Statistical algorithms for semi-parametric variance regression with application to biomarkers

By

Kristy Pamela Robledo

A thesis submitted to Macquarie University
for the degree of
Doctor of Philosophy

Department of Statistics
Faculty of Science and Engineering
December 2017

MACQUARIE
University
SYDNEY·AUSTRALIA

Contents

Abstract

Variance regression allows for heterogeneous variance, or heteroscedasticity, by incorporating a regression model into the variance. This thesis uses a variant of the Expectation-Maximisation (EM) algorithm to develop a new method for fitting additive variance regression models that allow for regression in both the mean and the variance. The algorithm is easily extended to allow for B-spline bases, thus allowing for the incorporation of a semi-parametric model in both the mean and variance. Although there are existing methods to fit these types of models, this new algorithm provides a reliable approach that is not susceptible to numerical instability that can be seen with other approaches.

We utilise the developed algorithm with a series of simulation studies and analysis of biomarker datasets. Various simulation studies show that the algorithm is capable of recovering the true model for a variety of scenarios. We also study automatic selection of model complexity based on various information criteria, and show that the Akaike information criterion (AIC) is useful for choosing the optimal number of knots in a B-spline model. It is also found that the ability to estimate the model complexity automatically is greatly improved with a larger sample size.

The algorithm is extended to allow for censored outcome data, and to allow for non-normal data with the incorporation of a skew regression model, using the skew-normal distribution. The algorithms developed in this thesis are available through an R package called `VarReg`, and a demonstration of the package is given using a biomarker dataset. This algorithm has wide capabilities for analysis of biomarker data, and provides a useful and stable additional tool for fitting variance regression models.

Statement of candidate

I certify that the work in this thesis entitled "Statistical algorithms for semi-parametric variance regression with application to biomarkers" has not previously been submitted for a degree, nor has it been submitted as part of requirements for a degree, to any university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me. Any assistance that I have received in my research work and the preparation of the thesis itself has been appropriately acknowledged.

In addition, I certify that all information sources and literature used are referenced in the thesis.

Kristy Pamela Robledo

Date

Acknowledgements

Firstly I would like to acknowledge my principal supervisor, Professor Ian Marschner, for allowing me the opportunity to undertake this PhD under his guidance. His door was always open whenever I had trouble and the time, patience and support he has provided has been monumental. Words cannot express my gratitude.

I would also like to thank my Associate supervisor, Professor Gillian Heller, for her thoughtful observations and input. The entire Department of Statistics, including fellow postgraduates, at Macquarie University made the entire research experience enjoyable. Additionally, thanks to my fellow colleagues at the NHMRC Clinical Trials Centre for their ongoing support.

Finally, I must express my very profound gratitude to my parents and to my husband for providing me with unfailing support and continuous encouragement throughout my years of study, and through this PhD. This accomplishment would not have been possible without them.
Thank you everyone for your support!
Kristy

List of figures

List of tables

1

Introduction

This thesis will introduce a new method for fitting semi-parametric variance regression models. Variance heterogeneity models in regression analysis allow the variance of the response variable to depend on covariates. This contrasts with standard regression models, which focus on allowing the mean of the response variable to depend on covariates. The traditional variance regression models typically assume a multiplicative model for the dependence of the error variance on the covariates (Aitkin, 1987; Smyth, 2002; Verbyla, 1993). This is primarily because covariate effects in variance heterogeneity models can be negative as well as positive, and a multiplicative model ensures that the overall error variance remains non-negative. Since this is simply a computational convenience, it may be that additive variance heterogeneity is more appropriate in some contexts. Indeed, additive decomposition of the variance is standard in other contexts, such as variance components models.

Although other methods for fitting additive variance heterogeneity models exist in

principle, in practice they can be numerically unstable due to the implicit parameter constraints that are required in an additive model. One of the motivations for this thesis is that it is of interest to develop new algorithms for problems that are computationally complex and numerically unstable, even in contexts where algorithms currently exist. This thesis will develop a new method for fitting an additive variance heterogeneity model, and extend this algorithm to incorporate a regression model in the skewness, as well as other complexities such as censored data. An advantage of our approach is that the additive variance structure naturally generalises to the inclusion of semi-parametric regression functions. An R (R Core Team, 2013) package entitled `VarReg` is presented which implements algorithms developed in this thesis (Robledo, 2017). The remainder of this chapter will cover the objectives, motivation and an overview of this thesis.

1.1 Objectives

This thesis will develop new statistical methodology for additive variance regression for modelling the effects of covariates on variance heterogeneity and study a range of applications of these techniques, including biomarker analysis. This thesis aims to develop a new method for fitting models with a regression in both the mean and the variance. These models can be used in measurement error analysis of biomarkers, and particularly an extension to allow for censored biomarker data in this setting is of interest. This thesis also aims to extend this method to allow for non-normal data with the development of a regression in the skewness of the distribution.

1.2 Motivation

Variance regression models arise in a variety of contexts, including measurement error and variance heterogeneity in standard linear regression analysis. Such models are necessary when the variance of the outcome measure changes as a covariate changes. The use of a variance regression model in these contexts allows the modelling of the variance in terms of covariates, either because the variance itself is of interest, or to increase the precision in estimation of the mean.

One area where the variance itself is of interest is in the analysis of biomarker data, which will be considered at various points in this thesis. Biomarkers are biological measurements that provide an indication of a disease state, such as cholesterol level in heart disease or viral load in HIV disease, and are an area of interest in clinical trials research. Models of biomarker variability and measurement error are important in practice because they allow an assessment of whether biomarker changes observed in response to new treatments are beyond what can be expected by chance, and whether the changes are related to patient-specific characteristics.

Other methods already exist for the fitting of additive variance heterogeneity models, such as the `gamlss` package in R. These other methods are covered in detail in Chapter 2. However, fitting these types of additive models can be complex, and finding a constrained non-stationary MLE can be difficult with Newton-type algorithms. Two simple examples are also introduced in this chapter, where the commonly used `gamlss` package is unable to converge. One is an example of a non-stationary MLE, while the other is surprisingly a stationary MLE. Both examples illustrate the need for new computational methodology for fitting these complex variance regression models.

1.3 Approach

Marschner (2014) recently suggested a possible method for fitting additive regression models with non-negativity and other parameter constraints, based on a variant of the Expectation-Maximisation (EM) algorithm. This approach will be adapted and generalised here for the use in variance heterogeneity models with additive covariate effects. The basic approach is to use the fact that an additive variance model can be viewed as having been generated from a latent outcome model in which the outcome variable is the sum of independent, unobserved, outcome variables. This allows us to use the EM algorithm which provides a numerically stable and flexible computational method that ensures the required non-negativity constraints are satisfied.

Once this approach has been developed, various generalisations are easily incorporated, such as treating censoring as an extra level of missingness in the EM formulation. Furthermore, the EM approach for the variance can be combined with other approaches for the mean and shape. This includes cycling between the mean and the variance

model fit using the Expectation/Conditional Maximisation Either (ECME) algorithm, or cycling between the shape model fit and the mean-variance model fit using a cyclic coordinate ascent approach. This allows the modelling of both normal and non-normal data, using regression models in the location, scale and shape. In the next section, we will give an overview of how these approaches are developed throughout this thesis.

1.4 Overview of the thesis

This thesis is composed of ten chapters of increasing complexity in the underlying model and data. This first chapter briefly introduces the problem, approach and structure of this thesis. Chapter 2 begins with an explanation of variance regression, and some motivating examples of why new computational algorithms are of interest. Multiplicative and additive models will be discussed, as well as other complexities with variance regression models. Chapter 3 provides some background on computational methods to be used in this thesis and an explanation of the example datasets. Chapter 4 will introduce the core idea of the thesis, which is that an additive variance model can be viewed as a latent outcome model, where the observed outcome is the sum of two latent outcomes. This will be explained by using a simplified variance regression algorithm, including a simulation study and an example dataset. This basic algorithm will then be extended to multiple regression in the mean and the variance in Chapter 5, where we will also cover standard error estimation, and again include a simulation study and an example dataset. Chapter 6 will further extend the algorithm into semi-parametric models with the use of B-spline basis functions, including monotonicity constraints, which can be of particular relevance in variance regression models. Chapter 7 introduces an algorithm to allow the outcome data to be censored. This censored data algorithm is then incorporated into Chapter 8, where we develop an algorithm to allow for non-normal data with the development of a skew or shape regression, again, with a simulation study and an example dataset. Chapter 9 demonstrates the use of the `VarReg` package to fit the algorithms developed in this thesis, with the use of an example biomarker dataset. An appendix to the thesis contains the package documentation. Lastly, Chapter 10 is a concluding chapter that consolidates the content of this thesis, and provides future research directions.

2

Variance regression

Constant variance, or homoscedasticity, is one of the standard assumptions of linear regression. Variance regression allows for heterogenous variance, or heteroscedasticity, by incorporating a regression model into the variance. This chapter will introduce variance regression models as an extension of standard linear regression. We will also explore some existing methods with which these types of models can be fit. Some problems with existing methodology will be demonstrated, and some additional complexities in this context, will be explored. The discussion in this chapter will motivate new methods for variance regression to be studied in this thesis.

2.1 Extension of linear regression

Linear regression is a standard analysis procedure for modelling the mean of a continuous outcome in terms of some covariates. The outcome variable, X_i, is associated with multiple covariates or predictors $\boldsymbol{z}_i = (z_{i1}, ..., z_{iP})$. The mean of X_i depends linearly

on the covariates z_{ip} with the corresponding regression coefficients β_p for each of the covariates ($p = 1, 2, ..., P$). We then assume normality and constant variance σ^2, giving

$$X_i \sim N\left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}, \sigma^2\right) \quad \text{for} \quad i = 1, 2, ..., n. \tag{2.1}$$

There are three main assumptions for standard linear regression:

1. the errors are assumed to be independently and identically distributed, including the assumption that the variance is constant,
2. the errors are assumed to follow a normal distribution, and
3. the regression function for the mean is linear in the predictors.

Violations of the first and second assumptions can sometimes be rectified with the use of a transformation of the response variable (Box and Cox, 1964), which can provide an approximation to normality. Violation of the third assumption has traditionally been met with the addition of polynomial terms, interactions, other non-linear transformations or semi-parametric models. It is of interest to note that least squares estimators give unbiased estimators of regression coefficients, even in the presence of heteroscedasticity.

Although transformations can sometimes be used to deal with heteroscedasticity, they are not always adequate. Furthermore, the heterogeneity of the variance could be of interest in itself. This leads to variance heterogeneity models which are an extension of the linear regression model. In these models, we fit a model to the mean, and a model to the variance,

$$X_i \sim N\left(f\left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}\right), h\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)\right), \tag{2.2}$$

with the use of two known link functions. The covariates in the variance model, x_{iq}, may be the same as the covariates in the mean model, or they may be different. The link function for the mean is typically the identity function as in (2.1), but in principle could be other functions. A commonly used link function for the variance is the log function which yields multiplicative variance models. The use of different link functions for the variance will be discussed in more detail in Section 2.4.

2.2 Motivating contexts

Variance regression models arise in a variety of contexts. In this section, we will focus on two specific motivating contexts. The first is the study of measurement error. In these studies, there may be a difference between a measured value and its true value, or perhaps readings from a new measurement technique versus the current standard technique. These data are typically graphically summarised with a Bland-Altman plot (Bland and Altman, 1986). These plots give the mean of the two observations on the x-axis, and the difference between the two observations on the y-axis, and allow visual insight into the variance of the difference over the data range. As an example, we will consider a dataset used by Bland and Altman (1986) on measurements of mean velocity of circumferential fibre shortening (VCF). This dataset has two measurements of VCF, the first measurement is taken by the long axis, and the second by the short axis in M-mode echocardiography (Darbela, Silayan, and Bland, 1986). Although the measurements were taken at the same point in time, the fact that the measurements are taken from different axes will introduce measurement error.

If $V_i^{(1)}$ is the VCF reading from the long axis, and $V_i^{(2)}$ is from the short, then the usual measurement error model for the measurements is

$$V_i^{(j)} = V_i^* + \epsilon_j \quad \text{where } \epsilon_j \sim N(0, \sigma^2) \quad j = 1, 2$$

and V_i^* is the 'true' VCF reading for the i^{th} patient. If we assume independence of the measurement errors for different measurements, then the difference in the measurements, $X_i = V_i^{(1)} - V_i^{(2)}$, follows a normal distribution with zero mean and variance $2\sigma^2$. This can be used to estimate the measurement error variance, σ^2.

Figure 2.1 shows the difference between the two measurements, versus the average of the measurements, typically referred to as a Bland-Altman plot. These data appear to have a zero mean and no change in the mean of the difference over the VCF observations. However, the scatter of the differences increases as the VCF increases, demonstrating larger variation in the difference in larger observations. A variance regression model could be used to model this increasing variance, and this is investigated in Section 4.5.

The second motivating context is variance heterogeneity in a standard linear regression analysis. This occurs when the variance of the outcome variable changes as a covariate in the model changes. An example based on CD4 data is shown in Figure 2.2. CD4 is a type of white blood cell, and here it has been measured in uninfected children born from HIV-1 infected women (Wade and Ades, 1994). In this dataset, it is clear that both the mean and the variance of CD4 counts decrease as age increases. Such data could be modelled by allowing age to be a covariate in both the mean and the variance models of CD4 counts.

A variety of different datasets from measurement error and variance heterogeneity contexts will be described in Chapter 3.

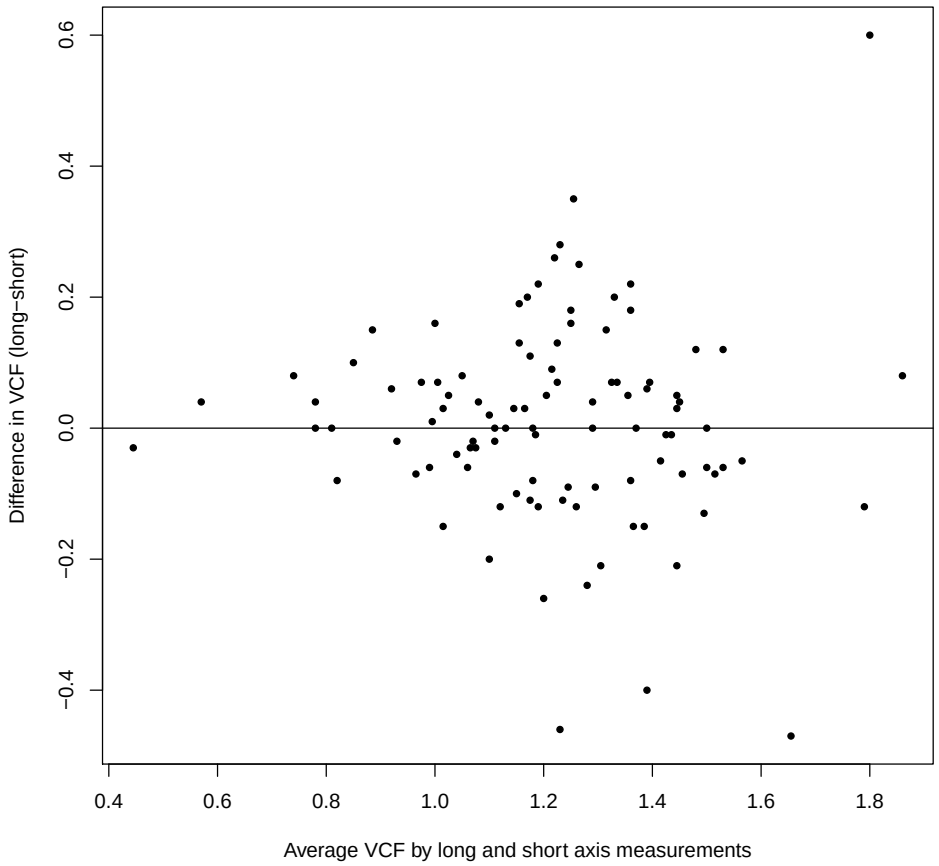

FIGURE 2.1: Bland-Altman plot: Mean VCF by long and short axes measurements, over the difference in the measurements.

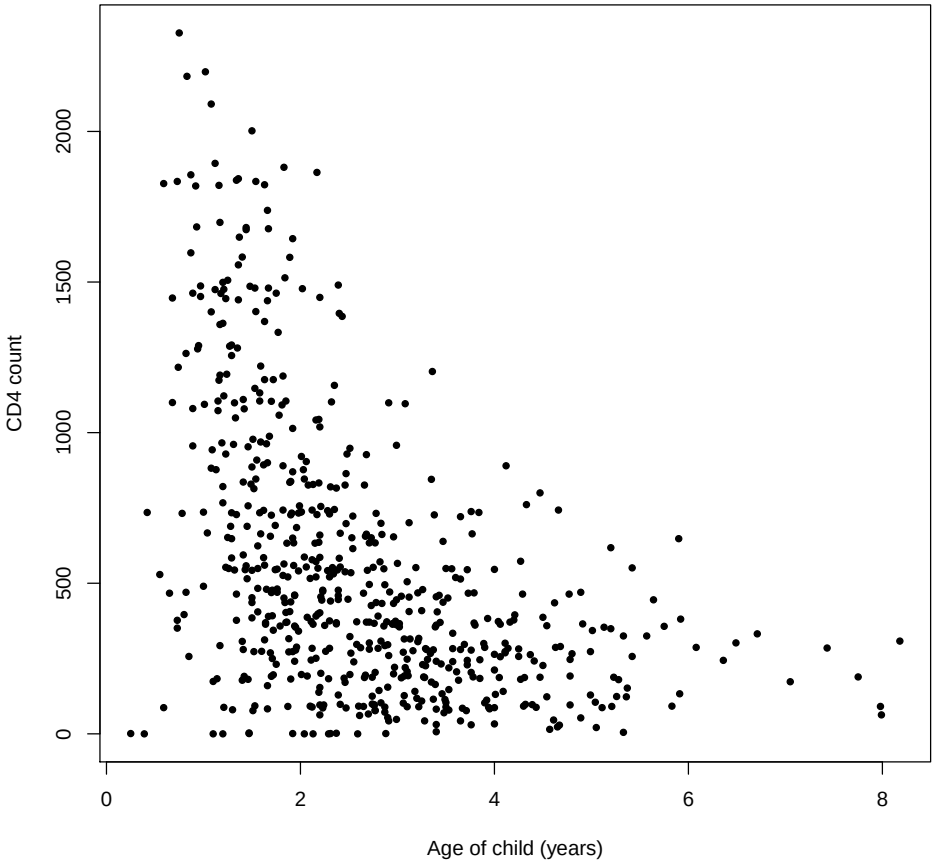

Figure 2.2: Plot of children's CD4 counts over age.

2.3 Existing methods

Variance regression models have previously been developed, but they typically assume a multiplicative model for the dependence of the error variance on the covariates (Aitkin, 1987; Smyth, 2002; Verbyla, 1993). These models generally use adaptations of Newton-type algorithms, which can struggle with additive variance regression models. In this section we will cover some of the more popular methodologies for fitting variance regression models; however, the methodology in this area is extensive.

In a simple linear variance scenario, Bland and Altman (1999) propose to model the variability in the standard deviation using a method based on the absolute residuals from a fitted regression line. Davidian and Carroll (1987) also speak of using methods based on residuals, but note that one should weight the residuals based on the variance

and iterate the process. They showed that unweighted least squares residuals yield unstable estimates of the variance function when the variance depends on the mean. They reviewed various methods of weighting the residuals including squared residuals, logarithms and absolute residuals. They then found that the latter were more robust to slight deviations from normality, and also observed that how well one models and estimates the variance will substantially impact the estimation of the mean.

Aitkin (1987) assumed that the mean and the variance depended upon the explanatory variables through parametric linear models. While parametric models may apply to some data, non-parametric and semi-parametric models provide a more flexible approach. Cole and Green (1992) use a maximum penalised likelihood approach, in which they estimate the Box-Cox power, the median and the coefficient of variation for centile reference curves. The key assumption for this model is that after a suitable power transformation, the data are normally distributed. The Mean And Dispersion Additive Model (MADAM) was introduced in Rigby and Stasinopoulos (1996) as an alternative method for fitting parametric, semi-parametric or non-parametric models for both the mean and variance. They utilise a successive relaxation algorithm for fitting the model. Rigby and Stasinopoulos (2005) later built upon their MADAM model and introduced the Generalised Additive Models for Location, Scale and Shape (GAMLSS) which has been built into the extensive R (R Core Team, 2013) package `gamlss` (Stasinopoulos and Rigby, 2007). This package provides a very general framework for fitting regression type models, allowing all the parameters of the distribution of the response variable to be modelled as linear, non-linear or smooth functions of the explanatory variables. GAMLSS was introduced to overcome some of the problems with generalised linear models (GLM) and generalised additive models (GAM) (Hastie and Tibshirani, 1990). The assumption for the response variable to be from the exponential family is relaxed and replaced by a more general distribution family, including highly skewed or kurtotic continuous and discrete distributions. The model allows any parameters of the distribution of the response variable to be modelled very flexibly as functions of the explanatory variables. A comprehensive review of the methods and software has recently been published (Stasinopoulos et al., 2017).

More recently, Bayesian approaches to regression in the variance and other parameters have been developed. They are based upon variational approximations (Menictas and Wand, 2015; Ormerod and Wand, 2010) and distributional regression (Klein et al., 2015; Klein et al., 2015), and has prompted discussion on the Bayesian approach. Much of this is centred upon the relative speed (Wand et al., 2011) and ease of implementation (Kneib, 2016), as opposed to the interpretation and philosophical distinctions between Bayesian and Frequentist approaches. Nonetheless, some people still prefer a Frequentist inference. We will be focusing on a Frequentist framework in this thesis, particularly to provide a stable and reliable method for variance regression. Currently, the GAMLSS framework provides the most flexible and popular approach for variance regression, and it will be utilised as a comparator method for evaluating the performance of the methods developed in this thesis.

2.4 Multiplicative versus additive models

Generally, the variance regression model (Aitkin, 1987; Verbyla, 1993; Smyth, 2002) uses a log-link model, $h(z) = \log(z)$, and thus the covariates affect the variance multiplicatively. This is primarily because covariate effects in variance heterogeneity models can be negative as well as positive, and the use of a multiplicative model ensures that the overall error variance remains non-negative. Since this is simply a computational convenience, it may be that additive variance heterogeneity is more appropriate in some contexts. Indeed, additive decomposition of the variance is standard in other contexts, such as variance components models and genome-wide association studies.

Additive variance can be achieved with the use of the identity link, rather than the log link. While additive models may be preferable in some contexts, they do have some complexities compared to multiplicative models. The maximum likelihood estimator (MLE) of the additive model will require constrained optimisation due to the non-negativity constraints on the variance. Also, it is important to remember that the MLE may be non-stationary, thus Newton-type algorithms that are searching for a stationary point may be problematic. Complexities with the additive variance model were discussed in some detail in Crisp and Burridge (1994), including the fact that the likelihood function may be unbounded.

An attractive property of additive variance models is that spline models are effectively additive models, so semi-parametric models are obtained with little additional effort.

2.5 Complexities with additive models

While additive models can in principle be handled within the GAMLSS framework, in practice there remain a number of problems. Two such models are presented below to demonstrate these problems using the `gamlss` package within R.

Example 1 is a simulated dataset with zero mean and linearly increasing variance (Figure 2.3). Note that from Figure 2.4, the data have a non-stationary MLE (a maximum with a non-zero derivative), with the shaded regions denoting values outside the parameter space. If the `gamlss` package is used to fit a model with zero mean and variance depending linearly on the covariate x, the algorithm does not converge and an error message is displayed, noting that an iteration of the MLE has moved out of the parameter space. The `gamlss` package allows the implementation of various algorithm methods, as well as various step-lengths, in order to achieve convergence within the data (Stasinopoulos and Rigby, 2007). While the use of step-size reduction did not achieve convergence with this example dataset, convergence was achieved with the implementation of step-halving. However, at times step-halving can achieve convergence at a suboptimal point (Lumley, Kronmal, and Ma, 2006).

Example 2 is also a simulated example, with zero mean and increasing variance (Figure 2.5). This dataset has a stationary MLE solution (Figure 2.6), however the same problem as above was observed with this dataset, with an iteration of the MLE leaving the parameter space. The implementation of step-halving or step-size reduction did not achieve convergence, even with various sizes used. Although it is perhaps understandable that the standard algorithms may struggle with the non-stationary MLE depicted in Figure 2.4, it is somewhat surprising that the stationary MLE depicted in Figure 2.6 can cause problems. These simple examples illustrate the usefulness of having a range of computational methods available in variance regression, and provide motivation for new approaches studied in this thesis.

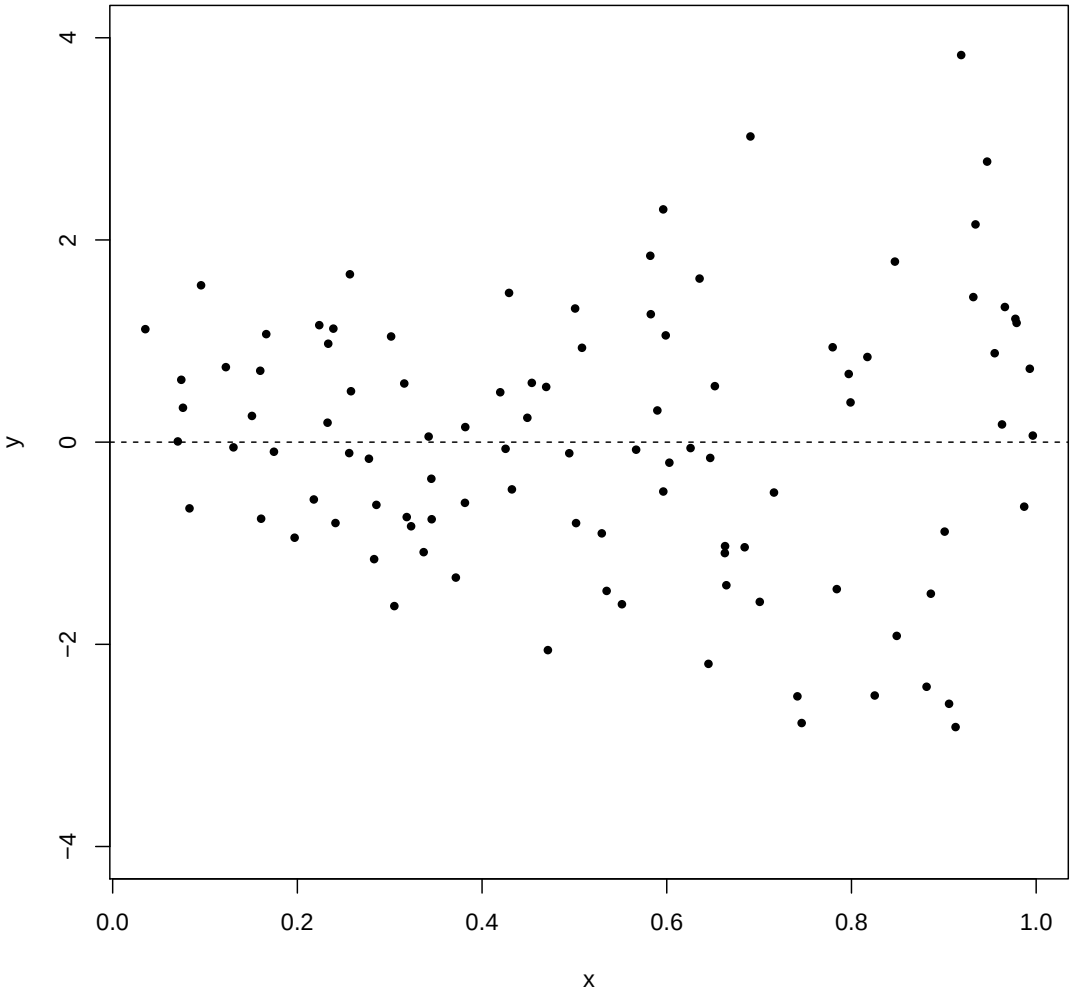

Figure 2.3: Example 1 simulated dataset: zero mean and linearly increasing variance.

These datasets will be covered again in Section 4.3 using the methodology developed in this thesis.

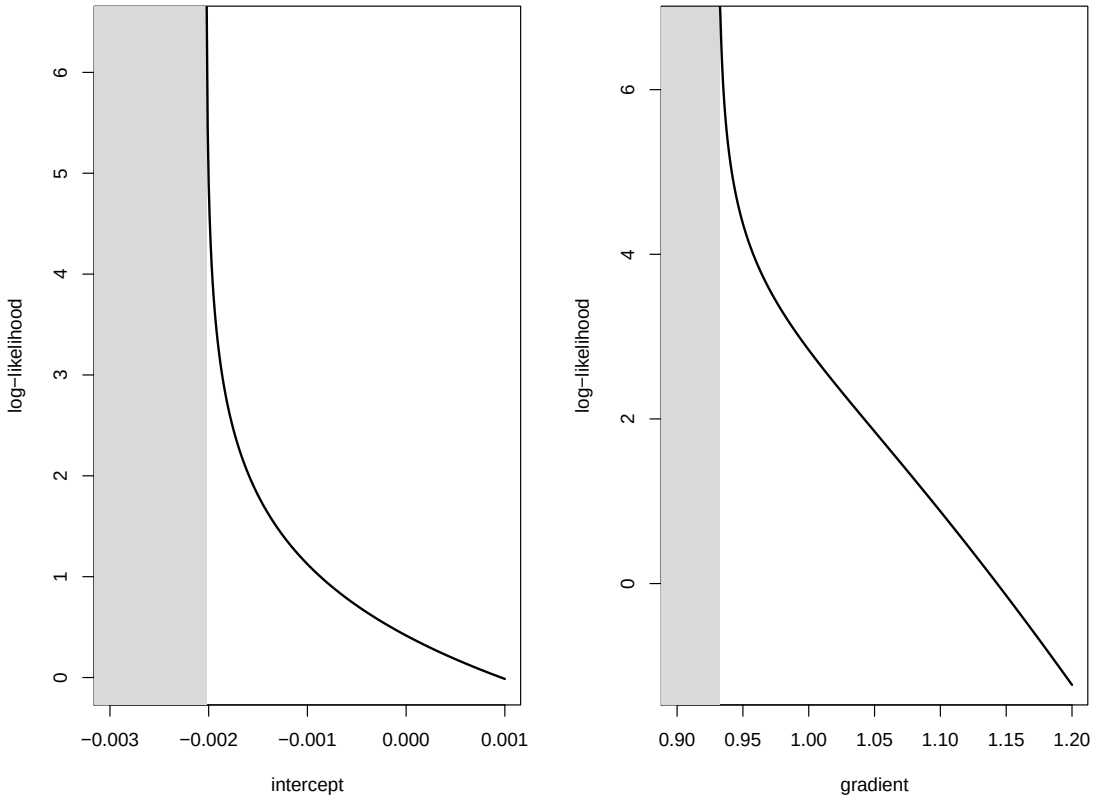

FIGURE 2.4: Profiles of the log-likelihood for Example 1 depicting a non-stationary MLE on the boundary of the parameter space. The shaded region denotes areas outside the parameter space.

2.6 Other complexities

This chapter has introduced variance regression, why it is important, and existing methods that are available to fit these models. We have also established some complexities with such models which may cause existing methods to fail.

There are various other complexities that can be explored in the context of variance regression. For example, biomarker data can often have values above or below a detectable level. This is called censored data, and at times large proportions of the samples may fall into the undetectable range. Another issue similar to this is truncation. Truncation is when the sample taken is restricted to lie between certain values, for example, in a clinical trial, patients may need to have certain cholesterol levels to be eligible for entry into the study.

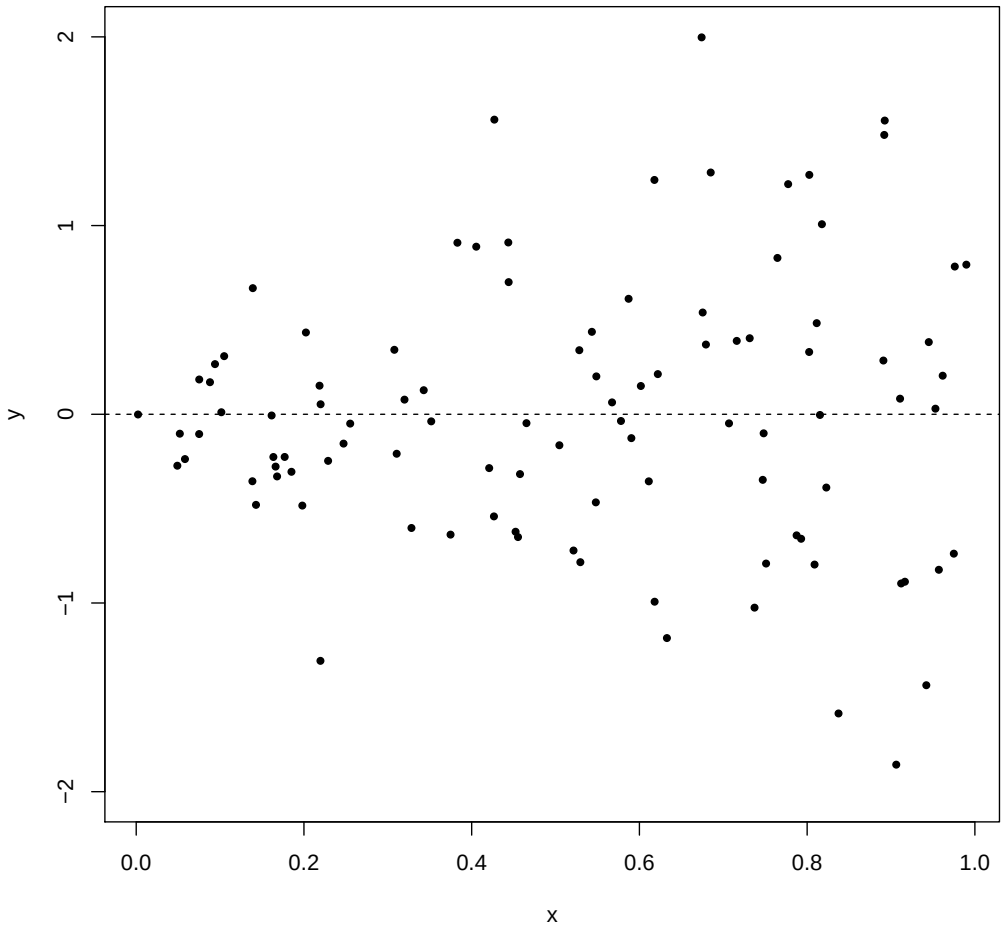

Figure 2.5: Example 2 simulated dataset: zero mean and linearly increasing variance.

Another common problem is that the variance is not linear over the covariates, so that non-linear regression functions are needed. Non-normality may also be a problem, particularly due to the presence of skewness in the data.

Lastly is the issue of monotonicity constraints, where it may be known that the variance is either increasing or decreasing, and it is not appropriate to assume otherwise.

This thesis will consider each of these complexities in turn, beginning with a basic approach that will be progressively generalised throughout the thesis. The main motivation is that new computational and model fitting methodology is of interest for complex models, where existing methods may encounter the sorts of problems discussed in this chapter.

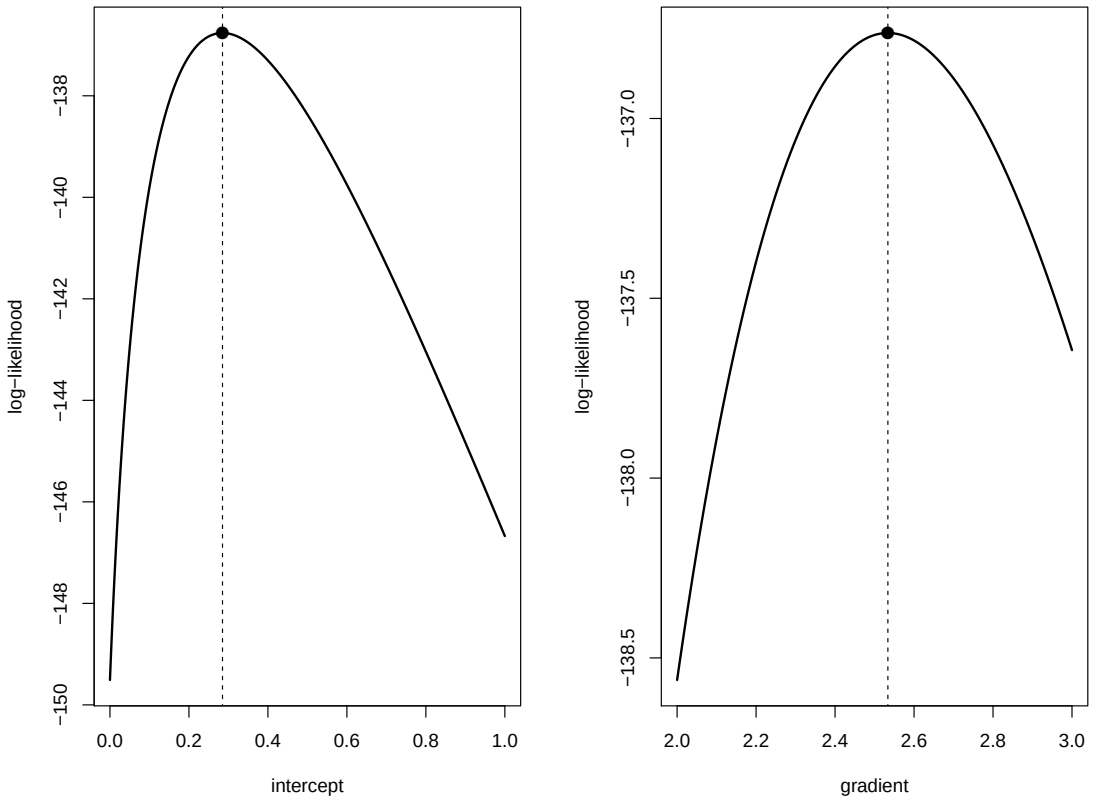

FIGURE 2.6: Profiles of the log-likelihood for Example 2 depicting a stationary MLE.

3

Overview of methods and datasets

In this chapter, we will provide some methodological background for the techniques developed subsequently in this thesis. We will explore the Expectation-Maximisation (EM) algorithm, and provide the computational basis for the algorithm that will be implemented in this thesis. Additionally, semi-parametric methods that will be used in subsequent chapters will be introduced, various criteria that will be used later to select the most appropriate model will be described, and the example datasets that feature in this thesis will be explained.

3.1 Computational methods

The Expectation-Maximisation (EM) algorithm is a method for performing maximum likelihood estimation when there is incomplete data (Dempster, Laird, and Rubin, 1977). We begin with a set of initial parameter estimates, and the algorithm uses these to 'fill in' the missing observations. It then updates the parameter estimates by

maximising the likelihood based on the hypothetical complete data. This constitutes one iteration, and each iteration increases the likelihood of the observed data monotonically. The process continues iterating until convergence, which is defined specifically below. As the EM algorithm is monotonic in the likelihood, it is therefore very stable and often overcomes problems with non-stationary MLEs, as mentioned previously in Section 2.5. There are a large number of variations and generalisations of the EM algorithm (Dempster, Laird, and Rubin, 1977; McLachlan and Krishnan, 2007). Here we will introduce the basic EM algorithm, and a variation used extensively throughout this thesis. Other variations will also be discussed later in this thesis.

3.1.1 EM algorithm

Let $\boldsymbol{\theta}$ be a vector of the parameters being estimated. For example, in the model presented in (2.2), $\boldsymbol{\theta} = (\beta_0, \beta_1, ..., \beta_P, \alpha_0, \alpha_1, ..., \alpha_Q)$. The concept of 'incomplete' data in the context of an EM algorithm refers to when we have an observed data vector $\boldsymbol{X}$, that is associated with a complete data vector $\boldsymbol{\mathcal{X}}$, through a many-to-one mapping from the sample space associated with $\boldsymbol{\mathcal{X}}$ to the sample space associated with $\boldsymbol{X}$. The complete data $\boldsymbol{\mathcal{X}}$ is only measured indirectly through $\boldsymbol{X}$.

Based on the log-likelihood $\ell(\boldsymbol{\theta}; \boldsymbol{X})$ for the observed data, we wish to find the MLE of $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. The EM algorithm is particularly useful in situations where the maximum likelihood estimation would be straightforward if we could maximise the complete data log-likelihood $L(\boldsymbol{\theta}; \boldsymbol{\mathcal{X}})$, if the complete data $\boldsymbol{\mathcal{X}}$ were available. The basic EM algorithm is made up of alternating expectation and maximisation steps (E- and M- steps), which iterate until convergence. Given the initial estimates, $\hat{\boldsymbol{\theta}}^{(0)}$, the E-step at the $(c+1)^{th}$ iteration requires calculation of

$$Q\left(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(c)}\right) = \mathbb{E}\left(L\left(\boldsymbol{\theta}; \boldsymbol{\mathcal{X}}\right) | \boldsymbol{X}, \hat{\boldsymbol{\theta}}^{(c)}\right).$$

The M-step then involves maximisation of Q with respect to $\boldsymbol{\theta}$, so the updated parameter estimate is

$$\hat{\boldsymbol{\theta}}^{(c+1)} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} Q\left(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(c)}\right)$$

A basic property of the EM algorithm is that it ensures that the likelihood will never decrease between iterations (McLachlan and Krishnan, 2007). As the M-step only considers estimates within the parameter space, this algorithm cannot iterate to estimates outside the parameter space. That is, for example, if the parameter space is positive then starting with positive estimates will guarantee that the updated parameter estimates remain positive. This makes this algorithm a useful approach in the context of variance regression, as estimates corresponding to negative variance are not possible.

The EM algorithm as described above updates $\hat{\boldsymbol{\theta}}^{(c)}$ at each iteration, to produce $\hat{\boldsymbol{\theta}}^{(c+1)}$ in the next iteration. A typical way of defining convergence is to use

$$\frac{||\hat{\boldsymbol{\theta}}^{(c+1)} - \hat{\boldsymbol{\theta}}^{(c)}||}{||\hat{\boldsymbol{\theta}}^{(c)}||} < \epsilon$$

where ϵ is some small constant, such as 10^{-6}. This is the definition of convergence that will be used in all algorithms described in this thesis.

The EM algorithm lends itself to situations where the outcome variable can be thought of as a function of a collection of unobserved latent outcome variables. In an EM algorithm, these underlying outcome variables can be thought of as missing data. This is how the EM algorithm will be utilised in this thesis.

3.1.2 Combinatorial EM algorithms

The basic concept behind the combinatorial EM (CEM) algorithm is explained in Marschner (2014). For our observed data $\boldsymbol{X}$, with log-likelihood $\ell(\boldsymbol{\theta}; \boldsymbol{X})$, let the parameter vector $\boldsymbol{\theta}$ lie in the parameter space $\boldsymbol{\Theta}$. For example, in the variance regression model, a natural complete data model has all coefficients non-negative, whereas $\boldsymbol{\Theta}$ has only the overall variances non-negative. In particular, for some finite set τ, we have $t \in \tau$ complete data models with log-likelihoods $L(\boldsymbol{\theta}(t); \boldsymbol{X})$ for $\boldsymbol{\theta} \in \Theta(t)$. The parameter space $\Theta(t)$ need not coincide with the parameter space for the observed model $\boldsymbol{\Theta}$, but it is assumed the collection of parameter spaces $\mathcal{T} = \{\Theta(t);\ t \in \mathcal{T}\}$ covers $\boldsymbol{\Theta}$ exactly. If a finite collection of such complete data models can be defined which together have parameter spaces that cover $\boldsymbol{\Theta}$ then this defines a collection of EM algorithms.

In the EM algorithm detailed in above in Section 3.1.1, our estimate $\hat{\boldsymbol{\theta}}^{(c)} \in \Theta(t)$ is

updated to estimate $\hat{\boldsymbol{\theta}}^{(c+1)} \in \Theta(t)$, such that $\ell(\hat{\boldsymbol{\theta}}^{(c+1)}) \geq \ell(\hat{\boldsymbol{\theta}}^{(c)})$. The E-step is the calculation of the conditional expectation

$$Q_t\left(\boldsymbol{\theta}(t) \mid \hat{\boldsymbol{\theta}}^{(c)}\right) = \mathbb{E}\left(L\left(\boldsymbol{\theta}(t); \boldsymbol{\mathcal{X}}\right) | \boldsymbol{X}, \hat{\boldsymbol{\theta}}^{(c)}\right) \quad t \in \tau. \tag{3.1}$$

The M-step for this tth EM algorithm is maximising

$$\hat{\boldsymbol{\theta}}^{(c+1)} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}(t)} Q_t\left(\boldsymbol{\theta}(t) \mid \hat{\boldsymbol{\theta}}^{(c)}\right) \quad t \in \tau, \tag{3.2}$$

which is over the parameter space of the tth complete data model $\Theta(t)$, not the observed data model $\boldsymbol{\Theta}$. Our family of EM algorithms specified in (3.1) and (3.2) give a collection of constrained maxima $\boldsymbol{T} = \{\theta(t); t \in \tau\}$, where

$$\hat{\boldsymbol{\theta}}(t) = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}(t)} \ell(\boldsymbol{\theta}(t)) \quad t \in \tau. \tag{3.3}$$

It then follows that the MLE $\hat{\boldsymbol{\theta}}$ is within $\boldsymbol{T}$, as $\hat{\boldsymbol{\theta}}$ must coincide with at least one of the constrained maxima $\hat{\boldsymbol{\theta}}(t)$ in (3.3). $\hat{\boldsymbol{\theta}}$ is then determined by finding the element in $\boldsymbol{T}$ which corresponds to the greatest $\ell(\hat{\boldsymbol{\theta}}(t))$.

This family of EM algorithms is the key to the CEM algorithm and the manner in which the parameter space is partitioned for the variance regression model, as we will see in Chapters 4, 5 and 7.

3.2 Semi-parametric methods

Semi-parametric models allow the inclusion of both parametric and nonparametric components (Ruppert, Wand, and Carroll, 2003). Semi-parametric regression models allow the dependence on a covariate x to be non-linear through some regression function $f(x)$. Spline models are a common way to implement semi-parametric regression.

There are many methods available to produce regression splines (De Boor, 1978), however the use of B-splines (or Basis splines) are computationally convenient in the regression spline approach and are widely used (Hastie and Tibshirani, 1990; Ruppert, Wand, and Carroll, 2003). A flexible alternative method is to consider fitting a step function to the data, which is also a useful model checking exercise.

These semi-parametric models are easily incorporated into our additive variance regression model, as mentioned in Section 2.4. Semi-parametric models will be explored later in this thesis in Chapter 6, and here we review some of the methodological background.

3.2.1 B-splines

B-splines are a family of polynomial splines constructed from the B-spline basis functions (De Boor, 1978). A series of polynomials joined end-to-end at a series of q fixed turning points $\xi_1 < .... < \xi_q$ make up a polynomial spline, where $\xi_1 = x_{min}$ and $\xi_q = x_{max}$ are the endpoints of range of the continuous covariate x. The spline is a polynomial of order k (or degree d, where $d = k-1$) between any two adjacent turning points.

If a sequence of knots t_j are placed equidistantly within the range, with a total of s internal knots, then for any given set of knots, the B-spline is unique. These B-splines with equidistant knots are called cardinal B-splines, and this is how knots will be determined in this thesis (see Section 3.2.3 for more information). The specification of the spline is made with the determination of the knot sequence $t_1 \leq ... \leq t_{M+k}$, where $M = k + q - 2$ is the number of free parameters that are required to determine the spline. We begin the knot sequence by placing k knots at the start (and end) of the sequence, $t_1 = ... = t_k = \xi_1$ and $t_{M+1} = ... = t_{M+k} = \xi_q$. We then place each of the s internal knots, starting at $t_{k+1} = \xi_2$ and ending at $t_M = \xi_{q-1}$. Note that we are only considering the most common case where the component polynomials of order k meet up at their $(k-1)^{th}$ derivative, where we only have each internal knot placed once at each of the turning points (Ramsay, 1988).

Given our set of $\boldsymbol{t}$ knots, ranging from $t_1, t_2, ..., t_{M+k}$, the M B-spline basis functions of order k for covariate x can be defined recursively as

$$B_m(x|k) = \begin{cases} 1, & \text{if } x \in [t_m, t_{m+1}) \\ 0, & \text{otherwise} \end{cases}$$

for $k = 1$ and

$$B_m(x|k) = \frac{x - t_m}{t_{m+k-1} - t_m} B_m(x|k-1) + \frac{t_{m+k} - x}{t_{m+k} - t_{m+1}} B_{m+1}(x|k-1)$$

for $k > 1$, where $M = k + q - 2$ is the number of basis functions. Note that if $t_{m+k-1} = t_m$ then $B_m(x|k-1) = 0$ for all x. As an example, for a sequence of values x, bases were computed of order $k = 3$ with two internal knots ($s = 2$), using the `bs()` function within the `splines` package (R Core Team, 2013). The knot sequence will be $t_1 \leq ... \leq t_8$, and given that x is a sequence ranging over $[0, 1]$, $\xi_1 = t_1 = t_2 = t_3 = 0$ and $\xi_4 = t_6 = t_7 = t_8 = 1$. The two internal knots will be at $\xi_2 = t_4 = 0.33$ and $\xi_3 = t_5 = 0.66$, as the knots are equally spaced. The B-spline basis functions ($M = 5$) are given in Figure 3.1.

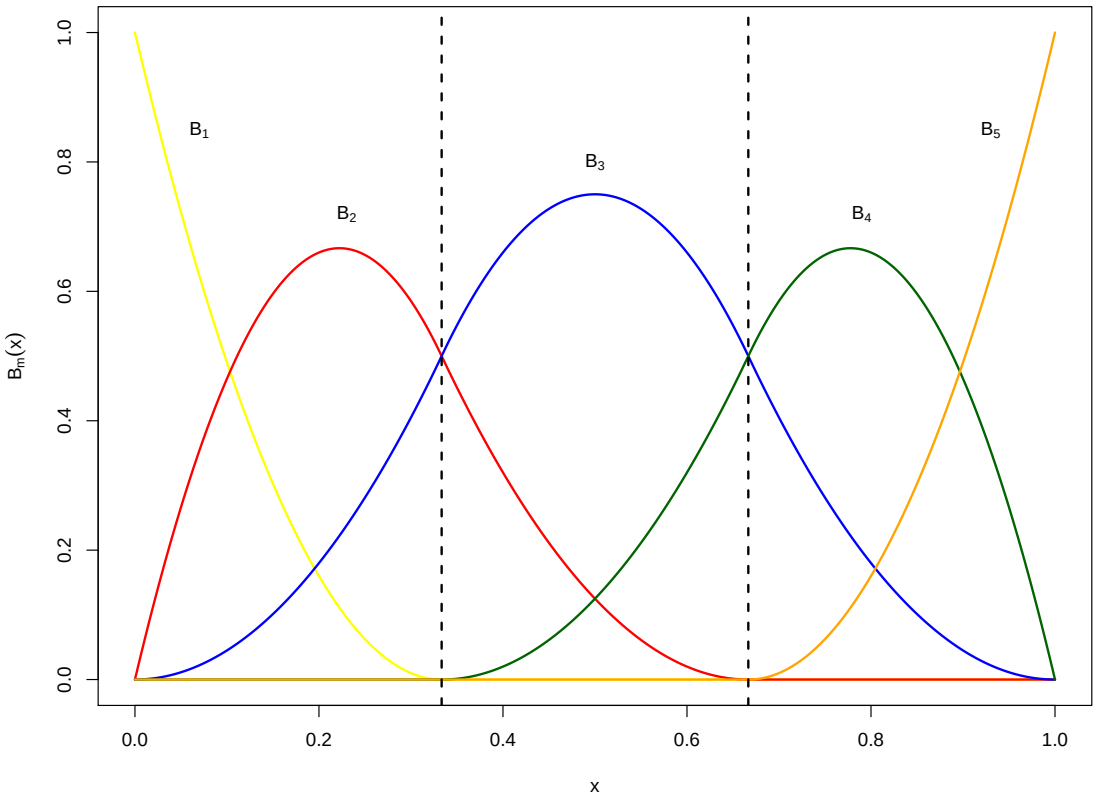

FIGURE 3.1: The M B-spline basis functions for a given x, of order $k = 3$ and with two evenly spaced internal knots ($s = 2$, dashed lines).

B-splines are normalised such that

$$\sum_{m=1}^{M} B_m(x) = 1$$

for all x. The first basis function, $B_1(x)$, is typically called the 'intercept' and with the function `bs`() the default is to not generate the intercept, in order to ensure identifiability of the B-spline coefficients. The regression function based on the B-spline bases can then be expressed as

$$f(x) = \gamma_0 + \sum_{m=2}^{M} \gamma_m B_m(x),$$

with γ_m the parameters to be estimated in the model, and the $B_m(x)$ values treated as the covariate values.

3.2.2 Monotonic splines

Sometimes it may be known that the relationship with a covariate is monotonic, and it is not appropriate to consider otherwise. To create a monotonically increasing spline, monotonic basis functions can be defined by summing the respective B-spline basis functions

$$I_m(x) = \sum_{k=m}^{M} B_k(x) \quad \text{where } m = 2, ..., M.$$

Note that the first basis function $m = 1$ is not included to ensure identifiability, as mentioned previously.

To create a monotonically decreasing spline, it is simply the sum of the respective basis functions

$$I_m(x) = \sum_{k=2}^{m} B_m(x) \quad \text{where } m = 2, ..., M.$$

Ramsay (1988) calls these bases I-splines, as they are obtained by integrating the B-spline bases.

These summed basis functions are then incorporated in our additive model as the covariates in the same manner as above:

$$f(x) = \gamma_0 + \sum_{m=2}^{M} \gamma_m I_m(x).$$

3.2.3 Knot selection

While B-splines offer many computational advantages, the difficulty of choosing the number and the position of the knots is one of their main drawbacks (Ruppert, Wand, and Carroll, 2003). However, Ramsay (1988) noted that the shape of the spline function is not very sensitive to the knot placement, and recommended that a useful preliminary knot placement is to use the median for a single interior knot, or tertiles for two etc., as required. This is how the knots will be determined in this thesis.

Information criteria can be used to then compare the models with differing numbers of knots, in order to find the optimal number of knots. These criteria do not require nested models, so the placement of the turning points is flexible. There are numerous criteria that can be used, but generally they measure the quality of the statistical model being fit to the data, relative to other models. For all of the criteria below, lower scores indicate models of better fit.

Akaike Information criterion

This is one of the most widely used information criteria to compare models that are not nested (Akaike, 1974). The Akaike Information criterion (AIC) rewards the goodness of fit as determined by the log-likelihood, however, it includes a penalty for the number of parameters being estimated. If k is the number of parameters in the model and l is the likelihood function, then the AIC is

$$AIC = -2\log(l) + 2k.$$

The optimum number of knots is then determined by minimising the AIC.

Akaike Information criterion (corrected)

The corrected Akaike Information criterion, or AICc, is the before mentioned AIC with a correction for small sample sizes (Sugiura, 1978). If n is the sample size, then

$$AICc = -2\log(l) + \frac{2kn}{n-k-1}.$$

Note that as n gets larger, the AICc converges to the AIC.

Bayesian Information criterion

The Bayesian Information criterion (BIC) has a larger penalty for the number of parameters than that used in the AIC (Schwarz, 1978). Assuming n is the sample size, l is the likelihood function and k is the number of parameters in the model,

$$BIC = -2\log(l) + k\log n.$$

Hannan-Quinn Information criterion

The Hannan-Quinn Information criterion (HQC) is an alternative to the BIC and AIC (Hannan and Quinn, 1979). This criterion imposes a smaller penalty to added parameters compared to the BIC, with

$$HQC = -2\log(l) + 2k\log\log n$$

These criteria detailed above provide a way to rank different models. It is important to note that using one particular criterion may give a different model as the most preferred model, compared to a different criterion. In subsequent chapters we will explore the use of these criteria for variance regression models.

3.3 Datasets

This section will provide a background for the various datasets that will be used to illustrate methods in this thesis. Several of these datasets contain variables that may be referred to as biomarkers. As defined by the Biomarkers Definitions Working Group

(2001), a biomarker is 'a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention'. So, common measures such as pulse or cholesterol, through to an inflammation marker such as C-reactive protein, can be called a biomarker. Biomarkers are increasingly used by clinicians in a variety of situations, from measuring disease progress, evaluating the most effective treatment for a disease, or establishing susceptibility to disease or its recurrence. The following biomarker datasets will be used to illustrate the methods that will be developed throughout this thesis.

3.3.1 VCF dataset

The VCF dataset has been used as a motivating example of measurement error in Section 2.2. It is a dataset of 100 measurements comparing two methods of measuring the mean velocity of circumferential fibre shortening (VCF), which is given in centimetres per second. VCF is a measure of the strength of the contraction of the left ventricle, when only a two-dimensional image is available.

This dataset contains two measurements of VCF, where the first measurement is taken by the long axis and the second by the short axis, in M-mode echocardiography (Darbela, Silayan, and Bland, 1986). A plot of the difference between the two measurements (long-short) and the average of the two measurements is shown in Figure 2.1.

3.3.2 CD4 dataset

CD4 is a type of white blood cell, and in this dataset, it has been measured in uninfected children born from HIV-1 infected women (Wade and Ades, 1994). The dataset has been used as a motivating example in Section 2.2 and contains 609 measurements of CD4 cell counts and the child's age at which the measurements were taken (Figure 2.2). In these data, it is clear that at younger ages there is more variation in the CD4 counts than at older ages, demonstrating heteroscedasticity. The dataset is stored within the `gamlss` package in R (Rigby and Stasinopoulos, 2005).

3.3.3 Viral load dataset

This is a dataset of the HIV viral load (blood concentration of HIV RNA on a $\log_{10}$ scale) in 285 participants. Prior to commencing a clinical trial, participants had their blood assayed twice during a short period of time (Kuritzkes et al., 1999). Although the underlying viral load is unchanged in this time, the readings will differ due to measurement error. Another important aspect is that measurements cannot be detected below a particular assay limit, in this case, 2.70 ($\log_{10}500$). In Figure 3.2, values below the limit have been set to the limit and the effect of censoring can be clearly seen at the left-hand side.

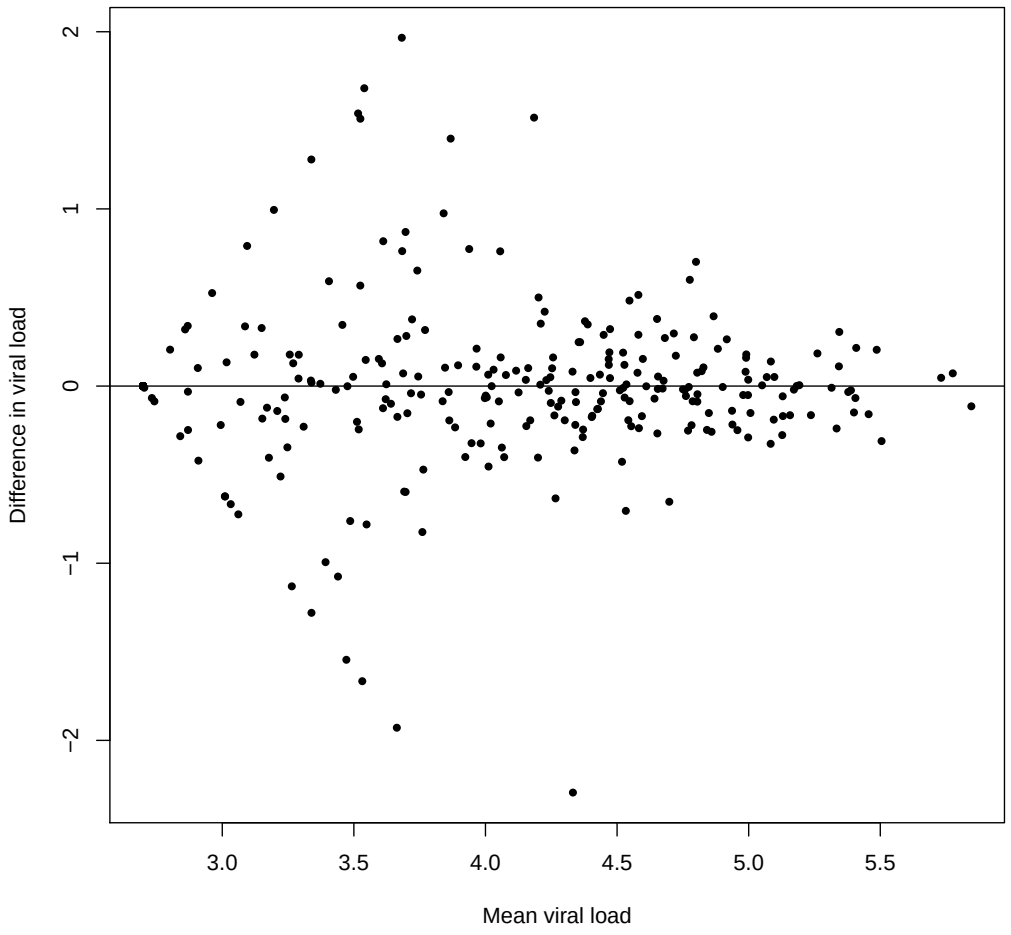

Figure 3.2: Bland-Altman plot of the RNA dataset: Plot of average viral load by difference in viral load, with the solid line depicting zero difference.

3.3.4 LIPID dataset

The Long-term Intervention with Pravastatin in Ischemic Disease (LIPID) study was a multi-centre randomised double-blind placebo-controlled trial that recruited 9014 patients with a history of myocardial infarction or unstable angina. The study found strong evidence that pravastatin (a cholesterol lowering medication) reduced the risk of death from coronary heart disease, cardiovascular disease, and all causes combined (The Long-Term Intervention with Pravastatin in Ischaemic Disease Study Group, 1998).

The dataset used in this thesis contains the 4502 patients who were allocated to the placebo treatment. The study measured various biomarkers at randomisation associated with coronary heart disease and cardiovascular disease, and a subset of these biomarkers in the placebo group will be used to demonstrate the `VarReg` package later in this thesis. The biomarkers to be investigated are lipoprotein-associated phospholipase A2 activity (LP-PLA2 activity) and Lipoprotein (a) (Lp(a)), and these are to be associated with LDL cholesterol. LP-PLA2 activity is associated with the presence of unstable plaque in arteries that are likely to break apart or rupture. Most heart attacks are caused by ruptured plaque or clots that cause blockages in the arteries that supply blood to the heart. Lp(a) has been shown to promote the uptake of LDL into blood vessel walls. It also may promote the development of plaque on the walls of blood vessels and the accumulation of clots in the arteries. Thus, the relationship between these biomarkers and LDL cholesterol is of interest.

3.3.5 Classic datasets

In addition to the biomarker datasets described above, a number of classic datasets from the literature are useful for illustrating methods to be developed in this thesis. Two such datasets are the so-called motorcycle crash dataset and the LIDAR dataset.

The LIDAR dataset contains 221 observations from a light detection and ranging (LIDAR) experiment. The range is the distance travelled before the light is reflected back to its source, and the log(ratio) is the logarithm of the ratio of the received light from the two laser sources (Sigrist, 1994). The LIDAR data is used in the textbook by Ruppert, Wand, and Carroll (2003) to illustrate heteroscedasticity, and Figure 3.3 is

in fact on the book's cover. The dataset is also stored within the `SemiPar` package in R (Wand, 2014).

The motorcycle crash dataset is a simulated dataset of a series of 133 measurements of head acceleration in a motorcycle accident involving crash dummies. The time in milliseconds is recorded, along with the acceleration of the head measured in gravitational force (g units). These data shown in Figure 3.4 also demonstrate heteroscedasticity. This dataset is stored within the `MASS` package in R (Venables and Ripley, 2002).

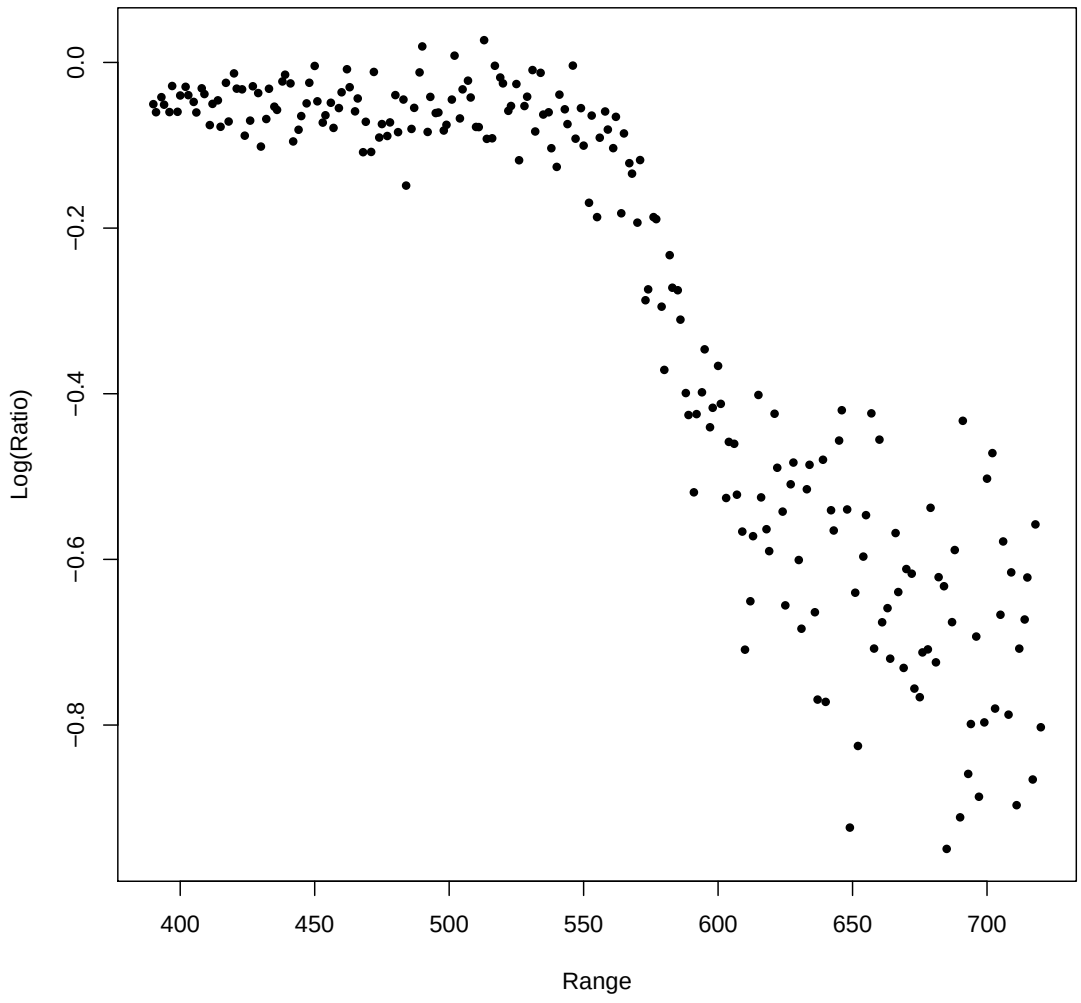

Figure 3.3: LIDAR dataset: Plot of log(ratio) of reflected light by range travelled.

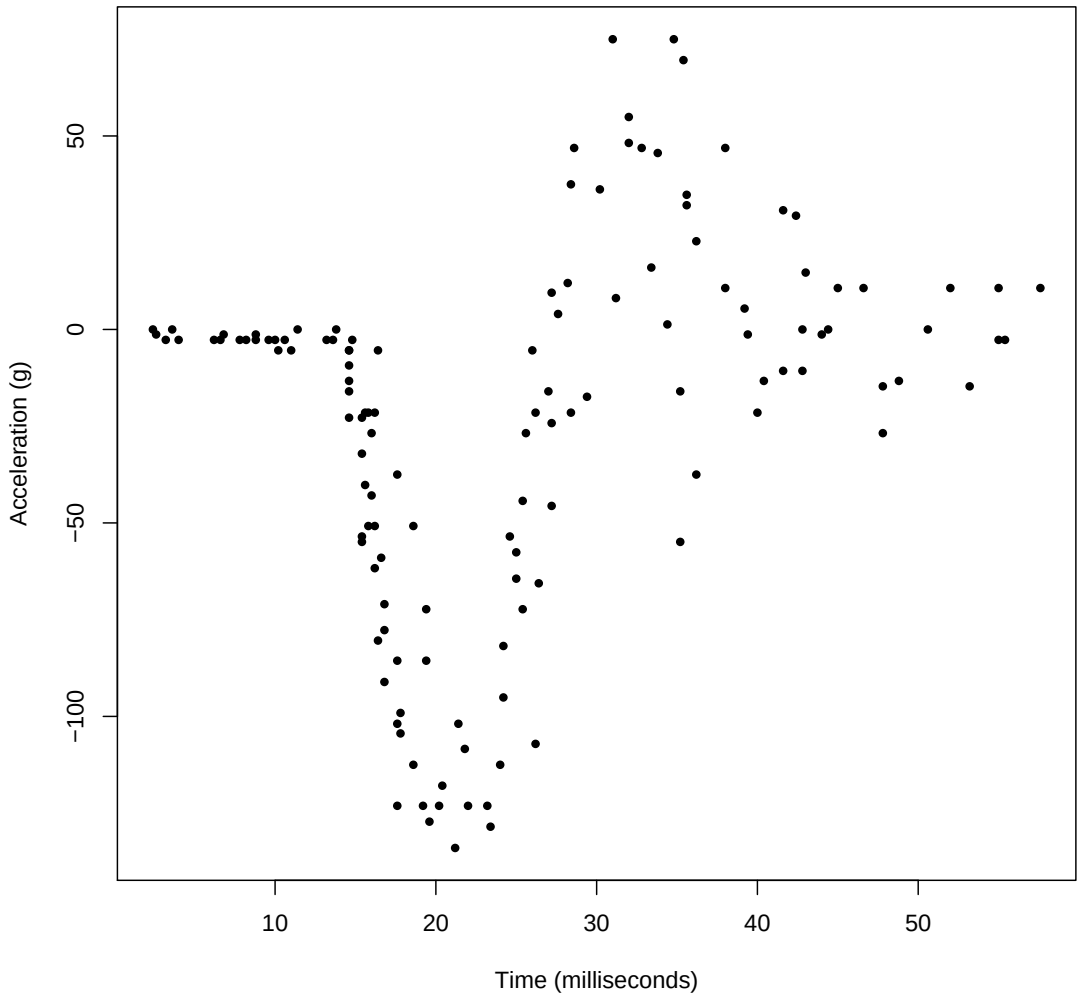

FIGURE 3.4: Motorcycle dataset: Plot of head acceleration over time in a motorcycle accident.

4

Basic method

In the previous chapters, we introduced the concept of modelling both the mean and the variance using a regression model, and the computational method of the EM algorithm. In this chapter, we present a simple special case to illustrate the basic EM algorithm that will be used and adapted throughout this thesis.

Suppose that the outcome variable X_i has two covariate vectors associated with it, with the mean being dependent on z_{ip} $(p = 1, ..., P)$ and the variance being dependent on x_{iq} $(q = 1, ..., Q)$. Note that these could be the same covariates, or different covariates. Then this gives the following general model

$$X_i \sim N\left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}, \quad \alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right) \quad \text{for } i = 1, 2, ..., n.$$

We will return to this model in Chapter 5 and more general models in subsequent chapters, but for now we will consider a simplified version of the model that illustrates the basic method.

4.1 Simplified model

Assume that the mean $\mu = 0$, and that there is just one continuous covariate x_i so that the variance has a simple linear form

$$X_i \sim N(0, \alpha_0 + \alpha_1 x_i). \tag{4.1}$$

Without any loss of generality, we will suppose that x_i has been scaled such that $x_i \in [0, 1]$. We will consider the standard likelihood-based estimation tools and then the EM algorithm that implements maximum likelihood estimation.

The likelihood for the model in (4.1) is

$$l(\alpha_0, \alpha_1) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi(\alpha_0 + \alpha_1 x_i)}} \exp\left(\frac{-X_i^2}{2(\alpha_0 + \alpha_1 x_i)}\right),$$

and the log-likelihood is therefore

$$\ell(\alpha_0, \alpha_1) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log(\alpha_0 + \alpha_1 x_i) - \frac{1}{2}\sum_{i=1}^{n}\frac{X_i^2}{\alpha_0 + \alpha_1 x_i}. \tag{4.2}$$

In order to create the likelihood equations, (4.2) must be differentiated with respect to α_0 and α_1:

$$\frac{\partial}{\partial\alpha_0}\ell(\alpha_0, \alpha_1) = -\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\alpha_0 + \alpha_1 x_i} + \frac{1}{2}\sum_{i=1}^{n}\frac{X_i^2}{(\alpha_0 + \alpha_1 x_i)^2} \tag{4.3}$$

and

$$\frac{\partial}{\partial\alpha_1}\ell(\alpha_0, \alpha_1) = -\frac{1}{2}\sum_{i=1}^{n}\frac{x_i}{\alpha_0 + \alpha_1 x_i} + \frac{1}{2}\sum_{i=1}^{n}\frac{x_i X_i^2}{(\alpha_0 + \alpha_1 x_i)^2}. \tag{4.4}$$

Then, in order to obtain the information matrix, we need to differentiate (4.3) and (4.4) each with respect to α_0 and α_1 in order to obtain our 2×2 matrix. This gives

$$
\begin{aligned}
I_{11} = -\frac{\partial^2}{\partial \alpha_0^2}\ell(\alpha_0, \alpha_1) \quad &= -\left(\frac{1}{2}\sum_{i=1}^{n}\frac{1}{(\alpha_0+\alpha_1 x_i)^2} - \sum_{i=1}^{n}\frac{X_i^2}{(\alpha_0+\alpha_1 x_i)^3}\right) \\
I_{12} = I_{21} = -\frac{\partial^2}{\partial \alpha_0 \alpha_1}\ell(\alpha_0, \alpha_1) &= -\left(\frac{1}{2}\sum_{i=1}^{n}\frac{x_i}{(\alpha_0+\alpha_1 x_i)^2} - \sum_{i=1}^{n}\frac{x_i X_i^2}{(\alpha_0+\alpha_1 x_i)^3}\right) \\
I_{22} = -\frac{\partial^2}{\partial \alpha_1^2}\ell(\alpha_0, \alpha_1) \quad &= -\left(\frac{1}{2}\sum_{i=1}^{n}\frac{x_i^2}{(\alpha_0+\alpha_1 x_i)^2} - \sum_{i=1}^{n}\frac{x_i^2 X_i^2}{(\alpha_0+\alpha_1 x_i)^3}\right),
\end{aligned}
$$

and our expected information matrix is then

$$
I(\boldsymbol{\theta}) = \frac{1}{2}\begin{bmatrix} \sum_{i=1}^{n}\frac{1}{(\alpha_0+\alpha_1 x_i)^2} & \sum_{i=1}^{n}\frac{x_i}{(\alpha_0+\alpha_1 x_i)^2} \\ \sum_{i=1}^{n}\frac{x_i}{(\alpha_0+\alpha_1 x_i)^2} & \sum_{i=1}^{n}\frac{x_i^2}{(\alpha_0+\alpha_1 x_i)^2} \end{bmatrix},
$$

where $\boldsymbol{\theta} = (\alpha_0, \alpha_1)$. Note that (4.3) and (4.4) can be further simplified to give our likelihood equations as

$$
\sum_{i=1}^{n} w_i \left[X_i^2 - (\alpha_0 + \alpha_1 x_i)\right] = 0 \tag{4.5}
$$

and

$$
\sum_{i=1}^{n} x_i w_i \left[X_i^2 - (\alpha_0 + \alpha_1 x_i)\right] = 0, \tag{4.6}
$$

where

$$
w_i = \frac{1}{(\alpha_0+\alpha_1 x_i)^2}.
$$

So, we see that the MLE corresponds to a weighted regression of X_i^2 on x_i, as was discussed in a general context in Davidian and Carroll (1987). As illustrated in Section 2.5, the additive variance model that solves (4.5) and (4.6) can lead to numerical difficulties, stemming from the non-negatively constrained parameter space, which motivates the study of the EM algorithm which offers a more stable approach.

4.2 EM algorithm

Given this simplified model, we now propose a hypothetical complete data model that can be used to motivate an EM algorithm. In particular, suppose that X_i is composed of two independent, unobserved, latent variables with complete data model

$$X_i = Y_i + Z_i$$
$$\text{where} \quad Y_i \sim N(0, \alpha_0) \quad \text{and} \quad Z_i \sim N(0, \alpha_1 x_i).$$

The log-likelihood corresponding to the complete data model is linear in Y_i^2 and Z_i^2,

$$L(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log(\alpha_0) - \frac{1}{2}\sum_{i=1}^{n}\frac{Y_i^2}{\alpha_0}$$
$$-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log(\alpha_1 x_i) - \frac{1}{2}\sum_{i=1}^{n}\frac{Z_i^2}{\alpha_1 x_i},$$

where $\boldsymbol{\theta}$ is a vector of the unknown parameters, $\boldsymbol{\theta} = (\alpha_0, \alpha_1)$.

Notice that under this latent variable model, we retain the observed data model given in (4.1). However, we can use the latent variable model to motivate an EM algorithm. In this case, the E-step is the calculation of the conditional expectations

$$\hat{Y}_i^2(\boldsymbol{\theta}) = \mathbb{E}(Y_i^2 | X_i; \boldsymbol{\theta}) \quad \text{and} \quad \hat{Z}_i^2(\boldsymbol{\theta}) = \mathbb{E}(Z_i^2 | X_i; \boldsymbol{\theta}).$$

These calculations of the conditional expectations may be obtained from the following conditional distributions:

$$Y_i | X_i \sim N\left(\frac{\alpha_0 X_i}{\alpha_0 + \alpha_1 x_i}, \frac{\alpha_0\left[\alpha_0 + \alpha_1 x_i - \alpha_0\right]}{\alpha_0 + \alpha_1 x_i}\right)$$

and

$$Z_i | X_i \sim N\left(\frac{\alpha_1 x_i X_i}{\alpha_0 + \alpha_1 x_i}, \frac{\alpha_1 x_i\left[\alpha_0 + \alpha_1 x_i - \alpha_1 x_i\right]}{\alpha_0 + \alpha_1 x_i}\right).$$

This then leads to

$$\hat{Y}_i^2(\boldsymbol{\theta}) = \alpha_0 + \frac{\alpha_0^2}{\alpha_0 + \alpha_1 x_i}\left(\frac{X_i^2}{\alpha_0 + \alpha_1 x_i} - 1\right)$$

and

$$\hat{Z}_i^2(\boldsymbol{\theta}) = \alpha_1 x_i + \frac{\alpha_1 x_i^2}{\alpha_0 + \alpha_1 x_i}\left(\frac{X_i^2}{\alpha_0 + \alpha_1 x_i} - 1\right).$$

Next, the M-step is the calculation of the updated estimates $\hat{\boldsymbol{\theta}}^{new} = (\hat{\alpha}_0^{new}, \hat{\alpha}_1^{new})$, where

$$\hat{\alpha}_0^{new} = n^{-1}\sum_{i=1}^{n} \hat{Y}_i^2(\hat{\boldsymbol{\theta}}^{old}) \quad \text{and} \quad \hat{\alpha}_1^{new} = n^{-1}\sum_{i=1}^{n} \frac{\hat{Z}_i^2(\boldsymbol{\theta}^{old})}{x_i}.$$

The EM algorithm detailed above will converge to the constrained maximum of the observed data log-likelihood, subject to the constraints $\alpha_0 \geq 0$ and $\alpha_1 \geq 0$. However, α_1 could be negative and this would demonstrate decreasing variance over x_i. In order to search the entire parameter space, we must accommodate both a positive or negative slope. To search the negative slope space, we fit another EM algorithm with covariate $1 - x_i$ in place of x_i. This leads to two $\hat{\theta}$ estimates, one from each implementation of the EM algorithm.

The MLE will then be the $\hat{\boldsymbol{\theta}}$ from the EM algorithm that achieved the highest log-likelihood. This process of using two constrained EM algorithms that together cover the entire parameter space is an instance of a CEM algorithm as described in Section 3.1.2.

4.3 Numerical example

As demonstrated in two data examples in Section 2.5, the `gamlss` package did not converge to the correct MLE, or did not converge at all. When the algorithm described above is applied to these two example datasets, convergence occurred reliably to the MLE. Here we discuss the results.

Example 1 was simulated data where the MLE for the intercept and the gradient are on the boundary of the parameter space, where there is an infinite likelihood. The

results from the EM-based approach are given in Figure 4.1, where we see the path that the EM algorithm takes along the log-likelihood contours, beginning with an initial estimate of $\hat{\boldsymbol{\theta}}_0 = (1, 1)$. The MLE for data in example 2 occurs at an interior stationary point, which the `gamlss` package also had difficulty converging to. The results from the EM-based approach detailed above are shown in Figure 4.2. These numerical examples provide a brief illustration of the stability of the method presented in this chapter in situations where a standard method may fail.

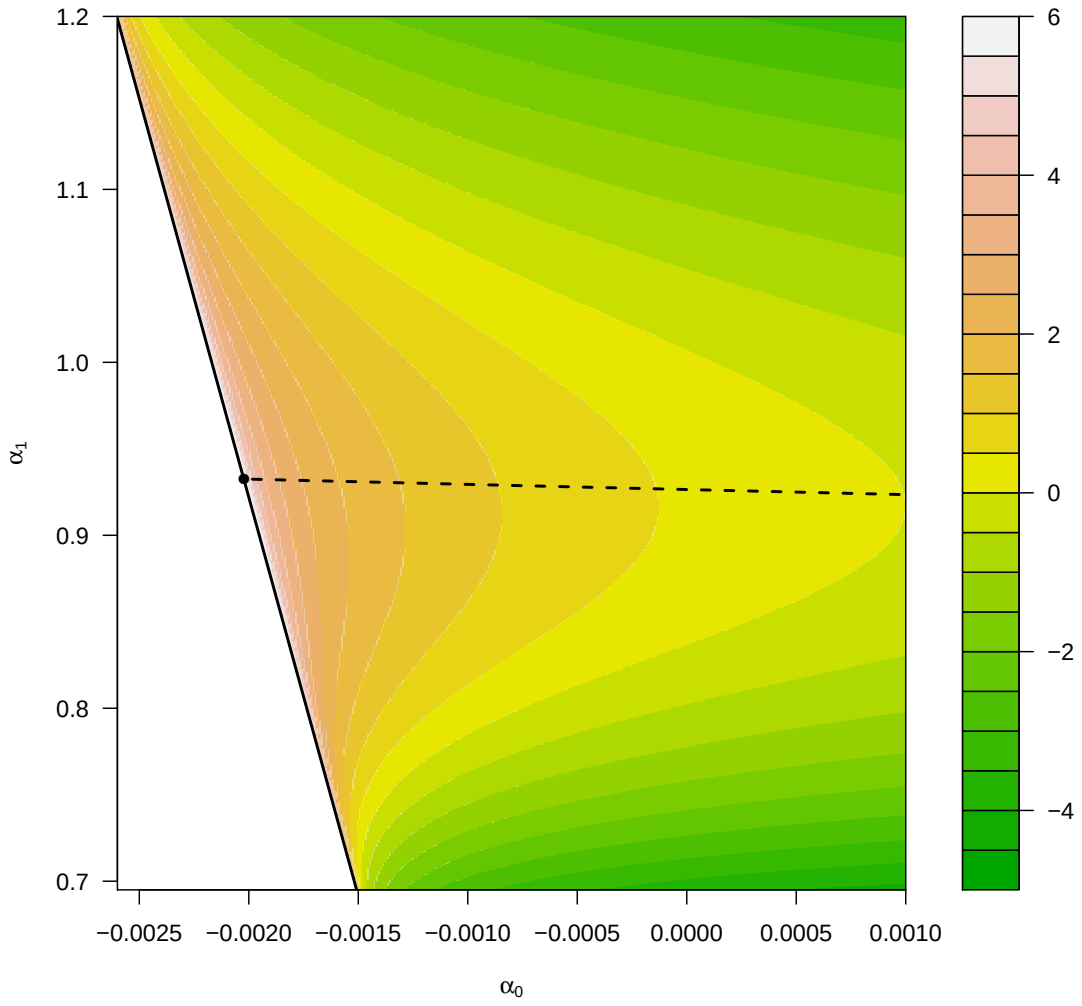

FIGURE 4.1: Log-likelihood of the EM algorithm for each combination of intercept and gradient for Example 1. The convergence path is the dotted line to the MLE on the boundary.

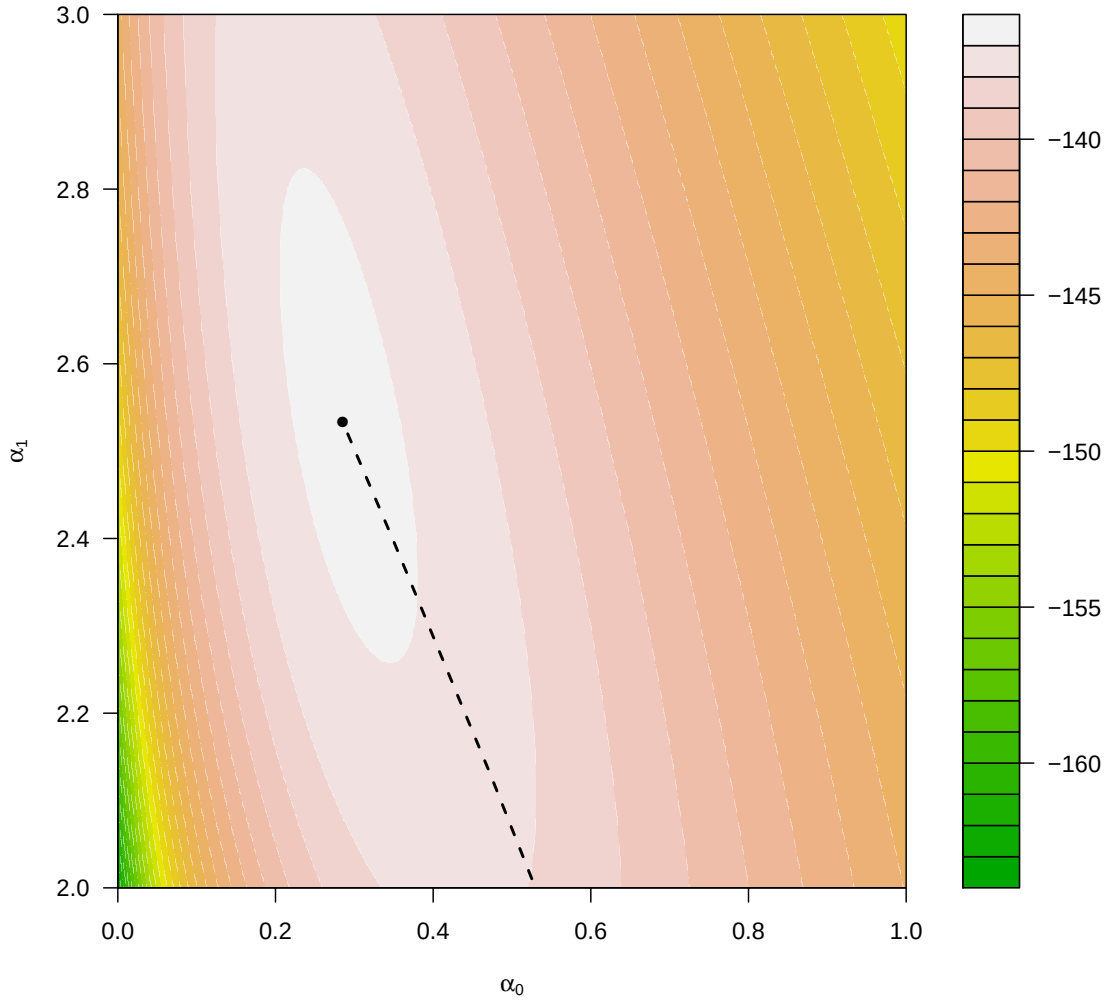

Figure 4.2: Log-likelihood of the EM algorithm for each combination of intercept and gradient for Example 2. The convergence path is the dotted line to the MLE.

4.4 Simulations

The weighted form of the estimating equations given in (4.5) and (4.6) suggest a simpler approach with $w_i = 1$, which has previously been considered by Bland and Altman (1999). In order to illustrate the use of the EM-based approach, a series of simulation studies have been conducted comparing the method detailed above with this crude least squares approach of regressing X_i^2 on x_i, without weights. In this simulation study, a zero mean model was used while the following variance models were compared: $0 + x_i$, $1 + x_i$, $2 - x_i$ and $1 - x_i$. Three different sample sizes were used: 100, 250 and 500 observations. The sampling distribution of x_i was also varied, and the x_i variable

was sampled uniformly, as well as from a negatively skewed, and a positively skewed distribution. Each simulation contained 1000 replications.

The efficiency based on the mean squared error (MSE) of the slope parameter over the various simulation studies is given in Figures 4.3 to 4.6.

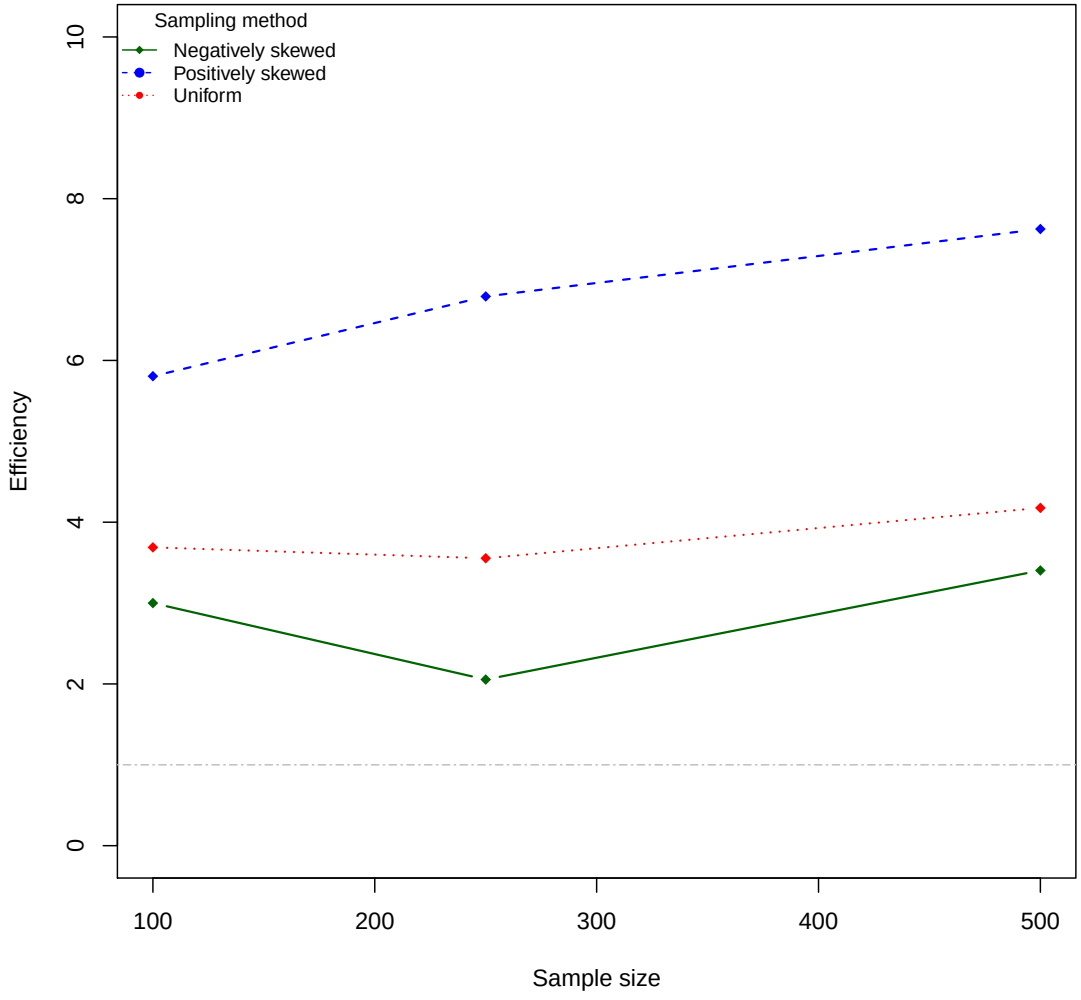

FIGURE 4.3: The MSE efficiency of the slope parameter from the simulation study with $\text{Var}(X_i) = x_i$. Values > 1 favour $\hat{\alpha}_{\text{ML}}$, and the grey dashed line indicates no difference.

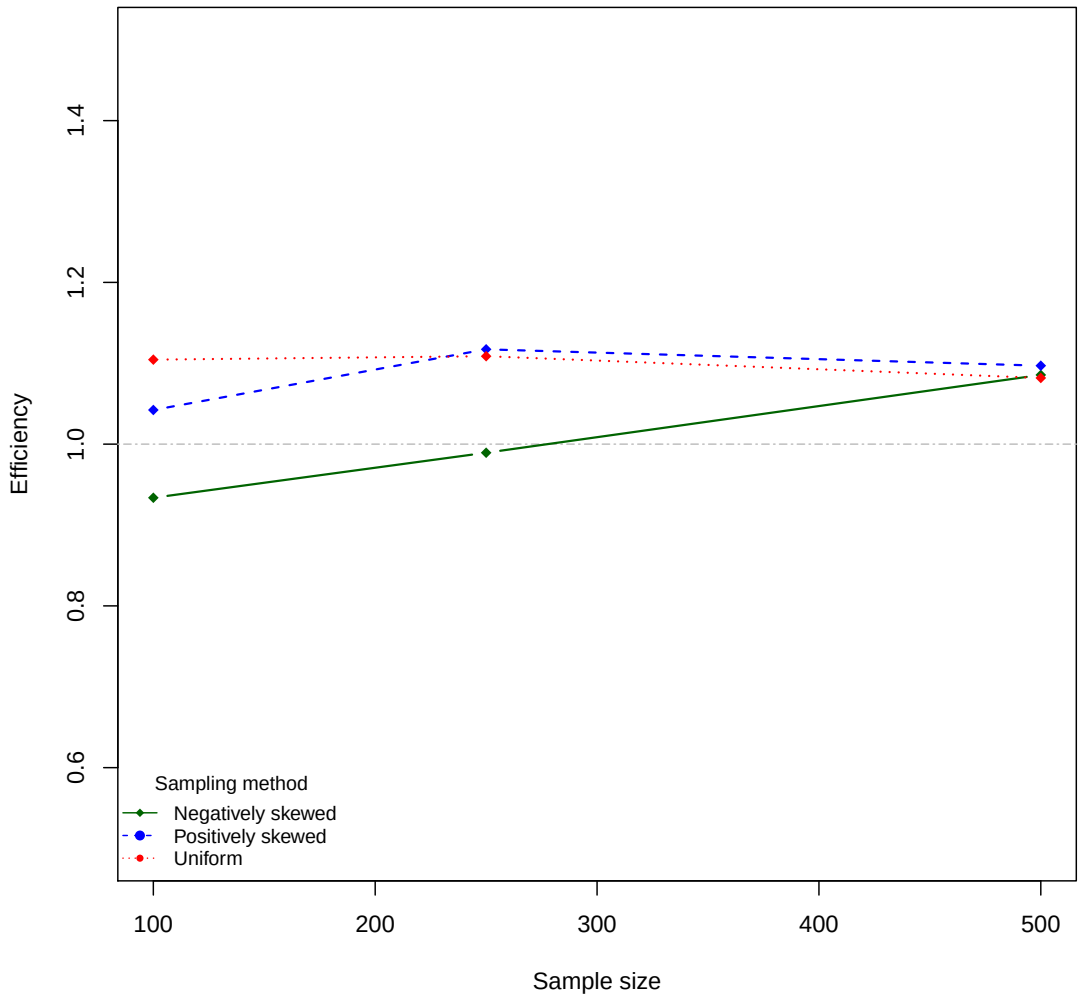

Figure 4.4: The MSE efficiency of the slope parameter from the simulation study with $\text{Var}(X_i) = 1 + x_i$. Values > 1 favour $\hat{\alpha}_{\text{ML}}$, and the grey dashed line indicates no difference.

The MSE efficiency was calculated as

$$\text{MSE efficiency} = \frac{(\text{mean}(\hat{\boldsymbol{\alpha}}_{\text{LS}}) - \boldsymbol{\theta})^2 + \text{variance}(\hat{\boldsymbol{\alpha}}_{\text{LS}})}{(\text{mean}(\hat{\boldsymbol{\alpha}}_{\text{ML}}) - \boldsymbol{\theta})^2 + \text{variance}(\hat{\boldsymbol{\alpha}}_{\text{ML}})},$$

where $\boldsymbol{\theta}$ is the vector of the true parameters the data were simulated from, $\hat{\boldsymbol{\alpha}}_{LS}$ is the vector of the slope parameters from the crude least squares approach and $\hat{\boldsymbol{\alpha}}_{\text{ML}}$ is the vector of the slope parameters from the EM-based approach described in this chapter.

In most instances, the EM-based estimates were just as efficient, and usually more efficient, than the crude least squares estimates. In particular, the EM-based approach

was up to 8 times more efficient in the slope parameter for the $\text{Var}(X_i) = x_i$ and $\text{Var}(X_i) = 1 - x_i$ models. For the model with $\text{Var}(X_i) = 1$ (constant variance) the crude least squares method was slightly more efficient for low sample sizes (results not shown). However, when the sample size increased to 500 or more, the two methods were equally efficient. This is to be expected, since the crude LS method is asymptotically equivalent to the MLE in the case of the constant variance model.

As well as providing a comparison of the efficiency of the weighted and unweighted approaches for variance regression, these simulations are useful in demonstrating the stability of the EM-based approach. In total, the method converged stably to the MLE in 45000 simulated datasets.

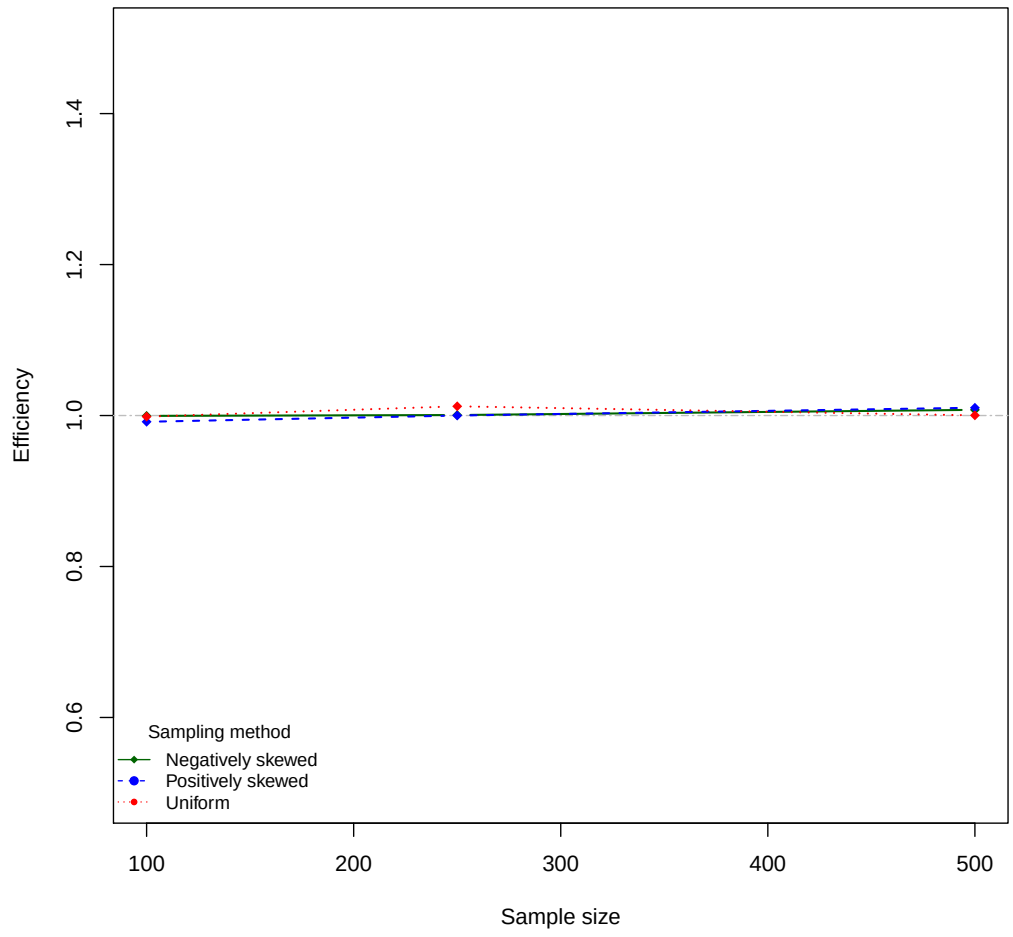

FIGURE 4.5: The MSE efficiency of the slope parameter from the simulation study with $\text{Var}(X_i) = 2 - x_i$. Values > 1 favour $\hat{\alpha}_{\text{ML}}$, and the grey dashed line indicates no difference.

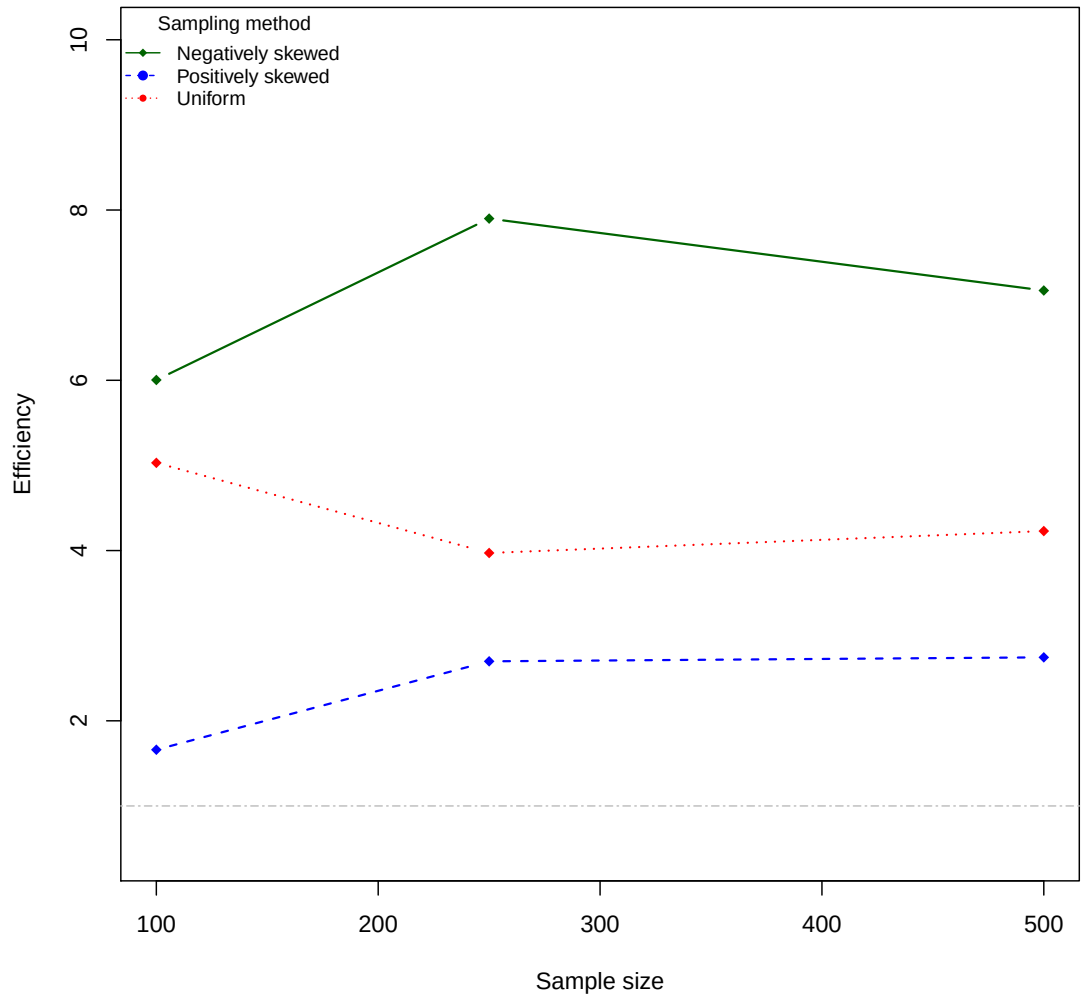

Figure 4.6: The MSE efficiency of the slope parameter from the simulation study with $\text{Var}(X_i) = 1 - x_i$. Values > 1 favour $\hat{\alpha}_{\text{ML}}$, and the grey dashed line indicates no difference.

4.5 Analysis example

The VCF dataset was introduced briefly in Section 3.3. The plot showed increasing variation over the average VCF, but there did not appear to be a change in the mean, which was approximately zero. The basic algorithm described in this chapter was used to fit the VCF data. A zero mean model was assumed, with the outcome as the difference in the two readings, and the average of the two readings as the covariate in the variance model.

The variance model result is shown in Figure 4.7, where there is close to zero variance at low VCF values, and this increases over the average VCF range. The residuals from the model are shown in Figure 4.8, which shows that the residuals appear to depart from a normal distribution, with a heavier tailed model perhaps being more appropriate.

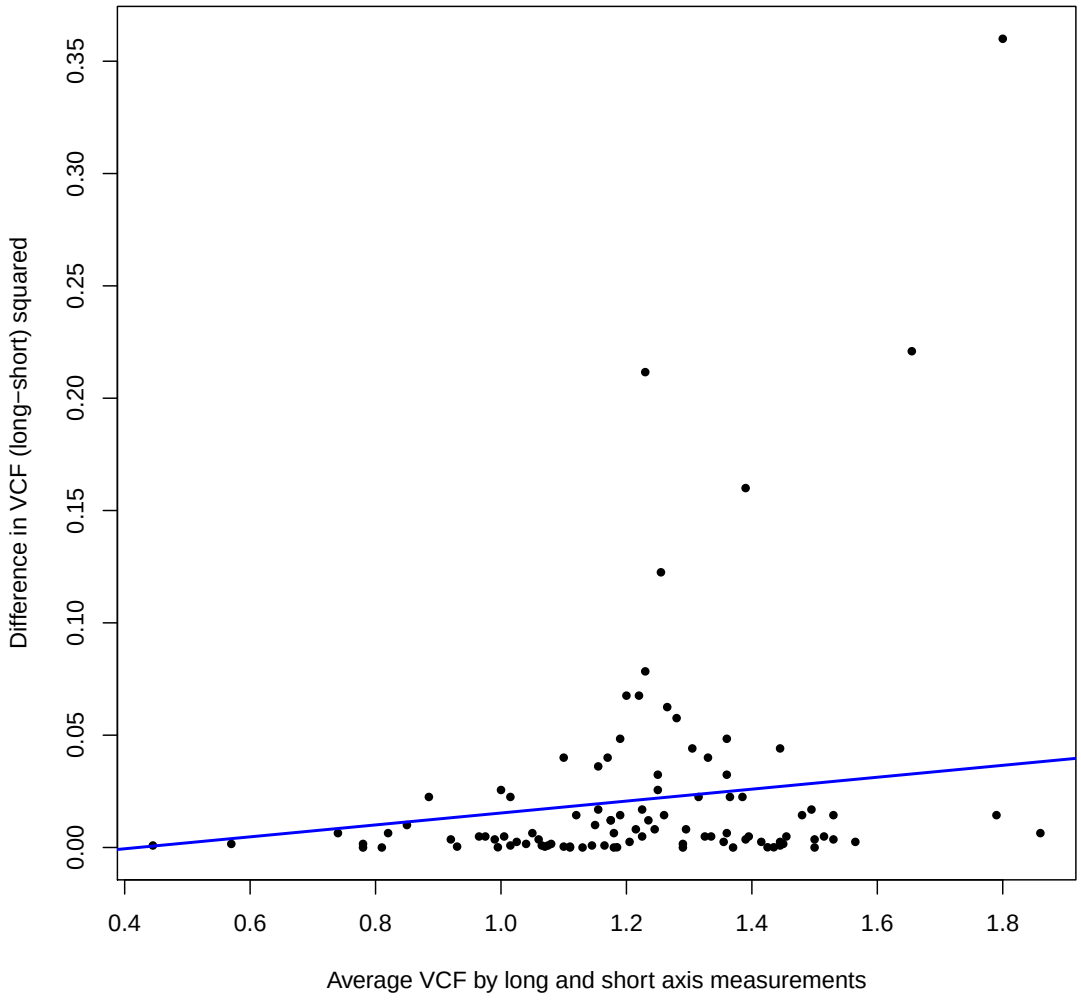

FIGURE 4.7: The linear variance (blue line) for the zero mean, linear variance model fit to the VCF data.

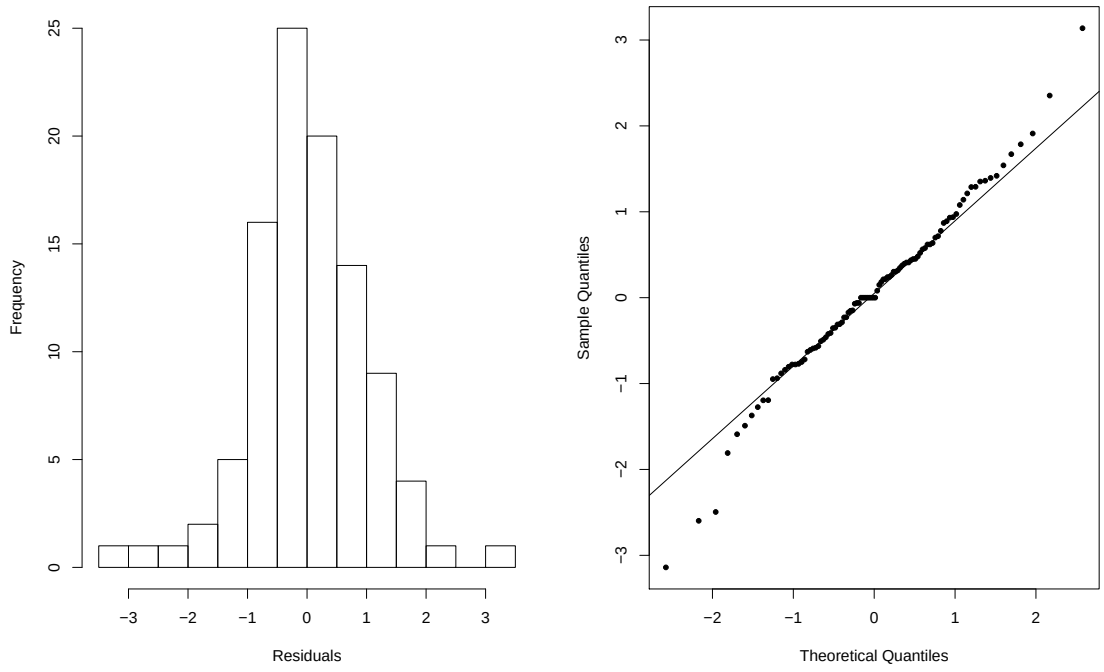

Figure 4.8: A histogram and Q-Q plot for the residuals from the zero mean, linear variance model fit to the VCF data.

4.6 Final comments

The simulation results presented in this chapter illustrate that the MLE approach detailed here has efficiency advantages over the crude unweighted approach. Furthermore, they illustrate that the EM-based approach for fitting the model can provide a reliable and stable approach that is not susceptible to the numerical instability that we have seen with other approaches. The EM-based approach was also shown to be applicable to the VCF data, although the simple linear model did not provide an adequate fit. In subsequent chapters, we will investigate its properties more generally using more realistic models, including analyses that seek to provide a more appropriate fit to the VCF data.

5

Multiple regression in mean and variance

In the previous chapter, we focused on a simplified model with zero mean,

$$X_i \sim N(0, \alpha_0 + \alpha_1 x_i) \quad \text{for} \quad i = 1, 2, ..., n.$$

Now let us now extend this model to include a regression model for the mean and the variance. Let the mean model have multiple covariates $\boldsymbol{z}_i = (z_{i1}, ..., z_{iP})$, and the variance model have multiple covariates, $\boldsymbol{x}_i = (x_{i1}, ..., x_{iQ})$. This gives the following more general model with regression in the mean and variance

$$X_i \sim N\left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}, \alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right) \quad \text{for} \quad i = 1, 2, ..., n. \tag{5.1}$$

In this chapter, we will explore the extension of the basic method introduced in Chapter 4 to model (5.1). As we will see in the next chapter, one of the main purposes of the methods presented here is for semi-parametric modelling using spline models, which can be expressed as additive multiple regression models in the mean and the

variance. We will also explore methods for obtaining standard error estimates, and present simulations and data analysis results.

5.1 Fitting details for more general models

In order to extend the method to fit a model in the mean, an additional step is added to the E-step and to the M-step of the EM algorithm. In this step, the mean is estimated using a linear regression model, weighted by the inverse of the current estimate of the variance. The covariates for the mean, $\boldsymbol{z}_i$, are used to fit this linear regression and the updated estimates for $\boldsymbol{\beta}$ are obtained, where $\boldsymbol{\beta} = (\beta_0, .., \beta_P)$. Adding this step to the EM algorithm converts it to an ECME (Expectation/Conditional Maximisation Either) algorithm (Liu and Rubin, 1994; McLachlan and Krishnan, 2007), because the additional step involves maximising the observed data likelihood over $\boldsymbol{\beta}$, given a current value for $\boldsymbol{\alpha}$.

The variance model is then extended to multiple covariates by incorporating additional latent variables for each covariate $\boldsymbol{x}_i$. If there are a total of Q covariates to be fit in the variance model, there are then $Q + 1$ independent, unobserved, latent variables,

$$Y_i \sim N(\beta_0 + \sum_{p=1}^{P} \beta_{ip} z_{ip}, \alpha_0), \quad Z_{i1} \sim N(0, \alpha_1 x_{i1}), \ ..., \quad Z_{iQ} \sim N(0, \alpha_Q x_{iQ}),$$

$$\text{where } X_i = Y_i + Z_{i1} + ... + Z_{iP}.$$

Similarly to Chapter 4, we have fit a constrained maximisation of the observed data log-likelihood. We need to search our entire parameter space, that is, non-decreasing variance ($\alpha_q \geq 0$) and decreasing ($\alpha_q < 0$) variance, for each α_q parameter. To maintain generalisability, assume that each continuous covariate is scaled such that $x_i \in [0, 1]$. Therefore, the constant term in the variance model (α_0) will be the variance when all other variance parameters are zero. In order to search for non-positive slope, an EM algorithm is fit using the covariate $1 - x_i$ in place of x_i, and thus the EM algorithm is maximising the log-likelihood over the parameter space estimates for $\alpha_0 \geq 0$ and $\alpha_q < 0$ for $q = 1, 2, ..., Q$. By repeating this for all possible covariate combinations, we

have a total of 2^Q EM algorithms to apply to these data in order to search the entire parameter space. The MLE will then be the $\hat{\boldsymbol{\theta}}$ from the EM algorithm that achieved the highest log-likelihood. These 2^Q models are referred to as the family of complete data models, and is another instance of a CEM algorithm as described in Section 3.1.2. While this is one method to search the entire parameter space for the MLE, other methods that may also be more efficient could be utilised (Donoghoe and Marschner, 2016; Marschner, 2014).

The log-likelihood corresponding to the complete data model is

$$\begin{aligned} L\left(\boldsymbol{\theta}\right) = &-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log\left(\alpha_0\right) - \frac{1}{2}\sum_{i=1}^{n}\frac{\left(Y_i - \beta_0 - \sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\alpha_0} \\ &-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log\left(\alpha_1 x_{i1}\right) - \frac{1}{2}\sum_{i=1}^{n}\frac{\left(Z_{i1}\right)^2}{\alpha_1 x_{i1}} + ... \\ &-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log\left(\alpha_Q x_{iQ}\right) - \frac{1}{2}\sum_{i=1}^{n}\frac{\left(Z_{iQ}\right)^2}{\alpha_Q x_{iQ}}, \end{aligned} \tag{5.2}$$

and is linear in Y_i^2 and Z_{iq}^2. The E-step is largely similar to that in the previous chapter, involving the calculation of the conditional expectations

$$\begin{aligned} &\hat{Y}_i^2(\boldsymbol{\theta}) = \mathbb{E}((Y_i - \beta_0 - \sum_{p=1}^{P}\beta_p z_{ip})^2 | X_i; \boldsymbol{\theta}), \\ &\hat{Z}_{i1}^2(\boldsymbol{\theta}) = \mathbb{E}(Z_{i1}^2 | X_i; \boldsymbol{\theta}) \quad ,..., \quad \hat{Z}_{iQ}^2(\boldsymbol{\theta}) = \mathbb{E}(Z_{iQ}^2 | X_i; \boldsymbol{\theta}) \end{aligned} \tag{5.3}$$

where $\boldsymbol{\theta} = (\beta_0, .., \beta_P, \alpha_0, .., \alpha_Q)$. This leads to

$$\hat{Y}_i^2(\boldsymbol{\theta}) = \alpha_0 + \frac{\alpha_0^2}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}}\left(\frac{\left(X_i - \left(\beta_0 + \sum_{p=1}^{P}\beta_p z_{ip}\right)\right)^2}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}} - 1\right). \tag{5.4}$$

The estimation of the latent variables Z_{iq} is also similar to the basic method with

$$\hat{Z}_{iq}^2(\boldsymbol{\theta}) = \alpha_q x_{iq} + \frac{\alpha_q x_{iq}^2}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} \left(\frac{\left(X_i - \left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip} \right) \right)^2}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} - 1 \right). \tag{5.5}$$

In the M-step, we calculate the updated estimates of $\boldsymbol{\theta}$, called $\hat{\boldsymbol{\theta}}^{new}$. Firstly for the mean, given the current estimate $\hat{\boldsymbol{\alpha}}^{old}$, we must fit a weighted linear regression for

$$X_i \sim N\left(\beta_0 + \sum \beta_p z_{ip}, \sigma_i^2(\hat{\boldsymbol{\alpha}}^{old}) \right)$$

where $\sigma_i^2(\hat{\boldsymbol{\alpha}}) = \alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}$, with weight as $w_i = \frac{1}{\sigma_i^2(\hat{\boldsymbol{\alpha}}^{old})}$. So the new estimate of $\boldsymbol{\beta}$ is obtained by the weighted least squares estimate

$$\hat{\boldsymbol{\beta}}^{new} = \underset{\hat{\boldsymbol{\beta}}^{new}}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \left(X_i - \left(\hat{\beta}_0 + \sum_{p=1}^{P} \hat{\beta}_p z_{ip} \right) \right)^2 \tag{5.6}$$

$$= \left(\boldsymbol{Z^T W Z} \right)^{-1} \boldsymbol{Z^T W X}, \tag{5.7}$$

where

$$\boldsymbol{Z} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{P1} & z_{P2} & \cdots & z_{Pn} \end{bmatrix}, \ \boldsymbol{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}.$$

The variance estimates are also updated with

$$\hat{\alpha}_0^{new} = n^{-1} \sum_{i=1}^{n} \hat{Y}_i^2 \left(\hat{\boldsymbol{\theta}}^{old} \right) \quad \text{and} \quad \hat{\alpha_q}^{new} = n^{-1} \sum_{i=1}^{n} \frac{\hat{Z}_{iq}^2(\hat{\boldsymbol{\theta}}^{old})}{x_{iq}}. \tag{5.8}$$

Once the current estimates have been calculated, the algorithm continues until convergence. The algorithm is summarised schematically in Figure 5.1. The likelihood

function that is being maximised for this more generalised model is

$$l(\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)}} \exp\left(\frac{-\left(X_i - \left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}\right)\right)^2}{2\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)}\right).$$

5.2 Standard error estimation

5.2.1 Information matrix

If $\boldsymbol{\theta} = (\beta_0, \beta_1, ..., \beta_P, \alpha_0, \alpha_1, ..., \alpha_Q)$, with a total of W parameters (where $W = P + Q + 2$) then the information matrix is the negative second matrix derivative of the log-likelihood function, which is a $W \times W$ matrix.

The log-likelihood for our general model discussed in the previous section is

$$\ell(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n} \log\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right) - \frac{1}{2}\sum_{i=1}^{n} \frac{\left(X_i - \beta_0 - \sum_{p=1}^{P} \beta_p z_{ip}\right)^2}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}}. \tag{5.9}$$

If we partially differentiate (5.9) with respect to β_0, we get the following likelihood equation

$$\frac{\partial}{\partial \beta_0}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \frac{X_i - \beta_0 - \sum_{p=1}^{P} \beta_p z_{ip}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}}.$$

This then follows on through each of the β_p parameters to give

$$\frac{\partial}{\partial \beta_p}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \frac{z_{ip}\left(X_i - \beta_0 - \sum_{p=1}^{P} \beta_p z_{ip}\right)}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}}.$$

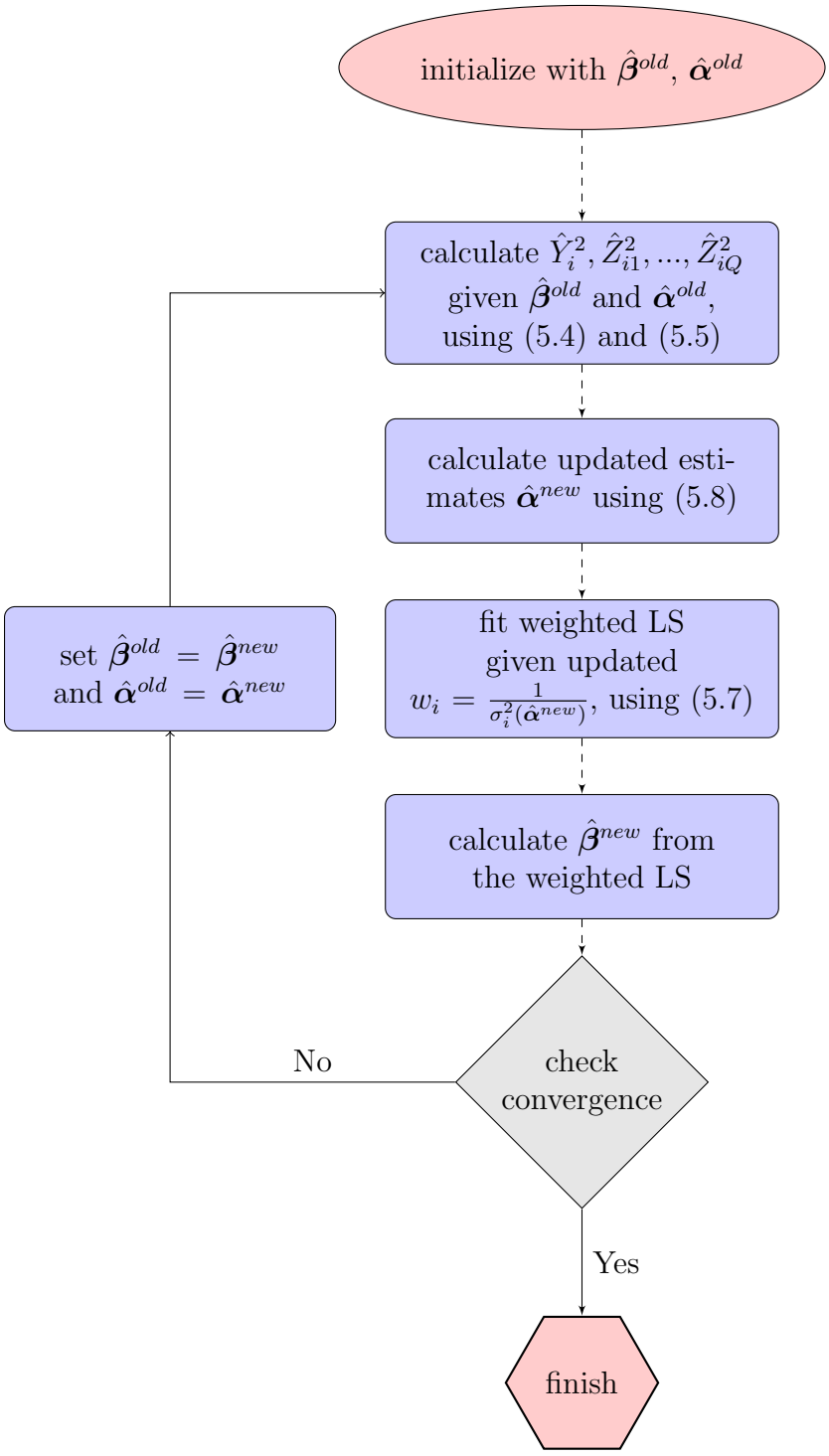

FIGURE 5.1: The ECME algorithm for the estimation of the mean and variance.

For the likelihood equation for α_0, we get

$$\frac{\partial}{\partial\alpha_0}\ell(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}} + \frac{1}{2}\sum_{i=1}^{n}\frac{\left(X_i - \beta_0 - \sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2},$$

and then for each of the α_q parameters we have

$$\frac{\partial}{\partial\alpha_q}\ell(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{i=1}^{n}\frac{x_{iq}}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}} + \frac{1}{2}\sum_{i=1}^{n}\frac{x_{iq}\left(X_i - \beta_0 - \sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2}.$$

Now, taking the second derivatives we obtain the following $(P+1)\times(P+1)$ matrix for the $\boldsymbol{\beta}$ parameters. We refer to this matrix as $\boldsymbol{B} = [B_{ij}]$:

$$B_{00} = -\frac{\partial^2}{\partial\beta_0^2}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\frac{1}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}}$$

$$B_{01} = B_{10} = -\frac{\partial^2}{\partial\beta_0\beta_1}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\frac{z_{i1}}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}}$$

$$B_{11} = -\frac{\partial^2}{\partial\beta_1^2}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\frac{z_{i1}^2}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}}$$

$$\vdots$$

$$B_{PP} = -\frac{\partial^2}{\partial\beta_P^2}\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\frac{z_{iP}^2}{\alpha_0 + \sum_{q=1}^{Q}\alpha_q x_{iq}}.$$

Now, the partial derivatives for the α_q parameters form a $(Q+1)\times(Q+1)$ matrix $\boldsymbol{A}$, where $\boldsymbol{A}=[A_{ij}]$:

$$A_{00}=-\frac{\partial^2}{\partial\alpha_0^2}\ell(\boldsymbol{\theta}) \quad =-\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2}+\sum_{i=1}^{n}\frac{\left(X_i-\beta_0-\sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^3}$$

$$A_{01}=-\frac{\partial^2}{\partial\alpha_0\alpha_1}\ell(\boldsymbol{\theta})=-\frac{1}{2}\sum_{i=1}^{n}\frac{x_{i1}}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2}+\sum_{i=1}^{n}\frac{x_{i1}\left(X_i-\beta_0-\sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^3}$$

$$A_{11}=-\frac{\partial^2}{\partial\alpha_1^2}\ell(\boldsymbol{\theta}) \quad =-\frac{1}{2}\sum_{i=1}^{n}\frac{x_{i1}^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2}+\sum_{i=1}^{n}\frac{x_{i1}^2\left(X_i-\beta_0-\sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^3}$$

$$\vdots$$

$$A_{QQ}=-\frac{\partial^2}{\partial\alpha_Q^2}\ell(\boldsymbol{\theta}) \quad =-\frac{1}{2}\sum_{i=1}^{n}\frac{x_{iQ}^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^2}+\sum_{i=1}^{n}\frac{x_{iQ}^2\left(X_i-\beta_0-\sum_{p=1}^{P}\beta_p z_{ip}\right)^2}{\left(\alpha_0+\sum_{q=1}^{Q}\alpha_q x_{iq}\right)^3}.$$

The partial derivatives of the combination of β_p and α_q parameters reduce to zero when we take the expectation, and thus the expected information matrix is block diagonal:

$$\begin{bmatrix}\boldsymbol{B} & \mathbf{0}^T\\ \mathbf{0} & \boldsymbol{A}\end{bmatrix},$$

where $\mathbf{0}$ is a $(Q+1)\times(P+1)$ matrix of zeroes.

Now, if we focus on the mean component of the expected information matrix, $\boldsymbol{B}$,

$$\begin{bmatrix} \sum_{i=1}^{n} \frac{1}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \sum_{i=1}^{n} \frac{z_{i1}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \cdots & \sum_{i=1}^{n} \frac{z_{iP}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} \\ \sum_{i=1}^{n} \frac{z_{i1}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \sum_{i=1}^{n} \frac{(z_{i1})^2}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \cdots & \sum_{i=1}^{n} \frac{z_{i1} z_{iP}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} \frac{z_{iP}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \sum_{i=1}^{n} \frac{z_{i1} z_{iP}}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} & \cdots & \sum_{i=1}^{n} \frac{(z_{iP})^2}{\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}} \end{bmatrix},$$

and for the variance component of the expected information matrix, $\boldsymbol{A}$,

$$\begin{bmatrix} \frac{1}{2} \sum_{i=1}^{n} \frac{1}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \frac{1}{2} \sum_{i=1}^{n} \frac{x_{i1}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \cdots & \frac{1}{2} \sum_{i=1}^{n} \frac{x_{iQ}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} \\ \frac{1}{2} \sum_{i=1}^{n} \frac{x_{i1}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \frac{1}{2} \sum_{i=1}^{n} \frac{(x_{i1})^2}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \cdots & \frac{1}{2} \sum_{i=1}^{n} \frac{x_{i1} x_{iQ}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2} \sum_{i=1}^{n} \frac{x_{iQ}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \frac{1}{2} \sum_{i=1}^{n} \frac{x_{i1} x_{iQ}}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} & \cdots & \frac{1}{2} \sum_{i=1}^{n} \frac{(x_{iQ})^2}{\left(\alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right)^2} \end{bmatrix}.$$

Lastly, these matrices need to be inverted in order to obtain the standard errors of the respective parameters. It is important to note that if the estimate is on the boundary, then the standard errors must be obtained by bootstrapping.

5.2.2 Bootstrapping

The standard error for these parameters can also be obtained by bootstrapping. If these data are randomly sampled (with replacement) to obtain a sample of size n, the algorithm described in Section 5.1 can be applied to this sample. This is a single bootstrap sample, which is repeated then a total of B times, where B is taken as one thousand. We then obtain B bootstrap samples, and thus B estimates for each of the parameters. The 2.5% and the 97.5% percentiles are then taken for each parameter in order to obtain the 95% confidence interval.

In practice, for some datasets with a large number of parameters in the variance model, this may take some time. This is because the entire parameter space is searched for each of the B bootstrap samples using the entire family of 2^Q EM algorithms. Nonetheless, due to the stability of the algorithm, reliable convergence will be obtained in all bootstrap replications.

5.3 Simulations

Similar to simulations performed in Chapter 4, a variety of variance models were compared in this simulation study. The variance models to be explored were $0 + x_i$, $1 + x_i$, $2 - x_i$ and $1 - x_i$, in three different sample sizes: 100, 250 and 500 observations. However, given the introduction of the mean model, there is an additional component to vary in the simulations. Both a constant mean model ($\beta_0 = 1$) and a linear mean model, $1 + x_i$, were explored. The mean squared error (MSE) of the four parameters were calculated: the intercept and slope for the mean (β_0 and β_1), and the intercept and slope for the variance (α_0 and α_1). As the MSE results from the constant mean model were similar to the results of the linear mean model, only the linear mean model results are shown (Figures 5.2 and 5.3).

The main conclusion from these simulations is that the proposed algorithm works well for fitting linear regression models in the mean and variance. From Tables 5.1 and 5.2, the simulation means are very close to the true values, even in the case with small sample sizes.

Comparing the variability results in these tables, it is clear that the slope parameters

Table 5.1: Results from a simulation study. Data from a constant mean model $\beta_0 = 1$, with various variance models, at 100, 250 or 500 observations.

	$\beta_0 = 1, \beta_1 = 0, \alpha_0 = 0, \alpha_1 = 1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.00	-0.00	-0.00	1.00	1.00	0.00	-0.00	0.99	1.00	0.00	0.00	0.99
SD	0.06	0.19	0.02	0.16	0.03	0.11	0.01	0.10	0.02	0.07	0.00	0.07
MSE	0.00	0.03	0.00	0.03	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00
	$\beta_0 = 1, \beta_1 = 0, \alpha_0 = 1, \alpha_1 = 1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	-0.02	0.98	0.99	1.00	0.00	0.99	1.00	1.00	0.01	0.98	1.01
SD	0.23	0.44	0.36	0.75	0.14	0.26	0.22	0.48	0.10	0.18	0.15	0.31
MSE	0.05	0.20	0.13	0.56	0.02	0.07	0.05	0.23	0.01	0.03	0.02	0.10
	$\beta_0 = 1, \beta_1 = 0, \alpha_0 = 1, \alpha_1 = 0$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	-0.01	0.99	-0.01	1.00	0.00	0.99	0.01	1.00	0.01	0.99	0.01
SD	0.21	0.37	0.30	0.52	0.12	0.22	0.18	0.32	0.09	0.15	0.13	0.22
MSE	0.04	0.14	0.09	0.27	0.02	0.05	0.03	0.10	0.01	0.02	0.02	0.05
	$\beta_0 = 1, \beta_1 = 0, \alpha_0 = 1, \alpha_1 = -1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.00	-0.00	1.00	-1.00	1.00	-0.00	0.99	-1.00	1.00	0.00	0.99	-0.99
SD	0.16	0.19	0.15	0.16	0.09	0.11	0.09	0.10	0.07	0.08	0.06	0.07
MSE	0.02	0.04	0.02	0.02	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00
	$\beta_0 = 1, \beta_1 = 0, \alpha_0 = 2, \alpha_1 = -1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	-0.01	1.99	-1.02	1.00	0.00	1.98	-0.98	1.00	0.01	1.98	-0.98
SD	0.28	0.45	0.51	0.78	0.16	0.26	0.31	0.46	0.12	0.18	0.22	0.32
MSE	0.08	0.20	0.26	0.60	0.03	0.07	0.09	0.21	0.01	0.03	0.05	0.10

(β_1 and α_1) have the largest variability. Other trends are as expected, such as decreasing variability with increasing sample size. Overall, these results are favourable for the proposed algorithm.

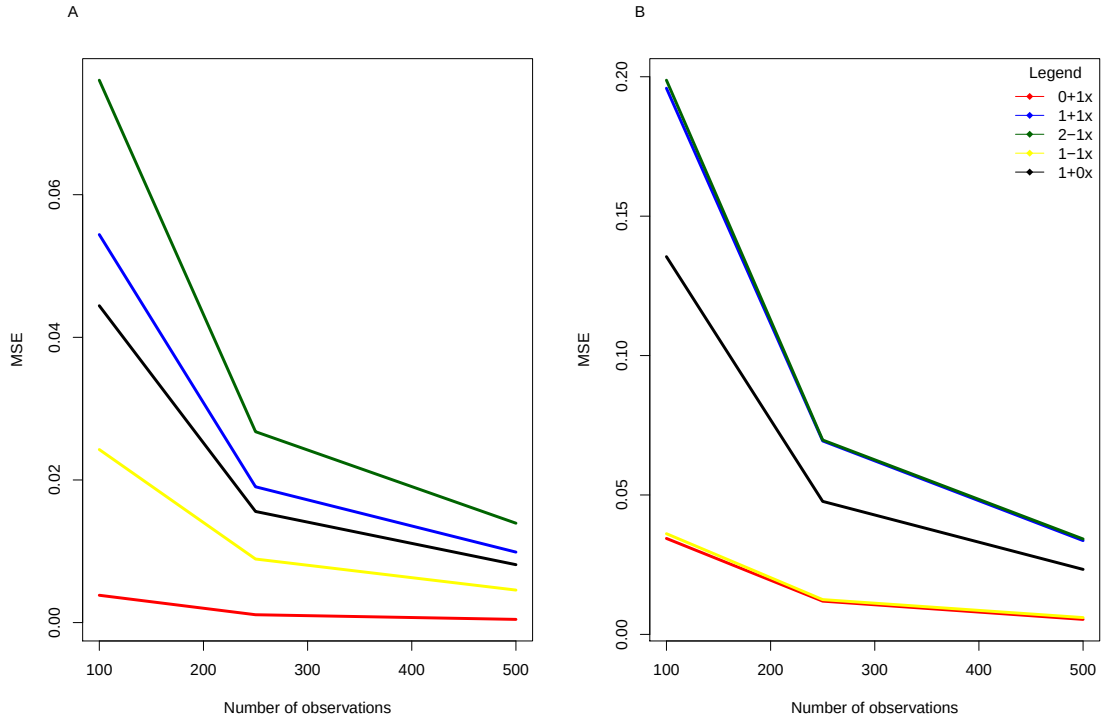

FIGURE 5.2: The MSE for the estimates of the mean. The intercept (A) and the slope (B), from the simulations performed of the mean model $1 + x_i$ and various variance models, for three sample sizes.

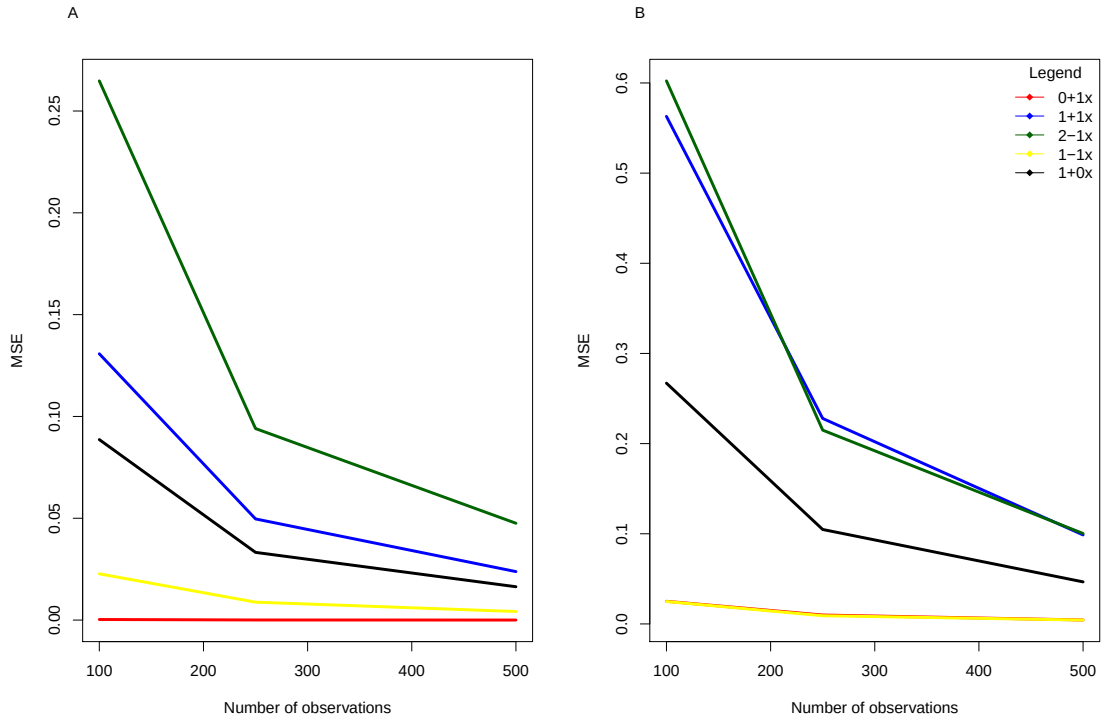

FIGURE 5.3: The MSE for the estimates of the variance. The intercept (A) and the slope (B), from the simulations performed of the mean model $1 + x_i$ and various variance models, for three sample sizes.

Table 5.2: Results from a simulation study. Data from a linear mean model $1 + x_i$, with various variance models, at 100, 250 or 500 observations.

	$\beta_0 = 1, \beta_1 = 1, \alpha_0 = 0, \alpha_1 = 1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.00	1.00	-0.00	1.00	1.00	1.00	-0.00	0.99	1.00	1.00	0.00	0.99
SD	0.06	0.19	0.02	0.16	0.03	0.11	0.01	0.10	0.02	0.07	0.00	0.07
MSE	0.00	0.03	0.00	0.03	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00
	$\beta_0 = 1, \beta_1 = 1, \alpha_0 = 1, \alpha_1 = 1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	0.98	0.98	0.99	1.00	1.00	0.99	1.00	1.00	1.01	0.98	1.01
SD	0.23	0.44	0.36	0.75	0.14	0.26	0.22	0.48	0.10	0.18	0.15	0.31
MSE	0.05	0.20	0.13	0.56	0.02	0.07	0.05	0.23	0.01	0.03	0.02	0.10
	$\beta_0 = 1, \beta_1 = 1, \alpha_0 = 1, \alpha_1 = 0$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	0.99	0.99	-0.01	1.00	1.00	0.99	0.01	1.00	1.01	0.99	0.01
SD	0.21	0.37	0.30	0.52	0.12	0.22	0.18	0.32	0.09	0.15	0.13	0.22
MSE	0.04	0.14	0.09	0.27	0.02	0.05	0.03	0.10	0.01	0.02	0.02	0.05
	$\beta_0 = 1, \beta_1 = 1, \alpha_0 = 1, \alpha_1 = -1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.00	1.00	1.00	-1.00	1.00	1.00	0.99	-1.00	1.00	1.00	0.99	-0.99
SD	0.16	0.19	0.15	0.16	0.09	0.11	0.09	0.10	0.07	0.08	0.06	0.07
MSE	0.02	0.04	0.02	0.02	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00
	$\beta_0 = 1, \beta_1 = 1, \alpha_0 = 2, \alpha_1 = -1$											
	n=100				n=250				n=500			
	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1	β_0	β_1	α_0	α_1
Mean	1.01	0.99	1.99	-1.02	1.00	1.00	1.98	-0.98	1.00	1.01	1.98	-0.98
SD	0.28	0.45	0.51	0.78	0.16	0.26	0.31	0.46	0.12	0.18	0.22	0.32
MSE	0.08	0.20	0.26	0.60	0.03	0.07	0.09	0.21	0.01	0.03	0.05	0.10

5.4 Analysis example

One of the main purposes of the methods presented here is for the introduction of semi-parametric modelling using spline models, which can be expressed as additive multiple regression models in the mean and the variance. However, we include here a linear model analysis based on data given in Chapter 2. A zero mean, linear variance model was explored in Section 4.5, and we will now build on these models with the incorporation of a mean model.

The first model given in Table 5.3 is the zero mean model presented in Chapter 4. The following two models build upon this, with a constant mean parameter, and then a linear mean model incorporated. Lastly, the final two rows have a constant variance

TABLE 5.3: Results from various mean and variance models of the VCF data.

Mean model	Variance model	Log-likelihood	AIC	AICc	BIC	HQC
None	Linear	54.8	-105.5	-105.4	-100.3	-103.4
Constant	Linear	54.8	-103.6	-103.4	-95.8	-100.4
Linear	Linear	59.8	-111.7	-111.3	-101.3	-107.5
Constant	Constant	46.2	-86.3	-86.1	-78.5	-83.2
Linear	Constant	46.2	-86.4	-84.0	-74.0	-80.2

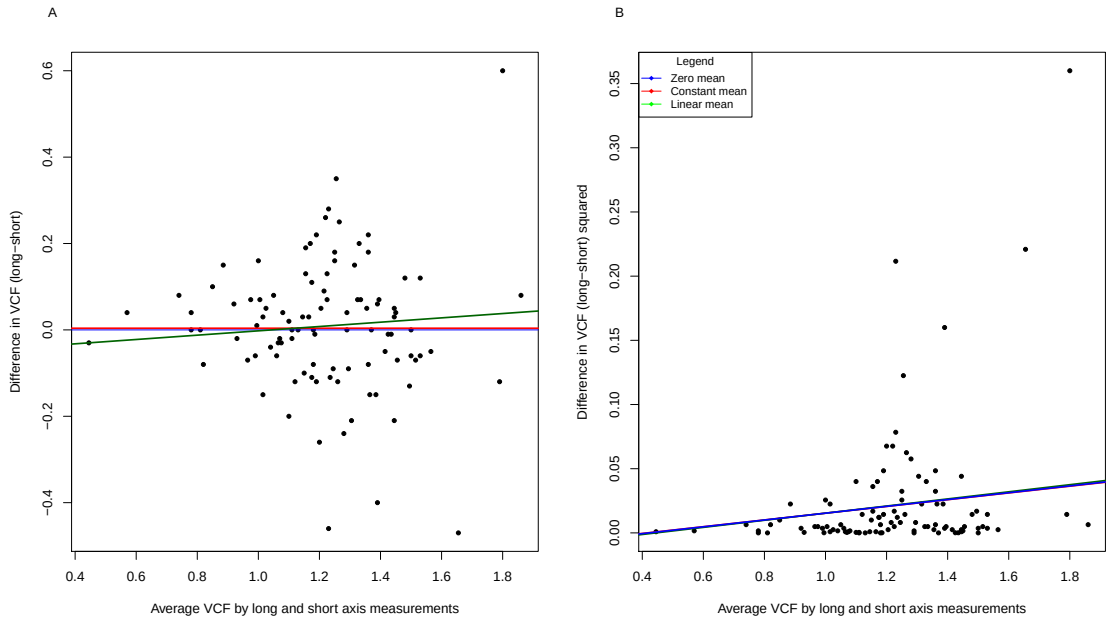

FIGURE 5.4: The mean (A) and the variance (B) for the three different mean models, each with linear variance models, fit to the VCF data. Note that the variance estimates are very similar and overlay each other.

parameter estimated, for completeness. From each of the information criteria, these two constant variance models are inferior to the linear variance models. When comparing the linear variance models, all of the criteria agree that the linear mean and linear variance model is the best model for these data. These models with linear variance and various mean models are further explored graphically in Figure 5.4. The differences in the mean models are apparent, while the variance estimates are very similar and overlay each other. Normal scores plots of the residuals from these three models are given in Figure 5.5, and again appear similar. As in the analysis presented in Chapter 4, there is some evidence of non-normality, suggesting the need for more flexible models such as semi-parametric or non-normal models.

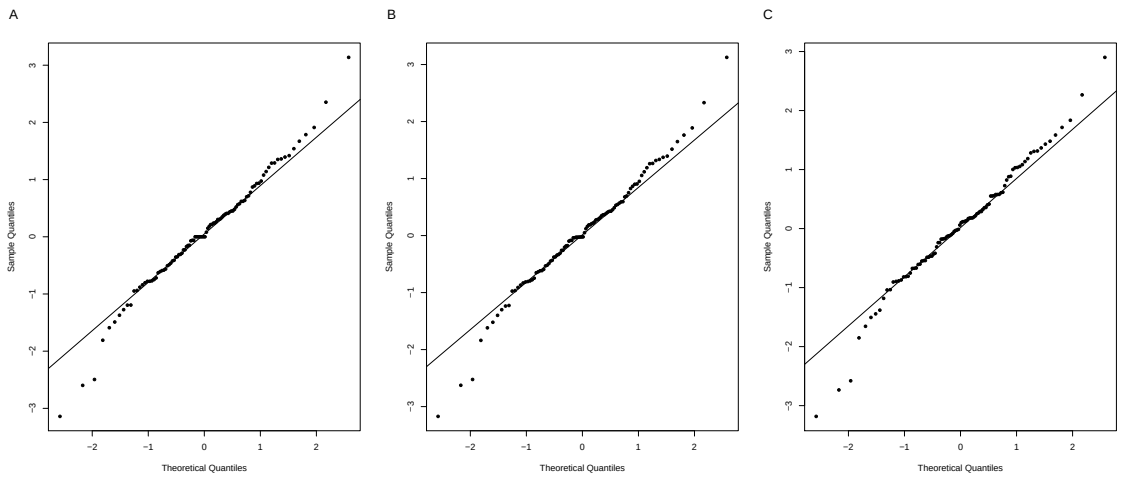

Figure 5.5: The residuals from the three different mean models, each with linear variance models, fit to the VCF data. A is the zero mean model, B is the constant mean model and C is the linear mean model.

5.5 Final comments

This chapter built on the basic method developed in Chapter 4 and detailed a more general model of fitting a regression in both the mean and the variance with multiple covariates in each. Simulations explored the efficiency of the estimates from both the mean and the variance model, and demonstrated that reliable algorithms are obtained. The method was applied to the VCF dataset, where it was shown that a linear model in the mean and variance was a good fit to these data. However, the residuals have heavy tails and perhaps the inclusion of additional parameters or flexibility may improve the fit. The next chapter will build upon this mean and variance model, and introduce the incorporation of semi-parametric models into this algorithm.

6

Semi-parametric models

In the previous chapter we explored a general model for fitting multiple covariates in the mean and the variance,

$$X_i \sim N\left(\beta_0 + \sum_{p=1}^{P} \beta_p z_{ip}, \alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}\right) \quad \text{for} \quad i = 1, 2, ..., n.$$

This chapter will utilise this general model in order to fit semi-parametric models in either the mean or the variance, or both. While the concept of semi-parametric models was introduced briefly in Chapter 3, in this chapter we will incorporate these methods in our CEM algorithm, show results from a simulation study and demonstrate the algorithm in example datasets.

6.1 Monotonic step functions

We begin the discussion of semi-parametric models by considering a specific model in which the variance is assumed to be an unrestricted monotonic function of a covariate. Later we will extend this to more general and flexible semi-parametric models.

At times, it may be appropriate to restrict the variance to be monotonically increasing or decreasing. When an ordered categorical or a continuous covariate is used in a regression model, the fitting of a semi-parametric step function model is easily achieved using the CEM algorithm presented in Chapter 5. This is largely due to the positively constrained nature of the algorithm.

Firstly, we must order the covariate x_i for $i = 1, 2, ..., n$, as unique observed values $w_1 < w_2 < ... < w_k$, where $k \leqslant n$. An unspecified, monotonic regression function $f(x_i)$ can then be added, with $f(w_i) < f(w_j)$ for all $i < j$. In practice, the unspecified f is only identifiable at each of the unique covariate values w_j, so the monotone step function jumps at each w_j. The size of this jump is estimated by including the covariate w in the CEM algorithm as a categorical covariate. The only difference is that it is unnecessary to cycle through parameter space subsets that correspond to negative increments, as we only need to search for increasing increments at each w_j. A simple amendment would allow for a monotonically decreasing variance, by reordering the unique observed values w_j as $w_k < w_{k-1} < ... < w_2 < w_1$.

Although this is a flexible method for extending a continuous covariate beyond a linear relationship, the disadvantage is the large number of parameters that need to be estimated. An alternative to this method is fitting B-spline basis functions that can be used to fit a smooth, flexible regression line, without requiring many degrees of freedom. This will be explored later in this chapter.

As an example of the step function approach, we consider the VCF data that was investigated previously in Chapter 5. The variance as a function of the mean VCF was increasing in Figure 2.1, therefore fitting a monotonic step function with increasing variance provides a natural model. The estimated step function for these data have been given in Figure 6.1, with tick marks on the inside of the x-axis indicating the w_k unique data points. It is interesting to note that there is not a step at each value,

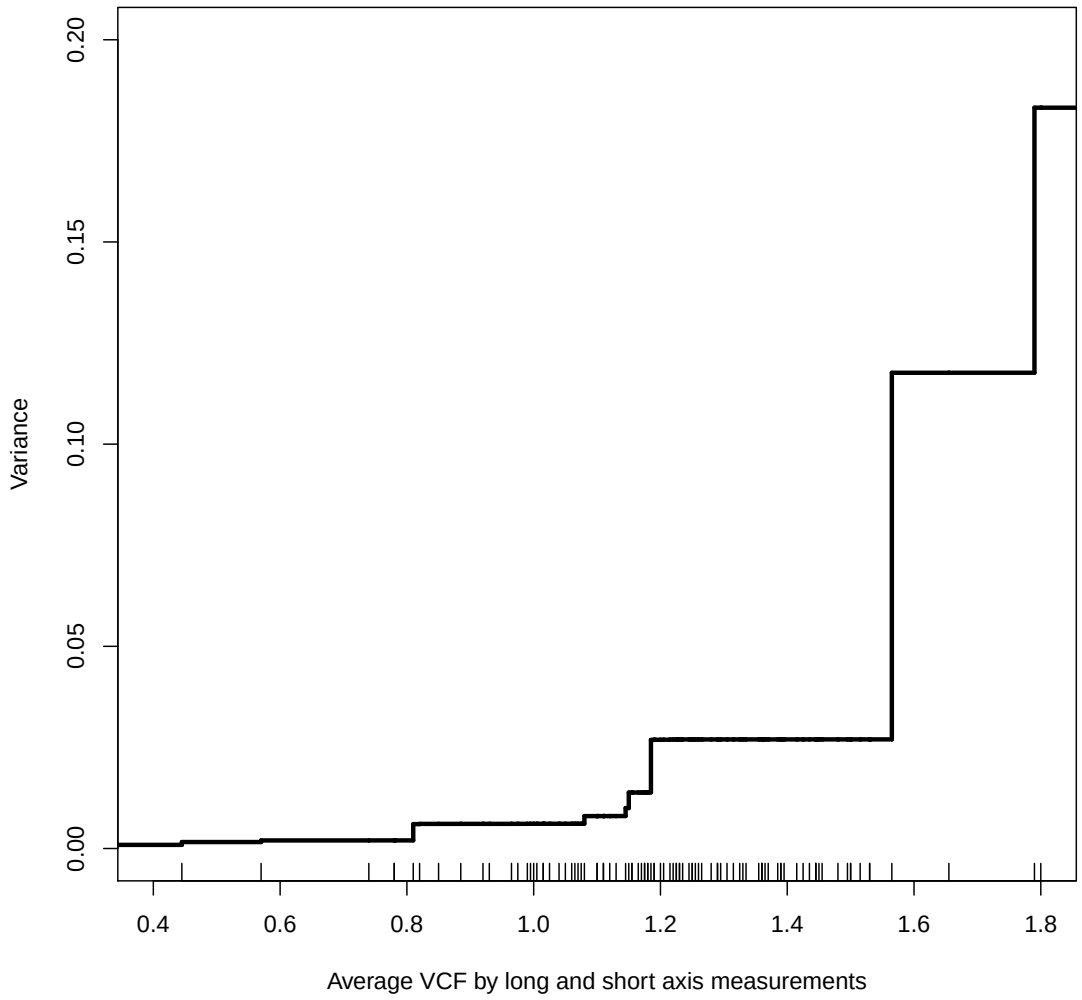

Figure 6.1: The step function for the variance of the VCF data. Each unique data point (w_k) is represented as a tick mark on the inside of the x-axis.

and the size of the step varies across the x values. This shows that some steps have been estimated to be zero, while others, particularly for larger VCF values, have been estimated to be quite large. This suggests that a non-linear VCF variance function may be the best fit for these data, and will be investigated further in this chapter.

6.2 Fitting details for semi-parametric models

For our data, let us assume that both the mean and the variance regression models follow unknown functions. Given our general additive model presented in Chapter 5, the incorporation of the B-spline basis functions from Section 3.2 for a continuous covariate x is straightforward. A sequence of knots t_j are placed equidistantly within the range $[x_{min}, x_{max}]$, with a total of s internal knots, to give

$$f(x) = \alpha_0 + \sum_{m=2}^{M} \alpha_m B_m(x|k)$$

where k is the order of the B-splines and $M = k + s$ is the number of basis functions, remembering we start at $m = 2$ to ensure identifiability. The values of the B-spline basis functions evaluated at x, $B_m(x)$, are then fit as covariates in the model instead of the x variable. Monotonic splines can be incorporated in the same manner, using the summed splines as discussed in Section 3.2.2.

These B-splines can be fit in both the mean and/or the variance model, and may differ in the number of internal knots in each model in order to give more flexibility to the respective curves. However, the more knots that are incorporated in the model, the more parameters that are fit (and hence less degrees of freedom). The use of information criteria will aid in the choice of the optimal number of knots, and thus the best model, for these data. This will be explored using the four criteria discussed in Section 3.2.3.

6.3 Standard error estimation

Since the spline model is just a special case of the multiple regression model, the estimation of the standard errors using the information matrix is the same as discussed previously in Section 5.2.1. Estimation of the standard error by bootstrapping is also straightforward, although care must be taken in how the spline basis values are determined. Rather than simply taking a sample by replacement of the basis functions $B_m(x)$, we instead take a sample with replacement of the covariate x. We then compute the B-spline basis functions for this sample, with s equidistant internal knots.

With this additional step to calculate the function for each bootstrap sample, the basis functions calculated for each of the bootstrap samples may be different.

Once the entire collection of bootstrap replications have been fit, the basis function for each of the samples is calculated over the interval $[x_{min}, x_{max}]$, producing a collection of bootstrap regression functions. A 95% CI at any given x value can be obtained by using the 2.5% and 97.5% percentiles of these regression functions evaluated at each x value. This will be illustrated in Section 6.6.

6.4 Monotonic splines

The VCF data has been presented as an example in previous chapters. Chapter 5 demonstrated that assuming a linear variance model was the optimal model among the models given in Table 5.3. However, the step function above in Figure 6.1 implies that perhaps a non-linear semi-parametric model in the variance may be more appropriate.

While monotonic step functions can be of use, it is important to note the large number of degrees of freedom that must be used in order to fit these models. The use of monotonic splines that were discussed previously in Section 3.2.2 can also enable a monotonic fit with a reduced number of degrees of freedom, compared to a step function.

For the VCF data, monotonic splines of increasing knots were fit and compared with the various information criteria. As per the AIC and the AICc, the optimal model is with four internal knots in the variance. This model is shown in Figure 6.2 along with the step function, and the B-spline basis function with no monotonic constraint (for completeness). It is seen that both spline functions follow the step function, in a smooth and more realistic way. It is also seen that the non-monotonic spline allows for decreases in the variance. These are probably spurious for this particular example, although in later examples we will see that non-monotonic variance functions are necessary in some contexts.

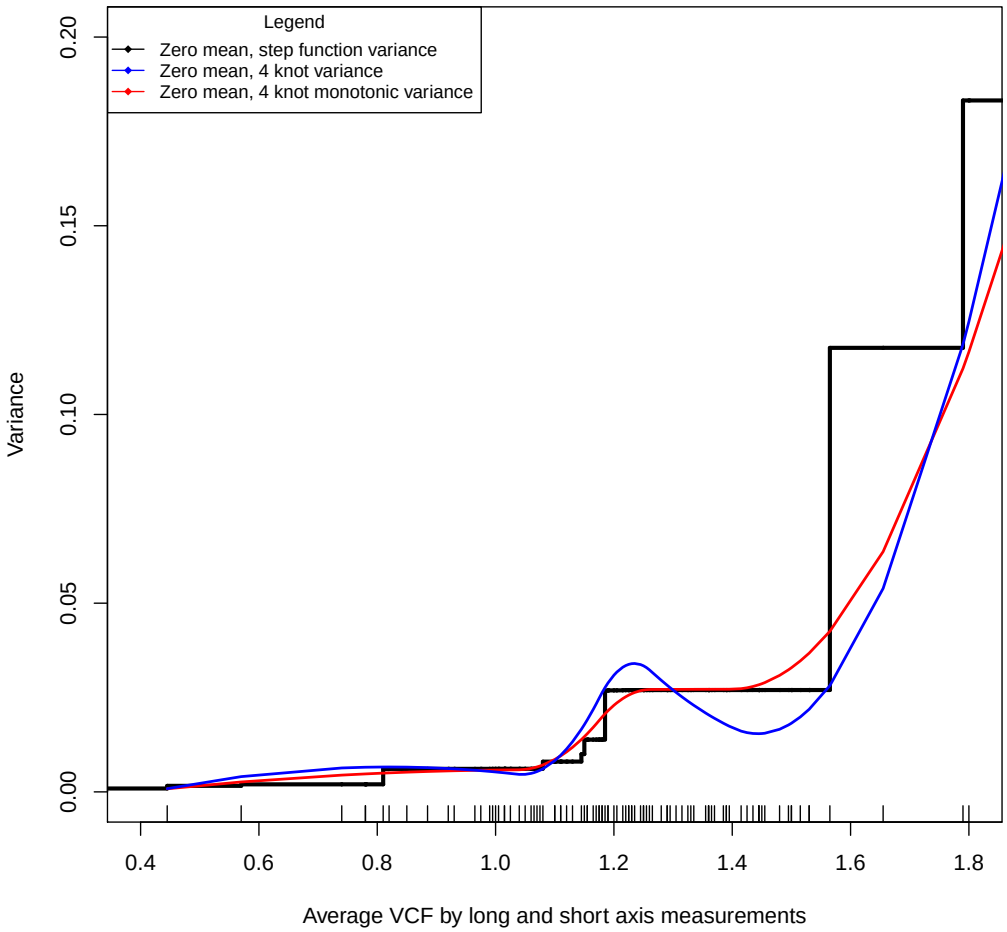

FIGURE 6.2: A comparison of various different variance models for the VCF dataset, each with zero mean. Each unique data point is represented as a tick mark on the inside of the x-axis.

6.5 Simulations

The data for the simulation study were sampled with a zero mean and two different variance functions: one monotonically increasing, and one increasing and decreasing periodically. Both functions were created from B-spline basis functions with two internal knots. Four different dataset sizes were also compared: 100, 250, 500 and 1000. For each combination, 500 simulations were performed.

The first component of the study was to ensure that the algorithm could reliably estimate the two variance functions, when the number of internal knots was known. The next component was to compare the various information criteria (as described in Section 3.2.3) with respect to their ability to select the optimal number of knots.

6.5.1 Estimating known functions

Two known functions, both with two internal knots, were sampled from with four different sample sizes, giving a total of eight simulation studies. A total of 500 simulations were performed, and the median and 2.5% and 97.5% percentiles for the estimates were obtained (Figure 6.3). From this figure, it is clear that the algorithm is capable of recovering the true model, even at small sample sizes of 100 observations.

For the monotonically increasing variance function, monotonic spline models were also fit with two internal knots to compare to the non-monotonic spline models. Figure 6.4 compares these two methods for the splines, over the four different sample sizes. At small sample sizes, the monotonic splines have slightly less variation than the non-monotonic splines, however at larger sample sizes, this difference is minimal. Both reliably estimate the function over the various sample sizes. This suggests that there is a small efficiency advantage in assuming a monotonic model, when the true variance function is monotonic.

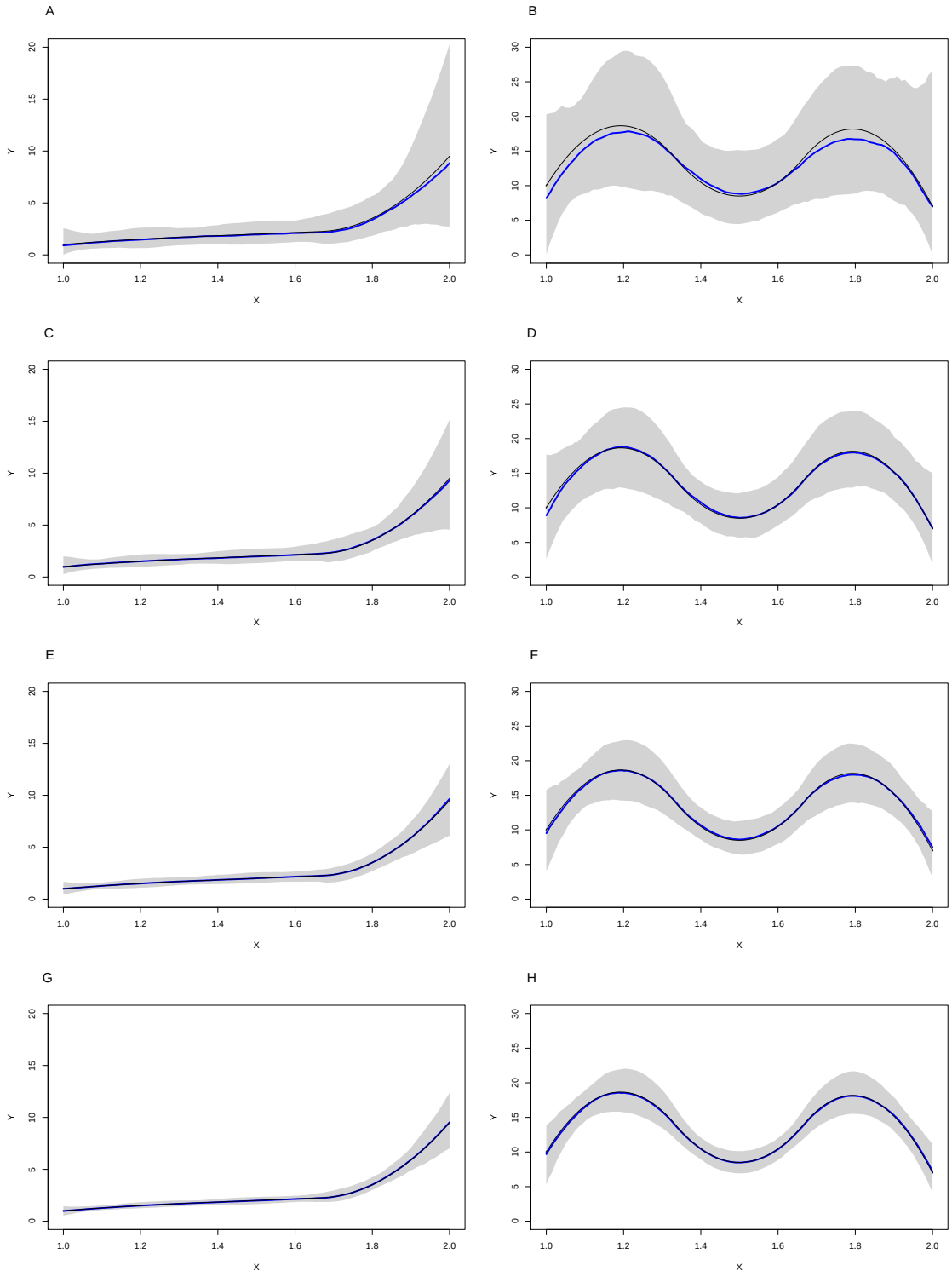

FIGURE 6.3: A comparison of the monotonically increasing variance function (A, C, E, G) and the periodic variance function (B, D, F, H) for 100 (A, B), 250 (C, D), 500 (E, F) and 1000 (G, H) observations. All models use two internal knots for each simulation, shown in blue. Black lines indicate the true function and grey areas indicate the respective 2.5% and 97.5% percentiles of the 500 simulations.

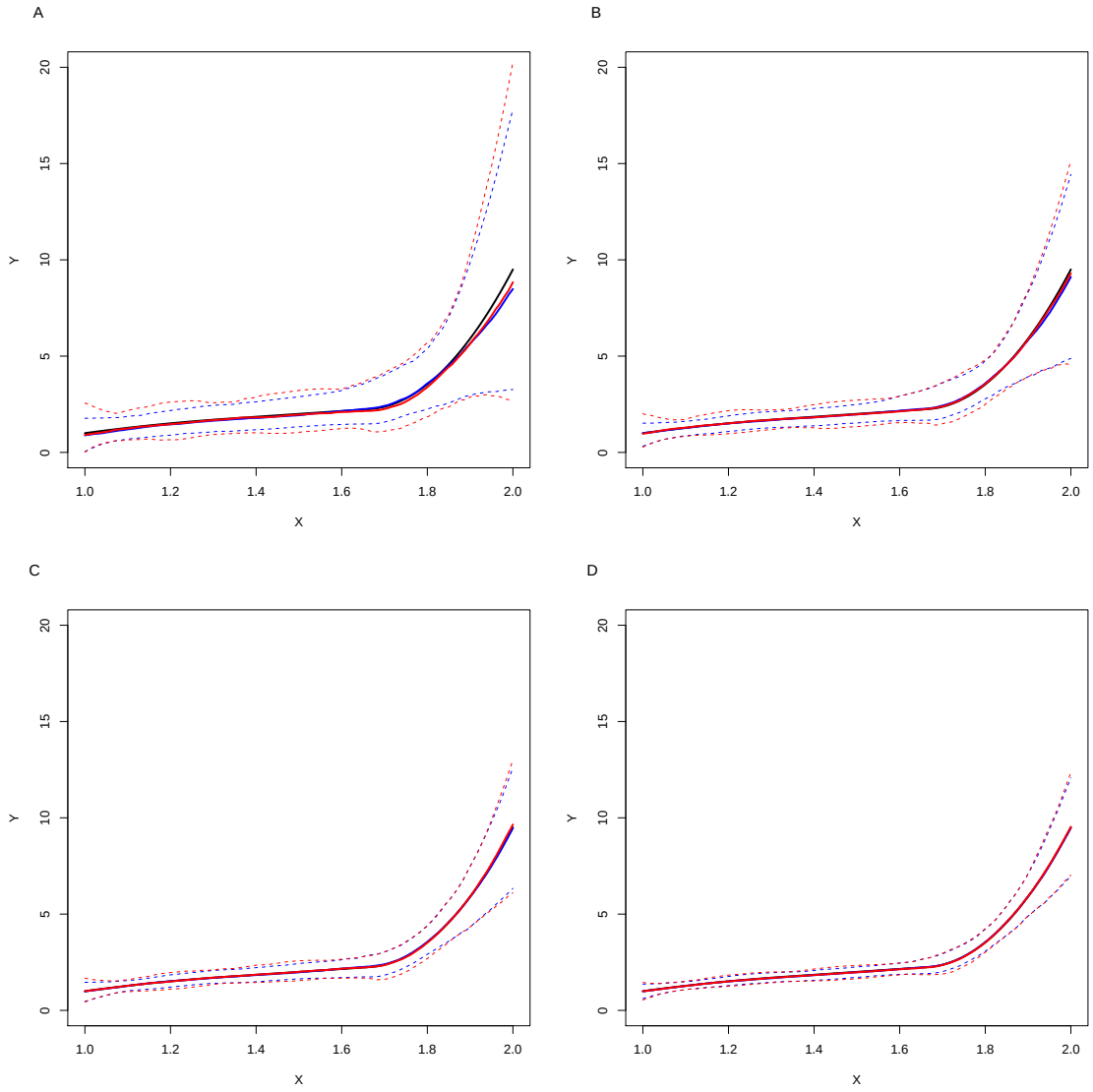

Figure 6.4: A comparison of the monotonically increasing variance function for normal splines (red) and monotonic splines (blue), for 100 (A), 250 (B), 500 (C) and 1000 (D) observations. All models use two internal knots for each simulation. Black lines indicate the true functions and dotted lines indicate the respective 2.5% and 97.5% percentiles for the 500 simulations.

6.5.2 Automatic choice of model complexity

The next aspect of these simulation studies was the choice of the optimal number of knots. Six models were performed for each simulated dataset, from two parameters (linear), three parameters (zero internal knots), up to seven parameters (four internal knots). The true model was the five parameter model with two internal knots. Once the six models were performed, the optimal number of knots was selected for each information criterion based on the lowest value for each of the four criteria; AIC, AICc, BIC and HQC. A summary of the optimal models for each of the information criteria is shown in Table 6.1. At small sample sizes, the BIC and HQC heavily favoured the simplistic linear model, while the AIC and AICc favoured a zero internal knot model (three parameters). When the number of observations increases to over 500, the AIC and AIC average out to choose the true number of parameters (five parameters). The ability to estimate the model complexity thus depends highly on the sample size. The HQC also averaged out to choose the true number of parameters for the periodic variance function, but not for the increasing variance function. For large n, the AIC, AICc and at times, the HQC, were all able to estimate the correct number of knots. However, the BIC still tended to oversmooth, particularly for the increasing variance. For small n, most of the criteria oversmoothed the data.

TABLE 6.1: Information criteria results from simulation studies. Numbers reported are the number of parameters in the model, with five parameters (two internal knots) the true model.

		Increasing variance				Periodic variance			
N	Statistic	AIC	AICc	HQC	BIC	AIC	AICc	HQC	BIC
100	Median	3.0	3.0	2.0	2.0	3.0	3.0	2.0	2.0
	Mean	3.5	3.3	2.9	2.5	3.8	3.5	2.8	2.2
	Mode	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
250	Median	4.0	4.0	4.0	3.0	5.0	5.0	5.0	2.0
	Mean	4.2	4.2	3.5	2.9	4.9	4.9	3.7	2.6
	Mode	4.0	4.0	4.0	2.0	5.0	5.0	5.0	2.0
500	Median	5.0	5.0	4.0	4.0	5.0	5.0	5.0	5.0
	Mean	4.6	4.6	4.1	3.5	5.2	5.2	4.6	3.6
	Mode	4.0	4.0	4.0	4.0	5.0	5.0	5.0	5.0
1000	Median	5.0	5.0	4.0	4.0	5.0	5.0	5.0	5.0
	Mean	5.0	5.0	4.5	4.1	5.4	5.4	5.1	4.9
	Mode	5.0	5.0	4.0	4.0	5.0	5.0	5.0	5.0

These results support the use of information criteria for the automatic selection of model complexity in larger sample sizes. The AIC was preferred, with the AICc not offering any advantages over AIC, and the BIC was subject to over-smoothing. In small samples, all criteria over-smoothed to some extent. This suggests that in small samples we should explore models with complexity greater than the automatically selected model.

6.6 Application of semi-parametric models

This section will focus on the use of example datasets to demonstrate the semi-parametric modelling of both the mean and the variance in the CEM algorithm. Various information criteria can be used to aid in the selection of the most appropriate model, with the previous simulation study suggesting in small sample sizes that the AIC may be preferable.

6.6.1 Analysis example 1

The motorcycle crash dataset was explained in Section 3.3. From the plot in Figure 3.4, it is clear that a non-linear model for both the mean and variance is required. To determine the optimal number of knots, a series of models were fit ranging from a linear model in each, up to eight knots in the mean and eight knots in the variance. The number of observations is 133, so therefore the AIC will be used in order to determine the model complexity. Table 6.2 gives the AIC for these 100 models that were fit, with the optimal model (lowest AIC) in bold. Note that for this dataset, the AIC, AICc, BIC and HQC all agreed that the model with six knots in the mean and six knots in the variance was the model of best fit. This optimal model with 6 knots in the mean and the variance is shown below in Figure 6.5. The normal scores plot and histogram of the residuals from this model are also given, and it can be seen that the residuals follow an approximately normal distribution.

TABLE 6.2: The AIC from the 100 different mean and variance models for the motorcycle crash data. The lowest AIC is in boldface.

Variance	Mean									
	Linear	0 knots	1 knot	2 knots	3 knots	4 knots	5 knots	6 knots	7 knots	8 knots
Linear	1377	1367	1347	1291	1263	1224	1204	1188	1196	1200
0 knots	1332	1355	1326	1289	1233	1203	1190	1175	1183	1186
1 knot	1363	1320	1327	1273	1226	1185	1163	1144	1151	1155
2 knots	1346	1309	1299	1277	1226	1187	1165	1147	1148	1152
3 knots	1341	1252	1299	1266	1220	1184	1160	1144	1146	1150
4 knots	1279	1264	1266	1244	1231	1179	1152	1138	1141	1146
5 knots	1252	1239	1241	1231	1225	1174	1141	1118	1130	1136
6 knots	1241	1234	1235	1235	1230	1145	1107	**1075**	1101	1112
7 knots	1241	1231	1232	1217	1219	1147	1122	1103	1115	1117
8 knots	1240	1220	1223	1202	1193	1149	1107	1089	1100	1104

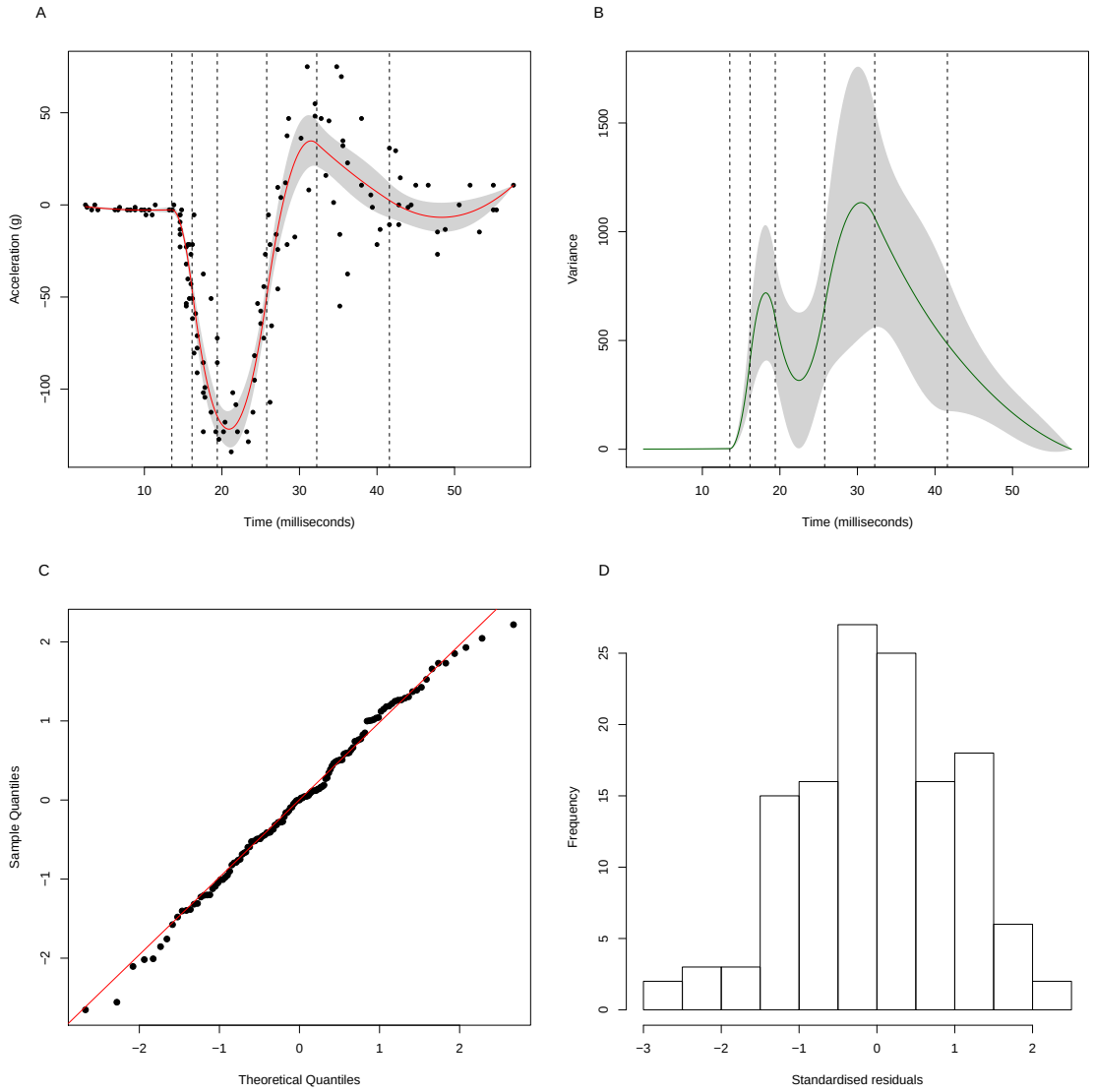

FIGURE 6.5: The optimal model for the motorcycle crash dataset. A: Mean model fitted in red with six internal knots (dashed vertical lines). Data represented as points, with 95% CI in grey. B: Variance model also with 6 internal knots (dashed vertical lines). 95% CI in grey. C: Q-Q plot of the standardised residuals and D: histogram of the standardised residuals.

Table 6.3: The AIC from the 100 different mean and variance models for the LIDAR data. The lowest AIC is in boldface.

Variance model	Mean model									
	Linear	0 knots	1 knot	2 knots	3 knots	4 knots	5 knots	6 knots	7 knots	8 knots
Linear	-270	-472	-482	-580	-558	-608	-595	-610	-605	-605
0 knots	-284	-504	-503	-609	-578	-636	-618	-636	-632	-632
1 knot	-339	-502	-501	-614	-577	-648	-625	-648	-644	-644
2 knots	-313	-506	-504	-620	-581	-649	-632	**-651**	-645	-645
3 knots	-330	-507	-505	-617	-577	-649	-630	-648	-644	-644
4 knots	-314	-509	-507	-617	-579	-647	-630	-646	-643	-642
5 knots	-317	-508	-507	-614	-577	-645	-630	-645	-641	-641
6 knots	-331	-504	-502	-612	-574	-643	-629	-645	-642	-641
7 knots	-322	-505	-513	-613	-577	-642	-628	-644	-640	-640
8 knots	-318	-510	-508	-611	-577	-639	-628	-644	-640	-640

6.6.2 Analysis example 2

The LIDAR dataset was introduced in Section 3.3. A series of models with increasing knots in the mean and variance were fit, in order to find the optimal model. Given that we have 221 observations, the AIC will be used again to determine the optimal model. The AIC for these 100 models are given in Table 6.3, with the lowest and therefore optimal model given in bold. The AIC, AICc and HQC all chose this model with six knots in the mean and two knots in the variance as the optimal model (Figure 6.6). The residuals are also shown in this plot, which show a small deviation from normality in the upper tail.

6.7 Final comments

This chapter illustrated the straightforward extension of the CEM algorithm developed in the previous chapter to semi-parametric modelling. This algorithm has broad applicability, as models can easily incorporate splines with varying degrees of knots that may be fit in the mean and variance. Additionally, bases could be created for more than one covariate and incorporated into the models. The flexibility of this algorithm therefore provides a broad framework for fitting semi-parametric models in the variance, and this was demonstrated in two example datasets.

From the simulation study, the CEM algorithm developed is capable of recovering the true model for all sample sizes. However, the ability to estimate the model complexity depends largely on the sample size. Automatic model selection works well for large

n, when the AIC should be used. For small n, the AIC should be supplemented with visual inspection of models with more parameters.

However, the algorithm developed here is restricted to outcome data that follows an approximately normal distribution. With the LIDAR data, there was a small deviation from normality in the standardised residuals. The next chapter will introduce an algorithm for use with censored outcome data. This censored data algorithm will then be built on in subsequent chapters, in order to develop another algorithm to simultaneously estimate the location, scale and skewness of the distribution, as a function of covariates. This will give the algorithm even broader applicability and flexibility for non-normal data.

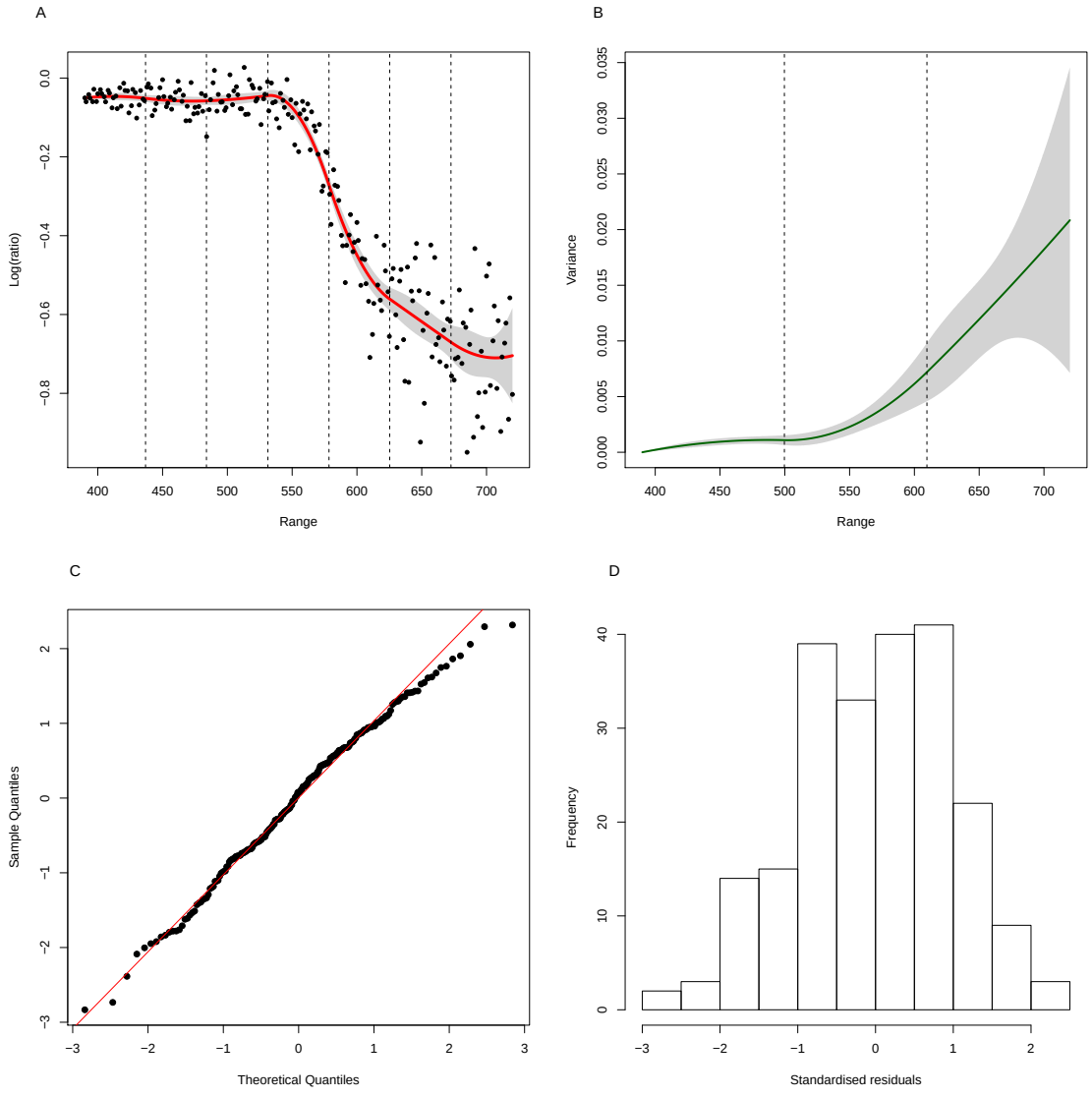

Figure 6.6: The optimal model for the LIDAR dataset. A: Mean model fitted in red with 6 internal knots (dashed vertical lines). Data represented as points, with 95% CI in grey. B: Variance model also with 2 internal knots (dashed vertical lines). 95% CI in grey. C: A Q-Q plot of the standardised residuals and D: a histogram of the standardised residuals.

7

Censored data

This chapter introduces censored data, and extends the CEM algorithm for analysis of the mean and variance when the data are left censored, right censored or a mixture of both. This chapter also provides the basis for the development of an algorithm in the next chapter to fit location, shape and scale regression models.

Censoring often occurs in the analysis of biomarker data. When samples are analysed and measured by an assay, these assays usually have a 'detectable limit' which determines the lowest and highest measurements that can be accurately made by the assay. For example, when measuring the blood concentration of HIV RNA (called HIV viral load) some assays cannot reliably measure the amount of viral load below 50 copies per ml of blood. Newer, more sensitive assays are always emerging that can detect lower quantities, however, they still have a limit beyond which the amount of viral load cannot be accurately measured. A similar situation arises with many laboratory assays.

There has been a steady increase in the number of biomarkers being investigated in

TABLE 7.1: Different types of censored data based on a difference in two measurements.

Measurement two	Measurement one	Difference (Measurement two - Measurement one)
Exact	Exact	No censoring
Exact	Right censored	Left censored
Exact	Left censored	Right censored
Left censored	Exact	Left censored
Right censored	Exact	Right censored
Left censored	Right censored	Left censored
Right censored	Left censored	Right censored
Right censored	Right censored	No information
Left censored	Left censored	No information

clinical trials in many different research fields. We need to be able to accurately model data with potentially large amounts of data below the detectable limit, or data with both lower and upper limits reached. Some studies report over one-third of their biomarker data as below the detectable limit (White et al., 2014).

When biomarker data can be below the detectable limit (left censored), and/or above the detectable limit (right censored), differences between two biomarker readings will also be censored. Biomarker data is commonly given on a log scale, therefore ranges over $(-\infty, \infty)$. Examples of the censoring of the difference between two readings of biomarker data are shown in Table 7.1.

In this chapter, we will consider the viral load dataset that was briefly explored in Section 3.3. These data are repeated measurements of blood concentration of HIV RNA on a $\log_{10}$ scale, and were obtained by assaying the blood of an infected individual twice in a short interval of time prior to the commencement of a clinical trial (Kuritzkes et al., 1999). Although the underlying viral load is unchanged in this short interval of time, the measurements will differ due to measurement error. Additionally, the assay has a lower detection limit of $\log_{10}(500)$, or 2.70. If $V_i^{(1)}$ is the first viral load reading, and $V_i^{(2)}$ is the second, then the usual measurement error model for the repeated measurements is

$$V_i^{(j)} = V_i^* + \epsilon_j \quad \text{where } \epsilon_j \sim N(0, \sigma^2) \quad j = 1, 2$$

where V_i^* is the underlying viral load for the ith patient. If we assume independence of the measurement errors for different measurements, we can assume that the difference

in the repeated measurements, $X_i = V_i^{(1)} - V_i^{(2)}$, follows a normal distribution with zero mean and variance $2\sigma^2$. This can be used to estimate the measurement error variance σ^2.

Figure 7.1 shows the difference between the repeated measurements, versus the average of the two measurements, where any data below the detectable limit is set at the detectable limit. While these data appear to have zero mean in the difference of the observations, the variance appears to have a reverse fanning pattern. This demonstrates larger variance in the difference between the observations in smaller observations. A variance regression model would need to be undertaken to model this decreasing variance, in particular to take account of the censored data that is clearly evident in the plot.

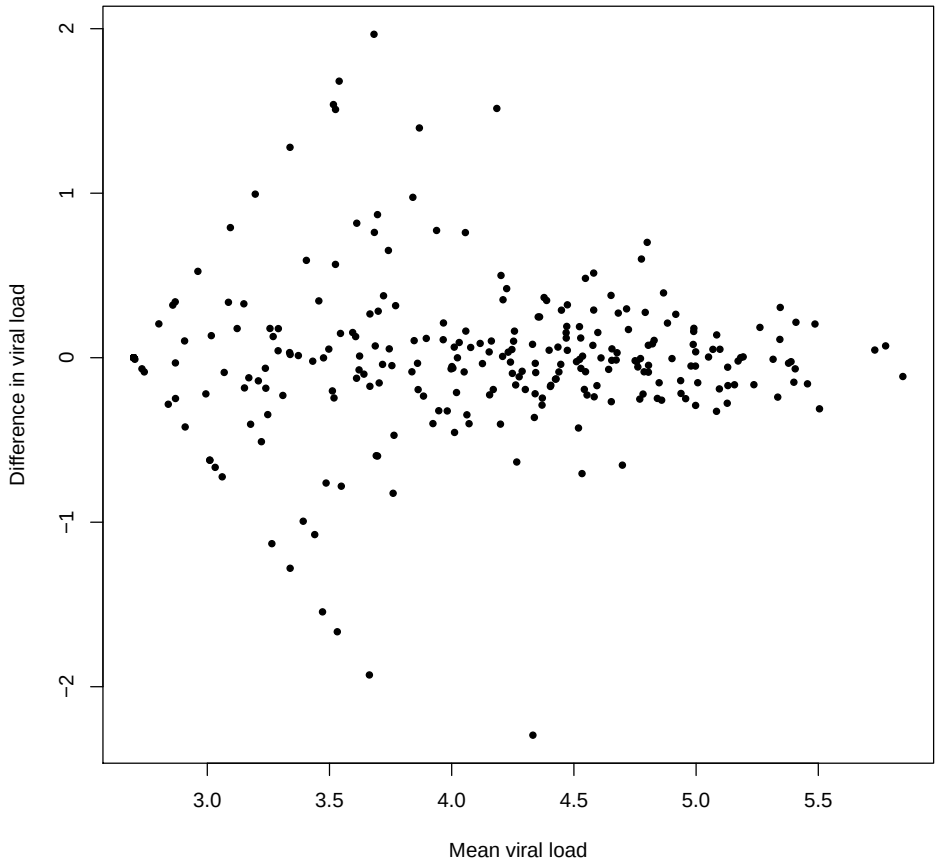

FIGURE 7.1: Bland-Altman plot of two HIV viral load measurements.

7.1 Fitting details

If X_i^* is our true outcome data, let X_i be the outcome data observed with censoring, and let us account for lower and upper limit censoring for completeness. This censored data gives an additional level of missingness to our CEM algorithm, and thus an extra level of our complete data. If $X^{(L)}$ is the lower limit of detection and $X^{(U)}$ is the upper limit of detection,

$$X_i = \begin{cases} X_i^*, & \text{if } X^{(L)} \leq X_i^* \leq X^{(U)} \\ X^{(L)}, & \text{if } X_i^* < X^{(L)} \\ X^{(U)}, & \text{if } X_i^* > X^{(U)} \end{cases} \qquad \text{for } i = 1, 2.., n. \tag{7.1}$$

We will also need to specify c, the censoring indicator, where

$$c_i = \begin{cases} 0, & \text{if } X^{(L)} \leq X_i^* \leq X^{(U)} \\ -1, & \text{if } X_i^* < X^{(L)} \\ 1, & \text{if } X_i^* > X^{(U)}. \end{cases}$$

Similarly to previous chapters, let the mean model have covariates $\boldsymbol{z}_i = (z_{i1}, ..., z_{iP})$, with coefficients $\boldsymbol{\beta} = (\beta_1, ..., \beta_P)$; and the variance model have covariates, $\boldsymbol{x}_i = (x_{i1}, ..., x_{iQ})$ with coefficients $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_Q)$. Also, let us assume that the covariates x_{iq} are scaled such that $x_{iq} \in [0, 1]$. For simplicity, let

$$\mu(\boldsymbol{\beta}) = \beta_0 + \sum_{p=1}^{P} \beta_p z_{ip} \quad \text{and} \quad \sigma^2(\boldsymbol{\alpha}) = \alpha_0 + \sum_{q=1}^{Q} \alpha_q x_{iq}.$$

If Φ is the standard normal cumulative distribution function and ϕ is the standard normal probability density function, then two useful functions are $R(z) = \dfrac{\phi(z)}{1 - \Phi(z)}$ and $Q(z) = \dfrac{-\phi(z)}{\Phi(z)}$ (Mills, 1926). Additionally, when z tends to negative infinity, the approximation $\dfrac{\phi(z)}{\Phi(z)} \approx -z$ can be used.

The likelihood function for the observed data model with censored data is

$$l(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} l_i$$

where

$$l_i = \begin{cases} \frac{1}{\sigma(\boldsymbol{\alpha})}\phi\left(\frac{X_i - \mu(\boldsymbol{\beta}))}{\sigma(\boldsymbol{\alpha})}\right), & \text{if } c_i = 0 \\ 1 - \Phi\left(\frac{\mu(\boldsymbol{\beta}) - X_i^{(L)}}{\sigma(\boldsymbol{\alpha})}\right), & \text{if } c_i = -1 \\ \Phi\left(\frac{\mu(\boldsymbol{\beta}) - X_i^{(U)}}{\sigma(\boldsymbol{\alpha})}\right), & \text{if } c_i = 1. \end{cases}$$

The corresponding log-likelihood is then

$$\ell(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \sum_{i=1}^{N} \ell_i,$$

where

$$\ell_i = \begin{cases} \log\left(\frac{1}{\sigma(\boldsymbol{\alpha})}\phi\left(\frac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}\right)\right), & \text{if } c_i = 0 \\ \log\left(1 - \Phi\left(\frac{\mu(\boldsymbol{\beta}) - X_i^{(L)}}{\sigma(\boldsymbol{\alpha})}\right)\right), & \text{if } c_i = -1 \\ \log\left(\Phi\left(\frac{\mu(\boldsymbol{\beta}) - X_i^{(U)}}{\sigma(\boldsymbol{\alpha})}\right)\right), & \text{if } c_i = 1. \end{cases}$$

Now in the CEM algorithm, the uncensored outcome variable X_i^* will be assumed to be composed of $Q + 1$ independent, unobserved, latent variables:

$$X_i^* = Y_i + Z_{i1} + ... + Z_{iQ} \quad \text{where} \quad Y_i \sim N(0, \alpha_0) \quad \text{and}$$
$$Z_{i1} \sim N(0, \alpha_1 x_{i1}), ..., Z_{iQ} \sim N(0, \alpha_Q x_{iQ}). \quad (7.2)$$

This means that we will need to find the conditional expectations of Y_i^2 and $Z_{i1}^2, ..., Z_{iQ}^2$, given the observed outcome value, X_i, which may be censored.

From Aitkin (1964), we see that for bivariate normal variables M and N, where

$$\begin{pmatrix} M \\ N \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right],$$

the conditional distribution for M given censoring in N can be obtained. In particular, given an upper limit, a, it was shown that:

$$\mathbb{E}(M^2 \mid N > a) = \frac{a\rho^2}{R(a)} + 1, \tag{7.3}$$

where ρ is the correlation between M and N and $R(a)$ was defined previously. We know from (7.2) that the variance for Y_i is α_0, and the variance for X_i is $\sigma^2(\boldsymbol{\alpha})$. So, given (7.2) and the general result (7.3), it follows that,

$$\begin{aligned} \mathbb{E}(Y_i^2 | X_i > X^{(U)}) &= \alpha_0 \mathbb{E}\left(\frac{Y_i^2}{\alpha_0}\middle| \frac{X_i}{\sigma(\boldsymbol{\alpha})} > \frac{X^{(U)}}{\sigma(\boldsymbol{\alpha})}\right), \\ &= \alpha_0 \left(\frac{\dfrac{X^{(U)}}{\sigma(\boldsymbol{\alpha})} Corr(Y_i, X_i)^2}{R\left(\dfrac{X^{(U)}}{\sigma(\boldsymbol{\alpha})}\right)} + 1\right). \end{aligned} \tag{7.4}$$

We can then determine the correlation between Y_i and X_i:

$$\begin{aligned} Corr(Y_i, X_i) &= \frac{cov(Y_i, X_i)}{\sqrt{\alpha_0 \sigma^2(\boldsymbol{\alpha})}} \\ &= \frac{cov(Y_i, Y_i + Z_i)}{\sqrt{\alpha_0 \sigma^2(\boldsymbol{\alpha})}} \\ &= \frac{cov(Y_i, Y_i) + cov(Y_i, Z_i)}{\sqrt{\alpha_0 \sigma^2(\boldsymbol{\alpha})}} \\ &= \frac{\alpha_0 + 0}{\sqrt{\alpha_0 \sigma^2(\boldsymbol{\alpha})}}, \\ &= \frac{\sqrt{\alpha_0}}{\sigma(\boldsymbol{\alpha})}. \end{aligned} \tag{7.5}$$

We can also determine the correlation for each of the Z_{iq} and X_i, using a similar argument:

$$Corr(Z_{iq}, X) = \frac{\sqrt{\alpha_q x_{iq}}}{\sigma(\boldsymbol{\alpha})}. \tag{7.6}$$

Now we apply the correlation found in (7.5) to (7.4), in order to obtain our conditional expectation of Y_i^2:

$$\mathbb{E}(Y_i^2 | X_i > X^{(U)}) = \alpha_0 + \frac{\alpha_0^2 X^{(U)}}{\sigma^3(\boldsymbol{\alpha}) R\left(\frac{X^{(U)}}{\sigma_X}\right)}. \tag{7.7}$$

The expectation of each of the Z_{iq}^2 follows the same argument, using (7.6).
In order to apply these expectations to left censored data, we use the function $Q(z)$ rather than $R(z)$.

The complete data log-likelihood is the same as given in Chapter 5 in Equation (5.2), and is linear in Y_i^2 and Z_{iq}^2. For censored outcome data, the E-step involves the calculation of the conditional expectations as defined in Chapter 5 in Equation (5.3):

$$\hat{Y}_i^2(\boldsymbol{\theta}) = \begin{cases} \alpha_0 + \dfrac{\alpha_0^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}\right)^2}{\sigma(\boldsymbol{\alpha})} - 1 \right), & \text{if } c_i = 0 \\ \alpha_0 + \dfrac{\alpha_0^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}}{Q\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}\right)} \right), & \text{if } c_i = -1 \\ \alpha_0 + \dfrac{\alpha_0^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}}{R\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}\right)} \right), & \text{if } c_i = 1, \end{cases} \tag{7.8}$$

remembering that $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\alpha})$, and X_i is defined in (7.1). The conditional expectations associated with the Z_{iq} follow the same principles,

$$\hat{Z}_{iq}^2(\boldsymbol{\theta}) = \begin{cases} \alpha_q x_{iq} + \dfrac{(\alpha_q x_{iq})^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})} \right)^2}{\sigma(\boldsymbol{\alpha})} - 1 \right), & \text{if } c_i = 0 \\ \alpha_q x_{iq} + \dfrac{(\alpha_q x_{iq})^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}}{Q\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})} \right)} \right), & \text{if } c_i = -1 \\ \alpha_q x_{iq} + \dfrac{(\alpha_q x_{iq})^2}{\sigma^2(\boldsymbol{\alpha})} \left(\dfrac{\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}}{R\left(\dfrac{X_i - \mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})} \right)} \right), & \text{if } c_i = 1. \end{cases} \tag{7.9}$$

The next step of the algorithm involves calculating the updated estimates of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}^{new}$. The estimates for $\hat{\boldsymbol{\alpha}}$ for fixed $\boldsymbol{\beta}$ are obtained by the following,

$$\hat{\alpha}_0^{new} = n^{-1} \sum_{i=1}^{n} \hat{Y}_i^2 \left(\hat{\boldsymbol{\theta}}^{old} \right) \quad \text{and} \quad \hat{\alpha_q}^{new} = \frac{\sum_{i=1}^{n} \hat{Z}_{iq}^2(\boldsymbol{\theta}^{old})}{n \sum_{i=1}^{n} x_{iq}}. \tag{7.10}$$

Previously in Chapter 5, a weighted linear regression was fit in order to obtain an updated estimate of the mean parameters ($\hat{\boldsymbol{\beta}}$) at each iteration, for fixed $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}^{new}$. In order to estimate the mean model with censored outcome data, we need to fit a heteroscedastic censored linear regression model. This can be achieved by first standardising the data, and then performing a homoscedastic censored linear regression at each iteration. In order to standardise the data, we divide by the standard deviation to obtain

$$\frac{X_i}{\sigma(\boldsymbol{\alpha})} \sim \text{censored } N\left(\frac{\mu(\boldsymbol{\beta})}{\sigma(\boldsymbol{\alpha})}, 1 \right). \tag{7.11}$$

A homoscedastic censored linear regression for $\dfrac{X_i}{\sigma(\boldsymbol{\alpha})}$ is then performed, against covariates $\dfrac{z_{ip}}{\sigma(\boldsymbol{\alpha})}$, for fixed $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}^{new}$. This can easily be implemented in standard software for censored normal linear regression, which can be performed with `survreg` in R. Once the censored regression is performed, the $\hat{\boldsymbol{\beta}}$ estimates are back transformed by

multiplying by the standard deviation $\sigma(\boldsymbol{\alpha})$ for our current fixed $\boldsymbol{\alpha}$. This process is continued until convergence of the parameter estimates.

This algorithm is an instance of the ECME algorithm, and is summarised schematically in Figure 7.2. As detailed in Section 3.1.2 and 5.1, the algorithm maximises the log-likelihood over a restricted parameter space, and will need to be run multiple times in order to maximise over the full parameter space. Thus a total of 2^Q ECME algorithms must be run, once for each combination of the q^{th} variance covariate taking the value x_i or $1 - x_i$, for $q = 1, 2, ..., Q$. The log-likelihood is then maximised over the entire parameter space with this family of ECME algorithms.

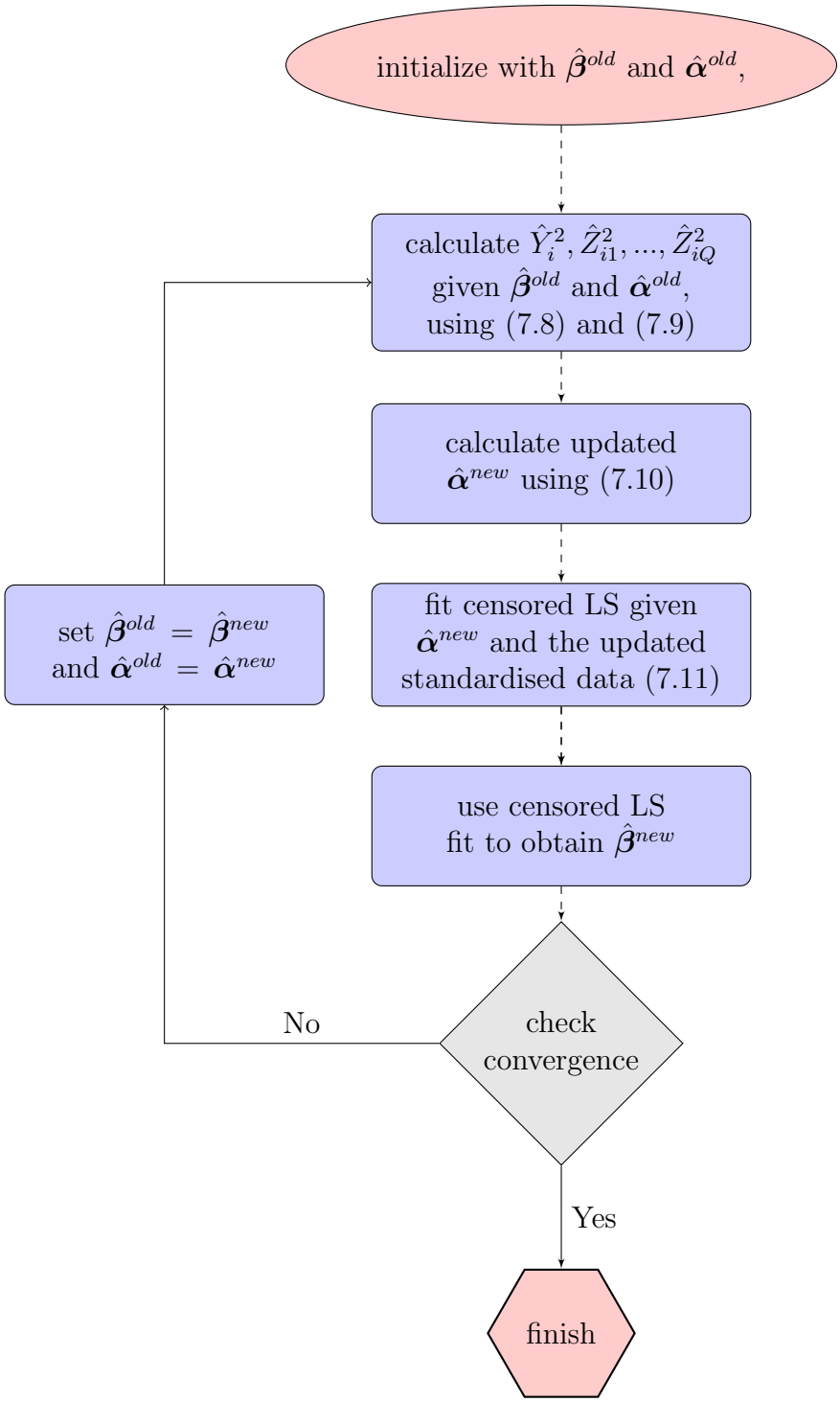

FIGURE 7.2: The ECME algorithm for the estimation of the mean and variance with censored outcome data.

7.2 Standard error estimation

The standard errors of the parameter estimates may be obtained by bootstrapping techniques, as presented in previous chapters. A total of B random samples with replacement of size n are taken, and the above algorithm is applied to each of the samples. Typically, B is set to 1000. We then obtain B bootstrap estimates, and take the 2.5% and 97.5% percentiles of these estimates in order to obtain the 95% confidence intervals. In practice, this may take some time for datasets with many parameters fit in the variance model. This is due to the number of parameter spaces, and thus ECME algorithms, that must be run.

7.3 Simulations

A series of simulations were performed to demonstrate this algorithm, with true parameters $\boldsymbol{\theta} = (\alpha, \beta) = (1, 1)$, that is, $X_i \sim (1 + x, 1 + x)$. Four scenarios with increasing left or right censoring were used, with on average 5%, 15%, 25% and 50% censoring, to demonstrate data either below or above a detectable limit. Additionally, simulations were performed with censoring on both the left and right sides, with on average 5%, 15% or 25% censoring on each side. Lastly, three sample sizes were explored; 100, 500 and 1000 observations. A total of 1000 simulations were performed for each combination. The results are summarised in Table 7.2.

Even with 50% censoring and only 100 observations, the mean estimates were unbiased with the highest relative bias being just 2%. For variance estimates, the relative bias was generally no more than 5%, and generally much smaller. The only exception was for cases with censoring at 50% and 100 observations, where we found relative bias in the range of 2.3% to 8.6%. However, if there are 500 observations or more, the relative bias did not exceed 3.5%. In each of the simulations, bias and precision improved with an increasing number of observations, and a lower degree of censoring.

Overall, these results illustrate that the algorithm can reliably estimate the mean and variance parameters, even in datasets with small sample size and a large amount of censoring.

TABLE 7.2: Results from the censoring simulation study, with 1000 simulations performed per row.

Censoring		Mean Intercept		Mean Slope		Variance Intercept		Variance Slope	
Left	Right	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
				N=100					
5%	0%	1.006	0.056	0.987	0.198	1.004	0.155	0.980	0.635
15%	0%	0.999	0.059	0.994	0.208	1.026	0.192	0.956	0.745
25%	0%	1.001	0.065	0.991	0.222	1.021	0.231	0.970	0.871
50%	0%	0.997	0.116	0.992	0.333	1.059	0.503	0.928	1.605
0%	5%	1.008	0.055	0.988	0.201	0.980	0.132	1.016	0.616
0%	15%	1.010	0.056	0.989	0.208	0.987	0.148	1.019	0.754
0%	25%	1.009	0.058	0.996	0.216	0.989	0.172	1.037	0.917
0%	50%	1.018	0.070	0.981	0.293	1.023	0.321	0.994	1.729
5%	5%	1.005	0.056	0.991	0.203	0.999	0.155	1.007	0.696
15%	15%	1.001	0.061	0.997	0.220	1.033	0.231	1.013	1.123
25%	25%	0.999	0.068	1.006	0.241	1.053	0.392	1.086	2.034
				N=500					
5%	0%	0.997	0.010	1.007	0.033	0.987	0.025	1.013	0.113
15%	0%	0.990	0.012	1.013	0.039	0.987	0.033	1.013	0.131
25%	0%	0.997	0.011	1.006	0.037	0.988	0.040	1.020	0.159
50%	0%	0.995	0.020	1.006	0.055	0.992	0.077	1.020	0.272
0%	5%	0.997	0.009	1.008	0.033	0.983	0.024	1.023	0.114
0%	15%	0.997	0.010	1.010	0.034	0.983	0.026	1.026	0.136
0%	25%	0.998	0.010	1.007	0.035	0.986	0.030	1.017	0.164
0%	50%	0.999	0.012	1.002	0.050	0.991	0.049	1.003	0.267
5%	5%	0.997	0.010	1.009	0.033	0.985	0.026	1.023	0.122
15%	15%	0.996	0.010	1.010	0.036	0.989	0.038	1.033	0.181
25%	25%	0.996	0.011	1.008	0.039	0.996	0.059	1.035	0.291
				N=1000					
5%	0%	0.998	0.005	1.005	0.017	0.999	0.013	0.995	0.057
15%	0%	0.996	0.005	1.005	0.017	0.999	0.016	1.002	0.065
25%	0%	0.997	0.005	1.005	0.019	1.001	0.020	0.997	0.077
50%	0%	0.998	0.010	1.002	0.028	1.000	0.037	1.002	0.126
0%	5%	0.997	0.005	1.006	0.017	0.998	0.012	0.997	0.057
0%	15%	0.998	0.005	1.004	0.017	1.000	0.013	0.993	0.065
0%	25%	0.999	0.005	1.004	0.019	1.002	0.015	0.990	0.078
0%	50%	1.000	0.006	1.000	0.026	1.006	0.024	0.981	0.123
5%	5%	0.996	0.005	1.008	0.018	1.005	0.014	0.992	0.066
15%	15%	0.998	0.005	1.004	0.018	1.002	0.018	0.999	0.087
25%	25%	0.997	0.005	1.005	0.020	1.010	0.031	0.994	0.143

7.4 Analysis example

The viral load dataset described earlier in this chapter will be used to demonstrate the proposed method. This dataset contains a total of 285 observations, however 28 have both readings as left censored and therefore the difference is uninformative. These observations are not included in these analyses. A total of 234 observations are measured exactly, while there are 14 that are left censored and 9 that are right censored (5% and 3% respectively). A monotonic step function will be explored first, then a monotonic splines model and finally a non-monotonic splines model.

Following the principles outlined in Section 6.1, a step function is easily incorporated into the above algorithm. The only addition on top of the algorithm detailed above in Section 7.1 is restricting these data to have the first x observation uncensored. For this viral load dataset, the first seven observations are censored (a mixture of left and right censoring), so therefore these observations were removed for the estimation of the monotonic step function. Given that the data are differences in repeat measurements, a zero mean model is assumed, and the variance estimates are restricted to be monotonically decreasing. This is consistent with the natural expectation that measurement error will increase in variability closer to the limit of detection (Álvarez Estévez et al., 2013).

Although this is a flexible method for examining the relationship between a covariate and the censored outcome, the disadvantages are the large number of parameters that need to be estimated, and the discontinuous step-function estimate. An alternative to this method is fitting B-spline basis functions that can be used to fit a smooth, flexible regression line, without requiring a large number of degrees of freedom. A series of models were fit to these data, with an increasing number of parameters in the mean and variance model. The AIC results for each of these models are shown in Table 7.3, with the optimal model that with a zero mean, and two knots fit in the variance.

A series of monotonic B-spline basis functions were also explored, with increasing knots in the mean and variance. From the AIC results of these models given in Table 7.4, the model with the lowest AIC was also that with zero mean and two knots in the variance. A comparison of the monotonic step model, this model with monotonic splines and the

Table 7.3: The AIC from the 36 different non-monotonic mean and variance models for the viral load data. The lowest AIC is in boldface.

Variance	Mean								
	Zero	Constant	Linear	0 knots	1 knot	2 knots	3 knots	4 knots	5 knots
Linear	343.7	345.0	346.6	339.9	342.0	344.2	345.8	346.5	349.2
0 knots	364.5	366.0	363.4	359.7	362.7	365.3	366.8	367.4	370.4
1 knot	322.9	324.8	326.7	327.4	329.0	330.7	332.9	332.7	334.2
2 knots	**299.3**	300.7	302.0	303.0	304.2	306.1	308.1	308.1	309.9
3 knots	299.9	301.0	302.4	303.4	304.6	306.5	308.5	308.5	310.1
4 knots	300.4	302.0	302.8	302.5	304.8	306.5	308.7	308.7	310.7
5 knots	301.4	302.9	303.9	304.6	305.7	307.7	309.7	310.3	311.9

TABLE 7.4: The AIC from the 49 different monotonic mean and variance models for the viral load data. The lowest AIC is in boldface.

Variance	Mean								
	Zero	Constant	Linear	0 knots	1 knot	2 knots	3 knots	4 knots	5 knots
Linear	343.7	345.0	346.6	339.9	342.0	344.2	345.8	346.5	349.2
0 knots	308.7	310.3	312.3	313.8	314.6	316.4	318.4	319.0	321.1
1 knot	301.3	303.0	304.9	306.7	307.8	309.2	311.4	311.7	313.5
2 knots	**299.9**	301.5	303.4	305.0	306.1	307.7	309.9	310.0	311.9
3 knots	300.7	302.3	304.3	305.8	306.9	308.6	310.7	310.7	312.7
4 knots	302.3	304.0	305.9	307.5	308.5	310.0	312.2	312.3	314.2
5 knots	301.4	303.1	305.0	306.4	307.4	309.2	311.3	311.6	313.3

model with non-monotonic splines is shown in Figure 7.3. While the monotonic splines follow the step function, the non-monotonic splines start at a lower variance than that in the monotonic models. While this change in variance may be appropriate in some scenarios, for viral load data it is known that there is higher variability at lower readings (Álvarez Estévez et al., 2013). Therefore the monotonic spline model is the most appropriate for these data.

Figure 7.4 compares the residuals from these models. Given that we have right-censored squared residuals, the distribution can be estimated using standard Kaplan-Meier curves. We can assess the model fit using the chi-squared distribution with one degree of freedom. From the figure, we have a relatively good fit with both monotonic and non-monotonic spline models.

7.5 Final comments

This chapter illustrated the extension of the CEM algorithm for analysis of the mean and variance when the outcome data may be censored. The data may be left censored, right censored or a combination of both. Through a simulation study, we demonstrated that the algorithm can reliably estimate mean and variance parameters with minimal bias, even with up to 50% censoring. We also illustrated its applicability for assessing measurement error for biomarker data that is subject to both left and right censoring. This chapter also provides the basis for the development of an algorithm in the next chapter to fit location, shape and scale regression models.

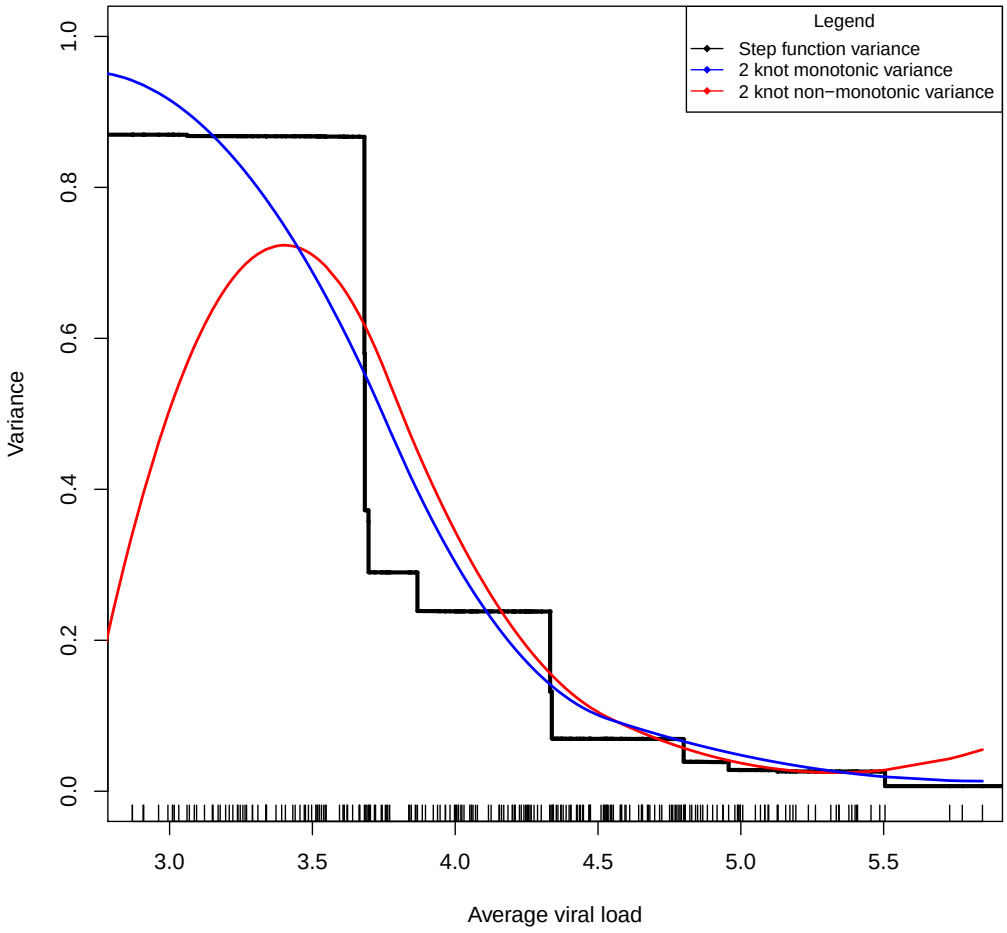

Figure 7.3: The optimal variance models for the viral load dataset. In black is the monotonic step function, in red is the spline model and in blue is the monotonic spline. Each unique data point (w_k) is represented as a tick mark on the inside of the x-axis.

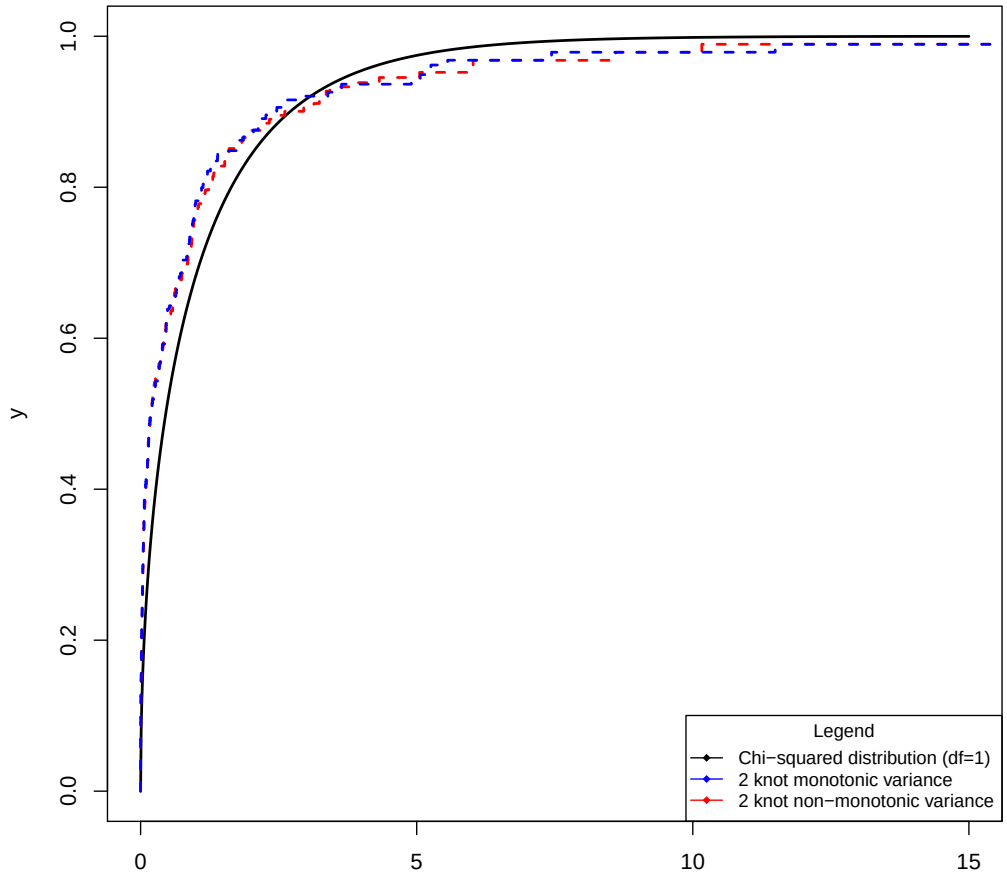

FIGURE 7.4: The censored squared residuals from the optimal variance models for the viral load dataset. In black is the chi-squared distribution with one degree of freedom. In red is the spline model residuals, and in blue is the monotonic spline model residuals. Each unique data point (w_k) is represented as a tick mark on the inside of the x-axis.

8

Skewness models

The focus of this chapter will be to introduce the skew-normal distribution and develop an algorithm to fit a regression model in the location, scale and shape parameters (LSS). The previous chapter introduced an extension of the CEM algorithm for censored data, and that algorithm will be utilised in this chapter. After the development of a simple algorithm to estimate the three parameters in the skew-normal model, the algorithm will be extended to accommodate regression models in location, scale and shape. A simulation study, as well as applications to datasets, will also be performed.

8.1 Skew-normal distribution

The skew-normal distribution is a distribution that extends the normal distribution to allow for non-zero skew (Azzalini, 2013). This distribution has three parameters, the location parameter ξ ($\xi \in (-\infty, \infty)$), the scale parameter ω ($\omega \in (0, \infty)$) and the shape parameter ν ($\nu \in (-\infty, \infty)$). If $\nu < 0$, the distribution is left skewed, and if

$\nu > 0$ then the distribution is right skewed. The normal distribution is recovered with $\nu = 0$ (Figure 8.1).

The probability density function of the skew normal is

$$f(x) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\nu\left(\frac{x-\xi}{\omega}\right)\right) \qquad -\infty < x < \infty$$

where ϕ and Φ are the density and distribution functions of the standard normal distribution, respectively. If a random variable X has a skew-normal distribution with parameters (ξ, ω, ν), this is written as

$$X \sim SN\left(\xi, \omega^2, \nu\right).$$

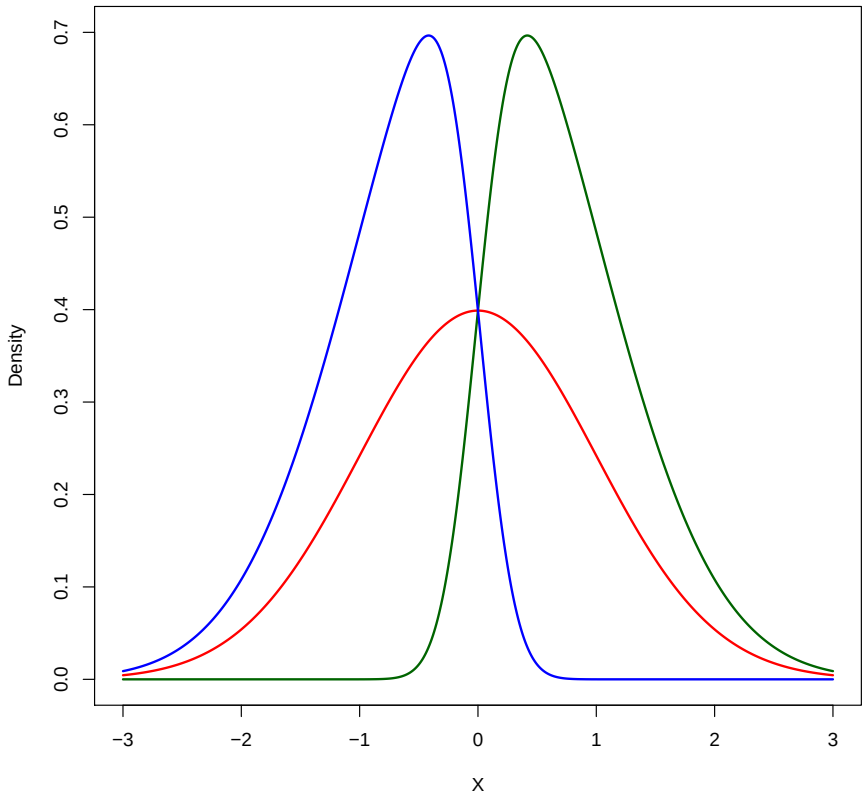

FIGURE 8.1: Examples of the skew-normal density function with $\alpha = 0$ in red, $\alpha = 3$ in green and $\alpha = -3$ in blue.

The mean μ, variance σ^2 and skewness γ can be obtained, in terms of the location, scale and shape parameters (ξ, ω, ν), as follows

$$\mu = \xi + \omega\delta\sqrt{\frac{2}{\pi}} \qquad \text{where } \delta = \frac{\nu}{\sqrt{1+\nu^2}} \tag{8.1}$$

$$\sigma^2 = \omega^2\left(1 - \frac{2\delta^2}{\pi}\right) \tag{8.2}$$

$$\gamma = \frac{4-\pi}{2}\frac{\left(\delta\sqrt{2/\pi}\right)^3}{\left(1-2\delta^2/\pi\right)^{3/2}}. \tag{8.3}$$

In general, members of this distribution have a skewed distribution, except for the special case $\nu = 0$ which corresponds to the normal distribution. While many properties of the skew-normal distribution are listed in Azzalini (2013), one important property is the fact that

$$\left(\frac{X-\xi}{\omega}\right)^2 \sim \chi^2_1$$

irrespective of ν. This property will be especially useful in residual analysis for the regression model presented later in this chapter.

8.2 Maximum likelihood estimation

If $X_1, ..., X_n$ are independent and identically distributed observations from $SN(\xi, \omega^2, \nu)$, the likelihood function for the sample is given as

$$\mathcal{L}(\xi, \omega, \nu) = \frac{2^n}{\omega^n}\prod_{i=1}^{n}\phi\left(\frac{X_i-\xi}{\omega}\right)\Phi\left(\nu\left(\frac{X_i-\xi}{\omega}\right)\right), \tag{8.4}$$

and the corresponding log-likelihood (omitting the constant term) reduces to

$$L(\xi, \omega, \nu) = -n\log(\omega) + \sum_{i=1}^{n}\log\phi\left(\frac{X_i-\xi}{\omega}\right) + \sum_{i=1}^{n}\log\Phi\left(\nu\left(\frac{X_i-\xi}{\omega}\right)\right). \tag{8.5}$$

As will be explained below, the likelihood in (8.4) is a censored Gaussian likelihood when viewed as a function of ξ and ω, and a probit regression likelihood when viewed as a function of ν. This is useful as it allows a straightforward cyclic coordinate ascent

algorithm to be used to obtain the MLE. For optimisation of a multi-variable function, a cyclic coordinate ascent algorithm involves the optimisation of a function with respect to one variable holding the other variables constant, and then repeating with respect to each of the variables (Lange, 2013). In our context, the cyclic coordinate ascent algorithm will cycle between a censored Gaussian model (keeping ν constant), and a probit regression (keeping ξ and ω constant).

Although there already exists an algorithm for obtaining the MLE in the skew-normal model in the `SN` package (Azzalini, 2016) in R, we introduce this new method as it more easily generalises to regression modelling.

In order to see that (8.4) is equivalent to a probit regression likelihood when viewed as a function of ν alone, we note that (8.4) is proportional to the following function of ν,

$$\mathcal{L}_2(\nu|\xi,\omega) = \prod_{i=1}^{n} \Phi\left(\nu\left(\frac{X_i-\xi}{\omega}\right)\right). \tag{8.6}$$

Now let the residuals be defined as follows, for fixed ξ and ω,

$$r_i = \frac{X_i-\xi}{\omega}.$$

Then, (8.6) is

$$\mathcal{L}_2(\nu|\xi,\omega) = \prod_{i=1}^{n} \Phi\left(\nu r_i\right) = \prod_{r_i\geq 0} \Phi(\nu|r_i|) \prod_{r_i<0} \left[1-\Phi(\nu|r_i|)\right].$$

This is exactly a probit regression likelihood with the binary outcome

$$V_i = \begin{cases} 1 & \text{if } r_i \geq 0 \\ 0 & \text{if } r_i < 0 \end{cases}$$

and the linear predictor being $\nu|r_i|$. Thus, for fixed ξ and ω, (8.4) can be maximised as a function of ν by fitting a probit regression on the binary outcome V_i, with no intercept and a single covariate $|r_i|$.

Next, let us look at the estimation of the ξ and ω parameters, holding ν constant. If we examine (8.4), we will now show how this can be considered to be a censored Gaussian likelihood, where the first component is with regards to the uncensored data, and the second component is with regards to the censored data (and multiplied by our constant ν). Note as well, that each observation contributes to both the censored and uncensored components of the likelihood.

For the purpose of performing a censored regression, we will utilise the algorithm developed in Section 7.1. In order to make use of this algorithm, the data needs to be manipulated to ensure that each observation is contributing to both components of the likelihood. Firstly, we create a censoring indicator, c, of length $2n$. The first n elements are set to 0, and then elements $n+1, ..., 2n$ are set to -1 to indicate left censoring. Next, we create a vector of 1s called I of length $2n$, and we also create a new outcome vector called D, where observations $1, ..., n$ are X, and observations $n+1, ..., 2n$ are also X. This corresponds to

$$D = \begin{pmatrix} X_1 \\ \vdots \\ X_n \\ X_1 \\ \vdots \\ X_n \end{pmatrix}, \quad I = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \text{and} \quad c = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Now, the likelihood for the observed data model is

$$l(\xi, \omega|\nu) = \prod_{i=1}^{2n} l_i,$$

where

$$l_i = \begin{cases} \frac{1}{\omega}\phi\left(\frac{D_i - I_i}{\omega}\right), & \text{if } c_i = 0 \\ 1 - \Phi\left(\nu\left(\frac{I_i - D_i}{\omega}\right)\right), & \text{if } c_i = -1, \end{cases}$$

which equates to

$$l_i = \begin{cases} \frac{1}{\omega}\phi\left(\frac{D_i - I_i}{\omega}\right), & \text{if } c_i = 0 \\ \Phi\left(\nu\left(\frac{D_i - I_i}{\omega}\right)\right), & \text{if } c_i = -1. \end{cases} \tag{8.7}$$

Now we can compare (8.4) and (8.7), and see that they follow a similar form, where $D_i = X_i$ and $I_i = \xi$.

So, we can fit our censored regression model to update $\hat{\xi}$ and $\hat{\omega}$ with our new outcome data D_i, using the censoring indicator (c_i) to indicate the censored observations. Note that the censored data observations require I_i and D_i to be multiplied by the constant ν. The location model contains only the vector I_i as a covariate, that is, with no intercept term. Meanwhile, the scale model contains an intercept only. The parameter from the I_i covariate is the estimate for $\hat{\xi}$ and the intercept estimate from the scale model is $\hat{\omega}$.

Once we have calculated our current estimates for $\hat{\xi}$, $\hat{\omega}$ and $\hat{\nu}$, we then examine convergence as per Section 3.1.1. If convergence has not been met, we iterate our cyclic coordinate ascent algorithm by re-calculating the residuals, performing another probit regression and then performing our censored Gaussian regression, until we do reach our convergence criteria. It is also of importance to note that whether we use the censored normal algorithm in Section 7.1, or another censored normal algorithm such as that provided in the `SN` package, we obtain the same result.

Numerical example

To compare the results from these two location, scale and shape (LSS) algorithms, that is, the `msn.mle` function in the `SN` package with the algorithm developed above, we consider a simple numerical example. Although this is just a single example, it is indicative of behaviour observed with other parameter values. Figure 8.2 provides the results from a random sample generated from $Y \sim SN(5, 9, 2)$, in which we compare the MLE results from the two methods. It can be seen that the two methods converge to the same parameter estimates and log-likelihood. Although the results from the two methods were the same, the `SN` package algorithm required 7 iterations while the

proposed algorithm required 40 iterations. Although the proposed algorithm required more iterations, an advantage of this algorithm is that it easily extends to allow for a regression model in the location, scale and shape parameters. In the next section, we will investigate how this can be achieved.

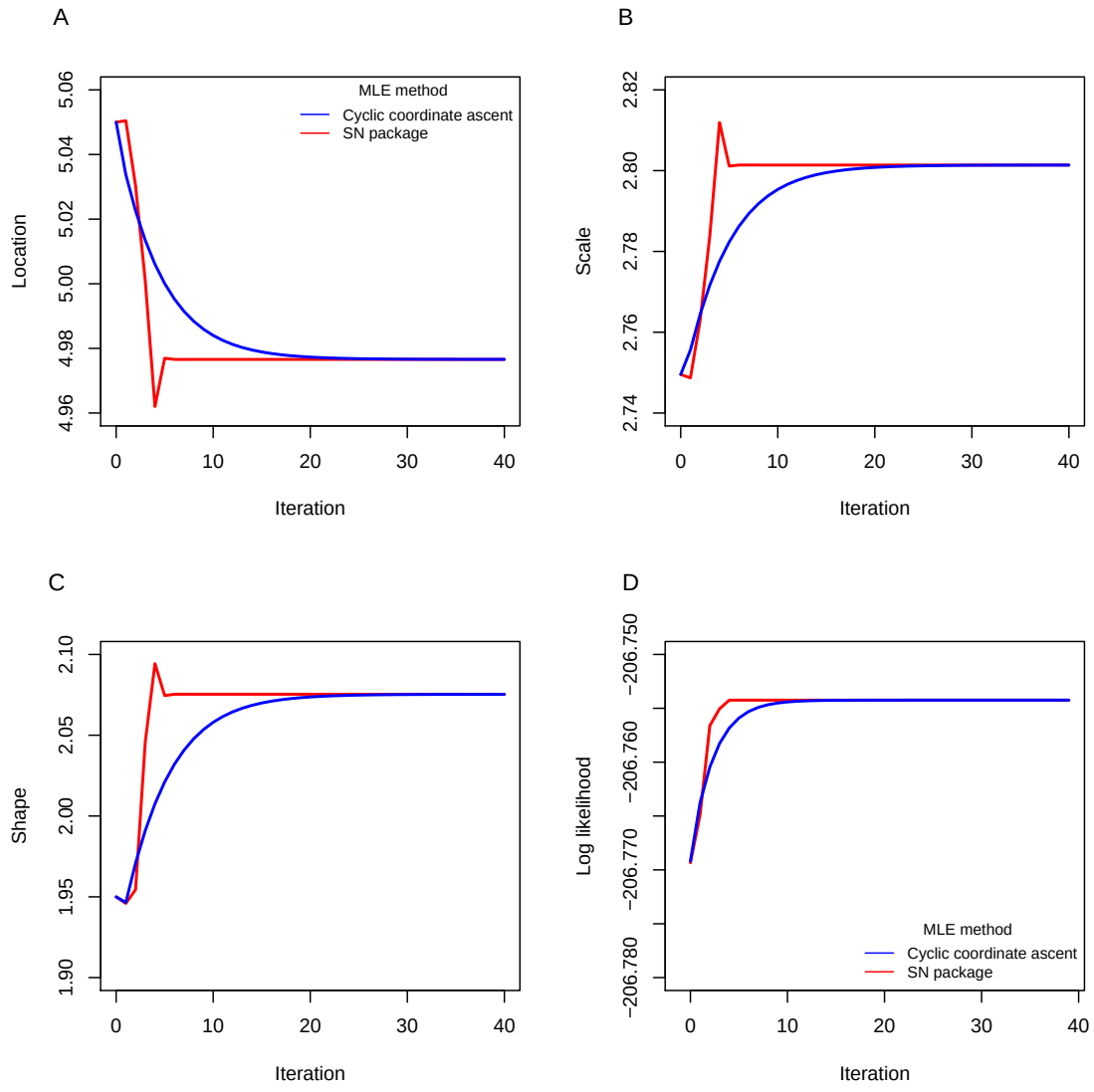

Figure 8.2: A numerical example to compare the two methods of obtaining the MLE. The location parameter (ξ) is given in A, the scale parameter (ω) in B, and the shape parameter (ν) in C over the iterations. The log-likelihood over the iterations is given in D.

8.3 Extension to LSS regression model

Now that an algorithm has been developed to estimate the ξ, ω and ν parameters in a skew normal model, let us extend this to incorporate a regression model for each of the three parameters. In this case the model becomes

$$X_i \sim SN\left(\xi_0 + \sum_{p=1}^{P}\xi_p s_{ip},\ \omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq},\ \nu_0 + \sum_{k=1}^{K}\nu_k u_{ik}\right) \quad \text{for} \quad i = 1, 2, ..., n,$$

where we have covariates s_{ip} $(p = 1, ..., P)$ for the location model, covariates l_{iq} $(q = 1, ..., Q)$ for the scale model and covariates u_{ik} $(k = 1, ..., K)$ for the shape model.

The likelihood for this model is

$$\mathcal{L}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\nu}) = \left(\frac{2}{\left(\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}\right)}\right)^n \prod_{i=1}^{n}\phi\left(\frac{X_i - \xi_0 - \sum_{p=1}^{P}\xi_p s_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}}\right) \Phi\left(\left(\nu_0 + \sum_{k=1}^{K}\nu_k u_{ik}\right)\left(\frac{(X_i - \xi_0 - \sum_{p=1}^{P}\xi_p s_{ip})}{\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}}\right)\right), \tag{8.8}$$

and the log-likelihood reduces to

$$L(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\nu}) = -n\log\left(\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}\right) + \sum_{i=1}^{n}\log\frac{\left(X_i - \xi_0 - \sum_{p=1}^{P}\xi_p s_{ip}\right)}{\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}} + \sum_{i=1}^{n}\log\Phi\left(\left(\nu_0 + \sum_{k=1}^{K}\nu_k u_{ik}\right)\frac{X_i - \xi_0 - \sum_{p=1}^{P}\xi_p s_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q}\omega_q l_{iq}}}\right),$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_P)$, $\boldsymbol{\omega} = (\omega_1, \omega_2, ..., \omega_Q)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, ..., \nu_K)$.

Similarly to the approach taken in the previous section, the likelihood in (8.8) can be

seen as a censored Gaussian likelihood when viewed as a function of $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$, and a probit regression likelihood when viewed as a function of $\boldsymbol{\nu}$. Therefore, we can take the same approach as before, with an extension of the cyclic coordinate ascent algorithm.

The steps outlined in the section above may be generalised as follows. First, the residuals can be defined as follows, for fixed $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$,

$$r_i = \frac{X_i - \xi_0 - \sum_{p=1}^{P} \xi_p s_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q l_{iq}}}.$$

Now,

$$\begin{aligned} l_2(\boldsymbol{\nu}|\boldsymbol{\xi}, \boldsymbol{\omega}) &= \prod_{i=1}^{n} \Phi\left(\left(\nu_0 + \sum_{k=1}^{K} \nu_k u_{ik}\right) r_i\right) \\ &= \prod_{r_i \geq 0} \Phi\left(\left(\nu_0 + \sum_{k=1}^{K} \nu_k u_{ik}\right) |r_i|\right) \prod_{r_i < 0} \left[1 - \Phi\left(\left(\nu_0 + \sum_{k=1}^{K} \nu_k u_{ik}\right) |r_i|\right)\right]. \end{aligned}$$

As in Section 8.2, this is exactly a probit regression likelihood with the binary outcome (V_i). Therefore for fixed $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$, the likelihood (8.8) can be maximised as a function of $\boldsymbol{\nu}$ by fitting a probit regression on the binary outcome V_i with the covariate $|r_i|$, and each covariate of interest u_{ik} multiplied by $|r_i|$, with no intercept term. The estimate of ν_0 is obtained from the estimate from the $|r_i|$ parameter in the model, and each of the estimates of ν_k are obtained from the parameter for the relevant u_{ik} covariate in the model.

Next, let us consider the estimation of $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$, while maintaining $\boldsymbol{\nu}$ constant. Firstly, similar to Section 8.2, we must create our censoring indicator c and our new outcome vector D. Also, we must create a new variable for each of the covariates s_{ip} which is of length $2n$. For each of these covariates, the values $i = 1, ..., n$ are the values from s_{ip} $i = 1, ..., n$, and the values $i = n+1, ..., 2n$ are also s_{ip}. Let us call these new covariates of length $2n$, z_{ip}.

In the same manner, the scale covariates are each manipulated to be of length $2n$. Each covariate, l_{iq}, is duplicated so that the first set of values, $i = 1, ..n$, are the respective value from l_{iq}. Then the next values, $i = n+1, .., 2n$, are also the respective value from

l_{iq}. Let us call these new covariates x_{iq}.

Chapter 7 describes the extension for the censored regression model for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$, so our observed data likelihood for this extended model is

$$l(\boldsymbol{\xi}, \boldsymbol{\omega}|\boldsymbol{\nu}) = \prod_{i=1}^{2n} l_i,$$

where

$$l_i = \begin{cases} \dfrac{1}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \phi\left(\dfrac{D_i - \xi_0 - \sum_{p=1}^{P} \xi_p z_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \right), & \text{if } c_i = 0 \\ 1 - \Phi\left(\left(\nu_0 + \sum_{k=1}^{K} \nu_k u_{ik} \right) \left(\dfrac{\xi_0 - \sum_{p=1}^{P} \xi_p z_{ip} - D_i}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \right) \right), & \text{if } c_i = -1, \end{cases}$$

which equates to

$$l_i = \begin{cases} \dfrac{1}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \phi\left(\dfrac{D_i - \xi_0 - \sum_{p=1}^{P} \xi_p z_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \right), & \text{if } c_i = 0 \\ \Phi\left(\left(\nu_0 + \sum_{k=1}^{K} \nu_k u_{ik} \right) \left(\dfrac{D_i - \xi_0 - \sum_{p=1}^{P} \xi_p z_{ip}}{\sqrt{\omega_0 + \sum_{q=1}^{Q} \omega_q x_{iq}}} \right) \right), & \text{if } c_i = -1. \end{cases} \tag{8.9}$$

Once the current estimates for $\boldsymbol{\xi}$, $\boldsymbol{\omega^2}$ and $\boldsymbol{\nu}$ are obtained, the cyclic coordinate ascent algorithm continues to cycle between a probit regression and a censored normal regression until convergence. The algorithm is summarised schematically in Figure 8.3. In this algorithm, estimation of standard errors is obtained by bootstrapping only.

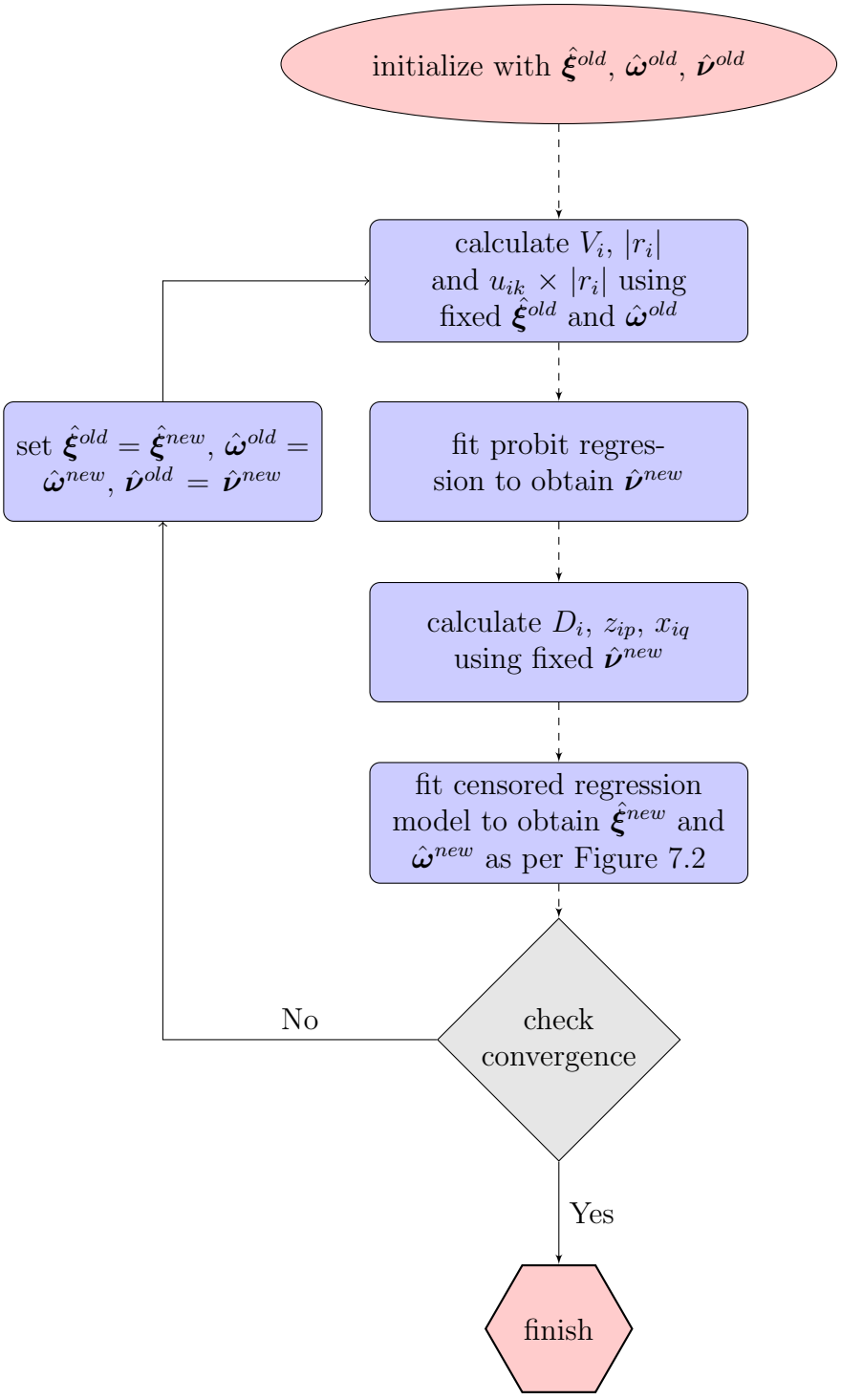

Figure 8.3: The cyclic coordinate ascent algorithm for the estimation of the location, scale and shape.

Note that with the development of the regression model in each of the parameters, a semi-parametric model with B-splines is now easily incorporated in the location, scale and shape parameters. As detailed in Section 6.2, the B-spline basis functions are fit as covariates in any or all models; the location, scale or the shape. The flexibility of each of the functions is determined by the number of knots specified, and can vary over the three models as required.

8.4 Simulations

A small simulation study was performed to investigate the performance of this algorithm for the estimation of the location, scale and shape parameters. For three sample sizes of 100, 500 and 1000 observations, data were randomly sampled from the following distribution $Y \sim SN\left(1 + 1x, 1 + 1x, 1\right)$.

For each simulation, a total of 500 repetitions were performed. The results are shown in Table 8.1. As expected, when the sample contains only 100 observations, the estimation of the parameters is more variable, with higher standard deviation and MSE. This improves with a larger sample, and at 1000 observations, the MSE is less than 10% for each of the parameters.

TABLE 8.1: Results from a simulation study. Data from a SN distribution with (ξ, ω^2 ν) as $Y \sim SN(1 + x, 1 + x, 1)$ at 100, 500 and 1000 observations.

	$n = 100$				
	ξ_0	ξ_1	ω_0^2	ω_1^2	ν_0
Mean	1.114	1.061	0.998	1.038	0.987
SD	0.421	0.417	0.479	0.860	0.994
MSE	0.190	0.177	0.230	0.741	0.987
	$n = 500$				
Mean	1.057	1.019	0.982	0.964	0.938
SD	0.223	0.183	0.241	0.375	0.433
MSE	0.053	0.034	0.059	0.142	0.191
	$n = 1000$				
Mean	1.025	1.007	0.985	0.998	0.974
SD	0.149	0.134	0.173	0.253	0.286
MSE	0.023	0.018	0.030	0.064	0.083

More importantly, for all sample sizes and all parameters, the average estimate is acceptably close to the true value. It also shows that, particularly for larger sample sizes, there is virtually no estimation bias.

The results are also shown graphically in Figure 8.4, now in terms of the mean, variance and skew. This clearly shows the increased precision with increasing sample size.

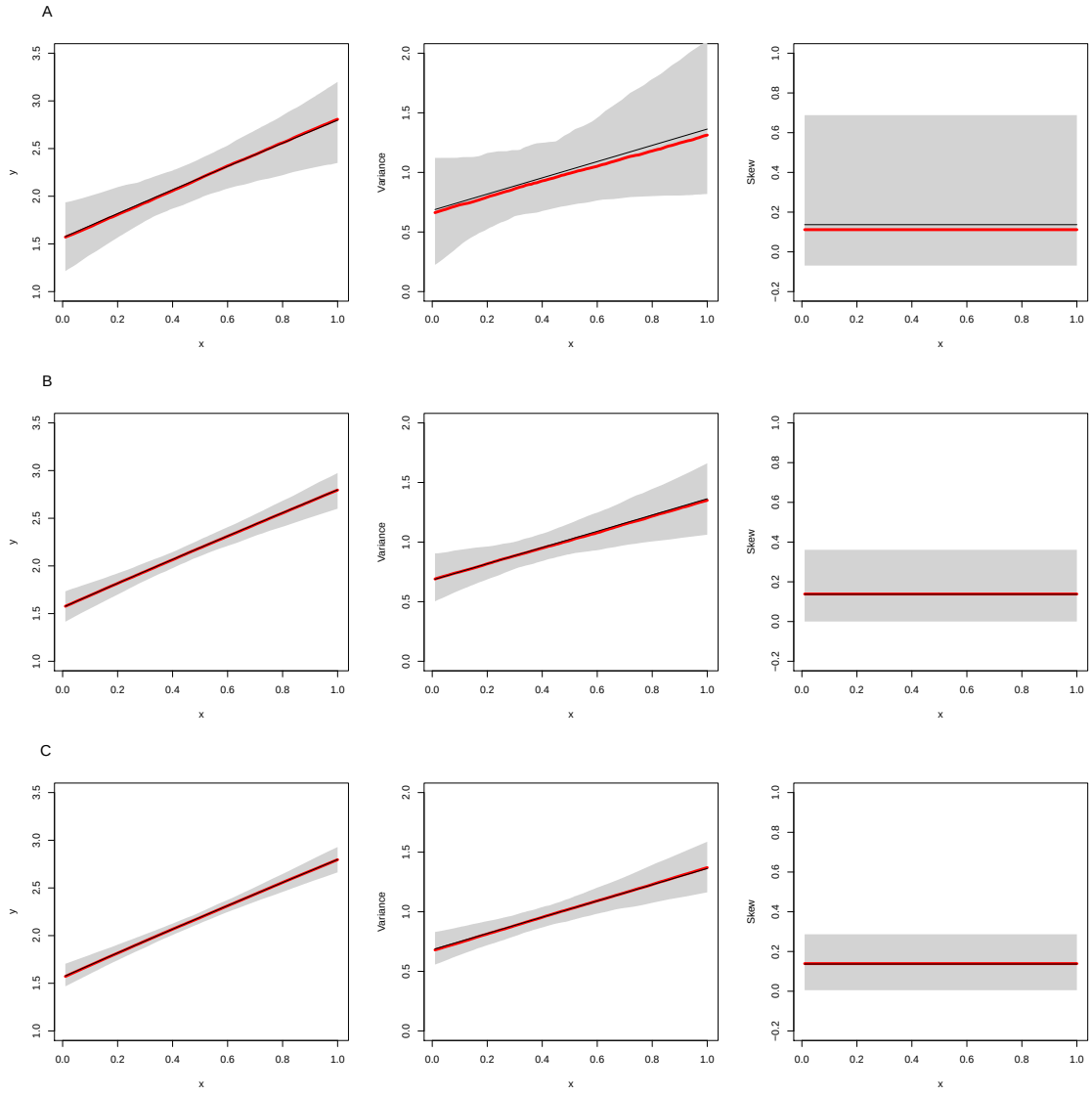

Figure 8.4: A summary of the simulation study. (A) The mean (left), variance (centre) and skew (right) over x for the 100 observation study. (B) For the 500 observation study and (C) the 1000 observation study. The area between the 2.5% and 97.5% percentiles is given in grey for each plot. The distribution from which these data were sampled is given in black.

8.5 Application of LSS models

8.5.1 Analysis example 1

The LIDAR dataset was explored in Section 6.6.2, where some small skew was apparent in the residuals (Figure 6.6). To explore this further, the EM-type algorithm for the location, scale and shape (LSS) developed above can be applied. The mean and variance model that was the best fit from the previous chapter had 4 internal knots in the mean and 2 internal knots in the variance. Assuming these models in the location and scale, a summary of the fit of two LSS models, compared to the model with no shape parameter from Section 6.6.2, is shown in Table 8.2. The AIC, BIC and HQC all agree that the model with a constant shape parameter (intercept only) is a better fit to these data than the model without a shape parameter. They also all agree that a regression model for the shape parameter is unnecessary. Although this model provides the best fit, from Figure 8.5 the model for the mean is clearly very similar for the three models. The variance regression is seen to be similar at lower values of the range variable, but deviates at high values. The residual plots appear to indicate an adequate fit (Figure 8.6).

TABLE 8.2: Results from various models fit to the LIDAR dataset.

Shape model	AIC	BIC	HQC
No shape model	-651.2	-603.6	-655.6
Constant shape model	-663.7	-612.8	-668.5
Linear shape model	-657.0	-602.6	-662.0

8.5.2 Analysis example 2

The next dataset to be analysed is the CD4 dataset from Section 3.3. Starting with the mean and variance, we can use the information criteria to find the optimal number of internal knots to fit in the models (see Table 8.3 for the AIC). The AIC, AICc and HQC agreed that the optimal model was 5 knots in the mean and 5 knots in the variance. As expected, the BIC favoured a model with less parameters, namely, 4 knots in the mean and 2 knots in the variance. Once the optimal model was determined, a shape

parameter was incorporated, both as a constant and a linear model over age. The comparison of these three models is shown in Table 8.4.

These results show that the linear shape model is the favoured model by all of the information criteria. A graphical summary of these models is shown in Figure 8.7, where there is only a small change in the mean models between the three models, but a change in the variance seen at one year of age. Note that while the shape model fit was a linear model, the transformation into the skew does not translate to a linear model in the skew. It can be seen from Figure 8.8 that the normal model is clearly inadequate with residuals that deviate substantially from normality. While the linear shape model is preferred by all criteria, the residuals for the constant model and the linear model appear very similar. This may indicate that the additional complexity of a linear shape model is not needed, and the inclusion of a constant term may be adequate.

Lastly, a comparison is given in Figure 8.9 of the different distributions of CD4 count over four ages, for the constant shape model. Clearly at the younger ages, the count has a moderately higher mean, and much larger variation.

Table 8.3: The AIC from different mean and variance models for the CD4 data. The lowest AIC is in boldface.

Variance model	Mean model									
	Zero	Constant	Linear	0 knots	1 knot	2 knots	3 knots	4 knots	5 knots	6 knots
Constant	9750.8	9205.7	9044.1	8995.6	8997.3	8977.0	8972.1	8967.6	8966.3	8964.8
Linear	9662.4	9156.5	8999.1	8919.5	8916.2	8893.4	8889.8	8886.3	8885.6	8884.5
0 knots	9552.6	9031.4	8882.9	8828.7	8827.5	8815.9	8815.1	8814.3	8814.7	8814.6
1 knot	9565.1	9062.4	8897.4	8838.8	8838.4	8823.6	8821.8	8820.3	8820.2	8822.0
2 knots	9528.2	8997.7	8890.3	8835.0	8836.0	8807.0	8801.0	8795.6	8794.3	8794.0
3 knots	9518.6	8987.6	8884.8	8834.5	8835.8	8805.6	8799.2	8794.0	8792.6	8792.6
4 knots	9521.1	8984.9	8883.5	8835.6	8837.1	8805.2	8799.0	8793.7	8792.7	8793.0
5 knots	9517.9	8983.8	8885.6	8838.5	8840.3	8804.8	8799.2	8790.4	**8789.1**	8795.7
6 knots	9523.9	8985.5	8888.2	8842.1	8844.0	8806.8	8801.1	8795.4	8794.4	8793.9

Table 8.4: Results from various models fit to the CD4 dataset.

Shape model	AIC	AICc	BIC	HQC
No shape model	8789.1	8790.0	8859.7	8816.6
Constant shape model	8681.4	8682.5	8756.4	8710.6
Linear shape model	8676.2	8677.4	8755.6	8707.1

8.6 Final comments

This chapter developed an algorithm with a regression model in the location, scale and shape parameters, utilising the censored data CEM algorithm developed in Chapter 7. Through a simulation study, we demonstrated that this algorithm can reliably estimate the three parameters with virtually no bias in large samples. We also illustrated its applicability for assessing variance heterogeneity in datasets, using the CD4 data as an example. The next chapter in this thesis will detail the R package that has been developed that incorporates this algorithm, along with other algorithms developed in this thesis.

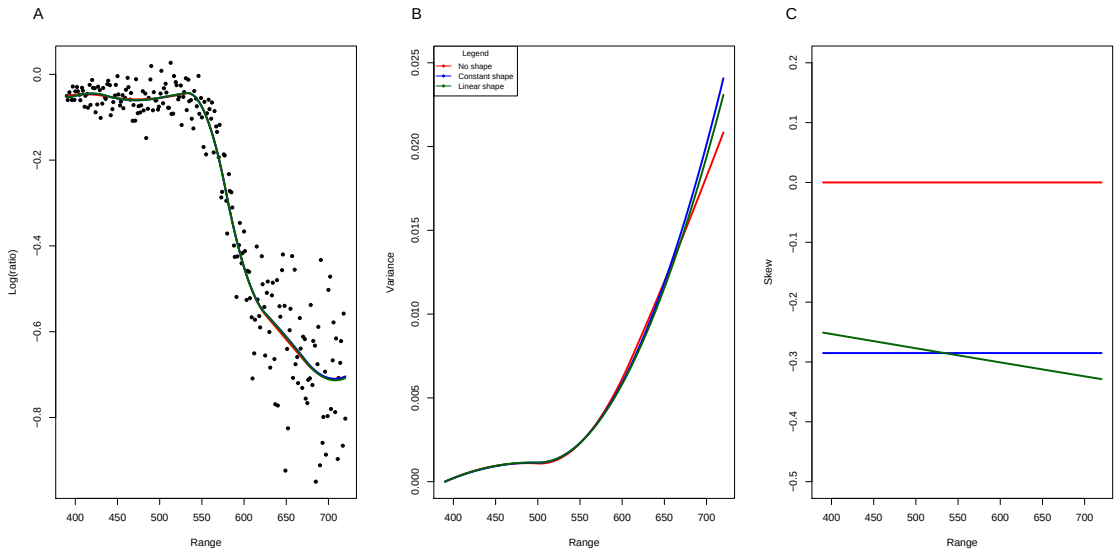

Figure 8.5: A summary of the various shape models fit to the LIDAR data. (A) A comparison of the mean models, with the data points shown in black. (B) A comparison of the variance models and (C) the skew models.

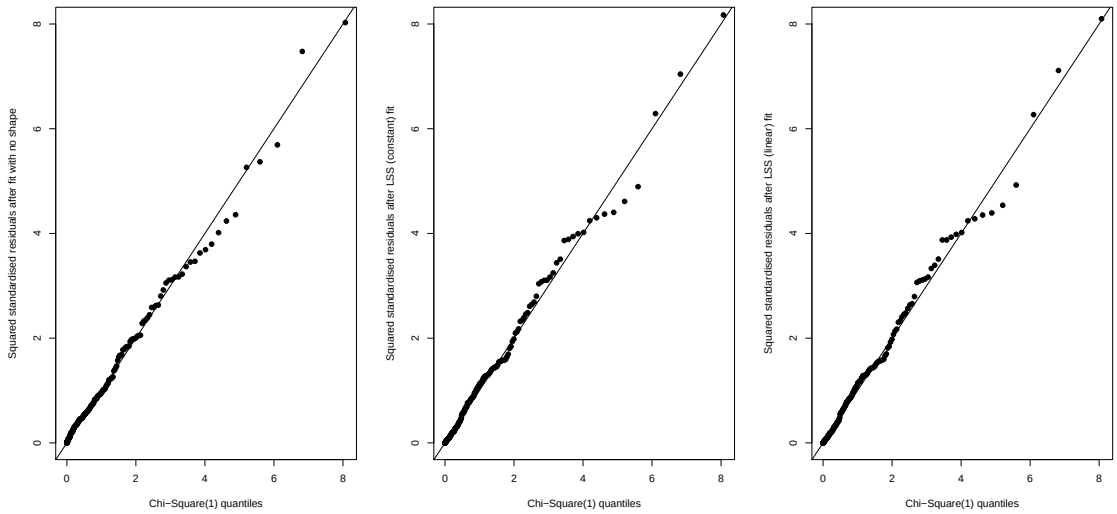

Figure 8.6: A summary of the residuals from the various shape models fit to the LIDAR data. (A) Model with no shape (normal model). (B) LSS model with constant shape parameter and (C) LSS model with linear shape parameters.

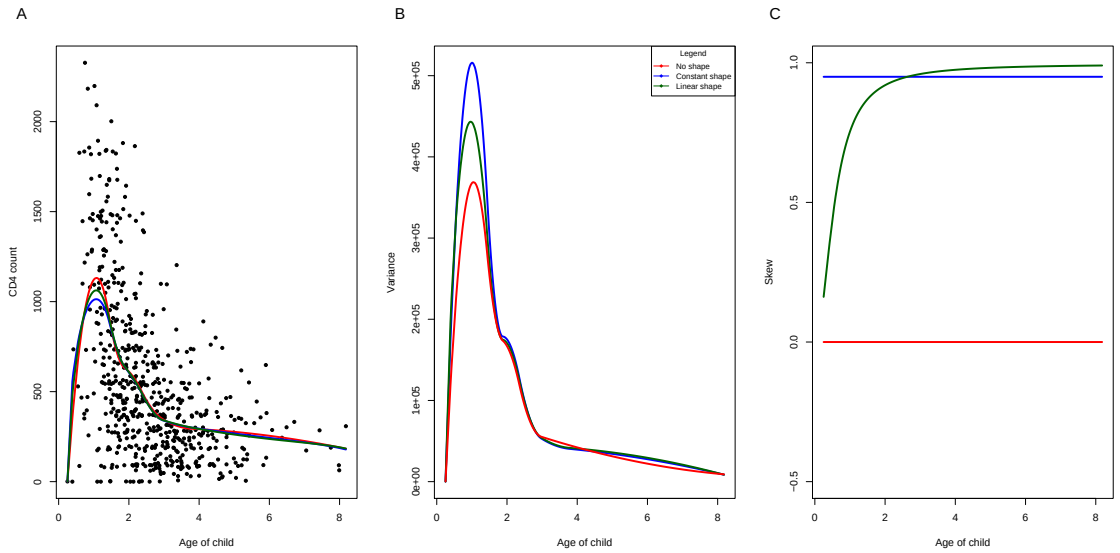

FIGURE 8.7: A summary of the various shape models fit to the CD4 data. (A) A comparison of the mean models, with the data points shown in black. (B) A comparison of the variance models and (C) the skew models.

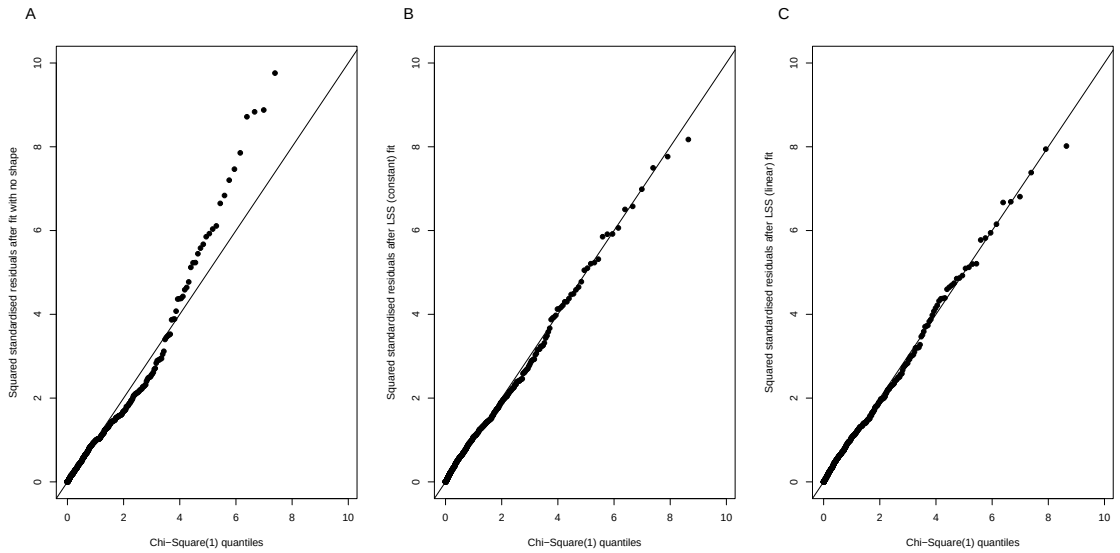

FIGURE 8.8: A summary of the residuals from the various shape models fit to the CD4 data. (A) Model with no shape parameter (normal model). (B) LSS model with constant shape parameter and (C) LSS model with a linear shape parameter.

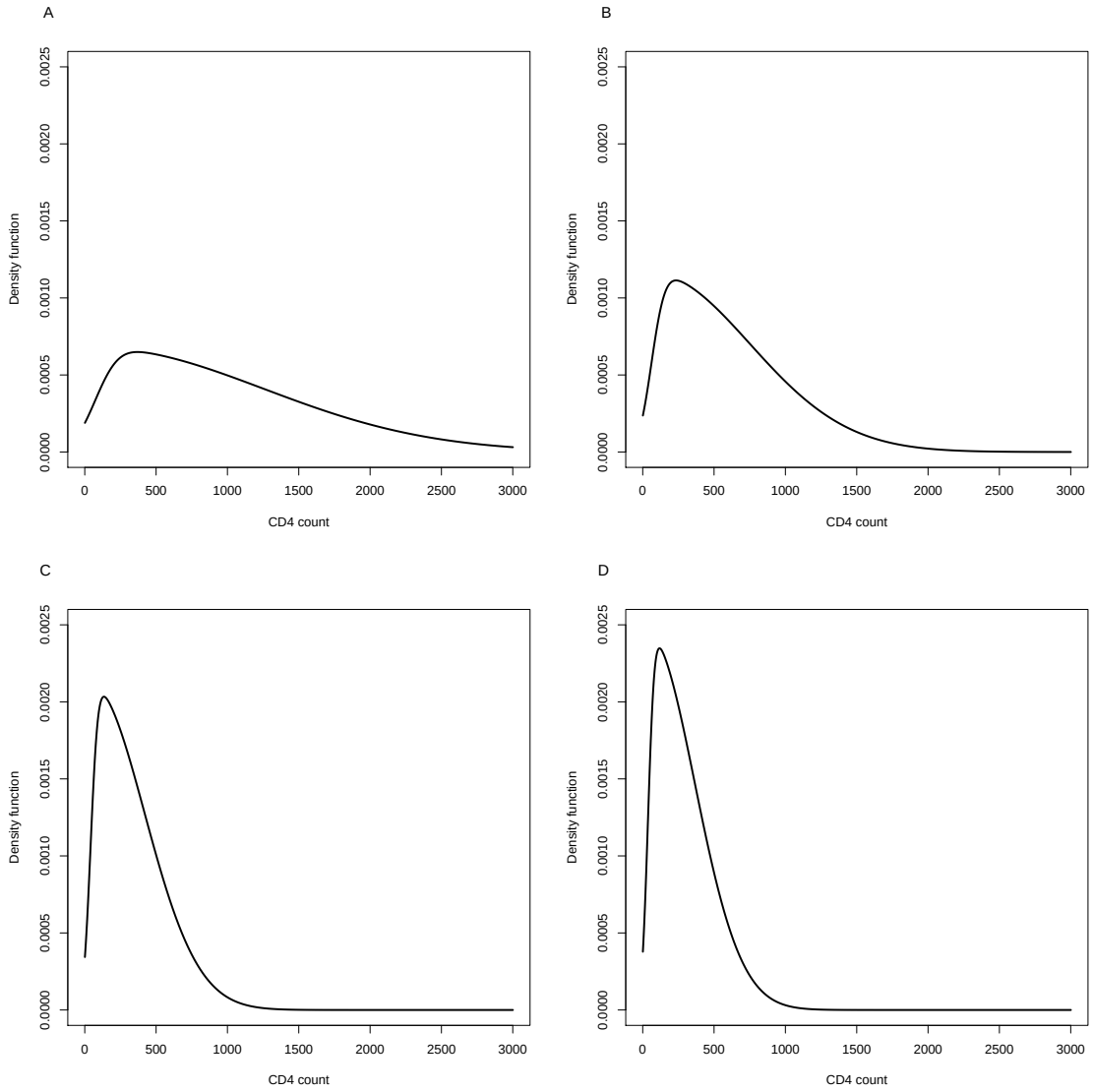

Figure 8.9: A summary of the CD4 count distributions over different ages for the constant skew model. (A) is at 1 year of age, (B) at 2 years, (C) at 3 years and (D) at 4 years of age.

9

Software and biomarker analysis

The algorithms presented in the previous chapters have been developed into an R package entitled `VarReg`. This chapter will demonstrate the main features of this package, using an example dataset from the Long-term Intervention with Pravastatin in Ischemic Disease (LIPID) study. The LIPID study was a multi-centre randomised double-blind placebo-controlled trial that recruited 9014 patients with a history of myocardial infarction or unstable angina. The study found strong evidence that pravastatin (a cholesterol lowering medication) reduced the risk of death from CHD, cardiovascular disease, and all causes combined (The Long-Term Intervention with Pravastatin in Ischaemic Disease Study Group, 1998). Various biomarkers were measured at randomisation and at one year, and a subset of these biomarkers in the placebo treated group will be used to demonstrate the `VarReg` package in this chapter. See Section 3.3 for more information about this dataset.

The biomarkers to be investigated are lipoprotein-associated phospholipase A2 activity (LP-PLA2 activity) and Lipoprotein (a) (Lp(a)), and these are to be associated with

LDL cholesterol. LDL cholesterol is low-density lipoprotein cholesterol and usually called the 'bad' cholesterol. In this study, treatment with pravastatin caused a reduction in LDL levels, and this was the main mechanism of the effect of treatment. As such, studying the relationship between the biomarkers and LDL may give insight into the extent to which the biomarkers are a mechanism of treatment action.

9.1 Overview of `VarReg` package

This package contains various functions to perform the algorithms developed in this thesis. Perhaps the main function in this package is `semiVarReg()`, which can be used to perform various linear or semi-parametric mean and variance regression models, for either uncensored or censored outcome data. The `searchVarReg()` function can also be used for uncensored or censored data, to search for the optimal model in the mean and variance, as determined by the information criterion of choice. Lastly, the `plotVarReg()` function plots the mean and variance models for censored or uncensored outcome data, as well as the residuals.

The location, scale and shape regression models that were developed in Chapter 8 are also a function in this package, called `lssVarReg()`. A plot function to partner this is the `plotlssVarReg()` function, which can be used to plot these three models and the residuals. Each of these functions will be explored in further detail with the use of the LIPID dataset.

9.2 The `semiVarReg()` function

The `semiVarReg()` function encapsulates a variety of models that can be fit in the mean and variance. At this time, only one covariate of interest can be included in these models. The input arguments and their default values are as follows:

- `y`: Vector containing the outcome data. There must be no missing data and censored values must be set to the limits of detection.
- `x`: Vector containing the covariate data. There must be no missing data and this vector should be the same length as `y`.

- `cens.ind`: Vector containing the censoring indicator for the outcome data (if applicable). The default is `NULL` which indicates there is no censored data. If applicable, this vector should be the same length as `y` and there must be no missing data. A value of `0` indicates uncensored data, `1` indicates right (or upper) censoring and `-1` indicates left (or lower) censoring.

- `meanmodel`: Text to specify the mean model to be fit to these data. The possible inputs are `zero`, `constant`, `linear` or `semi`. The option `semi` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.m`.

- `mean.intercept`: Logical argument (default=`TRUE`) to indicate if the mean model is to include an intercept term. This option is only available for censored outcome data in the mean model.

- `varmodel`: Text to specify the variance model to be fit to these data. The possible inputs are `constant`, `linear` or `semi`. The option `semi` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.v`.

- `knots.m`: Integer indicating the number of internal knots to be fit in the semi-parametric mean model. Knots are placed equidistantly over the covariate. The default value is 2.

- `knots.v`: Integer indicating the number of internal knots to be fit in the semi-parametric variance model. Knots are placed equidistantly over the covariate. The default value is 2.

- `degree`: Integer indicating the degree of the splines fit in the mean and the variance models. The default value is 2.

- `mono.var`: Text to indicate whether the variance model is monotonic. Note that this is not available for the `constant` variance model. Options are `none`, `inc` or `dec`, with the default being `none`. The option `inc` indicates increasing monotonic and `dec` indicates decreasing monotonic. For semi-parametric variance models (`varmodel=semi`), the appropriate monotonic B-splines are fit in the

semi-parametric variance model. If the variance model is `linear`, the parameter space is constrained (positive for increasing and negative for decreasing).

- `control`: List of control parameters for the algorithm. This includes `maxit` (the maximum number of iterations to be performed), `epsilon` (the positive convergence tolerance) and `bound.tol` (the positive tolerance for specifying the interior of the parameter space. This allows the algorithm to terminate early if an interior maximum is found).

An object of class `VarReg` is returned by the function and includes the following components:

- `modeltype`: Text indicating the model that was fit, noting if a censored approach was performed.
- `knots.m`, `knots.v`, `degree`, `meanmodel`, `varmodel`: Returning the input variables as described above
- `converged`: Logical argument indicating if convergence occurred.
- `iterations`: Total iterations performed.
- `reldiff`: Numeric value of the positive convergence tolerance that occurred at the final iteration.
- `loglik`: Numeric value of the maximised log-likelihood.
- `boundary`: Logical argument indicating if the MLE is on the boundary of the parameter space.
- `aic.c`: Numeric value of the Akaike information criterion corrected for small samples
- `aic`: Numeric value of the Akaike information criterion
- `bic`: Numeric value of the Bayesian information criterion
- `hqc`: Numeric value of the Hannan-Quinn information criterion

- `mean.ind`: Vector of integer(s) indicating the column number(s) in the dataframe `data` that were fit in the mean model.
- `mean`: Vector of the maximum likelihood estimates of the mean parameter(s).
- `var.ind`: Vector of integer(s) indicating the column(s) in the dataframe `data` that were fit in the variance model.
- `variance`: Vector of the maximum likelihood estimates of the variance parameter(s).
- `cens.ind`: Integer indicating the column in the dataframe `data` that corresponds to the censoring indicator.
- `data`: Dataframe containing the variables included in the model.

As an example, let us fit a model to LDL levels (`ldl0q`) at baseline and the LP-PLA2 activity (`PLA_ACTIVITY_0`) biomarker. To fit a model with a semi-parametric mean model (with three internal knots) and a linear variance model in the `VarReg` package, we use the command:

```
 pla_model<-semiVarReg(y=lipid$ldl0q, x=lipid$PLA_ACTIVITY_0,
meanmodel="semi", varmodel="linear", knots.m = 3 )
```

As we do not have any censored data, we do not need to input the `cens.ind` vector. This function gives the following output below. Note that a summary of the dataframe is given below, rather than the entire dataframe. The model converged after a total of 2006 iterations to give a maximised log-likelihood of -4288.1. The AIC for the model is 8592.1, and the MLE for the B-spline basis functions fit to the mean are given, as is the LP-PLA2 activity coefficient for the variance.

```
> pla_model[-21]
$modeltype
[1] "Mean and Variance regression"
$knots.m
[1] 3
$knots.v
NULL
$degree
[1] 2
```

```
$meanmodel
[1] "semi"
$varmodel
[1] "linear"
$converged
[1] TRUE
$iterations
[1] 2006
$reldiff
[1] 9.996968e-07
$loglik
[1] -4288.073
$boundary
[1] FALSE
$aic.c
[1] 8592.183
$aic
[1] 8592.146
$bic
[1] 8642.402
$hqc
[1] 8609.972
$mean.ind
[1] 3 4 5 6 7
$mean
   Intercept M_Knt3_Base1 M_Knt3_Base2 M_Knt3_Base3 M_Knt3_Base4
   3.6620290   -0.5021208    0.2401041    0.3163190    0.7519135
M_Knt3_Base5
   0.7257329
$var.ind
[1] 2
$variance
           Intercept lipid.PLA_ACTIVITY_0
        0.5908853898        -0.0002965638
$cens.ind
NULL
 > summary(pla_model[[21]])
  lipid.ldl0q   lipid.PLA_ACTIVITY_0  M_Knt3_Base1     M_Knt3_Base2
 Min.   :1.46   Min.   : 76.11       Min.   :0.0000   Min.   :0.0000
 1st Qu.:3.40   1st Qu.:229.24       1st Qu.:0.0000   1st Qu.:0.0000
 Median :3.88   Median :261.78       Median :0.0000   Median :0.3530
 Mean   :3.90   Mean   :262.31       Mean   :0.1083   Mean   :0.3752
 3rd Qu.:4.41   3rd Qu.:294.14       3rd Qu.:0.1752   3rd Qu.:0.7087
 Max.   :6.57   Max.   :500.74       Max.   :0.5480   Max.   :0.8701
  M_Knt3_Base3     M_Knt3_Base4      M_Knt3_Base5
 Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
 Median :0.3905   Median :0.00000   Median :0.00000
```

```
 Mean   :0.3961   Mean   :0.09153   Mean   :0.01173
 3rd Qu.:0.7440   3rd Qu.:0.13364   3rd Qu.:0.00000
 Max.   :0.8935   Max.   :0.53632   Max.   :1.00000
```

This function can also be used for censored outcome data. Using another example from the LIPID dataset, we can fit the same model for Lp(a) levels at baseline, which contains censored outcome data:

```
lp_model<-semiVarReg(y=lipid$ldl0q, x=lipid$LP_a_0,cens.ind = censor,
 meanmodel="semi", varmodel="linear", knots.m = 3, maxit=10000)
```

This gives the following output from our censored model:

```
 > lp_model[-21]
$modeltype
[1] "Censored Mean and Variance regression with mean.intercept="
[2] "TRUE"
$knots.m
[1] 3
$knots.v
NULL
$degree
[1] 2
$meanmodel
[1] "semi"
$varmodel
[1] "linear"
$converged
[1] TRUE
$iterations
[1] 1159
$reldiff
[1] 9.987434e-07
$loglik
[1] -4243.79
$boundary
[1] FALSE
$aic.c
[1] 8503.617
$aic
[1] 8503.58
$bic
[1] 8553.83
$hqc
[1] 8521.404
$mean.ind
[1] 3 4 5 6 7
```

```
$mean
   Intercept M_Knt3_Base1 M_Knt3_Base2 M_Knt3_Base3 M_Knt3_Base4
   3.2696067    0.6739793    0.5605446    0.8641551    0.2239174
M_Knt3_Base5
   1.7980594
$var.ind
[1] 2
$variance
   Intercept lipid.LP_a_0
0.5467657994 0.0007181958
$cens.ind
[1] 8
> summary(lp_model[[21]])
  lipid.ldl0q     lipid.LP_a_0    M_Knt3_Base1     M_Knt3_Base2
 Min.   :1.460   Min.   : 1.30   Min.   :0.0000   Min.   :0.0000
 1st Qu.:3.400   1st Qu.: 6.50   1st Qu.:0.0000   1st Qu.:0.0000
 Median :3.880   Median :13.40   Median :0.0000   Median :0.1741
 Mean   :3.901   Mean   :26.67   Mean   :0.1678   Mean   :0.3163
 3rd Qu.:4.410   3rd Qu.:43.20   3rd Qu.:0.3317   3rd Qu.:0.6534
 Max.   :6.570   Max.   :90.00   Max.   :0.6993   Max.   :0.8586
  M_Knt3_Base3      M_Knt3_Base4     M_Knt3_Base5       cens.ind
 Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :-1.00000
 1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.00000
 Median :0.06318   Median :0.0000   Median :0.0000   Median : 0.00000
 Mean   :0.18880   Mean   :0.1195   Mean   :0.1049   Mean   : 0.02608
 3rd Qu.:0.33588   3rd Qu.:0.1516   3rd Qu.:0.0000   3rd Qu.: 0.00000
 Max.   :0.73698   Max.   :0.6207   Max.   :1.0000   Max.   : 1.00000
```

9.3 The `plotVarReg()` function

This function produces plots of the mean and variance models, as well as the residuals to assess the assumptions of the model. The `plotVarReg()` function utilises the `seVarReg()` function internally in order to calculate the Fisher information matrix standard errors, as well as the `semiVarReg()` function if the bootstrapped 95% confidence intervals are requested. The input arguments and their default values are as follows:

- `x`: Object of class `VarReg` (output from `semiVarReg()`).
- `knot.lines`: Logical parameter indicating if lines showing where the internal knots are located should be placed on the graphics. Only relevant for semi-parametric models.

- `ci`: Logical argument indicating whether the 95% confidence intervals should be shown on the graphics. Default is `FALSE`.
- `ci.type`: Text to indicate the type of confidence interval that should be shown if `ci=TRUE`. Choices are `im` or `boot` for the Fisher information matrix or the bootstrapped confidence intervals, respectively. Default is `im`, however `im` is not an option for semi-parametric models.
- `bootreps`: Integer giving the number of bootstrap replications that should be performed for the bootstrapped confidence intervals (if `ci.type=boot`). Default is `1000`.
- `xlab`: Label for plots for the x variable
- `ylab`: Label for plots for the y variable
- `control`: List of control parameters, as documented above in Section 9.2.

The output from this function is a 2x2 panel of plots. If the outcome data is not censored, the four plots are:

- the mean function over the x variable, with or without 95% CI and with or without the knot lines indicated
- the variance function over the x variable, with or without 95% CI and with or without the knot lines indicated
- a Q-Q plot of the residuals from the model
- a histogram of the residuals from the model

Using the model produced above for LP-PLA2 activity, an example to produce these plots would be:

```
plotVarReg(pla_model, knot.lines = FALSE, ci=TRUE, ci.type = "im",
 ylab="Baseline LDL", xlab = "Baseline PLA2 Activity")
[1] "CI=true, type=information matrix"
[1] "Meanmodel='semi' so 95% CI cannot be given by information matrix"
```

The function output the plots given in Figure 9.1. The function also gave a note that as a semi-parametric model with B-spline basis functions was fit, the 95% CI cannot

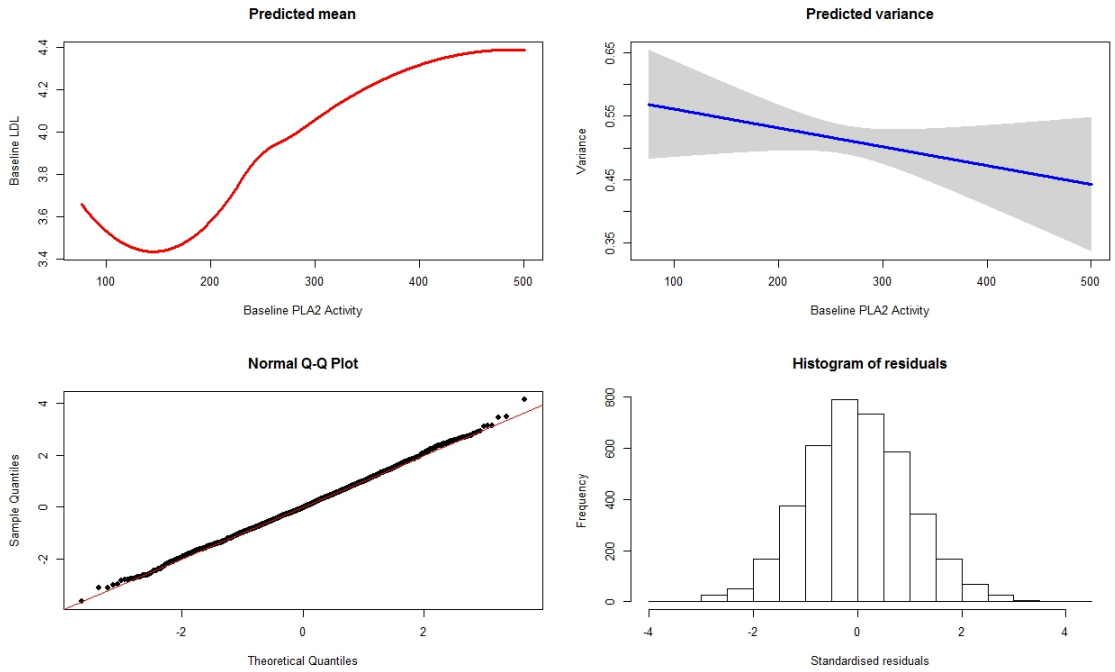

FIGURE 9.1: Plot produced from `plotVarReg()` function for PLA2 activity. Top left: predicted mean function. Top right: predicted variance with 95% CI in grey. Bottom left: Normal QQ plot of residuals (black) with the line of unity (red). Bottom right: histogram of residuals.

be given from the Fisher information matrix. Instead, the bootstrapping option would need to be used to produce the 95% CI for the mean function.

If an analysis of censored data were input, the plots created are slightly different. Using the results from the Lp(a) model from Section 9.2, we use the following command to plot the model:

```
> plotVarReg(lp_model, knot.lines = FALSE, ci=TRUE, ci.type = "im",
 ylab="Baseline LDL", xlab = "Baseline LP(a)")
[1] "CI=true, type=information matrix"
[1] "Meanmodel='semi' so 95% CI cannot be given by information matrix"
```

The mean and variance plots are given as previously, however, the other two residual plots are no longer appropriate for censored data. Given the censored residuals from these models, we can compare the squared standardised residuals (given in black), with their censoring indicator, to the chi-squared distribution with one degree of freedom (given in red). This is one of the plots given for censored data, and the other is a plot of the data, coloured by the censoring status. The plotted results are shown in Figure 9.2. The censored residuals can be seen to follow the chi-squared distribution well, and

the plot on the bottom left shows the data. The censored values are in red, and left censored data is given as a triangle, with right censored data given as an upside-down triangle.

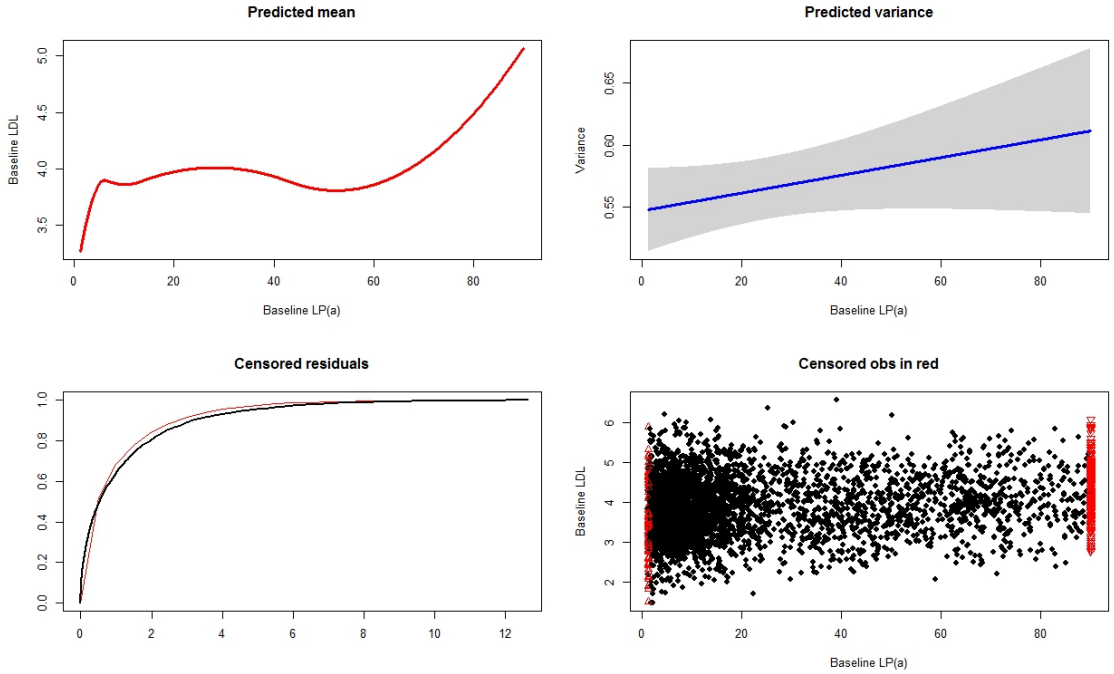

FIGURE 9.2: Plot produced from `plotVarReg()` function for censored data LP(a). Top left: predicted mean function. Top right: predicted variance with 95% CI in grey. Bottom left: The censored residuals are in black, with the chi-squared distribution (df=1) given in red. Bottom right: red triangles indicate left censored data and upside-down red triangles indicate right censored data.

9.4 The `searchVarReg()` function

This function is to aid the user to select a model with an appropriate number of internal knots in both the mean and the variance models. A series of models are fitted with increasing complexity, from zero mean up to a given total number of internal knots and constant variance up to a given total number of knots. The information criteria are calculated for each of these models, and the chosen criterion is then used to select the optimal model. Given the simulation results in Chapter 6, the AIC is the default information criterion for the model choice. The input arguments and their default values are as follows:

- `y`: Vector containing the outcome data. There must be no missing data and censored values must be set to the limits of detection.

- `x`: Vector containing the covariate data. There must be no missing data and this vector should be the same length as `y`.

- `cens.ind`: Vector containing the censoring indicator for the outcome data (if applicable). The default is `NULL` which indicates there is no censored data. If applicable, this vector should be the same length as `y` and there must be no missing data. A value of `0` indicates uncensored data, `1` indicates right (or upper) censoring and `-1` indicates left (or lower) censoring.

- `maxknots.m`: Integer indicating the maximum number of internal knots to be fit in the semi-parametric mean models. Knots are placed equidistantly over the covariate. The default value is 3.

- `maxknots.v`: Integer indicating the maximum number of internal knots to be fit in the semi-parametric variance model. Knots are placed equidistantly over the covariate. The default value is 3.

- `degree`: Integer indicating the degree of the splines fit in the mean and the variance models. The default value is 2.

- `mono.var`: Text to indicate whether the variance model is monotonic, (note this is not applied for the `constant` variance model). Options are `none`, `inc` or `dec` with the default as none. Note that `inc` indicates increasing monotonic and `dec` indicates decreasing monotonic. If the variance model is linear, the parameter space is constrained (positive for increasing and negative for decreasing). For semi-parametric variance models, the appropriate monotonic B splines are fit in the semi-parametric variance model.

- `selection` Text to indicate the information criterion that is to be used for the selection of the optimal model. Options are `AIC`, `AICc`, `HQC` and `BIC`. The `AIC` is the default.

- `print.it` Logical parameter to indicate if the results of the models should be printed as they occur. Default value is `FALSE`.

- control: List of control parameters for the algorithm. This includes maxit (the maximum number of iterations to be performed), epsilon (the positive convergence tolerance) and bound.tol (the positive tolerance for specifying the interior of the parameter space. This allows the algorithm to terminate early if an interior maximum is found).

The output from this function is a list containing the following items:

- ll: a dataframe of the log-likelihoods from each of the models that have been fit.
- AIC: a dataframe of the AIC from each of the models that have been fit. The parameters fit in the mean model are given in the columns, and the parameters in the variance are given in the rows.
- AICc: a dataframe of the AIC-c from each of the models that have been fit.
- BIC: a dataframe of the BIC from each of the models that have been fit.
- HQC: a dataframe of the HQC from each of the models that have been fit.
- best.model: an object of class VarReg containing the output from the optimal model (that model within the specified models in the mean and variance with the lowest information criterion according to the criterion selected).

We will now illustrate the use of this function with an example from the LIPID dataset. The LDL levels (ldl0q) at baseline and the LP-PLA2 activity (PLA_ACTIVITY_0) were looked at above for the semiVarReg function. The following code will look for the optimal model according to the AIC, allowing up to 5 knots in the mean and the variance:

```
best_pla <- searchVarReg(y=lipid$ldl0q, x=lipid$PLA_ACTIVITY_0,
maxknots.m = 5, maxknots.v = 5, maxit=10000,selection = "AIC")
```

The search begins with a zero mean and constant variance model, and fits up to 5 knots each in the mean and variance. This means that a total of 72 models are to be fit by this function, and for this example, took a total of 33 hours to complete. If a Windows computer is being used, a notification window appears to let the user know how the algorithm is progressing (Figure 9.3). The output is shown below, with the

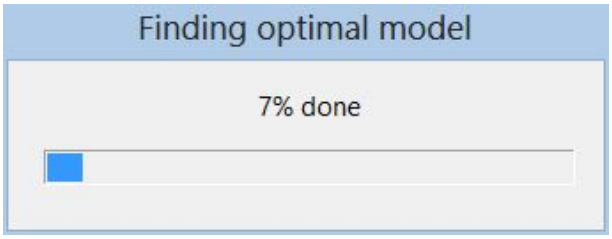

FIGURE 9.3: Example of the progress window for the `searchVarReg()` function.

last component `best.model` not shown. This is an object of class `VarReg`, the same as shown in Section 9.2 above.

```
> best_pla[-6]
$ll
             Mean_zero Mean_constant Mean_linear Mean_Knot0 Mean_Knot1
Var_constant -11057.51     -4447.035   -4297.194  -4294.288  -4292.935
Var_linear   -11047.72     -4445.369   -4296.535  -4293.372  -4292.143
Var_Knot0    -11047.69     -4433.626   -4295.276  -4292.313  -4291.175
Var_Knot1    -11047.53     -4433.313   -4295.239  -4292.120  -4291.040
Var_Knot2    -11047.33     -4433.139   -4295.139  -4292.147  -4291.014
Var_Knot3    -11047.29     -4432.838   -4295.074  -4292.134  -4291.008
Var_Knot4    -11047.27     -4432.660   -4294.952  -4292.023  -4290.883
Var_Knot5    -11047.19     -4432.309   -4294.514  -4291.615  -4290.465
             Mean_Knot2 Mean_Knot3 Mean_Knot4 Mean_Knot5
Var_constant  -4289.777  -4288.887  -4288.797  -4287.707
Var_linear    -4289.017  -4288.073  -4288.026  -4286.961
Var_Knot0     -4288.062  -4286.971  -4286.966  -4285.987
Var_Knot1     -4287.844  -4286.702  -4286.727  -4285.764
Var_Knot2     -4287.848  -4286.700  -4286.720  -4285.770
Var_Knot3     -4287.865  -4286.763  -4286.767  -4285.783
Var_Knot4     -4287.787  -4286.721  -4286.700  -4285.709
Var_Knot5     -4287.468  -4286.452  -4286.409  -4285.435

$AIC
             Mean_zero Mean_constant Mean_linear Mean_Knot0 Mean_Knot1
Var_constant  22117.02      8898.070    8600.387   8596.576   8595.871
Var_linear    22099.43      8896.738    8601.071   8596.744   8596.286
Var_Knot0     22101.38      8875.253    8600.551   8596.627   8596.350
Var_Knot1     22103.06      8876.627    8602.479   8598.240   8598.080
Var_Knot2     22104.67      8878.277    8604.278   8600.295   8600.028
Var_Knot3     22106.59      8879.676    8606.147   8602.268   8602.015
Var_Knot4     22108.54      8881.319    8607.903   8604.046   8603.767
Var_Knot5     22110.39      8882.618    8609.028   8605.231   8604.930
             Mean_Knot2 Mean_Knot3 Mean_Knot4 Mean_Knot5
Var_constant   8591.553   8591.774   8593.594   8593.414
Var_linear     8592.034   8592.146   8594.052   8593.922
Var_Knot0      8592.125   8591.943   8593.933   8593.974
Var_Knot1      8593.688   8593.405   8595.455   8595.528
```

```
Var_Knot2      8595.695   8595.401   8597.440   8597.539
Var_Knot3      8597.729   8597.526   8599.535   8599.566
Var_Knot4      8599.575   8599.441   8601.401   8601.417
Var_Knot5      8600.935   8600.903   8602.819   8602.870

$AICc
             Mean_zero Mean_constant Mean_linear Mean_Knot0 Mean_Knot1
Var_constant  22117.02      8898.073    8600.394   8596.586   8595.886
Var_linear    22099.43      8896.744    8601.081   8596.759   8596.308
Var_Knot0     22101.39      8875.263    8600.567   8596.648   8596.378
Var_Knot1     22103.07      8876.642    8602.500   8598.269   8598.117
Var_Knot2     22104.68      8878.298    8604.306   8600.331   8600.074
Var_Knot3     22106.61      8879.704    8606.184   8602.313   8602.071
Var_Knot4     22108.57      8881.356    8607.949   8604.102   8603.834
Var_Knot5     22110.42      8882.663    8609.084   8605.298   8605.009
             Mean_Knot2 Mean_Knot3 Mean_Knot4 Mean_Knot5
Var_constant   8591.575   8591.802   8593.630   8593.460
Var_linear     8592.062   8592.183   8594.098   8593.978
Var_Knot0      8592.161   8591.989   8593.989   8594.041
Var_Knot1      8593.733   8593.461   8595.522   8595.608
Var_Knot2      8595.751   8595.468   8597.520   8597.632
Var_Knot3      8597.796   8597.605   8599.627   8599.673
Var_Knot4      8599.654   8599.534   8601.507   8601.539
Var_Knot5      8601.028   8601.010   8602.941   8603.008

$BIC
             Mean_zero Mean_constant Mean_linear Mean_Knot0 Mean_Knot1
Var_constant  22123.30      8910.634    8619.233   8621.704   8627.281
Var_linear    22111.99      8915.584    8626.199   8628.154   8633.978
Var_Knot0     22120.23      8900.381    8631.961   8634.319   8640.324
Var_Knot1     22128.18      8908.036    8640.171   8642.214   8648.336
Var_Knot2     22136.08      8915.969    8648.251   8650.550   8656.566
Var_Knot3     22144.28      8923.650    8656.403   8658.806   8664.835
Var_Knot4     22152.52      8931.575    8664.441   8666.866   8672.869
Var_Knot5     22160.64      8939.156    8671.848   8674.333   8680.314
             Mean_Knot2 Mean_Knot3 Mean_Knot4 Mean_Knot5
Var_constant   8629.245   8635.747   8643.850   8649.952
Var_linear     8636.007   8642.402   8650.590   8656.742
Var_Knot0      8642.381   8648.481   8656.753   8663.076
Var_Knot1      8650.225   8656.225   8664.556   8670.912
Var_Knot2      8658.515   8664.503   8672.824   8679.205
Var_Knot3      8666.831   8672.909   8681.200   8687.514
Var_Knot4      8674.959   8681.107   8689.348   8695.647
Var_Knot5      8682.601   8688.851   8697.049   8703.382

$HQC
             Mean_zero Mean_constant Mean_linear Mean_Knot0 Mean_Knot1
Var_constant  22119.25      8902.527    8607.072   8605.489   8607.012
```

```
Var_linear    22103.89      8903.423    8609.983   8607.885   8609.655
Var_Knot0     22108.07      8884.166    8611.692   8609.996   8611.947
Var_Knot1     22111.97      8887.767    8615.848   8613.838   8615.905
Var_Knot2     22115.81      8891.646    8619.875   8618.120   8620.082
Var_Knot3     22119.95      8895.273    8623.973   8622.321   8624.297
Var_Knot4     22124.14      8899.144    8627.957   8626.328   8628.277
Var_Knot5     22128.21      8902.671    8631.310   8629.741   8631.668
             Mean_Knot2 Mean_Knot3 Mean_Knot4 Mean_Knot5
Var_constant   8604.922   8607.371   8611.419   8613.468
Var_linear     8607.631   8609.972   8614.106   8616.204
Var_Knot0      8609.950   8611.996   8616.215   8618.484
Var_Knot1      8613.741   8615.687   8619.964   8622.266
Var_Knot2      8617.977   8619.911   8624.178   8626.505
Var_Knot3      8622.239   8624.264   8628.501   8630.760
Var_Knot4      8626.313   8628.408   8632.595   8634.840
Var_Knot5      8629.902   8632.097   8636.241   8638.521
```

According to the AIC and the HQC, the optimal model for LP-PLA2 activity is that with two internal knots in the mean and constant variance. Alternatively, the BIC has the lowest value for the linear mean and constant variance model. The model can also be plotted from this with the following code (output suppressed).

```
plotVarReg(best_pla$best.model)
```

9.5 The `lssVarReg()` function

The `lssVarReg()` function fits the regression in the location, scale and shape for a skew-normal response distribution, as detailed in Chapter 8. At this time, only one covariate of interest can be included in these models. The input arguments and their default values are as follows:

- `y`: Vector containing the outcome data. There must be no missing data.
- `x`: Vector containing the covariate data. There must be no missing data and this vector should be the same length as `y`.
- `locationmodel`: Text to specify the location model to be fit to these data. The possible inputs are `constant`, `linear` or `semi`. The option `semi` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.l`.

- `scale2model`: Text to specify the scale (squared) model to be fit to these data. The possible inputs are `constant`, `linear` or `semi`. The option `semi` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.sc`.

- `shapemodel`: Text to specify the shape model to be fit to these data. The possible inputs are `constant`, `linear` or `semi`. The option `semi` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.sh`.

- `knots.l`: Integer indicating the number of internal knots to be fit in the semi-parametric location model. Knots are placed equidistantly over the covariate. The default value is 2.

- `knots.sc`: Integer indicating the number of internal knots to be fit in the semi-parametric scale (squared) model. Knots are placed equidistantly over the covariate. The default value is 2.

- `knots.sh`: Integer indicating the number of internal knots to be fit in the semi-parametric shape model. Knots are placed equidistantly over the covariate. The default value is 2.

- `degree`: Integer indicating the degree of the splines fit in the semi-parametric models. The default value is 2.

- `mono.scale`: Text to indicate whether the scale (squared) model is monotonic. Options are `none`, `inc` or `dec`, with the default as `none`. The option `inc` indicates increasing monotonically and `dec` indicates decreasing monotonically. If the model is linear, the parameter space is constrained (using appropriate `para.space`). For semi-parametric variance models, the appropriate monotonic B-splines are fit in the semi-parametric scale (squared) model.

- `para.space`: Text to specify the parameter space to be searched for the scale (squared) model parameters. `positive` means to only search positive parameter space, `negative` means to search only negative parameter space and `all` means search both. Default is `all`.

- `location.init`: Vector of initial parameter estimates for the location model. Defaults to a vector containing ones.

- `scale2.init`: Vector of initial parameter estimates for the scale (squared) model. Defaults to a vector containing ones.

- `shape.init`: Vector of initial parameter estimates for the shape model. Defaults to a vector of ones.

- `int.maxit`: Number of maximum iterations for the internal location and scale algorithm. Default is 1000 iterations.

- `print.it`: Prints progress of estimates through each iteration.

- `control`: List of control parameters for the algorithm. This includes `maxit` (the maximum number of iterations to be performed), `epsilon` (the positive convergence tolerance) and `bound.tol` (the positive tolerance for specifying the interior of the parameter space. This allows the algorithm to terminate early if an interior maximum is found).

An object of class `lssVarReg` is returned by the function and includes the following components:

- `modeltype`: Text indicating the model that was fit, always `LSS model` at this time.

- `locationmodel`, `scale2model`, `shapemodel`, `knots.l`, `knots.sc`, `knots.sh`, `degree`, `mono.scale`: Returning the input variables as described above

- `converged`: Logical argument indicating if convergence occurred.

- `iterations`: Numeric value of the total iterations performed of the main algorithm (not including the internal EM algorithm).

- `reldiff`: Numeric value of the positive convergence tolerance that occurred at the final iteration.

- `loglik`: Numeric value of the maximised log-likelihood.

- aic.c: Numeric value of the Akaike information criterion corrected for small samples
- aic: Numeric value of the Akaike information criterion
- bic: Numeric value of the Bayesian information criterion
- hqc: Numeric value of the Hannan-Quinn information criterion
- location: Vector of the maximum likelihood estimates of the location parameter(s).
- scale2: Vector of the maximum likelihood estimates of the scale (squared) parameter(s).
- shape: Vector of the maximum likelihood estimates of the shape parameter(s).
- data: Dataframe containing the variables included in the model.

As an example, let us extend our previous model fit to LDL levels (ldl0q) at baseline and the LP-PLA2 activity (PLA_ACTIVITY_0) biomarker. Given our previous optimal model, we fit a semi-parametric model with 2 internal knots in the location and a constant model for the scale squared. We now extend the model to incorporate a constant shape model by using the following command:

```
pla_lss_model<-lssVarReg(y=lipid$ldl0q, x=lipid$PLA_ACTIVITY_0,
locationmodel="semi", scale2model="constant", knots.l = 2,
shapemodel = "constant")
```

This gives the following output below. Note that the dataframe data is suppressed.

```
> pla_lss_model[-21]
$modeltype
[1] "LSS model"
$locationmodel
[1] "semi"
$knots.l
[1] 2
$scale2model
[1] "constant"
$knots.sc
NULL
$shapemodel
[1] "constant"
```

```
$knots.sh
NULL
$degree
[1] 2
$converged
[1] TRUE
$iterations
[1] 238
$reldiff
[1] 9.89781e-07
$loglik
[1] -4286.973
$aic.c
[1] 8587.975
$aic
[1] 8587.946
$bic
[1] 8631.92
$mono.scale
[1] "none"
$hqc
[1] 8603.543
$location
    mean.int L_Knt2_Base1 L_Knt2_Base2 L_Knt2_Base3 L_Knt2_Base4
   3.0241878   -0.1842498    0.5148115    0.8553864    1.0767670
$scale2
Intercept
0.6976118
$shape
[1] 0.8423662
```

The AIC from this model with a constant shape has a lower AIC than that of the model with no shape term, and therefore is the optimal model. A function to plot these three models will be introduced in the next section.

9.6 The `plotlssVarReg()` function

This function produces plots of the mean, variance and shape models, as well as the residuals to assess the assumptions of the model. However, at this time there are no bootstrapped confidence intervals produced for these types of models due to computational time. The input arguments and their default values are as follows:

- `x`: Object of class `lssVarReg`, as output from `lssVarReg()`.

- `knot.lines`: Logical parameter indicating if lines showing where the internal knots are located should be placed on the graphics. Only relevant for semi-parametric models.
- `xlab`: Label for plots for the x variable
- `ylab`: Label for plots for the y variable

The output from this function is a 2 by 2 panel of plots which are

- the mean function over the x variable, with or without the knot lines indicated;
- the variance function over the x variable, with or without the knot lines indicated;
- the skew function over the x variable, with or without the knot lines indicated; and
- a Q-Q plot of the squared residuals from the model, plotted against the Chi-squared (df=1) distribution. For data from a skew-normal distribution, these residuals should follow a Chi-squared (df=1) distribution, regardless of skew.

Using the model produced above for the location, scale and shape model for LP-PLA2 activity, the following code produces these plots:

```
pla.out<-plotlssVarReg(pla_lss_model, knot.lines = TRUE, ylab="Baseline
LDL", xlab = "Baseline PLA2 Activity")
```

The function output includes a dataframe as well as the plot given in Figure 9.4. The dataframe includes:

- `x` and `y` variables
- `eta` (η), the location parameter
- `omega` (ω), the scale parameter
- `shape` (ν), the shape parameter
- `predicted mean` (μ), as given in Equation (8.1)
- `predicted variance` (σ^2), as given in Equation (8.2)

- `predicted skewness` (γ), as given in Equation (8.3)
- `stand.res2`, the standardised residuals squared.

This chapter has demonstrated the development and use of the R package `VarReg` in which the algorithms presented in this thesis can be implemented. The various input and outputs for the main functions of the package have been shown, with the use of an example dataset. The R package documentation is given in the Appendix to this thesis. This package is freely available on the Comprehensive R Archive Network (CRAN), and enables the algorithms developed in this thesis to be used widely in various areas of application, such as measurement error and biomarker analyses.

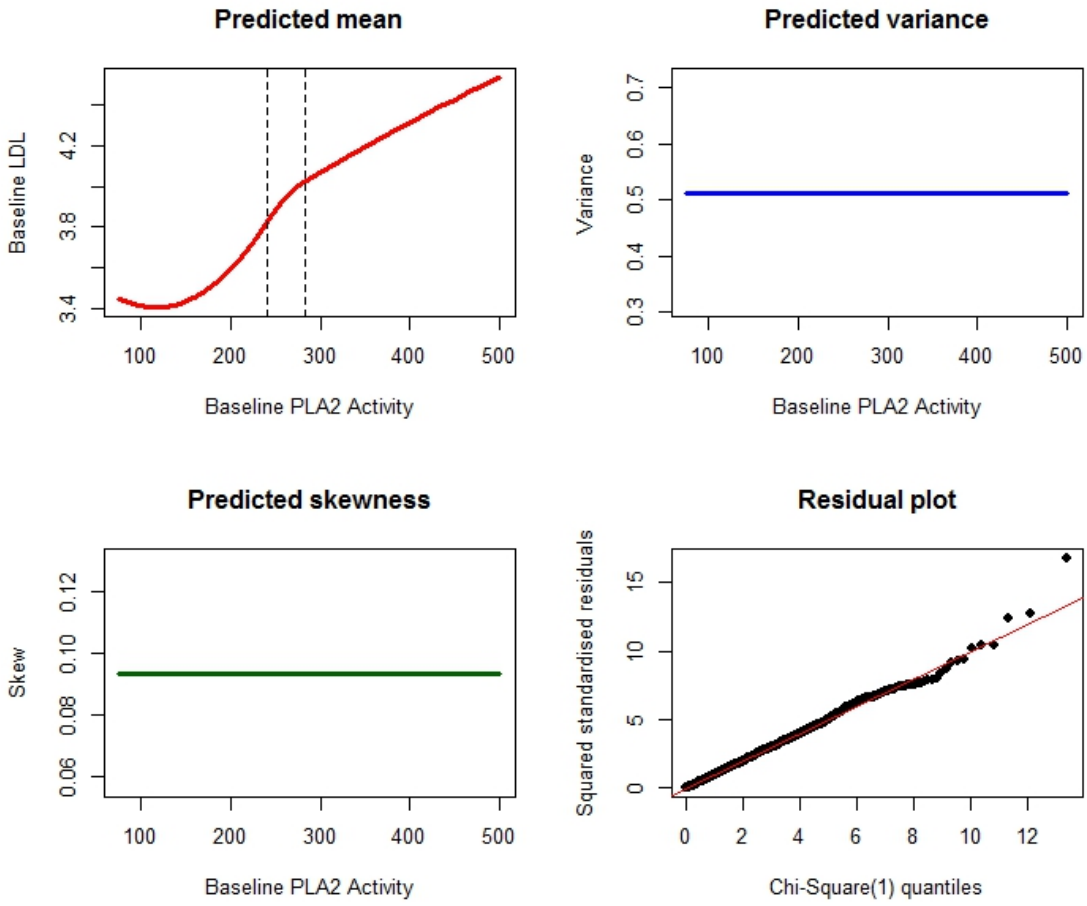

FIGURE 9.4: Plot produced from `plotlssVarReg()` function for PLA2 activity model. Top left: predicted mean function. Top right: predicted variance function. Bottom left: Predicted skew function. Bottom right: residual plot with residuals in black and line of unity in red.

10

Discussion and conclusions

In this thesis, we have developed new methodology for the fitting of semi-parametric variance regression models, and extended this to semi-parametric regression models for location, scale and shape. These methods aid in the analysis of heteroscedastic data, and are of particular use in the analysis of measurement error and biomarker data. This final chapter will review the algorithms produced in this thesis, their implementation and future work.

10.1 Summary of research

The first three chapters of this thesis provided an introduction and background for the new methodology presented in this thesis. In Chapter 2 we introduced the concept of variance regression and discussed some existing methods that are available to fit these models. We also considered some simple examples where existing methods experienced numerical instability and failed to converge. Chapter 3 then detailed some of the

computational methodology that was used in this thesis and an explanation of each of the example datasets.

With this background in place, we then considered a variety of complexities in turn, beginning with a basic approach that was progressively generalised throughout the thesis. The main motivation was that new computational and model fitting methodology are of interest for complex models, where existing methods may encounter the sorts of problems discussed in Chapter 2.

In Chapter 4 we introduced the basic approach, which was based on the idea that an additive variance regression model can be considered as a latent outcome model in which the observed outcome is the sum of two independent latent outcomes. Using a zero mean model and linear variance, the simulation results presented in this chapter illustrated that our approach is valid and has efficiency advantages over a crude unweighted approach. Furthermore, we illustrated that our EM-based approach for fitting the model provides a reliable and stable approach that is not susceptible to the numerical instability that we have seen with other approaches.

This basic method was then generalised in Chapter 5 to fit a regression in both the mean and the variance, with multiple covariates in each. Simulations explored the efficiency of the estimates from both the mean and the variance model, and demonstrated that reliable estimators are obtained. The method was applied to the VCF dataset, where it was shown that a linear model in the mean and the variance was a good fit to the data. However, the residuals had heavy tails, suggesting that the inclusion of additional parameters may improve the fit. This led to the investigation of semi-parametric models in Chapter 6. We showed that the proposed algorithm has broad applicability, as it allows models to be easily generalised to incorporate B-spline basis functions that may be fit in the mean and the variance. It also allowed monotonicity constraints which can be of particular relevance in variance models. Additionally, basis functions could be created for more than one covariate and incorporated into the appropriate model. The flexibility of this algorithm provides a broad framework for fitting semi-parametric models in the variance, and this was demonstrated in two example datasets. From the simulation study in this chapter, the CEM algorithm developed is capable of recovering the true model for all sample sizes. However, the ability to estimate the

model complexity depended largely on the sample size. Automatic model selection worked well for large n, when the AIC should be used. For small n, the AIC should be supplemented with visual inspection of models with more parameters.

Chapter 7 introduced an extra level of complexity, which was to allow the outcome data to be censored. Through a simulation study, we demonstrated that the algorithm can reliably estimate the mean and variance parameters with minimal bias, even with up to 50% censoring. We also illustrated its applicability for assessing measurement error for biomarker data that is subject to both left and right censoring. Although extension of the algorithm to account for censored data is of interest in its own right, we used this extension for an additional purpose, namely, to develop another algorithm that simultaneously allows estimation of location, scale and skewness of the distribution, as a function of covariates. This gave the algorithm even broader applicability and flexibility for non-normal data.

This extension was developed in Chapter 8, using a regression model in the location, scale and shape parameters. We utilised the skew-normal distribution, in conjunction with a coordinate ascent algorithm applied to our EM-type algorithm. Through a simulation study, we demonstrated that this algorithm can reliably estimate the parameters with virtually no bias in large samples. We also illustrated its applicability for assessing variance heterogeneity in real datasets, using the CD4 data example.

Lastly, Chapter 9 demonstrated the development and use of the R package `VarReg` in which the algorithms presented in this thesis are implemented. The various inputs and outputs for the main functions of the package were illustrated, with the use of an example biomarker dataset. In the Appendix to this thesis, the R package documentation is shown. This package is available on the Comprehensive R Archive Network (CRAN), and will enable the algorithms developed in this thesis to be used widely in various areas of application, such as measurement error and biomarker analyses.

10.2 Future work

There are at least three broad areas of future work for the algorithms developed in this thesis. The first is to extend the algorithm to more complex data and models. The second is to enhance the software implementation presented here. Additional theoretical improvements, particularly convergence acceleration, is a third area of potential future work.

In regards to the first area, since the basis of the methods is the EM algorithm, the same approach may be possible in other contexts where the model can be formulated as a missing data or latent variable model. Truncated data is another such example where the model can be thought of as a missing data problem. In contrast to censoring, where we know how many values have been found to be below or above a given detectable limit, for truncated data we have no count of how many values are outside of the bounds, since they are excluded from the sample. Thus, truncated samples can be thought of as being equivalent to a sample being taken with all values outside of the bounds entirely omitted. This type of sampling is common in a clinical trials context, where patients with say a cholesterol level between 155 and 171 mg per decilitre are eligible to be enrolled in a clinical trial. Data on patients that are not eligible to be enrolled would not typically be recorded, and therefore considered to be missing in an EM formulation. An extension of the algorithms in this thesis to truncated data may be useful, especially given the wide use of biomarkers in clinical trials.

In regards to the second area, at this stage the software only allows the incorporation of a single covariate in the mean and variance models. This could be extended with formula syntax to allow the incorporation of multiple covariates or factors. Lastly, there have been many recent theoretical developments to improve the speed of the EM algorithm (Donoghoe and Marschner, 2016; Varadhan and Roland, 2008; Zhou, Alexander, and Lange, 2011). The incorporation of semi-parametric models for the location, scale and shape means the CEM algorithm is computationally expensive, and would become more viable with the use of some of these methods to increase the speed.

10.3 Final remarks

This thesis has introduced new methods for semi-parametric variance regression in the frequentist framework, with extensions to censored data and regression models for the skew of the distribution. These algorithms are widely applicable to measurement error and biomarker analyses. An R package has been created to ensure that these methods can be implemented. Our methods are not presented as the only approach that could be taken for variance regression. Rather, our approach is a useful complement to existing methods for semi-parametric variance regression, particularly where numerical instability is encountered. We have shown how they can be easily extended and that they are reliable with good performance.

Appendix

Package 'VarReg'

May 31, 2017

Type Package

Title Semi-Parametric Variance Regression

Version 1.0.1

Maintainer Kristy Robledo <robledo.kristy@gmail.com>

Description Methods for fitting semi-parametric mean and variance models, with normal or censored data. Also extended to allow a regression in the location, scale and shape parameters.

License GPL-3

Depends R (>= 2.10)

LazyData TRUE

Suggests testthat

Imports splines, stats, graphics, sn, survival, utils

RoxygenNote 6.0.0

NeedsCompilation no

Author Kristy Robledo [aut, cre],
Ian Marschner [ths]

Repository CRAN

Date/Publication 2017-05-30 23:59:40 UTC

R topics documented:

`censlinVarReg` *Censored Linear mean and variance regression*

Description

`censlinVarReg` performs censored multivariate mean and multivariate variance regression. This function is designed to be used by the `semiVarReg` function.

Usage

```
censlinVarReg(dat, mean.ind = c(2), var.ind = c(2), cens.ind = c(3),
  mean.intercept = TRUE, para.space = c("all", "positive", "negative"),
  mean.init = NULL, var.init = NULL, control = list(...), ...)
```

Arguments

`dat`	Dataframe containing outcome and covariate data. Outcome data must be in the first column, with censored values set to the limits. Covariates for mean and variance model in next columns.
`mean.ind`	Vector containing the column numbers of the data in 'dat' to be fit as covariates in the mean model. 0 indicates constant mean option. NULL indicates zero mean option.
`var.ind`	Vector containing the column numbers of the data in 'dat' to be fit as covariates in the variance model. FALSE indicates constant variance option.
`cens.ind`	Vector containing the column number of the data in 'dat' to indicate the censored data. 0 indicates no censoring, -1 indicates left (lower) censoring and 1 indicates right (upper) censoring.
`mean.intercept`	Logical to indicate if an intercept is to be included in the mean model. Default is TRUE.
`para.space`	Parameter space to search for variance parameter estimates. "positive" means only search positive parameter space, "negative" means search only negative parameter space and "all" means search all. Default is all.
`mean.init`	Vector of initial estimates to be used for the mean model.
`var.init`	Vector of initial estimates to be used for the variance model.
`control`	List of control parameters. See `VarReg.control`.
`...`	arguments to be used to form the default control argument if it is not supplied directly

Value

`censlinVarReg` returns a list of output including:

- `converged`: Logical argument indicating if convergence occurred.
- `iterations`: Total iterations performed of the EM algorithm.
- `reldiff`: the positive convergence tolerance that occured at the final iteration.
- `loglik`: Numeric variable of the maximised log-likelihood.
- `boundary`: Logical argument indicating if estimates are on the boundary.
- `aic.c`: Akaike information criterion corrected for small samples
- `aic`: Akaike information criterion
- `bic`: Bayesian information criterion
- `hqc`: Hannan-Quinn information criterion
- `mean.ind`: Vector of integer(s) indicating the column number(s) in the dataframe `data` that were fit in the mean model.
- `mean`: Vector of the maximum likelihood estimates of the mean parameters.
- `var.ind`: Vector of integer(s) indicating the column(s) in the dataframe `data` that were fit in the variance model.
- `variance`: Vector of the maximum likelihood estimates of the variance parameters.
- `cens.ind`: Integer indicating the column in the dataframe `data` that corresponds to the censoring indicator.
- `data`: Dataframe containing the variables included in the model.

`censloop_em` *The Censored data EM loop*

Description

`censloop_em` is an EM loop function for censored data to be utilised by various other higher level functions.

Usage

```
censloop_em(meanmodel, theta.old, beta.old, p.old, x.0, X, censor.ind,
  mean.intercept, maxit, eps)
```

Arguments

`meanmodel`	Dataframe containing only the covariates to be fit in the mean model. NULL for zero mean model and FALSE for constant mean model.
`theta.old`	Vector containing the initial variance parameter estimates to be fit in the variance model.

beta.old	Vector containing the initial mean parameter estimates to be fit in the mean model.
p.old	Vector of length n containing the initial variance estimate.
x.0	Matrix of covariates (length n) to be fit in the variance model. All have been rescaled so zero is the minimum. If NULL, then its a constant variance model.
X	Vector of length n of the outcome variable.
censor.ind	Vector of length n of the censoring indicator. 0=uncensored, -1=left censored and 1 is right censored.
mean.intercept	Logical to indicate if mean intercept is to be included in the model.
maxit	Number of maximum iterations for the EM algorithm.
eps	Very small number for the convergence criteria.

Value

A list of the results from the EM algorithm, including:

- conv: Logical argument indicating if convergence occurred
- it: Total iterations performed of the EM algorithm
- reldiff: the positive convergence tolerance that occured at the final iteration.
- theta.new: Vector of variance parameter estimates. Note that these are not yet transformed back to the appropriate scale
- mean: Vector of mean parameter estimates
- fittedmean: Vector of fitted mean estimates
- p.old: Vector of fitted variance estimates

---

criterion *Calculation of information criterion*

Description

criterion calculates various information criterion for the algorithms in this package

Usage

```
criterion(n, loglik, param)
```

Arguments

n	Number of observations
loglik	Loglikelihood from model
param	Number of parameters fit in model

Value

A list of the four IC

- `aic.c`: Akaike information criterion corrected for small samples
- `aic`: Akaike information criterion
- `bic`: Bayesian information criterion
- `hqc`: Hannan-Quinn information criterion

`linVarReg` *Linear mean and variance regression function*

Description

`linVarReg` performs multivariate mean and multivariate variance regression. This function is designed to be used by the `semiVarReg` function.

Usage

```
linVarReg(dat, var.ind = c(2), mean.ind = c(2), para.space = c("all",
  "positive", "negative"), control = list(...), ...)
```

Arguments

`dat`	Dataframe containing outcome and covariate data. Outcome data must be in the first column. Covariates for mean and variance model in next columns.
`var.ind`	Vector containing the column numbers of the data in 'dat' to be fit as covariates in the variance model. FALSE indicates constant variance option.
`mean.ind`	Vector containing the column numbers of the data in 'dat' to be fit as covariates in the mean model. 0 indicates constant mean option. NULL indicates zero mean option.
`para.space`	Parameter space to search for variance parameter estimates. "positive" means only search positive parameter space, "negative" means search only negative parameter space and "all" means search all.
`control`	List of control parameters. See `VarReg.control`.
`...`	arguments to be used to form the default control argument if it is not supplied directly

Value

`linVarReg` returns a list of output including:

- `converged`: Logical argument indicating if convergence occurred.
- `iterations`: Total iterations performed of the EM algorithm.
- `reldiff`: the positive convergence tolerance that occured at the final iteration.

- loglik: Numeric variable of the maximised log-likelihood.
- boundary: Logical argument indicating if estimates are on the boundary.
- aic.c: Akaike information criterion corrected for small samples
- aic: Akaike information criterion
- bic: Bayesian information criterion
- hqc: Hannan-Quinn information criterion
- mean.ind: Vector of integer(s) indicating the column number(s) in the dataframe data that were fit in the mean model.
- mean: Vector of the maximum likelihood estimates of the mean parameters.
- var.ind: Vector of integer(s) indicating the column(s) in the dataframe data that were fit in the variance model.
- variance: Vector of the maximum likelihood estimates of the variance parameters.
- cens.ind: Integer indicating the column in the dataframe data that corresponds to the censoring indicator. Always NULL.
- data: Dataframe containing the variables included in the model.

---

loop_em *The EM loop for the main mean and variance function*

Description

loop_em is a basic EM loop function to be utilised by various other higher level functions.

Usage

```
loop_em(meanmodel, theta.old, p.old, x.0, X, maxit, eps)
```

Arguments

meanmodel	Dataframe containing only the covariates to be fit in the mean model. NULL for zero mean model and FALSE for constant mean model.
theta.old	Vector containing the initial variance parameter estimates to be fit in the variance model.
p.old	Vector of length n containing the containing the initial variance estimate.
x.0	Matrix of covariates (length n) to be fit in the variance model. All have been rescaled so zero is the minimum. If NULL, then its a constant variance model.
X	Vector of length n of the outcome variable.
maxit	Number of maximum iterations for the EM algorithm.
eps	Very small number for the convergence criteria.

Value

A list of the results from the EM algorithm, including

- `conv`: Logical argument indicating if convergence occurred
- `it`: Total iterations performed of the EM algorithm
- `reldiff`: the positive convergence tolerance that occured at the final iteration.
- `theta.new`: Vector of variance parameter estimates. Note that these are not yet transformed back to the appropriate scale
- `mean`: Vector of mean parameter estimates
- `fittedmean`: Vector of fitted mean estimates
- `p.old`: Vector of fitted variance estimates

`loop_lss` *The EM loop for the LSS model*

Description

`loop_lss` is the EM loop function for the LSS model to be utilised by various other higher level functions

Usage

```
loop_lss(alldat, xiold, omega2old, nuold, mean.ind, var.ind, nu.ind, para.space,
  maxit, eps, int.maxit, print.it)
```

Arguments

`alldat`	Dataframe containing all the data for the models. Outcome in the first column.
`xiold`	Vector of initial location parameter estimates to be fit in the location model.
`omega2old`	Vector of initial scale2 parameter estimates to be fit in the scale2 model.
`nuold`	Vector of initial nu parameter estimates to be fit in the nu model.
`mean.ind`	Vector containing the column numbers of the data in 'alldat' to be fit as covariates in the location model.
`var.ind`	Vector containing the column numbers of the data in 'alldat' to be fit as covariates in the scale2 model. FALSE indicates a constant variance model.
`nu.ind`	Vector containing the column numbers of the data in 'alldat' to be fit as covariates in the nu model. NULL indicates constant model.
`para.space`	Parameter space to search for variance parameter estimates. "positive" means only search positive parameter space, "negative" means search only negative parameter space and "all" means search all.
`maxit`	Number of maximum iterations for the main EM algorithm.
`eps`	Very small number for the convergence criteria.
`int.maxit`	Number of maximum iterations for the internal EM algorithm for the location and scale.
`print.it`	Logical to indicate if the estimates for each iteration should be printed.

Value

A list of the results from the algorithm, including conv, reldiff, it, mean, xi.new, omega2.new, nu.new, fitted.xi

- `conv`: Logical argument indicating if convergence occurred
- `it`: Total iterations performed of the EM algorithm
- `reldiff`: the positive convergence tolerance that occured at the final iteration
- `xinew`: Vector of location parameter estimates
- `omega2new`: Vector of scale squared parameter estimates
- `nunew`: Vector of shape parameter estimates
- `fitted.xi`: Vector of fitted location estimates

`lssVarReg` *Semi parametric location, shape and scale regression*

Description

`lssVarReg` performs a semiparametric location (ξ or xi), shape (ν or nu) and scale (ω or omega) regression model. Currently, this is only designed for a single covariate that is fit in the location, scale and shape models.

Usage

```
lssVarReg(y, x, locationmodel = c("constant", "linear", "semi"),
  scale2model = c("constant", "linear", "semi"), shapemodel = c("constant",
  "linear"), knots.l = 2, knots.sc = 2, knots.sh = 2, degree = 2,
  mono.scale = c("none", "inc", "dec"), para.space = c("all", "positive",
  "negative"), location.init = NULL, scale2.init = NULL,
  shape.init = NULL, int.maxit = 1000, print.it = FALSE,
  control = list(...), ...)
```

Arguments

`y`	Vector containing outcome data. Must be no missing data.
`x`	Vector containing the covariate data, same length as `y`. Must be no missing data.
`locationmodel`	Text to specify the location model to be fit. Options: `"constant"` = constant model (intercept only), `"linear"` = linear term with x covariate, `"semi"` = semi-parametric spline (specify with `knots.l`).
`scale2model`	Text to specify the scale^2 model to be fit. Options: `"constant"` = constant term only, `"linear"` = linear term with `x` covariate, `"semi"` = semiparametric spline (specify with `knots.sc`)
`shapemodel`	Text to specify the shape model to be fit. Options: `"constant"` = constant shape model, `"linear"` = linear term with x covariate, `"semi"` = semiparametric spline (specify with `knots.sh`).

knots.l	Integer indicating the number of internal knots to be fit in the location model. Default is '2'. (Note that the knots are placed equidistantly over x.)
knots.sc	Integer indicating the number of internal knots to be fit in the scale^2 model. Default is '2'. (Note that the knots are placed equidistantly over x.)
knots.sh	Integer indicating the number of internal knots to be fit in the shape model. Default is '2'. (Note that the knots are placed equidistantly over x.)
degree	Integer to indicate the degree of the splines fit in the location and scale. Default is '2'.
mono.scale	Text to indicate whether the scale2 model is monotonic. Default is "none" (no monotonic constraints). Options are "inc" for increasing or "dec" for decreasing. If this is chosen, the appropriate para.space is set autopmatically ("positive" for inc, "negative" for dec).
para.space	Text to indicate the parameter space to search for scale2 parameter estimates. "positive" means only search positive parameter space, "negative" means search only negative parameter space and "all" means search all parameter spaces. Default is all.
location.init	Vector of initial parameter estimates for the location model. Defaults to vector of 1's of appropriate length.
scale2.init	Vector of initial parameter estimates for the scale^2 model. Defaults to vector of 1's of appropriate length.
shape.init	Vector of initial parameter estimates for the shape model. Defaults to vector of 1's of appropriate length.
int.maxit	Integer of maximum iterations for the internal location and scale EM algorithm. Default is 1000 iterations.
print.it	Logical for printing progress of estimates through each iteration. Default is FALSE.
control	List of control parameters for the algorithm. See VarReg.control.
...	arguments to be used to form the default control argument if it is not supplied directly

Value

lssVarReg returns an object of class "lssVarReg", which inherits most from class "VarReg". This object of class lssVarReg is a list of the following components:

- modeltype: Text indicating the model that was fit, always "LSS model".
- locationmodel, scale2model, shapemodel, knots.l, knots.sc, knots.sh, degree,mono.scale : Returning the input variables as described above
- converged: Logical argument indicating if convergence occurred.
- iterations: Total iterations performed of the main algorithm (not including the internal EM algorithm).
- reldiff: the positive convergence tolerance that occured at the final iteration.
- loglik: Numeric variable of the maximised log-likelihood.
- aic.c: Akaike information criterion corrected for small samples

- aic: Akaike information criterion
- bic: Bayesian information criterion
- hqc: Hannan-Quinn information criterion
- location: Vector of the maximum likelihood estimates of the location parameters.
- scale2: Vector of the maximum likelihood estimates of the scale (squared) parameters.
- shape: Vector of the maximum likelihood estimates of the shape parameters.
- data: Dataframe containing the variables included in the model.

See Also

VarReg.control plotlssVarReg

Examples

```
## run a model with linear mean, linear variance and constant shape (not run):
## lssmodel<-lssVarReg(mcycle$accel, mcycle$times,  locationmodel="linear", scale2model="linear",
## shapemodel="constant",  maxit=10000)
```

---

lss_calc *Calculations for SN*

Description

lss_calc performs calculations for transforming SN data (location, scale and shape) to mean, variance and skew. This function is utilised by other, higher level functions.

Usage

```
lss_calc(x)
```

Arguments

x	Object of class lssVarReg (output from lssVarReg).

Value

dataframe containing:

- y: y variable
- x: x variable
- eta: η or fitted location estimates
- omega: ω or fitted scale estimates
- shape: α or fitted shape estimates
- predicted mean: fitted mean estimates
- predicted variance: fitted variance estimates
- Predicted skewness: fitted skewness estimates
- stand.res2: Squared standardised residuals

mcycle *mcycle dataset.*

Description

A dataset containing 133 observations from a simulated motorcycle accident, used to test crash helmets.

Usage

```
mcycle
```

Format

A data frame with 133 rows and 2 variables:

times in milliseconds from time of impact

accel in g, acceleration of the head ...

Source

Silverman, B. W. (1985) Some aspects of the spline smoothing approach to non-parametric curve fitting. Journal of the Royal Statistical Society series B 47, 1-52.

References

Venables, W. N. and Ripley, B. D. (1999) Modern Applied Statistics with S-PLUS. Third Edition. Springer.

Examples

```
library(VarReg)
data(mcycle)
attach(mcycle)
plot(times,accel)
```

plotlssVarReg *Plots graphics for a location, scale and shape regression model*

Description

`plotlssVarReg` is used to produce graphics for models fit in the `VarReg` package with the `lssVarReg` function. As the skew-normal distribution is used to fit this type of model, the data needs to be transformed from the SN parameters (location, scale and shape) to the typical mean, variance and skew parameters.

Usage

```
plotlssVarReg(x, knot.lines = FALSE, xlab = "x", ylab = "y")
```

Arguments

x	Object of class lssVarReg (output from lssVarReg).
knot.lines	Logical to show the knot lines on the graphics (if model is type "semi"). Default is TRUE
xlab	Label to be placed on the x axis of graphics (covariate)
ylab	Label to be placed on the y axis of graphics (outcome)

Value

A graphic is returned, as well as a dataframe. The graphic returned is a 2 by 2 plot of:

- the mean function over the x-variable, with or without the knot lines indicated
- the variance function over the x-variable, with or without the knot lines indicated
- the skew function over the x-variable, with or without the knot lines indicated
- a Q-Q plot of the squared residuals from the model, plotted against the Chi-squared (df=1) distribution. For data from a skew-normal distribution, these residuals should follow a Chi-squared (df=1) distribution, regardless of skew.

The dataframe returned contains the following columns:

- x: x variable
- y: y variable
- eta: (η), the location parameter
- omega: (ω), the scale parameter
- shape: (ν), the shape parameter
- predicted~mean: (μ), the mean
- predicted~variance: (σ^2), the variance
- predicted~skewness: (γ), the skew
- stand.res2: the standardised residuals squared.

See Also

lssVarReg

Examples

```
data(mcycle)
## not run. LSS model followed by the basic plot command
##lssmodel<-lssVarReg(mcycle$accel, mcycle$times, locationmodel="linear", scale2model="linear",
##shapemodel="constant", maxit=10000)
##lssplot_out<-plotlssVarReg(lssmodel, xlab="Time in seconds", ylab="Acceleration")
```

plotVarReg *Plots graphics for a mean and variance regression model*

Description

plotVarReg to produce graphics for models fit in this package.

Usage

```
plotVarReg(x, knot.lines = FALSE, ci = FALSE, ci.type = c("im", "boot"),
  bootreps = 1000, xlab = "x", ylab = "y", control = list(...), ...)
```

Arguments

x	Object of class VarReg (see semiVarReg).
knot.lines	Logical to indicate if knot lines should be shown on graphics (if model is type "semi"). Default is FALSE
ci	Logical indicate if 95% CI should be shown on the plots. Default is FALSE and ci.type="im".
ci.type	Text to indicate the type of CI to plot. Either "im" (information matrix) or "boot" (bootstrapped). Default is "im".
bootreps	Integer to indicate the number of bootstrap replications to be performed if ci.type="boot". Default is 1000.
xlab	Text for the label to be placed on the x axis of graphics (covariate)
ylab	Text for the label to be placed on the y axis of graphics (outcome)
control	list of control parameters to be used in bootstrapping. See VarReg.control.
...	arguments to be used to form the default control argument if it is not supplied directly

Value

This function returns a 2x2 plot, with slightly different plots given, depending on the outcome data. For uncensored data, the plots are:

- the mean function over the x-variable, with or without 95% CI, and with or without the knot lines indicated
- the variance function over the x-variable, with or without 95% CI and with or without the knot lines indicated
- a Q-Q plot of the residuals from the model
- a histogram of the residuals from the model

If the outcome data is censored, the last two plots are no longer appropriate. Given the censored residuals from the model, we can compare the squared standardised residuals (given in black) with their censoring indicator to the chi-squared distribution with one degree of freedom (given in red). This is one of the plots given for censored data, and the other is a plot of the data, coloured by the censoring status. The triangles with the point at the top are bottom censored and the triangles with the point at the bottom are top censored.

See Also

semiVarReg, VarReg.control

Examples

```
data(mcycle)
linmodel<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="linear", varmodel="linear",
maxit=10000)
plotVarReg(linmodel)
plotVarReg(linmodel, ci=TRUE, ci.type="im", ylab="Range", xlab="Time in seconds")
##not run
##plotVarReg(linmodel, ci=TRUE, ci.type="boot", bootreps=10,ylab="Acceleration",
##xlab="Time in seconds")

##not run
##semimodel<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="semi", varmodel="semi",
##knots.m=4, knots.v=2, maxit=10000)
##plotVarReg(semimodel, ci=TRUE, ci.type="boot",bootreps=10,ylab="Acceleration",
##xlab="Time in seconds", maxit=10000)
```

---

searchVarReg *Searches for best semi parametric mean and variance regression model*

Description

searchVarReg performs multiple semi-parametric mean and variance regression models for a covariate of interest, in order to search for the optimal number of knots. The best model is chosen based on the information criterion of preference ("selection"). At the moment, this is only designed for a single covariate that is fit in both the mean and variance models.

Usage

```
searchVarReg(y, x, cens.ind = NULL, maxknots.m = 3, maxknots.v = 3,
  degree = 2, mono.var = c("none", "inc", "dec"), selection = c("AIC",
  "AICc", "HQC", "BIC"), print.it = FALSE, control = list(...), ...)
```

Arguments

y	Vector containing outcome data. Must be no missing data and any censored values must be set to the limits of detection.
x	Vector containing the covariate data. Must be no missing data and same length as y.
cens.ind	Vector containing the censoring indicator, if applicable. There must be no missing data contained in the vector and this vector should be the same length as y. "0" values indicate uncensored data, "1" indicates right, or upper, censoring and "-1" indicates left, or lower, censoring. The default is NULL which indicates there is no censored data.

maxknots.m	Integer indicating the maximum number of internal knots to be fit in the mean model. Default is 3. (Note that the knots are placed equidistantly over x.)
maxknots.v	Integer indicating the maximum number of internal knots to be fit in the variance model. Default is 3. (Note that the knots are placed equidistantly over x.)
degree	The degree of the splines fit in the mean and variance. Default is 2.
mono.var	Text to indicate whether the variance model is monotonic (only applied to 'linear' or semi-parametric variance models). Default is "none" (no monotonic constraints). Options are "inc" for increasing or "dec" for decreasing. If the variance model is linear, the parameter space is constrained (positive for increasing and negative for decreasing). For semi-parametric variance models, the appropriate monotonic B splines are fit in the semi-parametric variance model.
selection	Text to indicate which information criteria is to be used for the selection of the best model. Choices are "AIC", "AICc", "BIC" and "HQC". Default is "AIC".
print.it	Logical to indicate whether to print progress from each model as the models are performed. Default is FALSE.
control	list of control parameters. See VarReg.control.
...	arguments to be used to form the default control argument if it is not supplied directly

Details

A matrix of models are performed, of increasing complexity. Mean models start at a zero mean model, then constant mean, linear, 0 internal knots, etc, up to a maximum internal knots as specified in maxknots.m. Variance models start at constant variance, linear variance, 0 internal knots, etc, up to max internal knots as specified in maxknots.v.

Note that this function can take some time to run, due to the number of models to be fit. A window will appear on windows based systems to show a progress bar for the function.

Value

searchVarReg returns an list, with the following components:

- ll: a dataframe of the log-likelihoods from each of the models that have been fit.
- AIC: a dataframe of the AIC from each of the models that have been fit. The parameters fit in the mean model are given in the columns, and the parameters in the variance are given in the rows.
- AICc: a dataframe of the AIC-c from each of the models that have been fit.
- BIC: a dataframe of the BIC from each of the models that have been fit.
- HQC: a dataframe of the HQC from each of the models that have been fit.
- best.model: an object of class VarReg (see semiVarReg) containing the output from the optimal model (that model within the specified models in the mean and variance with the lowest information criterion according to the criterion selected).

See Also

semiVarReg, VarReg.control

Examples

```
data(mcycle)
### not run
### find<-searchVarReg(mcycle$accel, mcycle$times, maxknots.v=3, maxknots.m=3,
### selection="HQC", maxit=10000)
```

`semiVarReg` *Semi parametric mean and variance regression*

Description

`semiVarReg` performs semi-parametric mean and variance regression models. Currently, this is only designed for a single covariate that is fit in the mean and variance models.

Usage

```
semiVarReg(y, x, cens.ind = NULL, meanmodel = c("zero", "constant",
  "linear", "semi"), mean.intercept = TRUE, varmodel = c("constant",
  "linear", "semi"), knots.m = 2, knots.v = 2, degree = 2,
  mono.var = c("none", "inc", "dec"), para.space = c("all", "positive",
  "negative"), control = list(...), ...)
```

Arguments

`y`	Vector containing outcome data. Must be no missing data and any censored values must be set to the limits of detection.
`x`	Vector containing the covariate data. Must be no missing data and same length as `y`.
`cens.ind`	Vector containing the censoring indicator, if applicable. There must be no missing data contained in the vector and this vector should be the same length as `y`. `"0"` values indicate uncensored data, `"1"` indicates right, or upper, censoring and `"-1"` indicates left, or lower, censoring. The default is `NULL` which indicates there is no censored data.
`meanmodel`	Text to specify the mean model to be fit to the data. The possible inputs are `"zero"`, `"constant"`, `"linear"` or `"semi"`. `"semi"` indicates a semi-parametric spline model, with the number of internal knots specified in `knots.m`.
`mean.intercept`	Logical argument to indicate if the mean model is to include an intercept term. This option is only available in the censored mean model, and the default=`TRUE`.
`varmodel`	Text to specify the variance model to be fit to the data. The possible inputs are `"constant"`, `"linear"` or `"semi"`. `"semi"` indicates a semi-parametric B-spline model, with the number of internal knots specified in `knots.v`.
`knots.m`	Integer indicating the number of internal knots to be fit in the semi-parametric mean model. Knots are placed equidistantly over the covariate. The default value is `2`.

knots.v	Integer indicating the number of internal knots to be fit in the semi-parametric variance model. Knots are placed equidistantly over the covariate. The default value is 2.
degree	Integer indicating the degree of the splines fit in the mean and the variance models. The default value is 2.
mono.var	Text to indicate whether the variance model is monotonic. Note that this is not available for the "constant" variance model. Options are "none", "inc" or "dec", with the default="none". "Inc" indicates increasing monotonic and "dec" indicates decreasing monotonic. If the variance model is linear, the parameter space is constrained (positive for increasing and negative for decreasing). For semi-parametric variance models, the appropriate monotonic B-splines are fit in the semi-parametric variance model.
para.space	Text to indicate the parameter space to search for scale2 parameter estimates. "positive" means only search positive parameter space, "negative" means search only negative parameter space and "all" means search all parameter spaces. Default is all.
control	list of control parameters. See VarReg.control.
...	arguments to be used to form the default control argument if it is not supplied directly

Value

semiVarReg returns an object of class "VarReg" which inherits some components from the class "glm". This object of class "VarReg" is a list containing the following components:

- modeltype: Text indicating the model that was fit, indicating if a censored approach or an uncensored approach was performed.
- knots.m, knots.v, degree, meanmodel, varmodel: Returning the input variables as described above
- converged: Logical argument indicating if convergence occurred.
- iterations: Total iterations performed.
- reldiff: the positive convergence tolerance that occurred at the final iteration.
- loglik: Numeric variable of the maximised log-likelihood.
- boundary: Logical argument indicating if the MLE is on the boundary of the parameter space.
- aic.c: Akaike information criterion corrected for small samples
- aic: Akaike information criterion
- bic: Bayesian information criterion
- hqc: Hannan-Quinn information criterion
- mean.ind: Vector of integer(s) indicating the column number(s) in the dataframe data that were fit in the mean model.
- mean: Vector of the maximum likelihood estimates of the mean parameters.
- var.ind: Vector of integer(s) indicating the column(s) in the dataframe data that were fit in the variance model.

- variance: Vector of the maximum likelihood estimates of the variance parameters.
- cens.ind: Integer indicating the column in the dataframe data that corresponds to the censoring indicator.
- data: Dataframe containing the variables included in the model.

Examples

```
data(mcycle)
## run a model with linear mean and linear variance:
linmodel<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="linear", varmodel="linear",
 maxit=10000)
## run a model with semi-parametric mean (4 internal knots) and semi-parametric variance (2 knots):
##not run
##semimodel<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="semi", varmodel="semi",
##knots.m=4, knots.v=2, maxit=10000)
## run a model with semi-parametric mean (4 internal knots) and semi-parametric monotonic
## variance (2 knots):
## not run
##semimodel_inc<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="semi", varmodel="semi",
##knots.m=4, knots.v=2, mono.var="inc")
```

---

seVarReg *SE calculations for mean and variance regression models*

Description

seVarReg calculates SE for an object of class VarReg. If the result is not on a boundary, the Fishers Information matrix SE are given. The bootstrapped 95% CI can also be calculated. Designed to be called by the plot function plotVarReg, rather than run by a user.

Usage

```
seVarReg(x, boot = FALSE, bootreps = 1000, vector.mean = x$data[, 2],
  vector.variance = x$data[, 2], control = list(...), ...)
```

Arguments

x	Object of class VarReg to determin the SE (eg. result from semiVarReg).
boot	Logical to indicate if bootstrapped CI should be calculated. Default is FALSE.
bootreps	Number of bootstraps to be performed if boot=TRUE. Default is 1000.
vector.mean	Vector of x values for which the SE of the mean is to be calculated. Default is the x covariate from the model.
vector.variance	Vector of x values for which the SE of the variance is to be calculated. Default is the actual x covariate from the model.
control	List of control parameters for the bootstrapped models. See VarReg.control.
...	arguments to be used to form the default control argument if it is not supplied directly

Value

The result is a list of results. This includes:

- `mean.est`: dataframe of overall results from the mean model, including parameter estimates from the model, SEs from information matrix (if `boundary=FALSE`) and if specified, the SE from bootstrapping with the bootstrapped 95% CI.
- `variance.est`: dataframe of overall results from the variance model, including parameter estimates from the model, SEs from information matrix (if `boundary=FALSE`) and if specified, the SE from bootstrapping with the bootstrapped 95% CI.
- `mean.im`: dataframe of the expected information matrices for the mean (as appropriate)
- `variance.im`: dataframe of the expected information matrices for the variance (as appropriate)
- `mean.outputs`: dataframe with complete output for mean graphics. Includes the `vector.mean` as input, and the mean vector (`mean.mean`) and the SE vector `mean.se.im`, and bootstrapping outputs as appropriate.
- `variance.outputs`: dataframe with complete output for variance graphics. Includes the `vector.variance` as input, and the mean vector (`var.mean`) and the SE vector `var.se.im`, and bootstrapping outputs as appropriate.

See Also

`semiVarReg`, `VarReg.control`

Examples

```
data(mcycle)
##Fit model with range as a covariate in the mean and the variance model
semimodel<-semiVarReg(mcycle$accel, mcycle$times, meanmodel="semi", varmodel="linear",
knots.m=4, maxit=10000)
##Calculate SE
se1<-seVarReg(semimodel, boot=FALSE)
##not run: with bootstrapping
##se2<-seVarReg(semimodel, boot=TRUE, bootreps=10)
##not run: calculate mean and SE for a given sequence
##test.seq<-seq(min(mcycle$times), max(mcycle$times),
##by=((max(mcycle$times)-min(mcycle$times))/999))
##se2<-seVarReg(semimodel, boot=TRUE, bootreps=10, vector.mean=test.seq)
```

`VarReg` *VarReg: Semi-parametric mean and variance regression*

Description

Methods for fitting semi-parametric mean and variance models, with normal or censored data. Also extended to allow a regression in the location, scale and shape parameters.

Details

This package provides functions to fit semi-parametric mean and variance regression models. These models are based upon EM-type algorithms, which can have more stable convergence properties than other algorithms for additive variance regression models.

The primary function to use for linear and semi-parametric mean and variance models is `semiVarReg`. This function also is able to fit models to censored outcome data. There is also a plot function for these models called `plotVarReg`. A search function has also been produced in order to assist users to find the optimal number of knots in the model (`searchVarReg`).

The other functions that are of particular use are `lssVarReg` and its plot function `plotlssVarReg`. This uses the skew-normal distribution and combines the EM algorithm with a coordinate-ascent type algorithm in order to fit a regression model in the location, scale and shape, therefore extending the semi-parametric models to non-normal data.

Author(s)

Kristy Robledo <robledo.kristy@gmail.com>

`VarReg.control` *Auxillary for controlling VarReg fitting*

Description

Use `VarReg.control` to determine parameters for the fitting of `semiVarReg`. Typically only used internally within functions.

Usage

```
VarReg.control(bound.tol = 1e-05, epsilon = 1e-06, maxit = 1000)
```

Arguments

`bound.tol`	Positive tolerance for specifying the interior of the parameter space. This allows the algorithm to terminate early if an interior maximum is found. If set to `bound.tol=Inf`, no early termination is attempted.
`epsilon`	Positive convergence tolerance. If θ is a vector of estimates, convergence is declared when $\sqrt{(\sum(\theta_{old} - \theta_{new})^2)}/\sqrt{\sum(\theta_{old})^2}$. This should be smaller than `bound.tol`.
`maxit`	integer giving the maximum number of EM algorithm iterations for a given parameterisation.

Details

This is used similarly to `glm.control`. If required, it may be internally passed to another function.

Value

A list of the three components: `bound.tol`, `epsilon` and `maxit` .

vcf *vcf dataset.*

Description

A dataset containing 100 observations of mean velocity of circumferential fibre shortening (vcf), made by long axis and short axis echocardiography.

Usage

```
vcf
```

Format

A data frame with 133 rows and 3 variables:

pid patient identifier

vcflong vcf measurement from long axis

vcfshort vcf measurement from short axis ...

Source

Data from Bland JM, Altman DG. (1986) Statistical methods for assessing agreement between two methods of clinical measurement. Lancet i, 307-310. (Supplied by Paul D'Arbela)

Examples

```
library(VarReg)
data(vcf)
attach(vcf)
plot(rowMeans(vcf[-1]),vcf$vcflong-vcf$vcfshort)
```

Index

Bibliography

Aitkin, M. A. (1964). Correlation in a singly truncated bivariate normal distribution. *Psychometrika* **29**(3): 263–270. DOI: 10.1007/BF02289723.

Aitkin, M. (1987). Modelling variance heterogeneity in normal regression using GLIM. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **36**(3): 332–339. DOI: 10.2307/2347792.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**(6): 716–723. DOI: 10.1109/TAC.1974.1100705.

Álvarez Estévez, M., N. Chueca Porcuna, V. Guillot Suay, A. Peña Monge, F. García García, L. Muñoz Medina, D. Vinuesa García, J. Parra Ruiz, J. Hernández-Quero, and F. García García (2013). Quantification of viral loads lower than 50 copies per milliliter by use of the Cobas AmpliPrep/Cobas TaqMan HIV-1 test, version 2.0, can predict the likelihood of subsequent virological rebound to >50 copies per milliliter. *Journal of Clinical Microbiology* **51**(5): 1555–1557. DOI: 10.1128/JCM.00100-13.

Azzalini, A. (2016). *The R package* **sn***: The skew-normal and skew-t distributions (version 1.4-0).* Università di Padova, Italia. URL: http://azzalini.stat.unipd.it/SN.

Azzalini, A. (2013). *The skew-normal and related families.* Cambridge University Press.

Biomarkers Definitions Working Group (2001). Biomarkers and surrogate endpoints: preferred definitions and conceptual framework. *Clinical Pharmacology and Therapeutics* **69**(3): 89–95. DOI: 10.1067/mcp.2001.113989.

Bland, J. M. and D. G. Altman (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet* **327**(8476): 307–310. DOI: 10.1016/S0140-6736(86)90837-8.

Bland, J. M. and D. G. Altman (1999). Measuring agreement in method comparison studies. *Statistical Methods in Medical Research* **8**(2): 135–60. DOI: 10.1191/096228099673819272.

Box, G. E. P. and D. R. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)* **26**(2): 211–252.

Cole, T. J. and P. J. Green (1992). Smoothing reference centile curves - the LMS method and penalized likelihood. *Statistics in Medicine* **11**(10): 1305–1319. DOI: 10.1002/sim.4780111005.

Crisp, A. and J. Burridge (1994). A note on nonregular likelihood functions in heteroscedastic regression models. *Biometrika* **81**(3): 585–587. DOI: 10.1093/biomet/81.3.585.

Darbela, P. G., Z. M. Silayan, and J. M. Bland (1986). Comparability of M-mode echocardiographic long axis and short axis left-ventricular function derivatives. *British Heart Journal* **56**(5): 445–449. DOI: 10.1136/hrt.56.5.445.

Davidian, M. and R. J. Carroll (1987). Variance function estimation. *Journal of the American Statistical Association* **82**(400): 1079–1091. DOI: 10.1080/01621459.1987.10478543.

De Boor, C. (1978). *A practical guide to splines*. New York: Springer-Verlag.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1): 1–38.

Donoghoe, M. W. and I. C. Marschner. Fast stable relative risk regression using an overparameterised EM algorithm. *Proceedings of the 31st International Workshop on Statistical Modelling*, 2016. **1**: 93–98.

Hannan, E. J. and B. G. Quinn (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)* **41**(2): 190–195.

Hastie, T. and R. Tibshirani (1990). *Generalized additive models*. 1st edition. London: Chapman and Hall.

Klein, N., T. Kneib, S. Klasen, and S. Lang (2015). Bayesian structured additive distributional regression for multivariate responses. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **64**(4): 569–591. DOI: 10.1111/rssc.12090.

Klein, N., T. Kneib, S. Lang, and A. Sohn (2015). Bayesian structured additive distributional regression with an application to regional income inequality in Germany. *The Annals of Applied Statistics* **9**(2): 1024–1052. DOI: 10.1214/15-AOAS823.

Kneib, T. (2016). Comment. *Journal of the American Statistical Association* **111**(516): 1563–1565. DOI: 10.1080/01621459.2016.1250576.

Kuritzkes, D. R., I. Marschner, V. A. Johnson, R. Bassett, J. J. Eron, M. A. Fischl, R. L. Murphy, K. Fife, J. Maenza, M. E. Rosandich, D. Bell, K. Wood, J. P. Sommadossi, and C. Pettinelli (1999). Lamivudine in combination with zidovudine, stavudine, or didanosine in patients with HIV-1 infection. A randomized, double-blind, placebo-controlled trial. National Institute of Allergy and Infectious Disease AIDS Clinical Trials Group Protocol 306 Investigators. *AIDS* **13**(6): 685–94. DOI: 10.1097/00002030-199904160-00009.

Lange, K. (2013). *Optimization*. Springer. ISBN: 9781461458388 (electronic bk.) 1461458382 (electronic bk.)

Liu, C. and D. B. Rubin (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika* **81**(4): 633–648. DOI: 10.2307/2337067.

Lumley, T., R. Kronmal, and S. Ma (2006). Relative risk regression in medical research: models, contrasts, estimators, and algorithms. *University of Washington Biostatistics Working Paper Series. Working Paper 293.* URL: http://biostats.bepress.com/uwbiostat/paper293/.

Marschner, I. C. (2014). Combinatorial EM algorithms. *Statistics and Computing* **24**(6): 921–940. DOI: 10.1007/s11222-013-9411-7.

McLachlan, G. J. and T. Krishnan (2007). *The EM Algorithm and extensions*. New York: Wiley. ISBN: 9780470191613. DOI: 10.1002/9780470191613.

Menictas, M. and M. P. Wand (2015). Variational inference for heteroscedastic semiparametric regression. *Australian and New Zealand Journal of Statistics* **57**(1): 119–138. DOI: 10.1111/anzs.12105.

Mills, J. P. (1926). Table of the ratio: area to bounding ordinate, for any portion of normal curve. *Biometrika* **18**(3/4): 395–400. DOI: 10.2307/2331957.

Ormerod, J. T. and M. P. Wand (2010). Explaining variational approximations. *The American Statistician* **64**(2): 140–153. DOI: 10.1198/tast.2010.09058.

R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org/.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science* **3**(4): 425–441. DOI: 10.1214/ss/1177012761.

Rigby, R. A. and D. M. Stasinopoulos (1996). A semi-parametric additive model for variance heterogeneity. *Statistics and Computing* **6**(1): 57–65. DOI: 10.1007/BF00161574.

Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **54**: 507–544. DOI: 10.1111/j.1467-9876.2005.00510.x.

Robledo, K. (2017). *VarReg: Semi-parametric variance regression*. R package version 1.0. URL: https://CRAN.R-project.org/package=VarReg.

Ruppert, D., M. P. Wand, and R. J. Carroll (2003). *Semiparametric Regression*. New York: Cambridge University Press.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6**(2): 461–464. DOI: 10.1214/aos/1176344136.

Sigrist, M. W. (1994). *Air monitoring by spectroscopic techniques*. New York: Wiley.

Smyth, G. K. (2002). An efficient algorithm for REML in heteroscedastic regression. *Journal of Computational and Graphical Statistics* **11**(4): 836–847. DOI: 10.1198/106186002871.

Stasinopoulos, D. M. and R. A. Rigby (2007). Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software* **23**(7): DOI: 10.18637/jss.v023.i07.

Stasinopoulos, M. D., R. A. Rigby, G. Z. Heller, V. Voudouris, and F. D. Bastiani (2017). *Flexible regression and smoothing : using GAMLSS in R*. Chapman and Hall/CRC. ISBN: 9781138197909 1138197904.

Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics - Theory and Methods* **7**(1): 13–26. DOI: 10.1080/03610927808827599.

The Long-Term Intervention with Pravastatin in Ischaemic Disease Study Group (1998). Prevention of cardiovascular events and death with pravastatin in patients with coronary heart disease and a broad range of initial cholesterol levels. *New England Journal of Medicine* **339**(19): 1349–1357. DOI: 10.1056/NEJM199811053391902.

Varadhan, R. and C. Roland (2008). Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scandinavian Journal of Statistics* **35**(2): 335–353. DOI: 10.1111/j.1467-9469.2007.00585.X.

Venables, W. N. and B. D. Ripley (2002). *Modern applied statistics with S*. Fourth edition. New York: Springer. URL: http://www.stats.ox.ac.uk/pub/MASS4.

Verbyla, A. P. (1993). Modelling variance heterogeneity: residual maximum likelihood and diagnostics. *Journal of the Royal Statistical Society. Series B (Methodological)* **55**(2): 493–508.

Wade, A. M. and A. E. Ades (1994). Age-related reference ranges: significance tests for models and confidence intervals for centiles. *Statistics in Medicine* **13**(22): 2359–67. DOI: 10.1002/sim.4780132207.

Wand, M. (2014). *SemiPar: Semiparametic regression*. R package version 1.0-4.1. URL: http://CRAN.R-project.org/package=SemiPar.

Wand, M. P., J. T. Ormerod, S. A. Padoan, and R. Fuhrwirth (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Analysis* **6**(4): 847–900. DOI: 10.1214/11-BA631.

White, H. D., A. Tonkin, J. Simes, R. Stewart, K. Mann, P. Thompson, D. Colquhoun, M. West, P. Nestel, D. Sullivan, A. C. Keech, D. Hunt, and S. Blankenberg (2014). Association of contemporary sensitive troponin I levels at baseline and change at 1 year with long-term coronary events following myocardial infarction or unstable angina. Results from the LIPID study (Long-Term Intervention With Pravastatin in Ischaemic Disease). *Journal of the American College of Cardiology* **63**(4): 345–354. DOI: 10.1016/j.jacc.2013.08.1643.

Zhou, H., D. Alexander, and K. Lange (2011). A quasi-Newton acceleration for high-dimensional optimization algorithms. *Statistics and Computing* **21**(2): 261–273. DOI: 10.1007/s11222-009-9166-3.