI: 10.1111/j.1740-0929.2005.00282.x.

Stack Overflow (2016). *r - Fixing variance values in lme4*. URL: https://stackoverflow.com/questions/39718754/fixing-variance-values-in-lme4 (visited on 07/02/2020).

Svensson, C., A. Linder, and S. -O. Olsson (Dec. 2006). "Mortality in Swedish Dairy Calves and Replacement Heifers". en. In: *Journal of Dairy Science* 89.12, pp. 4769–4777. ISSN: 0022-0302. DOI: 10.3168/jds.S0022-0302(06)72526-7.

Syrstad, O. (Agricultural Univ of Norway (1984). "Inheritance of Multiple Births in Cattle". English. In: *Livestock Production Science (Netherlands)*.

Tarrés, Joaquim et al. (May 2011). "Carcass Conformation and Fat Cover Scores in Beef Cattle: A Comparison of Threshold Linear Models vs Grouped Data Models". In: *Genetics Selection Evolution* 43.1, p. 16. ISSN: 1297-9686. DOI: 10.1186/1297-9686-43-16.

Tempelman, Robert J. (May 1998). "Generalized Linear Mixed Models in Dairy Cattle Breeding". en. In: *Journal of Dairy Science* 81.5, pp. 1428–1444. ISSN: 00220302. DOI: 10.3168/jds.S0022-0302(98)75707-8.

Van Tassell, C. P., L. D. Van Vleck, and K. E. Gregory (Aug. 1998). "Bayesian Analysis of Twinning and Ovulation Rates Using a Multiple-Trait Threshold Model and Gibbs Sampling". en. In: *Journal of Animal Science* 76.8, pp. 2048–2061. ISSN: 0021-8812. DOI: 10.2527/1998.7682048x.

Van Vleck, L. D. and K. E. Gregory (Mar. 1996). "Genetic Trend and Environmental Effects in a Population of Cattle Selected for Twinning". en. In: *Journal of Animal Science* 74.3, pp. 522–528. ISSN: 0021-8812. DOI: 10.2527/1996.743522x.

Varona, L., C. Moreno, and J. Altarriba (Apr. 2009). "A Model with Heterogeneous Thresholds for Subjective Traits: Fat Cover and Conformation Score in the Pirenaica

Beef Cattle". en. In: *Journal of Animal Science* 87.4, pp. 1210–1217. ISSN: 0021-8812. DOI: 10.2527/jas.2008-1361.

Vazquez, Ana Ines, Douglas Bates, et al. (2010). "Technical Note: An R Package for Fitting Generalized Linear Mixed Models in Animal Breeding". en. In: *Journal of Animal Science* 88.2, pp. 497–504. ISSN: 0021-8812. DOI: 10.2527/jas.2009-1952.

Vazquez, Ana Ines, D. Gianola, et al. (Feb. 2009). "Assessment of Poisson, Logit, and Linear Models for Genetic Analysis of Clinical Mastitis in Norwegian Red Cows". en. In: *Journal of Dairy Science* 92.2, pp. 739–748. ISSN: 00220302. DOI: 10.3168/jds.2008-1325.

Vazquez, Ana Ines, M.A. Perez-Cabal, et al. (Apr. 2012). "Predictive Ability of Alternative Models for Genetic Analysis of Clinical Mastitis: Comparison of Models for Mastitis". en. In: *Journal of Animal Breeding and Genetics* 129.2, pp. 120–128. ISSN: 09312668. DOI: 10.1111/j.1439-0388.2011.00950.x.

Wang, Wei (2013). "Identifiability of Linear Mixed Effects Models". EN. In: *Electronic Journal of Statistics* 7, pp. 244–263. ISSN: 1935-7524. DOI: 10.1214/13-EJS770.

Willam, Alfons and Henner Simianer (2017). *Tierzucht*. ger. 2., vollständig überarbeitete und erweiterte Auflage. UTB Agrarwissenschaften, Veterinärmedizin 3526. Stuttgart: Verlag Eugen Ulmer. ISBN: 978-3-8252-4805-5.

Yin, T. et al. (Apr. 2014). "Genetic Analyses of Binary Longitudinal Health Data in Small Low Input Dairy Cattle Herds Using Generalized Linear Mixed Models". en. In: *Livestock Science* 162, pp. 31–41. ISSN: 1871-1413. DOI: 10.1016/j.livsci.2014.01.021.

Zhang, T. et al. (1994). "Diagnosis of Freemartinism in Cattle: The Need for Clinical and Cytogenetic Evaluation". English. In: *Journal of the American Veterinary Medical Association (USA)*. ISSN: 0003-1488.

Zhao, Y. et al. (Feb. 2006). "General Design Bayesian Generalized Linear Mixed Models". en. In: *Statistical Science* 21.1, pp. 35–51. ISSN: 0883-4237. DOI: 10.1214/088342306000000015.

Zihlmann, Reto (2020). *cowfit: LMMs and GLMMs in genetic evaluation*. URL: https://github.com/retodomax/cowfit (visited on 08/22/2020).

Declaration of Originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

Title of work (in block letters):

Generalized Linear Mixed Effects Models in Genetic Evaluations

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Zihlmann	Reto

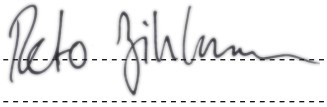
With my signature I confirm that

- I have committed none of the forms of plagiarism described in the Citation etiquette information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .
- I am aware that the work may be screened electronically for plagiarism.
- I have understood and followed the guidelines in the document *Scientific Works in Mathematics*.

Place, date:

Zürich, 31.08.2020

Signature(s):



For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.