Dipl.-Ing. Sanela Omerovic, BSc

Fitting Mixtures of Generalized Nonlinear Models

DOCTORAL THESIS

to achieve the university degree of

Doktorin der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Herwig Friedl

Institute of Statistics

Graz, March 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date

Signature

Contents

Abbreviations **V**

Symbols **VII**

Figures **IX**

Listings **XI**

Tables **XIII**

Introduction **1**

1 Generalized Nonlinear Models **5**
1.1 Nonlinear Regression Models 6
1.2 Linear Exponential Family 8
1.2.1 Members of the Exponential Family 10
1.2.2 Maximum Likelihood Estimation 12
1.3 Dispersion Parameter 14
1.3.1 Deviance 14
1.3.2 Pearson Statistics 15
1.4 Model Specification 16

2 Finite Mixtures of Generalized Nonlinear Models **19**
2.1 Model Specification 20
2.2 Identifiability 21
2.3 Maximum Likelihood Estimation 24
2.4 Expectation-Maximization (EM) Algorithm 24
2.4.1 Incomplete-Data Structure Problem 25
2.4.2 Definitions of the EM Algorithm 26
2.4.3 Expectation-Step 27
2.4.4 Maximization-Step 28
2.4.5 Iteration Procedure and Convergence Aspects 34

2.4.6 Geometric Properties and Gradient-Based Methods 36
2.5 Standard Errors 37
2.5.1 Score Statistics and Missing Information 39
2.5.2 Louis' Method for Standard Error Computation 40
2.6 Extensions of the EM Algorithm 40
2.6.1 Classification-Expectation-Maximization (CEM) Algorithm 41
2.6.2 Stochastic-Expectation-Maximization (SEM) Algorithm 41
2.7 Number of Components 42
2.7.1 Model Selection Criteria 43

3 Mixtures of Generalized Nonlinear Models in R 45
3.1 S4 Language in R 46
3.2 Generalized Nonlinear Models in R 46
3.2.1 The Function `nls()` in R 47
3.2.2 The Package **gnm** in R 48
3.3 Finite Mixture Models in R 52
3.3.1 The Package **flexmix** in R 53
3.3.2 The M-Step Driver for Mixtures of Generalized Linear Models . . . 56
3.4 Extending **flexmix** for Mixtures of Generalized Nonlinear Models 58
3.4.1 The M-Step Driver for Mixtures of Generalized Nonlinear Models . 59
3.5 Standard Errors 62
3.5.1 Standard Errors in **flexmix** 63
3.5.2 Exact Computation 63
3.5.3 Numerical Derivation 64

4 Monte Carlo Simulation Study for Gamma Mixture Models 65
4.1 Simulation Setup 66
4.1.1 Model Specification 66
4.1.2 Mean Function 66
4.1.3 Initial Configuration 67
4.1.4 Data Generation 68
4.2 Parameters of Interest and Measures of Algorithm Performance 69
4.2.1 Parameter Estimates 69
4.2.2 Measures of Algorithm Performance 71
4.3 Fitting Procedure 71
4.3.1 Assessment of Components 72
4.3.2 Exemplary Application of the Fitting Procedure 73
4.3.3 Simulation Procedure 77
4.4 Simulation Results (Sample Size $n = 1\,000$) 77
4.4.1 Parameter Estimates 78
4.4.2 Standard Errors and Confidence Intervals 83
4.5 Simulation Results (Sample Size $n = 500$) 87
4.5.1 Parameter Estimates 87
4.5.2 Standard Errors and Confidence Intervals 92
4.6 Conclusions 95

5 Modeling Gas Flow on Exits of Gas Transmission Networks 97
5.1 Model Framework . . . 98
5.1.1 Sigmoid Mean Functions . . . 98
5.1.2 Mixture Distributions . . . 101
5.1.3 Predictions . . . 104
5.2 Gas Flow Data 1 . . . 106
5.2.1 Nonlinear Regression . . . 107
5.2.2 Two-Component Mixtures of GNMs . . . 111
5.2.3 Model Comparison . . . 119
5.2.4 Predictions of Gas Flow on Design Temperatures . . . 119
5.2.5 Final Remarks . . . 121
5.3 Gas Flow Data 2 . . . 123
5.3.1 Two-Component Gamma Mixture Models . . . 124
5.3.2 Three-Component Gamma Mixture Models . . . 129
5.3.3 Model Comparison . . . 132
5.3.4 Predictions of Gas Flow on Design Temperatures . . . 134
5.3.5 Final Remarks . . . 135
5.4 Gas Flow Data 3 . . . 136
5.4.1 Two-Component Mixtures of GNMs . . . 137
5.4.2 Model Comparison . . . 140
5.4.3 Predictions of Gas Flow on Design Temperatures . . . 141
5.4.4 Final Remarks . . . 141
5.5 Conclusions . . . 143

6 Modeling Multiple Regimes in Economic Growth 145
6.1 Solow Model with Human Capital Accumulation . . . 146
6.2 Country Data Set . . . 147
6.3 Starting Configuration and Simulation . . . 149
6.4 Nonlinear Regression . . . 150
6.5 Two-Component Mixtures of GNMs . . . 151
6.6 Conclusions . . . 158

Final Remarks 161

Appendices 163
A Definitions . . . 163
B Equilibrium or Steady State in the Solow Model . . . 164
C Packages in R . . . 166

Bibliography 167

Abbreviations

AIC	Akaike Information Criterion
AGCS	Gas Clearing and Settlement AG
BIC	Bayesian Information Criterion
cdf	cumulative density function
CEM	Classification-Expectation-Maximization
CRAN	Comprehensive R Archive Network
df	degrees of freedom
EFNM	Exponential Family Nonlinear Model
EIC	Energy Identification Code
EM	Expectation-Maximization
ENTSOG	European Network of Transmission System Operators for Gas
ESS	Explained Sum of Squares
FMM	Finite Mixture Model
GDP	Gross Domestic Product
GLM	Generalized Linear Model
GNM	Generalized Nonlinear Model
ICL	Integrated Classification Likelihood
IWLS	Iteratively Re-weighted Least Squares
iid	identically and independently distributed
LS	Least Squares
LR	Likelihood Ratio
MC	Monte Carlo
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
NLS	Nonlinear Least Squares
NPMLE	Nonparametric Maximum Likelihood Estimator
OOP	Object-Oriented Programming
pdf	probability density function
pmf	probability mass function
Q-Q plot	Quantile-Quantile plot

RSS	Residual Sum of Squares
RHS	Right Hand Side (`formula` argument in R)
SEM	Stochastic-Expectation-Maximization
TSS	Total Sum of Squares
UML	Unified Modeling Language
WNLS	Weighted Nonlinear Least Squares
ZAMG	Zentralanstalt für Meteorologie und Geodynamik

Symbols

$\boldsymbol{\theta}$	natural parameter
$\boldsymbol{\mu}$	mean function
$\boldsymbol{y}$	response vector, dependent variable
$\boldsymbol{x}$	vector of explanatory (independent) variables
$\boldsymbol{X}$	design matrix
$\boldsymbol{\epsilon}$	error term vector
$\boldsymbol{\beta}$	regression coefficient vector
$\boldsymbol{\eta}$	linear predictor
ϕ	dispersion parameter
$\boldsymbol{\Psi}$	finite mixture model parameter vector
I	index set for sample
K	index set for mixture components
P	index set for regression coefficients
$\mathcal{Y}$	sample space
$\mathcal{Y}^c$	complete-data space
$\mathcal{Z}$	component label vector space
$\mathcal{B}$	regression coefficient vector space
$f^M(\cdot)$	mixture model probability density function
$h(\cdot)$	regression function
$f(\cdot)$	probability density function
$f^c(\cdot)$	complete-data probability density function
$\mathcal{L}(\cdot)$	likelihood function
$\ell(\cdot)$	log-likelihood function
$\ell^c(\cdot)$	complete-data log-likelihood function
$S(\cdot)$	score function
$\tilde{\ell}(\cdot)$	penalized likelihood function
$g(\cdot)$	link function
$\mathrm{V}(\cdot)$	variance function
$\mathrm{Cov}(\cdot)$	variance-covariance matrix
$\mathrm{I}(\cdot)$	observed information matrix

$\mathcal{I}(\cdot)$	Fisher (expected) information matrix
$\mathrm{Q}(\cdot)$	expectation function for complete-data log-likelihood
$\mathrm{H}(\cdot)$	expectation function for missing-data log-likelihood
$D(\cdot)$	deviance
X^2	Pearson statistic
$\mathrm{SE}(\cdot)$	Standard error
$\mathrm{CI}(\cdot)$	Confidence interval
$\mathrm{SD}(\cdot)$	Standard deviation
$\mathrm{BIAS}(\cdot)$	Bias
$\mathrm{ACI}(\cdot)$	asymptotic confidence interval
$\mathrm{ASE}(\cdot)$	asymptotic standard error
$\bar{\beta}$	Monte-Carlo mean (analogous for $\bar{\nu}$ ad $\bar{\pi}$)
$\mathbb{R}^{\text{temp}}$	real values within interval $[-10, 20]$
$G(\nu, \lambda)$	Gamma distribution
λ	rate parameter (Gamma distribution)
ν	shape parameter (Gamma distribution)
$\Gamma(\cdot)$	Gamma function
α	capital share in economic growth model
β	human capital share in economic growth model
g	technical advancement
δ	depreciation rate
A_0	country specific technological endowment
s_K	saving rate for physical capital
s_H	saving rate for human capital
λ^{nls}	convergence rate to steady state for nonlinear regression
λ^{mix}	convergence rate to steady state for mixture model

List of Figures

3.1 Unified Modeling Language (UML) class diagram for **flexmix** 54
3.2 UML class diagram for `"FLXM"` in **flexmixNL** 59

4.1 Simulated data ($n = 1000$ and $n = 500$) 69
4.2 Component specific regression functions (initial configuration) 72
4.3 Fitted two-component Gamma mixture model ($n = 1000$) 74
4.4 Rootogram for two-component Gamma mixture ($n = 1000$) 75
4.5 Component 1 78
4.6 Histograms and box-plots for fitted coefficients (component 1, $n = 1000$) . 79
4.7 Component 2 80
4.8 Histograms and box-plots for fitted coefficients (component 2, $n = 1000$) . 81
4.9 Histograms and box-plots for fitted shape parameter estimates ($n = 1000$) 82
4.10 Histograms and box-plots for fitted prior weight estimates ($n = 1000$) . . . 83
4.11 Component 1 87
4.12 Histogram and box-plots for fitted coefficients (component 1, $n = 500$) . . 88
4.13 Component 2 89
4.14 Histogram and box-plots for fitted coefficients (component 2, $n = 500$) . . 90
4.15 Histograms and box-plots for fitted shape parameter estimates ($n = 500$) . 91
4.16 Histograms and box-plots for fitted prior weight estimates ($n = 500$) . . . 92

5.1 Gas flow data 1 106
5.2 Q-Q plot of the residuals for the fitted nonlinear model 108
5.3 Fitted mean function for gas flow data 1 and models (5.1) and (5.2) . . . 110
5.4 Starting configuration for gas flow data 1 112
5.5 Fitted two-component mixtures of normal (left) and Gamma (right) for Model (5.1) 114
5.6 Rootogram for two-component normal mixtures for Model (5.1) 115
5.7 Rootogram for two-component Gamma mixtures for Model (5.1) 115
5.8 Fitted two-component mixtures of normal (left) and Gamma (right) for Model (5.2) 116
5.9 Rootogram for two-component normal mixture for Model (5.2) 117

5.10 Rootogram for two-component Gamma mixture for Model (5.2) 117
5.11 AIC,BIC and ICL for fitted models for gas flow data 1 119
5.12 Gas flow data 2 123
5.13 Gas flow data 2 with distinction working days (right) and holidays (left) . 124
5.14 Starting configuration for gas flow data 2 125
5.15 Fitted two-component Gamma mixture for Model (5.1) 127
5.16 Rootogram for two-component Gamma mixture for Model (5.1) 127
5.17 Fitted two-component Gamma mixture for Model (5.2) 128
5.18 Rootogram for two-component Gamma mixture and Model (5.2) 128
5.19 Fitted three-component Gamma mixture for Model (5.1) 131
5.20 Rootogram for three-component Gamma mixture and Model (5.1) 131
5.21 Fitted three-component Gamma mixture with Model (5.2) for working days (left) and holidays (right) 132
5.22 Rootogram for three-component Gamma mixture with Model (5.2) 132
5.23 AIC, BIC and ICL for fitted models for gas flow data 2 134
5.24 Gas flow data 3 136
5.25 Gas flow data 3 with distinction working days (right) and holidays (left) . 137
5.26 Starting configuration for gas flow data 3 138
5.27 Fitted two-component normal mixture for Model (5.1) 139
5.28 Rootogram for fitted two-component normal mixture for Model (5.1) . . . 139
5.29 Fitted two-component Gamma mixture for Model (5.1) 140

6.1 Cross-correlation for GDP growth and production factors 148
6.2 Density function of GDP growth 149
6.3 Rootogram for two-component normal mixture model 154
6.4 Fitted two-component mixture distribution for the GDP growth 155
6.5 Cross-correlation GDP growth with investment share 155
6.6 Cross-correlation GDP growth with country population growth 156
6.7 Cross-correlation GDP growth with education level 156

Listings

3.1 Fitting the Michaelis-Menten model with `nls()` . . . 50
3.2 Fitting the Michaelis-Menten model with `gnm()` . . . 50
3.3 Fitting the linearized Michaelis-Menten model with `glm()` . . . 51
3.4 Comparison of the estimates for the Michaelis-Menten model in R . . . 52
3.5 `"FLXcontrol"` default values . . . 55
3.6 M-step driver for mixtures of GLMs in **flexmix** (excerpt) . . . 56
3.7 Function call `flexmix()` for Gaussian mixtures of GNMs . . . 60
3.8 M-step driver for Gaussian mixtures of GNMs in **flexmixNL** (excerpt) . . . 60
3.9 Outsourced fitting function for Gaussian mixtures in **flexmixNL** . . . 61
3.10 Function call `flexmix()` for mixtures of GNMs with Gamma responses . . 61
3.11 Fitting function for Gamma mixtures in **flexmixNL** . . . 62
3.12 Outsourced fitting function for Gamma mixtures in **flexmixNL** . . . 62

4.1 Specification of mean function with **gnm** . . . 67
4.2 Function call for nonlinear Gamma mixture model in **flexmixNL** . . . 72
4.3 `flexmix()` output for nonlinear two-component Gamma mixture model . 73
4.4 Model selection criteria for exemplary fitting . . . 73
4.5 `summary()` output in **flexmix** for exemplary fitting . . . 74

5.1 Fitting command with `nls()` for Model (5.1) . . . 99
5.2 Fitting command with `nls()` for Model (5.2) . . . 100
5.3 Fitting command with `gnm()` for Model (5.1) . . . 100
5.4 Specification of Model (5.2) with **gnm** . . . 101
5.5 Fitting command with `gnm()` for Model (5.2) . . . 101
5.6 Fitting command for two-component normal mixtures with `flexmix()` . . 102
5.7 Fitting command for two-component Gamma mixtures with `flexmix()` . . 103
5.8 `summary()` output for two-component Gaussian mixtures for Model (5.1) . 116
5.9 `summary()` for two-component Gamma mixtures for Model (5.1) . . . 116
5.10 `summary()` for two-component Gaussian mixtures for Model (5.2) . . . 117
5.11 `summary()` for two-component Gamma mixtures for Model (5.2) . . . 117
5.12 `summary()` output for three-component mixture model and Model (5.1) . 132
5.13 `summary()` output for three-component mixture model and Model (5.2) . 132

6.1 Regression function for GDP growth model in R 150
6.2 Nonlinear regression output for GDP growth model 150
6.3 Fitting command and output for two-component normal mixture with Model (6.3) 152
6.4 `summary()` output for for two-component normal mixture and Model (6.3) 152

List of Tables

3.1 Arguments in `nls()` . . . 48
3.2 Arguments in `gnm()` . . . 49
3.3 Arguments for symbolic functions of class `"nonlin"` in `gnm()` . . . 49
3.4 Arguments in `"FLXM"` . . . 56

4.1 Initial configuration for two-component Gamma mixture model . . . 68
4.2 Fitted coefficients, standard errors and confidence intervals ($n = 1000$) . . 76
4.3 MC results, standard errors and coverage rates (component 1, $n = 1000$) . 85
4.4 MC results, standard errors and coverage rates (component 2, $n = 1000$) . 86
4.5 MC results, standard errors and coverage rates (component 1, $n = 500$) . . 93
4.6 MC results, standard errors and coverage rates (component 2, $n = 500$) . . 94

5.1 Parameters of interest for two-component mixture models . . . 102
5.2 Parameters of interest for three-component mixture models . . . 104
5.3 Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 1 (nonlinear regression) . . . 110
5.4 Ranges for starting values for gas flow data 1 . . . 111
5.5 Simulation results for two-component mixtures with Model (5.1) . . . 113
5.6 Simulation results for two-component mixtures with Model (5.2) . . . 113
5.7 Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 1 (two components) . . . 118
5.8 Prediction of gas flow for design temperatures for gas flow data 1 . . . 121
5.9 Ranges for starting values for gas flow data 2 . . . 125
5.10 Simulation results for two-component Gamma mixtures with Models (5.1) and (5.2) . . . 126
5.11 Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 2 (two components) . . . 129
5.12 Simulation results for three-component Gamma mixtures with Models (5.1) and (5.2) . . . 130
5.13 Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 2 (three components) . . . 133

5.14 Prediction of gas flow for design temperatures for gas flow data 2 135
5.15 Ranges for starting values for gas flow data 3 137
5.16 Simulation results for two-component mixtures with Model (5.1) 139
5.17 Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 3 (two components) 141
5.18 Prediction of gas flow for design temperatures for gas flow data 3 142

6.1 Cross-country regression variables for time period 1960 - 1985 147
6.2 Ranges for starting values 150
6.3 Regression coefficients for economic growth model (6.3) 154
6.4 Convergence rates towards steady state (component 1) 157
6.5 Convergence rates towards steady state (component 2) 158

Introduction

Finite Mixture Models (FMMs)

Finite mixture models represent a suitable method to deal with heterogeneity in data. They allow to find subgroups within a data set without a definite knowledge on the membership of the groups. The underlying probabilistic clustering methodology allows to detect distinct components along similar distributional shapes. This procedure enables to provide a detailed statistical analysis on identified subgroups and to derive conclusions on the overall population. Mixture models are being applied in different fields of research as they cover a broad range of different models. They comprise complex distributional shapes allowing for different underlying distributions. Advanced mixture models cover mixtures of regression models as, for example, the Generalized Linear Model (GLM). While this model class has been widely studied and is constantly being extended, nonlinear regression was not emphasized within mixture models in the past. As practical applications buttress the use of nonlinear regression functions, the present work introduces the new model class of mixtures of Generalized Nonlinear Models (GNMs).

Generalized Nonlinear Models (GNMs)

A Generalized Nonlinear Model (GNM) comprises nonlinear regression in a flexible way by embedding the classical nonlinear regression model within the exponential family distribution. Nonlinear regression is strongly influenced by a specific problem which impacts the parametrization of the regression function and the determination of the underlying statistical distribution. The regression function has in general no standardized form. It consists of arbitrary functional elements where the coefficients are linked to predictors through a nonlinear functional relationship. Despite the variety of functional forms, specific functions emerged as typical forms representing, for example, (biological) growth processes. The classical nonlinear regression model is based on the normal distribution. Due to the characteristics of specific applications, an extension to other distributions seems obvious. Therefore, an extension for the broad range of exponential family distributions can be adopted which allows to apply well-known results derived

from Generalized Linear Models. Fitting nonlinear regression functions depends in general on numerical procedures where complex functional structures may provide numerical problems. A crucial difficulty in the fitting process arises through the appropriate choice of starting values which remains a necessary requirement within the framework of mixture models for GNMs.

Implementation in R

The implementation of the new model class is provided within the statistical computing environment R. The technical implementation of mixtures of GNMs builds on the efficient package **flexmix** which was developed by Friedrich Leisch and Bettina Grün. The package allows for possible extensions due to its modular framework. It covers a broad repertoire on various models comprising the GLM, where mixtures of GNMs represent a suitable extension. The implementation of mixtures of GNMs constitutes a key result within the present work. Therefore, the procedure of extending **flexmix** is outlined in detail. The implementation of the new model class is wrapped up in the new package **flexmixNL**.

Structure of the present work

The objective of the present work is to give a coherent view on mixtures of GNMs by presenting their construction in consecutive steps. Building on this, the theoretical background is technically implemented in R and subsequently applied to synthetic and real data. The following paragraphs give a general overview on the content of the subsequent chapters.

Chapter 1 outlines the construction and components of GNMs as an extension of the well-known nonlinear regression model for the normal distribution. The exponential family serves as the main distributional framework where special attention is given to the mean and variance. While the modeling of the mean follows a nonlinear regression function, the computation of the variance requires a proper estimation of the dispersion parameter. Key results on the specifics of GNMs serve as basis for the subsequent chapters.

Chapter 2 presents the construction of mixture models with GNMs as basic model class. It outlines furthermore technical difficulties arising from the nonlinear functional structure within the mean function. Due to the complexity of their nature, mixture models comprise a large amount of unknown parameters. The chapter presents a general derivation of the necessary parameters in order to fit a mixture of GNMs. Special focus is given to the EM algorithm as the underlying fitting procedure. As the adequate choice of the number of components remains a problem specific task, the chapter concludes with appropriate model selection criteria.

Chapter 3 presents the technical framework in R as a special focus of the present work. A main part of the thesis consists of the derivation of a fitting procedure for the new

mixture model class. The package **flexmix** serves as the main base for the technical implementation and incorporates an efficient EM algorithm for mixture models. The chapter introduces an appropriate extension for mixtures of GNMs resulting in the new package **flexmixNL**. The derivation of the new fitting approach is outlined in detail and enables the fitting of arbitrary nonlinear regression functions. The current implementation allows for the normal and Gamma distribution and can be extended to further members of the exponential family. The chapter presents furthermore two approaches for the calculation of standard errors for the derived parameter estimates.

In order to underpin the statistical reliance of the derived results and the efficiency of the new fitting approach, a Monte Carlo (MC) simulation study is performed. The detailed procedure and the results are summarized in Chapter 4. In accordance to the specific sigmoid pattern of nonlinear data structures, a synthetic data set with a typical decreasing structure is chosen for the fitting. The simulation study is performed under a Gamma distribution following the idea of fitting maximum values. Chapter 4 gives a detailed overview on the setup and framework of the performed simulation study. It highlights furthermore the limitations of the algorithm which occur due to numerical obstacles. Main results outline the quality of the MC estimates for two different sample sizes.

Chapter 5 discusses applications of the new methodology for mixtures of GNMs to the specific problem of gas flow modeling. Within the present thesis, the generalization of the new mixture model class is provided for arbitrary nonlinear regression functions within the framework of exponential family distributions. In this context, the mixing of typical sigmoid gas flow curves is enabled. In order to assess the performance and reliability of the new package, the implemented methods are applied to typical data exhibiting a sigmoid decreasing pattern, where the new models succeed to identify definite subgroups even for dense data structures.

Chapter 6 outlines the application of mixtures of GNMs to economic growth models. The economic growth model arises from a nonlinear dependency structure between economic factors motivating the use of nonlinear regression models. Within this aspect, mixtures of GNMs face the occurring heterogeneity of economies by modeling the economic state for different groups of countries.

CHAPTER 1

Generalized Nonlinear Models

Introduction

The object of this chapter is to introduce a special class of statistical models linking the well-known exponential family distributions to the group of nonlinear regression models. Nonlinear regression models have been widely explored in Bates and Watts (1988) and Seber and Wild (2003). Their work represents the main literature on this topic. Classical nonlinear regression analysis, as analyzed by Bates and Watts (1988) and Seber and Wild (2003), builds on normally distributed and homoscedastic error terms. Wei (1998) extends the classical nonlinear regression models by embedding them into the framework of the linear exponential family similar to Generalized Linear Models (GLMs). By addressing conceptual similarities to GLMs, well-known results from McCullagh and Nelder (1989) can be adopted to the new model class. The resulting model class is denoted as Generalized Nonlinear Models (GNMs) or Exponential Family Nonlinear Models (EFNMs) in literature and will provide a basis for further models in the remaining work. Main reference is made to Wei (1998) representing the primary literature concerning GNMs within this work.

The classical nonlinear regression model builds a key part of the GNM and will be discussed in Section 1.1. The distributional framework is given by the linear exponential family in Section 1.2 which gives also an overview on general aspects regarding the Maximum Likelihood (ML) estimation. The estimation of the dispersion parameter is outlined in Section 1.3 and completes the specification of the GNM. The final Section 1.4 concludes the chapter with the summarized model specifications of GNMs and reflects key assumptions for the subsequent analysis within the remaining work.

1.1 Nonlinear Regression Models

While linear regression models comprise either constants and possibly products of regression coefficients with explanatory variables, nonlinear regression models arise with at least one nonlinear relationship between a regression coefficient and a predictor. The non-linearity is given particularly in the regression coefficient leading to a nonlinear combination of predictors. Typical examples for nonlinear terms build on exponential or power functions. The fitting of nonlinear regression models is strongly related to iterative optimization techniques. The wide area of applications yields a diverse range of possible nonlinear regression functions building on various relationship structures. Common functions are also summarized in Bates and Watts (1988, p. 329). All applications share the property of a known functional relationship given by the nonlinear regression function and the unknown regression coefficients as parameters of interest. The problem specific and diverse nonlinear structure of the regression functions prohibits the derivation of an unified fitting procedure in order to obtain the regression coefficients. The underlying iterative optimization methods are moreover problem specific and rely often on modified Newton-Raphson and Gauss-Newton procedures. Disadvantages occur concerning the convergence of the numerical procedures to desired solutions which is not easy to establish.

Bates and Watts (1988, p. 33) define a nonlinear regression model as

$$\mathrm{E}[y_i] = h(\boldsymbol{x}_i, \boldsymbol{\beta}), \quad i = 1, \ldots, n, \ \boldsymbol{x}_i \in \mathbb{R}^m, \ \boldsymbol{\beta} \in \mathbb{R}^P, \tag{1.1}$$

where the additive error terms follow a normal distribution with zero mean and constant variance. The vectors $\boldsymbol{x}_i$ comprise explanatory variables and are considered as fixed. The function $h(\boldsymbol{x}_i, \boldsymbol{\beta})$ represents a nonlinear regression function and is in general entirely known except for the regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_P)^\top$. Graphical illustrations of a given data set and prior knowledge facilitate the choice and parametrization of $h(\boldsymbol{x}_i, \boldsymbol{\beta})$. As the use of the phrase *nonlinear* has been extended to topics beyond regression functions in different literature, the following definition states its general use in this work, see also Bates and Watts (1988, p. 32).

Definition 1.1 *Nonlinear Function*
The function $h(\boldsymbol{x}_i, \boldsymbol{\beta})$ is nonlinear in the regression parameter vector $\boldsymbol{\beta}$ if at least one partial derivative $\frac{\partial h}{\partial \beta_p}$ depends on β_p for $p = 1, \ldots, P$.

Unlike in linear regression models the dimensions of the vectors of explanatory variables $\boldsymbol{x}_i$ and those of the regression coefficient vectors $\boldsymbol{\beta}$ do not necessarily coincide. The regression coefficients and their meaning depend on the functional structure of the relationship between responses and explanatory variables. The remaining task is given by the appropriate and precise estimation of the regression coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^P$. The usual fitting method used for nonlinear regression models is given by Nonlinear Least Squares (NLS) based on minimizing the Residual Sum of Squares (RSS)

$$\mathrm{RSS}(\boldsymbol{\beta}) = \|\boldsymbol{y} - h(\boldsymbol{x}, \boldsymbol{\beta})\|^2. \tag{1.2}$$

The RSS is minimized for the regression parameter estimator $\hat{\boldsymbol{\beta}}$, i.e.

$$\hat{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^P} \mathrm{RSS}(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^P} \|\boldsymbol{y} - h(\boldsymbol{x}, \boldsymbol{\beta})\|^2.$$

The appearance of non-constant variances for the observations y_i can be handled by linear transformations or the use of Weighted Nonlinear Least Squares (WNLS) methods. Transforming nonlinear regression models with mean function (1.1) can eliminate the heteroscedasticity by variance-stabilization or even establish a linear relationship between the transformed variables. Checking whether the given nonlinear regression model in Equation (1.1) can be transformed into a linear one is important as linear regression models can be analytically solved and are easier to apply. The following well-known model exemplifies such a transformation procedure.

Example 1.1 *Michaelis-Menten Model, introduced by Bates and Watts (1988, p. 33)*
The Michaelis-Menten model describes the mean rate of enzymatic reactions y_i as a nonlinear regression model depending on the concentration x_i of a substrate given by the equation

$$\mathrm{E}[y_i] = h(x_i, \boldsymbol{\beta}) = \frac{\beta_0 x_i}{\beta_1 + x_i}, \quad i = 1, \ldots, n, \; x_i \in \mathbb{R}. \tag{1.3}$$

The regression coefficients consist of $\boldsymbol{\beta} = (\beta_0, \beta_1)$. The numerator in (1.3) represents the maximum achievable reaction rate as upper asymptote. The regression coefficient β_1 denotes the specific concentration in substrate indicating half of the maximum achievable reaction rate. The original nonlinear regression function (1.3) can be modified to

$$\mathrm{E}[y_i] = \frac{1}{\frac{1}{\beta_0} + \frac{\beta_1}{\beta_0}\frac{1}{x_i}} = \frac{1}{\tilde{\beta}_0 + \tilde{\beta}_1 \frac{1}{x_i}}$$

abrogating the nonlinear model as the reciprocal mean function can be expressed as linear regression model in the predictor $\tilde{x}_i = \frac{1}{x_i}$ given by

$$g(\mu_i) = \frac{1}{\mathrm{E}[y_i]} = \tilde{\beta}_0 + \tilde{\beta}_1 \tilde{x}_i. \tag{1.4}$$

Equation (1.4) demonstrates the transformation of the original nonlinear Michealis-Menten model into a linear regression model. The reciprocal mean reaction rate depends on the reciprocal concentration in substrate $\tilde{x}_i$ but is linear in the regression coefficients $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_0, \tilde{\beta}_1)^\top$.

As the previous example clearly demonstrates, a transformation of the nonlinear regression problem involves a change in the regression parameters as well. The original model assigns often a specific meaning to the regression parameters with a certain scientific or physical interpretation. A transformation abrogates these meanings and exacerbates the final interpretation of the estimators which may be of less interest compared to the original ones. For example, the original regression coefficient β_1 relates to the concentration in substrate indicating half of the maximum achievable reaction rate which is again characterized by the regression coefficient β_0. The transformed regression coefficients $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_0, \tilde{\beta}_1)^\top$ do not allow any physical interpretation. A possible solution is to re-transform the coefficients $\tilde{\boldsymbol{\beta}}$, which can be obtained by estimation methods for linear

regression models, and to interpret the original, or respectively re-transformed, coefficients. Particular caution is required concerning the error structure within such transformations. Bates and Watts (1988, p. 25) postulate the assumption of normally distributed responses y_i enabling the use of Least Squares (LS) methods which are known for their tractability and mathematical advantages. According to the preconditions for the LS method, efficient estimators can be derived based on homoscedastic responses resulting from a constant variance for all y_i. While transformations of nonlinear regression models result in the required linear functional form they may destruct the presumed error structures. Conversely, a linearization of the nonlinear regression function may stabilize the error variance but can possibly lead to undesirable dependency structures as Seber and Wild (2003) explain. The difficulties encourage the use of efficient numerical approaches in order to fit the original nonlinear regression problem in Equation (1.1). As Seber and Wild (2003, p. 5) stress, nonlinear regression models sometimes lead to better interpretable regression parameters while they definitively complicate the calculus in the background of the parameter estimation. Therefore it is necessary to provide numerical techniques for the derivation of an accurate regression parameter estimator $\hat{\boldsymbol{\beta}}$. Potential solutions and fitting techniques depend to a great extend on the specific parametrization of Function (1.1). It is conceivable that complicated functions comprising several regression coefficients in a nonlinear term afford a higher level in fitting effort and analysis than a regression function with one nonlinear regression coefficient. Use can be made of occurring linear terms in Equation (1.1) which are denoted as *conditionally linear parameters* in Bates and Watts (1988, p. 36) as they can be estimated by linear regression conditional on the remaining nonlinear regression coefficients. These considerations reflect the variety of potential approaches for obtaining solutions of nonlinear regression models which can take various forms due to theoretical considerations regarding the specific problem and the given data set.

1.2 Linear Exponential Family

According to Casella and Berger (2001, p. 111), the probability distribution for a (continuous or discrete) random variable $y \in \mathbb{R}$ belongs to the exponential family if its probability density function (pdf) or probability mass function (pmf) can be expressed as

$$f(y;\boldsymbol{\theta}) = h(y)c(\boldsymbol{\theta}) \exp\left(\sum_{i=1}^{s} w_i(\boldsymbol{\theta})t_i(y)\right). \tag{1.5}$$

Expression (1.5) consists of the function $h(y) \geq 0$ and real-valued functions $t_1(y), \ldots, t_s(y)$ in y. The remaining functions, denoted as $w_1(\boldsymbol{\theta}), \ldots, w_s(\boldsymbol{\theta})$, are real-valued functions depending exclusively on the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$. Exponential families are widely used in statistics due to advantageous statistical and mathematical properties as Casella and Berger (2001, p. 112) explain. The exponential family comprises a big number of distributions (continuous and discrete). The following theorem specifies the first two moments of this family.

Theorem 1.1 *Casella and Berger (2001, Theorem 3.4.2)*
For any random variable y with pdf or pmf as in Equation (1.5) the following holds for $j = 1, \ldots, d$,

$$\begin{aligned} \mathrm{E}\left[\sum_{i=1}^{s} \frac{\partial w_i(\boldsymbol{\theta})}{\partial \theta_j} t_i(y)\right] &= -\frac{\partial}{\partial \theta_j} \log c(\boldsymbol{\theta}), \\ \mathrm{Var}\left(\sum_{i=1}^{s} \frac{\partial w_i(\boldsymbol{\theta})}{\partial \theta_j} t_i(y)\right) &= -\frac{\partial^2}{\partial \theta_j^2} \log c(\boldsymbol{\theta}) - \mathrm{E}\left[\sum_{i=1}^{s} \frac{\partial^2 w_i(\boldsymbol{\theta})}{\partial \theta_j^2} t_i(y)\right]. \end{aligned}$$

With reference to the number of terms s, Casella and Berger (2001) distinguish between two different groups of exponential families. In the case of $d < s$ with d denoting the dimension of the natural parameter space Θ, the resulting exponential family is called a *curved exponential family*. For $s = d$, the resulting family is denoted as a *full exponential family*. It is on the other side impossible for the number of terms s to fall below the number of parameters d. s most often reduces to one term. The specific case of exponential families with $s = 1$ is also known as *linear exponential family* and will be addressed in more detail in the present work. Exponential family distributions with one term, respectively $s = 1$, allow a modified representation of their pdf or pmf compared to the given form in Equation (1.5) where a random variable y follows a distribution specified by the pdf expressed as

$$f(y; \theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right). \tag{1.6}$$

$b(\cdot)$ and $c(\cdot, \cdot)$ are specified functions and θ represents the natural parameter with $\theta \in \Theta \subseteq \mathbb{R}$. The dispersion parameter ϕ takes values within the subset $\Phi \subseteq \mathbb{R}$ and is specified by the dispersion function $a(\phi)$, occurring in (1.6), which is often simplified to

$$a(\phi) = a \cdot \phi. \tag{1.7}$$

It is common practice in literature to set the dispersion parameter equal to $\phi = \sigma^2$ which is treated as nuisance parameter. The family of distributions with pdf as specified in (1.6) comprising the previously discussed characteristics is called the *linear exponential family*. The dispersion parameter ϕ is assumed to be known in (1.6). Wei (1998, p. 2) suggests the phrase exponential family for distributions with pdf (1.6) regardless of ϕ being known or unknown in accordance with general practice in literature.

The specification of the dispersion function in (1.7) enables the use of varying weights for different observations. Expressing the weight parameter through $a = w^{-1}$ leads to the particular specification of the dispersion function given by

$$a(\phi) = \frac{\phi}{w}. \tag{1.8}$$

McCullagh and Nelder (1989, p. 29) postulate that the weight parameters w in (1.8) require already available information in terms of known *prior weights*. Therefore the main focus within computations reduces to the analysis of the global dispersion parameter ϕ. Wei (1998, p. 12) introduces the notation *weighted exponential family* for distributions

with dispersion function (1.8) and pdf

$$f(y;\theta,\phi) = \exp\left(w\frac{y\theta - b(\theta)}{\phi} + c(y,\phi/w)\right). \tag{1.9}$$

The weighted exponential family with specific weights for different observations denotes a special and not very common case within the exponential family. Nevertheless, it will be of high importance for further analysis in this thesis.

1.2.1 Members of the Exponential Family

The linear exponential family represents a commonly used group of distributions in statistics. As previously mentioned, pdfs of the linear exponential family take the form (1.6). The following examples show well-known members of this class of distributions by transforming the specific pdfs to the mathematical form of (1.6).

Example 1.2 (Normal Distribution)
A random variable $y \sim N(\mu,\sigma^2)$ has the pdf $f(y;\mu,\sigma^2)$ given by

$$\begin{aligned} f(y;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \\ &= \exp\left(\frac{y\mu - \mu^2/2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right). \end{aligned}$$

For $\theta = \mu$ and $\phi = \sigma^2$ the resulting exponential family has the components

$$a(\phi) = \phi, \;\; b(\theta) = \frac{\theta^2}{2} \quad \textit{and} \quad c(y,\phi) = -\frac{y^2}{2\phi} - \frac{1}{2}\log(2\pi\sigma^2).$$

Example 1.3 (Gamma Distribution)
A random variable $y \sim G(a,\lambda)$ has the pdf $f(y;a,\lambda)$ given by

$$f(y;a,\lambda) = \exp\{-\lambda y\}\lambda^a y^{a-1}\frac{1}{\Gamma(a)}.$$

After re-parametrization of the mean $\mu = \nu/\lambda$ and shape parameter $\nu = a$ the pdf takes the form

$$\begin{aligned} f(y;\mu,\nu) &= \exp\left(-\frac{\nu}{\mu}y\right)\left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1}\frac{1}{\Gamma(\nu)} \\ &= \exp\left(-\frac{\nu}{\mu}y + \nu\log(\nu) - \nu\log(\mu) + (\nu-1)\log(y) - \log\Gamma(\nu)\right) \end{aligned}$$

with $\mu,\nu,y > 0$. For $\theta = -1/\mu$ and $\phi = 1/\nu$ the resulting exponential family has the components

$$a(\phi) = \phi, \;\; b(\theta) = -\log(-\theta) \quad \textit{and} \quad c(y,\phi) = \frac{1}{\phi}\log\frac{1}{\phi} + \left(\frac{1}{\phi} - 1\right)\log y - \log\Gamma\left(\frac{1}{\phi}\right).$$

Example 1.4 (Poisson Distribution)
A random variable $y \sim P(\mu)$ has the pmf $f(y;\mu)$ given by

$$f(y;\mu) = \frac{\mu^y}{y!}e^{-\mu} = \exp\left(y\log\mu - \mu - \log y!\right).$$

For $\theta = \log\mu$ and $\phi = 1$ the resulting exponential family has the components

$$a(\phi) = \phi, \quad b(\theta) = \exp(\theta) \quad \textit{and} \quad c(y,\phi) = -\log y!.$$

Example 1.5 (Binomial Distribution)
A random variable $my \sim Bin(m,\pi)$ has the pmf $f(y;m,\pi)$ given by

$$\begin{aligned} f(y;m,\pi) &= \binom{m}{my}\pi^{my}(1-\pi)^{m-my} \\ &= \exp\left(\log\binom{m}{my} + my\log\pi + m(1-y)\log(1-\pi)\right) \\ &= \exp\left(\frac{y\log\frac{\pi}{1-\pi} - \log\frac{1}{1-\pi}}{1/m} + \log\binom{m}{my}\right), \quad y = 0, \frac{1}{m}, \frac{2}{m}, \ldots, 1. \end{aligned}$$

For $\theta = \log\frac{\pi}{1-\pi}$ and $\phi = 1$ the resulting exponential family has the components

$$a(\phi) = a\cdot\phi = \frac{1}{m}\phi, \quad b(\theta) = \log\frac{1}{1-\pi} = \log(1+\exp\theta) \quad \textit{and} \quad c(y,\phi) = \log\binom{\frac{1}{\phi}}{\frac{y}{\phi}}.$$

Further examples can be found in McCullagh and Nelder (1989).

The linear exponential family has advantageous technical properties which can be applied for the random variable y. A specific representation of the mean and variance of y is given with the following derivation. As Casella and Berger (2001, pp. 335-339) outline, the following equations hold for the linear exponential family,

$$\mathrm{E}\left[\frac{\partial\log f(y;\theta,\phi)}{\partial\theta}\right] = 0, \tag{1.10}$$

$$\mathrm{Var}\left(\frac{\partial\log f(y;\theta,\phi)}{\partial\theta}\right) = \mathrm{E}\left[\frac{\partial\log f(y;\theta,\phi)}{\partial\theta}\right]^2 = \mathrm{E}\left[\frac{-\partial^2\log f(y;\theta,\phi)}{\partial\theta^2}\right]. \tag{1.11}$$

The expected value of the derivative of (1.6) with respect to the natural parameter θ, using the simplification $a(\phi) = \phi$, results in the expression

$$\mathrm{E}\left[\frac{\partial\log f(y;\theta)}{\partial\theta}\right] \overset{(1.6)}{=} \mathrm{E}\left[\frac{y-b'(\theta)}{\phi}\right] = \frac{1}{\phi}\mathrm{E}[y-b'(\theta)] \overset{!}{=} 0$$

which can be further advanced to the general expression for the expected value (also *mean*) of y given by

$$\mathrm{E}[y] = b'(\theta) = \mu. \tag{1.12}$$

Equation (1.12) points out the functional relationship between the mean μ and natural parameter. The terms b' and b'' refer to the first and second order derivatives of $b(\cdot)$ with

respect to θ, respectively

$$b'(\theta) := \frac{\partial b(\theta)}{\partial \theta}, \qquad b''(\theta) := \frac{\partial^2 b(\theta)}{\partial \theta^2}.$$

The variance Equation (1.11) can be rearranged and advanced using (1.6) to

$$\mathrm{E}\left[\frac{\partial \log f(y;\theta)}{\partial \theta}\right]^2 + \mathrm{E}\left[\frac{\partial^2 \log f(y;\theta)}{\partial \theta^2}\right] \overset{(1.6)}{=} \frac{\mathrm{Var}(y)}{\phi^2} + \frac{-b''(\theta)}{\phi} \overset{!}{=} 0$$

enabling the following representation of the variance of y

$$\mathrm{Var}(y) = \phi \cdot b''(\theta) = \phi \cdot \mathrm{V}(\mu). \tag{1.13}$$

Equation (1.13) outlines the decomposition of the variance of y into the dispersion parameter ϕ and the variance function $\mathrm{V}(\mu)$, which relates the variance of y to its mean μ. Typical variance functions are given in Example 1.6.

Example 1.6 (Variance Function)

1. *For any random variable $y \sim N(\mu, \sigma^2)$ the variance function equals* 1.
2. *For any random variable $y \sim G(a, \lambda)$ with mean $\mu = \nu/\lambda$ and shape parameter $\nu = a$ the variance function is given by $\mathrm{V}(\mu) = \mu^2$.*
3. *For any random variable $y \sim P(\mu)$ with mean μ the variance function is given by $\mathrm{V}(\mu) = \mu$.*
4. *For any random variable $my \sim Bin(m, \pi)$ with mean μ the variance function is given by $\mathrm{V}(\mu) = \frac{\mu \cdot (1-\mu)}{m}$.*

For further examples reference is made to McCullagh and Nelder (1989).

1.2.2 Maximum Likelihood Estimation

This section outlines the general concept of the Maximum Likelihood (ML) estimation for the linear exponential family and presents its key results. The work of Casella and Berger (2001) serves as the main reference for deeper involvement for ML estimation.

Considering a vector $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ where the n elements y_i follow an exponential family with $y_i \overset{iid}{\sim} f(y_i; \theta_i, \phi)$ and natural parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$, then $\boldsymbol{y}$ follows the joint pdf expressed as

$$\begin{aligned} f(\boldsymbol{y}; \boldsymbol{\theta}, \phi) &= \prod_{i=1}^{n} f(y_i; \theta_i, \phi) \\ &\overset{(1.6)}{=} \prod_{i=1}^{n} \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right) \\ &= \exp\left(\sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right) \end{aligned}$$

$$= \quad \exp\left(\frac{\boldsymbol{y}^\top \boldsymbol{\theta} - b^*(\boldsymbol{\theta})}{\phi} + c^*(\boldsymbol{y}, \phi)\right) \tag{1.14}$$

with functional forms

$$b^*(\boldsymbol{\theta}) := \sum_{i=1}^{n} b(\theta_i) \quad \text{and} \quad c^*(\boldsymbol{y}, \phi) := \sum_{i=1}^{n} c(y_i, \phi). \tag{1.15}$$

As (1.14) clearly shows, the joint pdf of $\boldsymbol{y}$ takes again the general form of a linear exponential family. Therefore the canonical parameter vector $\boldsymbol{\theta}$ and the dispersion parameter ϕ specify the distributional parameters. The dispersion parameter ϕ is in general treated as a known scaling parameter in order to derive an estimator for $\boldsymbol{\theta}$. For this reason the following proceeding will focus on the estimation of $\boldsymbol{\theta}$, whereas adequate computation methods for ϕ will be discussed in the subsequent Section 1.3. The likelihood function will be denoted as a function in the unknown distributional parameter vector $\boldsymbol{\theta}$ for a sample $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^n$ as

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi) = f(\boldsymbol{y}; \boldsymbol{\theta}, \phi) = \prod_{i=1}^{n} f(y_i; \theta_i, \phi).$$

$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$ is a function in the distributional parameter $\boldsymbol{\theta}$ corresponding in appearance to the joint pdf $f(\boldsymbol{y}; \boldsymbol{\theta}, \phi)$ in (1.14). The underlying concept of ML estimation aims to choose those values for the model parameters for which the likelihood function attains its maximum.

Definition 1.2 *Maximum Likelihood Estimator (MLE), Casella and Berger (2001, p. 316) The MLE of the parameter $\boldsymbol{\theta}$ based on a sample $\boldsymbol{y}$ is given by $\hat{\boldsymbol{\theta}}$ and corresponds to the parameter value where $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$ attains its maximum.*

Hence, searching for the MLE affords maximizing the likelihood function $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$ with respect to $\boldsymbol{\theta}$. It is common practice to carry the maximization problem over to the logarithmic likelihood function $\ell(\boldsymbol{\theta}; \boldsymbol{y}, \phi) = \log \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$ due to technical advantages within the computation. The strictly monotone behavior of the logarithm on $(0, \infty)$ ensures coinciding maxima of $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$ and $\ell(\boldsymbol{\theta}; \boldsymbol{y}, \phi)$. The log-likelihood function is given by

$$\begin{aligned} \ell(\boldsymbol{\theta}; \boldsymbol{y}, \phi) &= \log \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \phi) = \log \prod_{i=1}^{n} f(y_i; \theta_i, \phi) \\ &= \sum_{i=1}^{n} \log f(y_i; \theta_i, \phi) \overset{(1.6)}{=} \sum_{i=1}^{n} \left(\frac{y_i \theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right). \end{aligned}$$

According to this usual practice, the analysis within this work focuses mainly on the log-likelihood functions of the underlying statistical models. In order to obtain possible maxima of $\ell(\cdot)$, the gradient of the log-likelihood with respect to $\boldsymbol{\theta}$, also denoted as *score function* $S(\boldsymbol{y}; \boldsymbol{\theta}, \phi)$, is analyzed. In order to obtain a maximum, first order conditions imply the necessary condition

$$S(\boldsymbol{y}; \hat{\boldsymbol{\theta}}, \phi) \overset{!}{=} \mathbf{0}. \tag{1.16}$$

Casella and Berger (2001, p. 316) outline the difficulty of finding and verifying a global maximum as a drawback of the ML estimation. Further problems arise if the maximum is located on the boundary of the domain of the parameter space Θ which affords separate analysis. For complex models, Equation (1.16) obviates analytical solutions and appropriate numerical approaches have to be found, which requires further analysis on whether local or global maxima were derived.

The MLE $\hat{\boldsymbol{\theta}}$ is asymptotically normally distributed with $\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}^*, \mathcal{I}(\boldsymbol{\theta}^*)^{-1})$, see Casella and Berger (2001, p. 472), with the asymptotic variance given by the inverse of the *expected information* $\mathcal{I}(\boldsymbol{\theta}^*)$ (also *Fisher information*). Therefore the Fisher information can be used as an approach to compute the variance-covariance matrix of the derived MLE $\hat{\boldsymbol{\theta}}$. According to Casella and Berger (2001, p. 338), the *Fisher information* for linear exponential families can be expressed as

$$\mathcal{I}(\boldsymbol{\theta}) = -\mathrm{E}\left[\frac{\partial^2 \ell(\boldsymbol{\theta}; \boldsymbol{y}, \phi)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}\right] \overset{(1.6)}{=} \mathrm{E}\left[S(\boldsymbol{y}; \boldsymbol{\theta}, \phi) S^\top(\boldsymbol{y}; \boldsymbol{\theta}, \phi)\right]. \tag{1.17}$$

1.3 Dispersion Parameter

Section 1.2.2 outlines the ML estimation as an appropriate method in order to obtain the estimate $\hat{\boldsymbol{\theta}}$ for a fixed dispersion parameter ϕ. As the variance of the response y depends also on the dispersion parameter in (1.13), its estimation is crucial for the complete specification of GNMs. The dispersion parameter ϕ can be derived in a sequential step to the fitting of $\hat{\boldsymbol{\theta}}$. This section summarizes two approaches in order to obtain estimates for the dispersion parameter $\hat{\phi}$.

1.3.1 Deviance

The deviance represents a measure for the goodness-of-fit of statistical models. It basically compares the fitted model to the so-called *full model* as McCullagh and Nelder (1989, p. 33) outline. The latter matches the given data $\boldsymbol{y}$ exactly through n parameters so that it yields the maximum likelihood achievable. The deviance will be denoted in dependence of $\boldsymbol{\mu}$ instead of $\boldsymbol{\theta}$ which is common practice as McCullagh and Nelder (1989, p. 33) point out. Therefore, $\hat{\boldsymbol{\mu}}$ denotes the MLE for the unknown mean vector $\boldsymbol{\mu}$. The deviance is constructed as twice the difference between the log-likelihood functions of both models and takes the following form for members of the exponential family

$$\begin{aligned} D(\boldsymbol{y}, \boldsymbol{\mu})/\phi &= 2\left(\ell(\boldsymbol{\theta}, \phi; \boldsymbol{y})_{\boldsymbol{\mu}=\boldsymbol{y}} - \ell(\boldsymbol{\theta}, \phi; \boldsymbol{y})_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}}\right) \\ &= 2\sum_{i=1}^{n}\left(\{y_i\theta_i - b(\theta_i)\}_{\boldsymbol{\mu}=\boldsymbol{y}} - \{y_i\theta_i - b(\theta_i)\}_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}}\right)/\phi. \end{aligned}$$

For the specific case of weighted exponential family distributions with pdf (1.9) and dispersion parameter (1.8), McCullagh and Nelder (1989, p. 33) specify the deviance function as

$$D_w(\boldsymbol{y}; \boldsymbol{\mu}) = 2\sum_{i=1}^{n} w_i\left(\{y_i\theta_i - b(\theta_i)\}_{\boldsymbol{\mu}=\boldsymbol{y}} - \{y_i\theta_i - b(\theta_i)\}_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}}\right). \tag{1.18}$$

The deviance $D(\cdot)$ is a function of the data only, while the scaled deviance $D^*(\cdot)$ represents a modified measure scaled by the dispersion parameter, as McCullagh and Nelder (1989, p. 34) emphasize, respectively

$$D^*(\boldsymbol{y}, \boldsymbol{\mu}) := D(\boldsymbol{y}, \boldsymbol{\mu})/\phi.$$

An adequate estimator for the dispersion parameter is given by

$$\hat{\phi} = \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\mu}})}{n - P},$$

where P denotes the number of unknown parameters corresponding to the specifications of nonlinear regression models in Section 1.1.

The following list summarizes the deviance functions for the exemplary members of exponential family distributions listed in Section 1.2.1.

Example 1.7 (Deviance)
Considering a statistical model with $\mathrm{E}[y_i] = \mu_i$*, the following distributional properties hold:*

1. *For a sample where* $y_i \sim N(\mu_i, \sigma^2)$ *the deviance function is given by*

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = \sum_{i=1}^{n} (y_i - \mu_i)^2.$$

2. *For a sample where* $y_i \sim G(a, \lambda_i)$ *with mean* $\mu_i = \nu/\lambda_i$ *and shape parameter* $\nu = a$ *the deviance takes the form*

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = 2 \sum_{i=1}^{n} \left(-\log\left(\frac{y_i}{\mu_i}\right) + \frac{(y_i - \mu_i)}{\mu_i} \right).$$

3. *For a sample where* $y_i \sim P(\mu_i)$ *with mean* μ_i *the deviance is given by*

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = 2 \sum_{i=1}^{n} \left(y_i \log\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i) \right).$$

4. *For a sample where* $m y_i \sim Bin(m, \pi_i)$ *with mean* μ_i *the deviance is given by*

$$D(\boldsymbol{y}, \boldsymbol{\mu}) = 2 \sum_{i=1}^{n} m_i \left(y_i \log\left(\frac{y_i}{\mu_i}\right) + (1 - y_i) \log\left(\frac{1 - y_i}{1 - \mu_i}\right) \right).$$

1.3.2 Pearson Statistics

Wei (1998, p. 22) lists the use of the generalized Pearson X^2 statistic as another suitable method in order to estimate the dispersion parameter ϕ within GNMs which was also proposed by McCullagh and Nelder (1989, p. 358) and Smyth (2003). For given responses y_i, optional weights w_i and fitted values μ_i the Pearson statistic is given by the

sum

$$X^2 = \sum_{i=1}^{n} w_i \frac{(y_i - \hat{\mu}_i)^2}{\mathrm{V}(\hat{\mu}_i)}. \tag{1.19}$$

The dispersion parameter can be estimated by means of the Pearson statistics

$$\hat{\phi} = \frac{X^2}{n - P}, \tag{1.20}$$

corrected by the residual degrees of freedom (df) $n - P$. Equation (1.20) represents the default estimator in statistical programs, particularly in R, as Smyth (2003, p. 123) points out.

1.4 Model Specification

This section intends to reflect the previously discussed components of a GNM in order to state the framework for the further work. The GNM consists basically of the following three components.

Systematic Component: A GNM includes independent predictor variables $\boldsymbol{x}_i$ by a nonlinear regression function in $\boldsymbol{\beta}$ as

$$g(\mu_i) = h(\boldsymbol{x}_i, \boldsymbol{\beta}), \qquad i = 1, \ldots, n, \ \boldsymbol{\beta} \in \mathbb{R}^P.$$

The regression function $h(\boldsymbol{x}_i, \boldsymbol{\beta})$ models in general a known functional relationship between the response y_i and predictor variables $\boldsymbol{x}_i$. The nonlinear regression functions complies with the specification in Section 1.1.

Link Function: Within GNMs the link function establishes a relationship between y_i and its mean μ_i through

$$\mu_i = g^{-1}(h(\boldsymbol{x}_i, \boldsymbol{\beta})), \qquad i = 1, \ldots, n,$$

where $g(\cdot)$ represents a monotonic function and the function $h(\cdot)$ is in general nonlinear as specified in the *systematic component*. The link function is previously defined ensuring that the mean function maps into the set of plausible values of μ_i considering the distributional properties. In the case of nonlinear regression functions the often problem specific nonlinear functional structure allows for specifications enabling a plausible mapping domain. Therefore, the identity link can be used where the mean vector is directly modeled by the regression function given by $g(\mu_i) = \mu_i$ for $i = 1, \ldots, n$.

Stochastic Component: The random part of GNMs is defined by distributional properties of the response variable y_i. The generalization of the response distribution to the broad class of the exponential family is a key point in the theory of GLMs and can be applied to GNMs. Whereas the assumption for the standard error term distribution is normal, the class of GNMs allows to specify other distributions like Gamma or Poisson. The response follows the exponential family with pdf (1.6). Based on (1.10) and

(1.11), the mean and variance are given by

$$\mathrm{E}[y_i] = \mu_i, \quad \mathrm{Var}(y_i) = \phi \cdot \mathrm{V}(\mu_i), \quad i = 1, \ldots, n,$$

with dispersion parameter ϕ and variance function $\mathrm{V}(\mu_i)$. This setup shows that GNMs allow for modeling non-constant variances depending on the variance function $\mathrm{V}(\mu_i)$. Distributional properties are specified through the characteristics of the linear exponential family in Section 1.2 and the specifications for the dispersion parameter in Section 1.3.

In the scope of the remaining work, GNMs will serve as an important base for further models. In particular, responses are assumed to stem from an exponential family with nonlinear mean function $\mu_i(\boldsymbol{\beta})$. Further computations and analysis on GNMs will focus particularly on the main parameter of interest $\boldsymbol{\beta}$ relating to the mean through the link function. Within nonlinear regression models the fitted values range in general within an adequate mapping domain due to the often problem specific functional structure. Therefore, the link function is restricted to the identity link within the remaining work which actually facilitates the handling of the subsequent mixture models. Consequently, the mean function is modeled directly through $\mu_i(\boldsymbol{\beta}) = h(\boldsymbol{x}_i, \boldsymbol{\beta})$. For reasons of comparability the pdf will be denoted as $f(y_i; \mu_i(\boldsymbol{\beta}), \phi)$ following the scientific literature on GNMs. A further focus of the remaining work lies on estimation methods for the dispersion parameter ϕ as necessary distributional parameters for GNMs and supplementary information on the variability of the responses.

CHAPTER 2

Finite Mixtures of Generalized Nonlinear Models

Introduction

A Finite Mixture Model (FMM) represents a highly accomodative statistical model which gained strong interest in recent years. Due to their flexibility FMMs cover a large area of application. They allow to model complex distributional shapes as well as evident group structures in heterogeneous data sets. Complexity in distributional shapes is in general driven by heterogeneous patterns which are attributed to latent classes. The assumption on the existence of latent classes affords prior knowledge on their specifics as the fitting of FMMs requires distributional assumptions. Latent classes yield to the assignment of the data to distinct components which can be viewed as clusters. Pearson (1894) adapted in a first report about FMMs a mixture of two heteroscedastic normal pdfs with different means to a data set of crabs body lengths. Presuming the existence of two subgroups in the data he was able to handle the skewed pdf properly by the resulting mixture distribution. The assumption on latent structures within a given population is an essential feature that distinguishes FMMs from other statistical models like nonparametric models. The latter comprise distribution free fitting methods and functions with structures that are highly data driven. Therefore nonparametric models may attach more weight to specific data structures or outliers while the fitting of FMMs places emphasis on the parametrization of a predefined distributional setup. FMMs are in general also referred to as *probability-based* clustering methods.

Even though FMMs offer a high level of flexibility in modeling, they comprise various components which afford an appropriate specification and in a further step accurate estimation techniques. The general estimation method for the unknown parameters in FMMs is the ML estimation. With the work of Dempster et al. (1977) a suitable and computationally advantageous fitting method for FMMs was found in the EM algorithm. Based on the likelihood function, the EM algorithm enabled an efficient computation as it approached to solve the original problem as an incomplete data structure problem. With the application of the EM algorithm to the fitting of FMMs the latter attracted in-

creasing interest among statisticians in different fields. The fitting of FMMs still leads to challenges which remain problem specific. The choice of the final number of components in the FMM has impact on the computational results of the EM algorithm as well as the selection of proper starting values. Nevertheless, the advantages of FMMs clearly outbalance the challenges as the increased number of applications show. The probability-based clustering by means of the EM algorithm enables the fitting of mixtures of regression models which have contributed to the gain in popularity of FMMs. In this context, latent classes are assumed to exhibit specific functional structures which are modeled by component specific mean regression functions. In the general model the mixture components follow a specified distribution which will be restricted to the exponential family in further analysis. Additionally, the underlying mean is assumed to follow a specified nonlinear regression function.

The main objective of this chapter is to introduce the class of mixtures of GNMs (as described in Chapter 1) which have not been investigated up to now as no previous research on this topic is available. Furthermore, an implementation of these models was provided in R and will be addressed in this work. Section 2.1 defines the associated model specification and necessary terms for further analysis while Section 2.2 discusses identifiability aspects. The general estimation method is based on ML estimation which is outlined in Section 2.3. The subsequent Section 2.4 derives the estimation approach of the EM algorithm. The computation of standard errors is outlined in Section 2.5 while Section 2.6 introduces two well-known extensions of the EM algorithm. The final Section 2.7 comprises information criteria for the optimal choice of the number of components within mixtures of GNMs.

2.1 Model Specification

Let $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ be a random vector with independent but not necessarily identically distributed components y_i. Each component is assumed to have a mixture distribution denoted by $f^M(\cdot)$. The mixture distribution consists of K components each from an exponential family. The K-component mixture distribution is given by its mixture pdf

$$f^M(y_i; \mu_i(\boldsymbol{\beta}), \boldsymbol{\phi}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k), \quad i = 1 \ldots, n \tag{2.1}$$

with vectors of dispersion parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_K)^\top$ and weight parameters $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top$. The mixture distribution consists of K single distributions which are often referred to as *components* in literature. In fact, a mixture model with $K < \infty$ is also known as *Finite Mixture Model (FMM)*. The weights π_k are often referred to as *component weights* and follow the necessary conditions

$$0 \leq \pi_k \leq 1 \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1, \quad k = 1, \ldots, K,$$

in order to guarantee probabilistic properties of the mixture density function $f^M(\cdot)$. The component specific density functions are denoted by $f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k)$ for each y_i,

$i = 1, \ldots, n$. The component specific density functions stem in general from the same parametric family with pdf as given in (1.6) and differ solely in the component specific means $\mu_i(\boldsymbol{\beta}_k)$ and dispersion parameter ϕ_k for each component $k = 1, \ldots, K$. The mean $\mu_i(\boldsymbol{\beta}_k)$ is modeled through the GNM

$$\mu_i(\boldsymbol{\beta}_k) = g^{-1}(h_i(\boldsymbol{\beta}_k)) = g^{-1}(h(\boldsymbol{x}_i, \boldsymbol{\beta}_k)), \quad i = 1, \ldots, n, \tag{2.2}$$

with component specific regression coefficient vectors $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_K^\top)^\top$ and for each component $\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kP})^\top$. The component specific parameters can vary between all components, known as *varying effects* in literature, as Grün and Leisch (2006, p. 2) point out, or some of them may be fixed over all components (*fixed effects*). Therefore, it is also possible to allow for a variation between groups of components due to specific knowledge of the problem, which corresponds to a nesting. While (2.2) models the mean within each component, the general mean over all components $k = 1, \ldots, K$ can be derived from (2.1) as

$$\mu_i(\boldsymbol{\beta}) = \mathrm{E}[y_i] = \sum_{k=1}^{K} \pi_k \mu_i(\boldsymbol{\beta}_k), \quad i = 1, \ldots, n. \tag{2.3}$$

The FMM with mixture pdf as given in (2.1) is fully specified through the component weights, the component specific mean functions and the dispersion parameters. As Wei (1998, p. 15) stresses, it is common in literature to analyze the statistical behavior of the regression parameter $\boldsymbol{\beta}_k$ directly since there is a mapping between the component specific means $\mu_i(\boldsymbol{\beta}_k)$ and their regression coefficient vectors $\boldsymbol{\beta}_k$ given by (2.2) for $i = 1, \ldots, n$. For further discussion, all parameters will be pooled in one parameter vector

$$\boldsymbol{\Psi} = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_K^\top, \phi_1, \ldots, \phi_K)^\top \tag{2.4}$$

with component specific regression coefficients $\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kP})^\top$. The parameter vector (2.4) is of main interest in further calculations and comprises all occurring parameters in the FMM for the sake of clarity. The main goal of further analysis is to compute suitable estimates for $\boldsymbol{\Psi}$. This includes in particular the computation of estimators for the component specific regression coefficients $\boldsymbol{\beta}_k$ and dispersion parameters ϕ_k for each component $k = 1, \ldots, K$ for any given mixture of GNMs.

2.2 Identifiability

The identifiability of FMMs represents an important aspect in the theory about FMMs as it has direct influence on the fitting procedures and the interpretation of the component parameters. For K components with component specific pdfs $f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k)$, the mixture model (2.1) is determined through the parameter vector $\boldsymbol{\Psi}$ as specified in Equation (2.4). Considering two mixture densities

$$f^M(y_i; \boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k) \quad \text{and} \quad f^M(y_i; \boldsymbol{\Psi}^*) = \sum_{k=1}^{K^*} \pi_k^* f(y_i; \mu_i(\boldsymbol{\beta}_k^*), \phi_k^*)$$

with parameter vectors $\boldsymbol{\Psi}, \boldsymbol{\Psi}^* \in \Omega$, the mixture model is *not identifiable* if for any two distinct parameters $\boldsymbol{\Psi} \neq \boldsymbol{\Psi}^*$ the respective mixture distributions equal, respectively

$$f^M(\boldsymbol{y}; \boldsymbol{\Psi}) \equiv f^M(\boldsymbol{y}; \boldsymbol{\Psi}^*), \quad \boldsymbol{y} \in \mathcal{Y}. \tag{2.5}$$

Within the context of mixture models identifiability problems arising through the equivalence (2.5) can be attributed to several reasons. The mixture distribution (2.1) is in general invariant to the *labeling of the components*. Therefore, two parameter vectors $\boldsymbol{\Psi} \neq \boldsymbol{\Psi}^*$ which differ solely in their components' labeling yield to equal mixture distributions. A K-component FMM is in general exposed to $K!$ ways of arranging the components yielding the same mixture distribution. The general interchangeability of the components $f(\cdot)$ can be avoided by imposing a distinct restriction on the order of the component labels by establishing a relationship between the components' parameters $(\pi_k, \boldsymbol{\mu}(\boldsymbol{\beta}_k), \phi_k)$ for $k = 1, \ldots, K$. As Frühwirth-Schnatter (2006, p. 19) points out, FMMs are identifiable concerning the interchangeability of their labels under a weak constraint where the component specific parameters differ at least in one parameter. A possible approach deals with the ordering according to the size of the mixing proportions through the condition

1. $\pi_1 < \pi_2 < \ldots < \pi_K$

which is appropriate for mixtures with different component sizes. Stricter constraints may be given, for example, by ordering the component specific distribution parameters $(\boldsymbol{\mu}(\boldsymbol{\beta}_k), \phi_k)$. Nevertheless, FMMs require a careful choice of component ordering constraints. Frühwirth-Schnatter (2006, p. 19) draws attention to related difficulties whereas a restriction on variances might involve the non-idenfiability of mixtures with equal variances. Identifiability problems as specified in Equation (2.5) for two distinct parameter vectors $\boldsymbol{\Psi} \neq \boldsymbol{\Psi}^*$ may also arise due to a potential *overfitting*. This situation can be attributed to empty components or several equally parametrized components. Requiring a different parametrization within the components rules out the possibility of equal component pdfs. Similar to the ordering constraint, a weak inequality requirement may suffice for the identifiability of distinct components by imposing, for example, a constraint on the component specific regression coefficients. As a suitable measure for dealing with overfitting, Grün and Leisch (2008c) state the following necessary conditions:

2. $\pi_k > 0 \; \forall k = 1, \ldots, K$ and

3. $k \neq l \Rightarrow \boldsymbol{\beta}_k \neq \boldsymbol{\beta}_l \; \forall \; k, l = 1, \ldots, K.$

Despite these considerations regarding the interchangeability of components and potential overfitting, FMMs may possibly remain non-identifiable. *Generic identifiability* has been studied for different mixtures of several specific distributions apart from mixtures of regression models. Teicher (1963) proved the identifiability for finite mixtures of the univariate normal and Gamma distribution. These results were advanced for multivariate cases and further distributions by Yakowitz and Spragins (1968) and Titterington et al. (1985). Yakowitz and Spragins (1968, p. 210) showed that FMMs of a specific distributional family are identifiable if the members of the underlying distributional family

are linearly independent over the field of real numbers. General identifiability results were derived for the Gaussian, Gamma and Poisson distribution. Special cases comprise mixtures of binomial distributions which are identifiable under a restriction on the number of components.

Identifiability of mixtures of regression models has been discussed by Wang et al. (1998), Hennig (2000), Grün and Leisch (2008a,c) and Frühwirth-Schnatter (2006). The generic identifiability of mixtures of normal regression models does not necessarily follow from generic identifiability of mixtures of distributions as Hennig (2000, p. 276) and Frühwirth-Schnatter (2006, p. 243) point out. Within the context of mixtures of linear regression models, Hennig (2000) completed the requirement on the a full rank covariate matrix $\boldsymbol{X}$ by further conditions. Therefore, regression parameters are identifiable if the number of distinct hyperplanes generated by the covariates (exlcuding the intercept) exceeds the number of clusters K. This is especially critical if the regression model includes categorical variables or in the case of low variability of the explanatory variables, as Grün and Leisch (2008a, p. 8) and Frühwirth-Schnatter (2006, p. 244) outline. Grün and Leisch (2008a) generalized the identifiability results for the class of mixtures of GLMs.

Theorem 2.1 *Identifiability of mixtures of GLMs, Grün and Leisch (2008a, p. 7)*
The mixture model defined by

$$h(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{\Psi}) = \prod_{t=1}^{N} \left[\sum_{k=1}^{K} \pi_k \prod_{i \in I_t} f(y_i; \mu_{ik}, \phi_k) \right] \tag{2.6}$$

and

$$g^{-1}(\mu_{ik}) = \boldsymbol{x}_i^\top \boldsymbol{\beta}_k \tag{2.7}$$

is identifiable if the following conditions are fullfilled:

1. *(a)* $\exists\, \tilde{I} \neq \emptyset : \tilde{I} \subseteq \bigcup_{t=1}^{N} I_t$ *: The mixture of distributions given by* $\sum_{k=1}^{K} \pi_k f(y_i; \mu_{ik}, \phi_k)$ *is identifiable* $\forall\, i \in \tilde{I}$.

   (b) $q^* > K$ *with*

   $$q^* := \left\{ q : \forall\, i^* \in \tilde{I} : \exists H_j \in \{H_1, \ldots, H_q\} : \{x_i : i \in I_{t(i^*)} \cap \tilde{I}\} \subset H_j \wedge H_j \in \mathcal{H}_U \right\}$$

   where $\mathcal{H}_U$ *is the set of* $H(\alpha) := \{\boldsymbol{x} \in \mathbb{R}^U : \alpha^\top \boldsymbol{x} = \boldsymbol{0}\}$ *with* $\alpha \neq \boldsymbol{0}$.

2. *The matrix* $\boldsymbol{X}$ *has full rank.*

I_t comprises the set of indices from repeated observations with fixed component membership corresponding to individual t. The respective observations are given by $(y_t, \boldsymbol{x}_t) = (y_i, \boldsymbol{x}_i)_{i \in I_t}$. The present identifiability results on mixtures of GLMs can be adopted to mixtures of GNMs only to a limited extent. A crucial difference arises through the substitution of the mean function in (2.7) through a nonlinear regression function (1.3). The fitting of GNMs depends strongly on the specification of the nonlinear mean function (2.2) which is in general problem specific. The specification of the nonlinear functional form affects the dimension of the regression coefficient vector and the quality of the numerical fitting procedures. The latter might be complicated through arising colinearities

within the optimization methods. Within mixtures of GNMs different starting values are applied to the numerical fitting methods in order to detect possible multiple solutions. A related identifiability aspect on general latent class models is given by the *local identifiability* concept. As Kim and Lindsay (2015, p. 746) outline, local identifiability refers to an open neighborhood of parameter estimates where every parameter of the respective neighborhood generates an unique distribution. The verification of locally identifiable regions is obtained by checking the Fisher information on the estimated parameters. Local identifiability of parameter estimates depends on the non-singularity of the information matrix as discussed in Allman et al. (2009) and Rothenberg (1971).

2.3 Maximum Likelihood Estimation

Let $\boldsymbol{y} \in \mathcal{Y}$ be a random vector with mixture distribution as given in Section 2.1. The unknown parameter vector $\boldsymbol{\Psi}$ in (2.4) specifies the FMM and has to be estimated. The general method for finding an appropriate estimator $\hat{\boldsymbol{\Psi}}$ is the ML estimation. The likelihood function of the FMM corresponds to the joint pdf of the observed data vector viewed as a function in the parameters. Based on the mixture pdf given in (2.1) the likelihood function results in

$$\mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{y}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k). \tag{2.8}$$

The likelihood function is assumed to be twice differentiable in the unknown parameter vector $\boldsymbol{\Psi}$. For the sake of clarity the component specific pdfs will be denoted as

$$f_{ik} := f(y_i; \mu_i(\boldsymbol{\beta}_k), \phi_k)$$

for further analysis and $i = 1, \ldots, n$ and $k = 1, \ldots, K$. The corresponding log-likelihood function takes the specific form

$$\ell(\boldsymbol{\Psi}; \boldsymbol{y}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k f_{ik} \right\}. \tag{2.9}$$

In order to derive the ML estimator for the unknown parameter vector $\boldsymbol{\Psi}$, the score function of the log-likelihood function (2.9) has to fulfill the necessary condition

$$S(\boldsymbol{y}; \boldsymbol{\Psi}) = \frac{\partial \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \boldsymbol{\Psi}} \stackrel{!}{=} \mathbf{0}. \tag{2.10}$$

2.4 Expectation-Maximization (EM) Algorithm

The EM algorithm was introduced by Dempster et al. (1977) as an iterative computation method for ML estimation. The iteration scheme is straightforward and general as it can be applied to various statistical problems. Along to the original work of Dempster et al. (1977), McLachlan and Krishnan (2008) represents a detailed source for the areas of application as well as for different extensions of the EM algorithm.

The log-likelihood function for mixtures of GNMs (2.9) turns out to be computationally highly complex and time-expensive due to the need of extensive numerical methods. Because of its complexity the maximization of the likelihood function is difficult to provide and may be solved iteratively. Therefore the EM algorithm is applied for an iterative computation of the MLE $\hat{\boldsymbol{\Psi}}$. The algorithm is based on an incomplete-data structure problem and applies repeatedly two steps until an appropriate result is reached. The incomplete-data structure problem will be discussed in more detail in Section 2.4.1. Subsequent sections present the derivation of the EM algorithm for FMMs of GNMs.

2.4.1 Incomplete-Data Structure Problem

The basic idea of the EM algorithm is to view the given data vector $\boldsymbol{y} \in \mathcal{Y}$ as being incomplete. The data vector $\boldsymbol{y}$ is also denoted as *observable* or *incomplete* data vector in the context of the EM algorithm. As McLachlan and Peel (2000, p. 19) explain, $\boldsymbol{y}$ is assumed to have an underlying mixture pdf (2.1) denoted by $f^M(\boldsymbol{y}; \boldsymbol{\Psi})$ with the unknown parameter vector $\boldsymbol{\Psi} \in \Omega$. Dempster et al. (1977, p. 15) made the assumption about the existence of a finite set of unobservable variables associated to the vector $\boldsymbol{y}$. Actually, each observation y_i is linked to K indicator variables $z_{i1}, \ldots, z_{iK}$ assigning the membership of y_i to the kth mixture component for $k = 1, \ldots, K$. The indicator variables are also referred to as *missing* or *hidden* information within the framework of the EM algorithm. Let $\boldsymbol{z}_i \in \mathcal{Z} = \{0, 1\}^K$ denote the component label vectors with $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iK})^\top$ where all components equal to zero except one. Thus, $\boldsymbol{z}_i$ realize in the set of binary numbers as

$$z_{ik} = \begin{cases} 1, & \text{if } y_i \in k\text{th component,} \\ 0, & \text{if } y_i \notin k\text{th component,} \end{cases}$$

assigned with the component weights π_k as probabilities of occurrence for $i = 1, \ldots, n$ and $k = 1, \ldots, K$. The component weights π_k are viewed as the prior probability of y_i stemming from the kth component. Accordingly, the allocation of the component memberships of y_i is binomially distributed with probability π_k for each component $k = 1, \ldots, K$. Therefore the label vectors $\boldsymbol{z}_i$ are multinomially distributed

$$\boldsymbol{z}_i \overset{\text{iid}}{\sim} Mult_K(1, \boldsymbol{\pi}), \quad i = 1, \ldots, n,$$

with the component weights as event probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top$. Let $\boldsymbol{y}^c$ denote the data vector consisting of the augmented samples

$$\boldsymbol{y}^c = (\boldsymbol{y}^\top, \boldsymbol{z}^\top)^\top$$

defined on the set $\mathcal{Y}^c = \mathcal{Y} \times \mathcal{Z}$. $\boldsymbol{y}^c$ is also known as *complete* data vector in the context of the EM algorithm. The pdf of $\boldsymbol{y}^c$ is denoted by $f^c(\boldsymbol{y}^c; \boldsymbol{\Psi})$ with parameter vector $\boldsymbol{\Psi}$. The pdf of the complete vector $\boldsymbol{y}^c$ represents the marginal distribution of the observable and hidden information given by

$$f^c(\boldsymbol{y}^c; \boldsymbol{\Psi}) = f(\boldsymbol{y}; \boldsymbol{\Psi}) \cdot g(\boldsymbol{y}^c | \boldsymbol{y}; \boldsymbol{\Psi}).$$

Modifying the pdf of the complete data vector results in the following relationship of the corresponding log-likelihood function

$$\ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) = \ell(\boldsymbol{\Psi}; \boldsymbol{y}) + \log g(\boldsymbol{y}^c|\boldsymbol{y}; \boldsymbol{\Psi}). \tag{2.11}$$

The unobserved but complete log-likelihood function has the specific form

$$\begin{aligned} \ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) &= \sum_{i=1}^{n} \log \left\{ \prod_{k=1}^{K} (\pi_k f_{ik})^{z_{ik}} \right\} = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \{\pi_k f_{ik}\} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \{\log \pi_k + \log f_{ik}\}. \end{aligned} \tag{2.12}$$

2.4.2 Definitions of the EM Algorithm

Since the component label vectors cannot be observed, they represent missing or hidden information and obviate a direct computation of the complete log-likelihood function (2.12). The EM algorithm provides an indirect optimization of (2.12) by taking the conditional expectation on observable information and providing an iterative maximization with respect to the unknown parameter vector $\boldsymbol{\Psi}$, as the following description shows.

Let $\boldsymbol{\Psi}^{(j)}$ be the current estimate in the jth iteration. In the following $\mathrm{E}[\cdot|\boldsymbol{y}, \boldsymbol{\Psi}^{(j)}]$ denotes the expectation operator taking the current estimate $\boldsymbol{\Psi}^{(j)}$ for $\boldsymbol{\Psi}$ in computations. Rearranging (2.11) results in

$$\ell(\boldsymbol{\Psi}; \boldsymbol{y}) = \ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) - \log g(\boldsymbol{y}^c|\boldsymbol{y}; \boldsymbol{\Psi}). \tag{2.13}$$

Let the following conventions hold for the expectations of the log-likelihood functions

$$\begin{aligned} \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}) &:= \mathrm{E}[\ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)|\boldsymbol{y}, \boldsymbol{\Psi}^{(j)}], \\ \mathrm{H}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}) &:= \mathrm{E}[\log g(\boldsymbol{y}^c|\boldsymbol{y}; \boldsymbol{\Psi})|\boldsymbol{y}, \boldsymbol{\Psi}^{(j)}]. \end{aligned}$$

Therefore, the incomplete-data log-likelihood can be expressed as

$$\ell(\boldsymbol{\Psi}; \boldsymbol{y}) = \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}) - \mathrm{H}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}). \tag{2.14}$$

McLachlan and Krishnan (2008, p. 78) show that the EM algorithm guarantees an increase of the incomplete data log-likelihood after every iteration step due to

$$\begin{aligned} \ell(\boldsymbol{\Psi}^{(j+1)}; \boldsymbol{y}) - \ell(\boldsymbol{\Psi}^{(j)}; \boldsymbol{y}) &= \mathrm{Q}(\boldsymbol{\Psi}^{(j+1)}; \boldsymbol{\Psi}^{(j)}) - \mathrm{Q}(\boldsymbol{\Psi}^{(j)}; \boldsymbol{\Psi}^{(j)}) \\ &\quad + \mathrm{H}(\boldsymbol{\Psi}^{(j)}; \boldsymbol{\Psi}^{(j)}) - \mathrm{H}(\boldsymbol{\Psi}^{(j+1)}; \boldsymbol{\Psi}^{(j)}) \geq 0. \end{aligned}$$

For the iterative computation of a series of estimates $\boldsymbol{\Psi}^{(1)}, \ldots, \boldsymbol{\Psi}^{(j)}, \boldsymbol{\Psi}^{(j+1)}$ the inequality

$$\mathrm{Q}(\boldsymbol{\Psi}^{(j+1)}; \boldsymbol{\Psi}^{(j)}) \geq \mathrm{Q}(\boldsymbol{\Psi}^{(j)}; \boldsymbol{\Psi}^{(j)})$$

holds as $\boldsymbol{\Psi}^{(j+1)}$ is chosen in order to maximize $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$.The remaining term is non-

negative due to the the inequality

$$\mathrm{H}(\boldsymbol{\Psi}^{(j+1)};\boldsymbol{\Psi}^{(j)}) - \mathrm{H}(\boldsymbol{\Psi}^{(j)};\boldsymbol{\Psi}^{(j)}) \leq 0,$$

which is valid due to the Jensen inequality and the concavity of the logarithmic function, see also McLachlan and Krishnan (2008, p. 78). This result is equivalent with an increasing log-likelihood function when updating the parameter vector $\boldsymbol{\Psi}$,

$$\ell(\boldsymbol{\Psi}^{(j+1)};\boldsymbol{y}) \geq \ell(\boldsymbol{\Psi}^{(j)};\boldsymbol{y}), \quad \forall\, j \geq 0. \tag{2.15}$$

These results prove the monotonicity and convergence of the sequence of estimators provided by the EM algorithm if the likelihood is bounded above.

2.4.3 Expectation-Step

As the complete log-likelihood function $\ell^c(\boldsymbol{\Psi};\boldsymbol{y}^c)$ comprises unknown information, its computation is provided by taking the conditional expectation given the observed data $\boldsymbol{y}$ and a current estimate $\boldsymbol{\Psi}^{(j)}$. $\boldsymbol{\Psi}^{(j)}$ denotes the approximated value of the unknown parameter vector $\boldsymbol{\Psi}$ after the jth EM iteration step. The expectation function of the complete data log-likelihood $\mathrm{Q}(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(j)})$ results by substituting the unknown parameter vector $\boldsymbol{\Psi}$ by its current estimate $\boldsymbol{\Psi}^{(j)}$ in the computation of the log-likelihood function $\ell^c(\boldsymbol{\Psi};\boldsymbol{y}^c)$. The expectation $\mathrm{Q}(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(j)})$ is given in the jth iteration as

$$\begin{aligned}
\mathrm{Q}(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(j)}) &= \mathrm{E}\left[\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\left\{\log\pi_k + \log f_{ik}\right\}\Big|\boldsymbol{y},\boldsymbol{\Psi}^{(j)}\right] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathrm{E}\left[z_{ik}\big|y_i,\boldsymbol{\Psi}^{(j)}\right]\left\{\log\pi_k + \log f_{ik}\right\}.
\end{aligned}$$

The complete data log-likelihood $\ell^c(\boldsymbol{\Psi};\boldsymbol{y}^c)$ is linear in the classification indicators z_{ik}. Therefore, the computation of the expectation $\mathrm{Q}(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(j)})$ affords in particular the calculation of the conditional expectation of z_{ik} given the observable data $\boldsymbol{y}$ and the current estimate $\boldsymbol{\Psi}^{(j)}$ in terms of

$$w_{ik}^{(j)} := \mathrm{E}[z_{ik}=1|y_i,\boldsymbol{\Psi}^{(j)}] = \mathbb{P}[z_{ik}=1|y_i,\boldsymbol{\Psi}^{(j)}] = \frac{\pi_k^{(j)} f_{ik}^{(j)}}{\sum_{l=1}^{K}\pi_l^{(j)} f_{il}^{(j)}}, \tag{2.16}$$

for $i = 1,\ldots,n$ and $k = 1,\ldots,K$. Using result (2.16) for the computation of the conditional expectation yields

$$\mathrm{Q}(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(j)}) \overset{(2.16)}{=} \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}^{(j)}\left\{\log\pi_k + \log f_{ik}\right\}, \quad j \geq 0. \tag{2.17}$$

The specific conditional expectation of the complete log-likelihood function given by (2.17) serves as the objective function for the maximization within the EM algorithm. McLachlan and Peel (2000, p. 20) denote the proportions w_{ik} as the *posterior probabilities* for y_i being drawn from the kth component.

2.4.4 Maximization-Step

The Maximization (M)-step provides the subsequent approximation $\boldsymbol{\Psi}^{(j+1)}$ of the unknown parameter vector $\boldsymbol{\Psi}$ in the iteration step $(j+1)$. Calculating the updated estimate $\boldsymbol{\Psi}^{(j+1)}$ requires the maximization of the underlying conditional expectation function $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$ in (2.17) with respect to $\boldsymbol{\Psi}$ and subject to the condition

$$\mathrm{Q}(\boldsymbol{\Psi}^{(j+1)}; \boldsymbol{\Psi}^{(j)}) \geq \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}), \qquad j \geq 0.$$

As the complete specification of mixtures of GNMs requires the estimation of the component weights, the component specific regression coefficients, in order to compute the mean functions, and the component specific dispersion parameters, their estimation is addressed separately within this section.

2.4.4.1 Component Weights

Considering the component weights or mixing proportions π_k, the maximization of the log-likelihood has to be provided subject to the condition $\sum_{k=1}^{K} \pi_k = 1$. This constrained maximization can be performed by including a Lagrangian multiplier λ. The Lagrangian function will be denoted as $\tilde{\mathrm{Q}}$ and is given by

$$\tilde{\mathrm{Q}}(\boldsymbol{\Psi}; \lambda; \boldsymbol{\Psi}^{(j)}) = \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}) - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right). \tag{2.18}$$

The maximization of the unconstrained Lagrangian function $\tilde{\mathrm{Q}}$ replaces the maximization of the former likelihood function with constraints. Thus the estimates $\hat{\pi}_k$ can be derived by zeroing the derivative of (2.18) which results in the equation

$$\frac{\partial \tilde{\mathrm{Q}}(\boldsymbol{\Psi}, \lambda; \boldsymbol{\Psi}^{(j)})}{\partial \pi_k} = \sum_{i=1}^{n} \frac{w_{ik}^{(j)}}{\pi_k} - \lambda \stackrel{!}{=} 0 \tag{2.19}$$

for the components $k = 1, \ldots, K$. The classification variables $w_{ik}^{(j)}$ as given in (2.16) are also known as component proportions or posterior weights and denote the chance of an observation y_i belonging to the kth component as a proportion of the weighted component density to the overall mixture density. Therefore, the following equation holds for the mixing proportions

$$\pi_k = \frac{1}{\lambda} \sum_{i=1}^{n} w_{ik}^{(j)}.$$

The Lagrangian parameter λ maintains the component weights condition of summing up to one in (2.18). In order to obtain $\hat{\lambda}$, both sides are added up over all components yielding

$$\sum_{k=1}^{K} \frac{1}{\lambda} \sum_{i=1}^{n} w_{ik}^{(j)} = \sum_{k=1}^{K} \pi_k = 1 \Longleftrightarrow \hat{\lambda} = n.$$

The estimator for the Lagrangian parameter λ results in the total number of observations n. Consequently, the component weights estimates $\hat{\pi}_k$ are given by the mean value of the classification data w_{ik} given in Equation (2.20) with $\lambda = n$. The respective component weight estimator for the iteration step $(j+1)$ results in

$$\hat{\pi}_k^{(j+1)} = \frac{1}{n} \sum_{i=1}^{n} w_{ik}^{(j)}. \tag{2.20}$$

2.4.4.2 Component Specific Regression Coefficients

The specification of the component specific pdfs affords the fitting of the component specific mean functions and dispersion parameters. The aim of this section is to enable the fitting of component specific mean functions through the estimation of the regression coefficient vectors $\boldsymbol{\beta}_k$ for $k = 1, \ldots, K$. The dispersion parameters are treated as fixed values. Within the EM algorithm, the parameter estimates $\boldsymbol{\beta}_k^{(j)}$ are determined by maximizing the conditional expectation function $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$ with respect to the regression coefficients. The resulting score function with respect to the component specific regression coefficients $\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kP})^\top$ is given by

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \beta_{kp}} = \sum_{i=1}^{n} w_{ik}^{(j)} \frac{\partial \log f_{ik}}{\partial \beta_{kp}} = \sum_{i=1}^{n} w_{ik}^{(j)} \frac{\partial \log f_{ik}}{\partial \mu_{ik}} \frac{\partial \mu_{ik}}{\partial \beta_{kp}}.$$

Replacing the component pdfs by their exponential family expression leads to

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \beta_{kp}} \overset{(1.6)}{=} \sum_{i=1}^{n} w_{ik}^{(j)} \frac{y_i - \mu_{ik}}{\phi_k V(\mu_{ik})} \frac{\partial \mu_{ik}}{\partial \beta_{kp}} \tag{2.21}$$

for the component specific regression coefficients with $p = 1, \ldots, P$ and $k = 1, \ldots, K$. The $\boldsymbol{\beta}_k^{(j)}$ can be obtained by the Iteratively Re-weighted Least Squares (IWLS) algorithm which will be discussed in the subsequent section.

2.4.4.3 Iteratively Re-weighted Least Squares Algorithm

Within the EM algorithm, updating $\boldsymbol{\beta}_k^{(j)}$ by the subsequent estimate $\boldsymbol{\beta}_k^{(j+1)}$ affords in general an iterative procedure. In the following, a general iteration scheme will be derived for the fitting of $\boldsymbol{\beta}_k^{(j+1)}$ which is suitable within mixtures of GNMs. As the fitting procedure is being performed in the same way for the distinct components $k = 1, \ldots, K$, the component specific labeling will be waived in the interest of greater clarity within the subsequent derivation. The necessary iteration steps will be labeled by the control variable $t = 1, \ldots, t^*$ where t^* denotes the last iteration step after reaching a predefined convergence criteria. The related estimate $\boldsymbol{\beta}^{(t^*)}$ represents the EM update through $\boldsymbol{\beta}^{(j+1)} := \boldsymbol{\beta}^{(t^*)}$. The current estimate $\boldsymbol{\beta}^{(j)}$ serves as initial value through $\boldsymbol{\beta}^{(0)} := \boldsymbol{\beta}^{(j)}$ or rather $\boldsymbol{\Psi}^{(0)} := \boldsymbol{\Psi}^{(j)}$ where the component weights and dispersion parameters are presumed as fixed. A suitable estimate for the regression coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_P)^\top$ can be derived by applying the Fisher scoring method given the iteration step $(t+1)$,

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathcal{I}\left(\boldsymbol{\beta}^{(t)}\right)^{-1} \frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t)}}, \quad t \geq 0. \tag{2.22}$$

The scoring procedure affords an initial vector $\boldsymbol{\beta}^{(0)}$ and the computation of the expected information matrix $\mathcal{I}(\cdot)$ with respect to the regression coefficient vector $\boldsymbol{\beta}$. The negative of the Hessian matrix represents the *observed information matrix*, respectively

$$\mathrm{I}(\boldsymbol{\beta}; \boldsymbol{y}) := -\frac{\partial^2 \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}.$$

The expected value of the observed information matrix $\mathrm{I}(\boldsymbol{\beta}; \boldsymbol{y})$ leads to the *Fisher information* given by

$$\mathcal{I}(\boldsymbol{\beta}) := \mathrm{E}\left[\mathrm{I}(\boldsymbol{\beta}; \boldsymbol{y})\right] = \mathrm{E}\left[-\frac{\partial^2 \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}\right].$$

The application of the iteration procedure (2.22) with initial value $\boldsymbol{\beta}^{(0)}$ generates a sequence of estimates $\boldsymbol{\beta}^{(t)}$ converging to the MLE. Thus the algorithm is stopped if a predefined convergence criterion is reached. The component specific mean functions are going to be denoted in further analysis as

$$\boldsymbol{\mu} := \boldsymbol{\mu}(\boldsymbol{\beta}) = (\mu_1(\boldsymbol{\beta}), \ldots, \mu_n(\boldsymbol{\beta}))^\top,$$

with $\mu_i := \mu_i(\boldsymbol{\beta})$, $i = 1, \ldots, n$, for the sake of clarity. For mixtures of GNMs the score functions with respect to the component specific regression coefficients β_p are given by

$$\frac{\partial \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \beta_p} = \sum_{i=1}^{n} w_i \frac{\partial \log f_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_p} \overset{(1.6)}{\underset{(1.13)}{=}} \sum_{i=1}^{n} w_i \frac{y_i - \mu_i}{\phi V(\mu_i)} \frac{\partial \mu_i}{\partial \beta_p} = \sum_{i=1}^{n} w_i^* (y_i - \mu_i) \frac{\partial \mu_i}{\partial \beta_p}$$

for weights $w_i^* = w_i(\phi\, \mathrm{V}(\mu_i))^{-1}$ and $p = 1, \ldots, P$. The corresponding matrix form is given by

$$\frac{\partial \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \boldsymbol{\beta}} = \frac{1}{\phi} J(\boldsymbol{\beta})^\top W (\boldsymbol{y} - \boldsymbol{\mu}) \tag{2.23}$$

with diagonal matrix $W = \mathrm{diag}(w_1^*, \ldots, w_n^*)$ and the Jacobian matrix $J(\boldsymbol{\beta}) \in \mathbb{R}^{n \times P}$ containing the partial derivatives

$$J(\boldsymbol{\beta}) = \left(\frac{\partial \boldsymbol{\mu}(\boldsymbol{\beta})}{\partial \beta_1}, \ldots, \frac{\partial \boldsymbol{\mu}(\boldsymbol{\beta})}{\partial \beta_P}\right).$$

The observed information can be obtained by computing the second derivative of the expectation function (2.17) for mixtures of GNMs with respect to the regression coefficients vector. Taking the expectation yields the Fisher information, respectively

$$\mathcal{I}(\boldsymbol{\beta}) = -\mathrm{E}\left[\frac{\partial^2 \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}\right] \overset{(2.23)}{=} \frac{1}{\phi}\left[J(\boldsymbol{\beta})^\top W J(\boldsymbol{\beta})\right], \tag{2.24}$$

which can be inverted in order to derive the Fisher scoring iteration scheme (2.22). For complex likelihood functions, as it is the case with mixtures of GNMs, using the expected information instead of the observed information matrix leads to less calculation effort

and a decrease in computational time. In fact, within the iteration it is not necessary to explicitly compute the Hessian matrix.

Inverting the Fisher Information (2.24) leads to model-specific difficulties when dealing with nonlinear functional structures for the component specific mean functions. Based on the specific functional form of the regression function $h(\boldsymbol{x}_i, \boldsymbol{\beta})$ in (2.2), a possibly singular Jacobian matrix may occur in (2.24). Under these circumstances the matrix product $J(\boldsymbol{\beta})^\top W J(\boldsymbol{\beta})$ is in general not invertible which complicates the use of the Fisher scoring method in (2.22). Nevertheless, numerical solutions have been derived to deal with these rank deficient matrices by replacing the nonexistent inverse matrices by appropriate approaches. In connection with deriving roots of nonlinear equation systems, Ben-Israel (1966, p. 95) proposed to substitute the inverse of the singular Jacobian matrix by its generalized inverse matrix in the underlying Newton-Raphson algorithm. The generalized inverse matrix, also known as pseudoinverse or the Moore-Penrose inverse in literature, was derived by Penrose (1955) as a generalization of the inverse matrix for singular and non-quadratic matrices. For a definition of a generalized inverse matrix, reference is made to the Appendix. Applying the suggestion in Ben-Israel (1966, p. 95) to the inverse of the Fisher information leads to the generalized inverse matrix product

$$\mathcal{I}(\boldsymbol{\beta})^{-1} \approx \mathcal{I}(\boldsymbol{\beta})^{+} = \phi \left[J(\boldsymbol{\beta})^\top W J(\boldsymbol{\beta}) \right]^{+}$$

and enables the use of the Fisher scoring (2.22). Substituting $\mathcal{I}(\boldsymbol{\beta})^{-1}$ by its generalized inverse matrix $\mathcal{I}(\boldsymbol{\beta})^{+}$ yields to an iteration scheme. The resulting scoring method for updating the component specific regression coefficient vector $\boldsymbol{\beta}$ is prescribed as

$$\begin{aligned}
\boldsymbol{\beta}^{(t+1)} &= \boldsymbol{\beta}^{(t)} + \left[\frac{1}{\phi} J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} J\left(\boldsymbol{\beta}^{(t)}\right) \right]^{+} \frac{1}{\phi} J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} \left(\boldsymbol{y} - \boldsymbol{\mu}^{(t)}\right) \\
&= \boldsymbol{\beta}^{(t)} + \left[J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} J\left(\boldsymbol{\beta}^{(t)}\right) \right]^{+} J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} \left(\boldsymbol{y} - \boldsymbol{\mu}^{(t)}\right) \\
&= \left[J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} J\left(\boldsymbol{\beta}^{(t)}\right) \right]^{+} J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} \left[J\left(\boldsymbol{\beta}^{(t)}\right) \boldsymbol{\beta}^{(t)} + \left(\boldsymbol{y} - \boldsymbol{\mu}^{(t)}\right) \right] \\
&= \left[J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} J\left(\boldsymbol{\beta}^{(t)}\right) \right]^{+} J\left(\boldsymbol{\beta}^{(t)}\right)^\top W^{(t)} \boldsymbol{z}^{*(t)} \qquad (2.25)
\end{aligned}$$

with $\boldsymbol{\mu}^{(t)} := \mu(\boldsymbol{\beta}^{(t)})$. The resulting iteration step (2.25) reflects the typical form of an IWLS update. In each iteration step the new regression coefficient update $\boldsymbol{\beta}^{(t+1)}$ depends on the Jacobian matrix $J(\boldsymbol{\beta}^{(t)})$ and the adjusted dependent variables which are set as

$$\boldsymbol{z}^{*(t)} := J\left(\boldsymbol{\beta}^{(t)}\right) \boldsymbol{\beta}^{(t)} + \left(\boldsymbol{y} - \boldsymbol{\mu}^{(t)}\right). \qquad (2.26)$$

The iteration scheme comprises the two subsequent steps within each iteration $t \geq 0$:

1. Update the adjusted dependent variables (2.26) for a current approximation $\boldsymbol{\beta}^{(t)}$.
2. Obtain the regression coefficient vector $\boldsymbol{\beta}^{(t+1)}$ as given in the IWLS iteration (2.25) with the current adjusted variables and the approximation $\boldsymbol{\beta}^{(t)}$.

The fitting procedure consists of subsequent repetitions of these two steps until a prede-

fined convergence criterion is reached. In case of convergence, the regression coefficient vector after the final iteration step $\boldsymbol{\beta}^{(t^*)}$ is assigned as subsequent value within the EM algorithm, respectively $\boldsymbol{\beta}^{(j+1)} := \boldsymbol{\beta}^{(t^*)}$.

Example 2.1 (Normal Distribution)
For normally distributed components with $\phi_k = \sigma_k^2$ the score equation with respect to the regression coefficient vector is given by

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \boldsymbol{\beta}_{kp}} = \sum_{i=1}^{n} w_{ik}^{(j)} \frac{(y_i - \mu_{ik})}{\sigma_k^2} \frac{\partial \mu_{ik}}{\partial \beta_{kp}}.$$

Modifying the score equation leads to the WNLS approach, respectively

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \beta_{kp}} = \frac{1}{\sigma_k^2} \sum_{i=1}^{n} w_{ik}^{(j)} (y_i - \mu_{ik}) \frac{\partial \mu_{ik}}{\partial \beta_{kp}}.$$

Assuming a fixed dispersion parameter σ_k^2 for each component, yields also to the WNLS problem by the weighted RSS

$$\mathrm{RSS}(\boldsymbol{\beta}_k) := \|W_k^{1/2}(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\beta}_k))\|^2, \tag{2.27}$$

with diagonal weight matrix $W_k = \mathrm{diag}(w_{1k}, \ldots, w_{nk})$. The WNLS estimator satisfies $\hat{\boldsymbol{\beta}}_k := \arg\max_{\boldsymbol{\beta}_k} \mathrm{RSS}(\boldsymbol{\beta}_k)$. In order to derive $\hat{\boldsymbol{\beta}}_k$, the $\mathrm{RSS}(\boldsymbol{\beta}_k)$ has to be minimized with respect to $\boldsymbol{\beta}_k$ and equated by zero, respectively

$$\frac{\partial \, \mathrm{RSS}(\boldsymbol{\beta}_k)}{\partial \boldsymbol{\beta}_k} = 2J(\boldsymbol{\beta}_k)^\top W_k(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\beta}_k)) \stackrel{!}{=} \mathbf{0}$$

corresponding to the score function in (2.23).

A well-known problem when fitting mixtures with underlying normal distribution is the possibility of an occurring infinite likelihood as $\sigma_k^2 \to 0$. Within this case one variance tends to zero and causes estimation problems.

Example 2.2 (Gamma Distribution)
For Gamma distributed components with $\phi_k = \nu_k^{-1}$ and $\mu_k = -\theta_k^{-1}$ and $\mathrm{V}(\mu_{ik}) = \mu_{ik}^2$ the score function results in

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \beta_{kp}} = \sum_{i=1}^{n} w_{ik}^{(j)} \frac{\nu_k}{\mu_{ik}^2} (y_i - \mu_{ik}) \frac{\partial \mu_{ik}}{\partial \beta_{kp}} = \nu_k \sum_{i=1}^{n} w_{ik}^{*(j)} (y_i - \mu_{ik}) \frac{\partial \mu_{ik}}{\partial \beta_{kp}}.$$

According to (2.23) the weight matrix contains entries $w_{ik}^{*(j)} = w_{ik}^{(j)} / \mu_{ik}^2$.

For distributions with dispersion parameters differing from 1, ϕ_k is treated as fixed value in (2.21) and thus in the IWLS procedure. Appropriate estimates for the components dispersion parameters ϕ_k will be presented in the subsequent section.

2.4.4.4 Component Specific Dispersion Parameters

In order to specify the component specific pdfs f_{ik} completely, the dispersion parameters ϕ_k have to be estimated for each component $k = 1, \ldots, K$. The dispersion parameters are allowed to differ through all components inducing heteroscedastic mixture models but stay constant within a fixed component. The aim of this section is to conclude the estimation of the unknown parameter vector $\boldsymbol{\Psi}$ by sketching an appropriate estimation method for ϕ_k.

For members of the exponential family, the score function of (2.17) with respect to the dispersion parameter is given by

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \phi_k} = \sum_{i=1}^{n} w_{ik}^{(j)} \frac{\partial \log f_{ik}}{\partial \phi_k} = \sum_{i=1}^{n} w_{ik}^{(j)} \left(-\frac{y_i \theta_i - b(\theta_i)}{\phi_k^2} + \frac{\partial c(y_i, \phi_k)}{\partial \phi_k} \right). \quad (2.28)$$

Thus, the ML approach does not provide any general fitting method as the estimating function depends in particular on the structure of the term $c(y_i, \phi_k)$. Depending on the underlying distribution, $c(y_i, \phi_k)$ can take a complexity which cannot be handled by analytical solutions. Moreover the meaning and interpretation of ϕ_k changes for different distributions due to different choices of variance functions, as McCullagh and Nelder (1989, p. 357) explain. As Section 1.2.1 showed, the Poisson and binomial distribution have fixed $\phi_k = 1$, while other distributions like the normal and Gamma distribution certainly afford an appropriate estimation of the dispersion parameter. These practical difficulties concerning (2.28) disable the derivation of a general MLE $\hat{\phi}_k$ for all members of the exponential family and motivate the use of alternative estimation techniques as presented in Section 1.3. The usual approach for estimating the dispersion parameter for a non-tractable score function (2.28) is based on the deviance. Wei (1998, p. 21) emphasizes its adequacy within GNMs. Equation (1.18) allows to embed the weight components w_{ik} for the component specific dispersion parameters through $w_i = w_{ik}$. The final estimator for ϕ_k requires averaging over the component specific weights with w_{ik} defined as in (2.16),

$$\phi_k = \frac{D_w^*(\boldsymbol{y}; \hat{\boldsymbol{\mu}}_k)}{\sum_{i=1}^{n} w_{ik}},$$

where $\hat{\boldsymbol{\mu}}_k := \boldsymbol{\mu}(\hat{\boldsymbol{\beta}}_k)$ holds for $k = 1, \ldots, K$. Section 1.3 introduced the (weighted) Pearson statistic X^2 (1.19) which is also convenient within FMMs. Including the classification variables as weight components by $w_i := w_{ik}$ yields a modified Pearson statistics which can be used as an estimator for the computation of the dispersion parameter in mixture models. Within the EM algorithm, the estimation of the component specific dispersion parameters is provided independently of the computation of the component weights $\pi_k^{(j)}$ or the component specific regression coefficient vectors $\boldsymbol{\beta}_k^{(j)}$. Replacing the weight parameters by the posterior proportions obtained within the E-step (2.16), represents a necessary step for the computation of $\phi_k^{(j)}$ in the jth EM iteration step. Exemplary, normal and Gamma distributed components are mentioned as examples in the following.

Example 2.3 (Normal Distribution)
For normally distributed components the dispersion parameter equals $\phi_k = \sigma_k^2$. The score equation with respect to the dispersion parameter is given by

$$\frac{\partial \, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})}{\partial \sigma_k^2} = \sum_{i=1}^{n} w_{ik}^{(j)} \left(\frac{(y_i - \mu_{ik})^2}{2\sigma_k^4} - \frac{1}{2\sigma_k^2} \right)$$

and enables the derivation of an analytical solution for $\hat{\sigma}_k^2$. Equating the score function to zero leads to

$$\hat{\sigma}_k^{2\ (j+1)} = \frac{\sum_{i=1}^{n} w_{ik}^{(j)} (y_i - \hat{\mu}_{ik})^2}{\sum_{i=1}^{n} w_{ik}^{(j)}},$$

where $\hat{\mu}_{ik}$ corresponds to the fitted values provided by the component specific regression function $\mu_i(\hat{\boldsymbol{\beta}}_k)$ for observation y_i.

Using the weighted deviance (1.18) enables the computation of the component specific dispersion parameter for Gamma distributed components.

Example 2.4 (Gamma Distribution)
For Gamma distributed components, as defined in Example 1.3, the dispersion parameter equals the reciprocal value of the shape $\phi_k = \nu_k^{-1}$. The common approach to estimate the dispersion parameter refers to the deviance for weighted exponential families as the score function with respect to ν_k prohibits an analytical solution. The estimator for the dispersion parameter is derived by

$$\hat{\phi}_k^{(j+1)} = \frac{D_w^*(\boldsymbol{y}; \hat{\boldsymbol{\mu}}_k)}{\sum_{i=1}^{n} w_{ik}^{(j)}} = \frac{\sum_{i=1}^{n} w_{ik}^{(j)} (-\log(y_i/\hat{\mu}_{ik}) + (y_i - \hat{\mu}_{ik})/\hat{\mu}_{ik})}{\sum_{i=1}^{n} w_{ik}^{(j)}},$$

where $\hat{\mu}_{ik}$ corresponds to the fitted values provided by the component specific regression function $\mu_i(\hat{\boldsymbol{\beta}}_k)$ for observation y_i. The estimator can be applied to heteroscedastic mixtures of GNMs with underlying Gamma distributions.

2.4.5 Iteration Procedure and Convergence Aspects

The EM algorithm maximizes the conditional expectation of the complete log-likelihood function given by $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$ with respect to the unknown parameter vector $\boldsymbol{\Psi}$. Simultaneously, an increase in the incomplete log-likelihood function $\ell(\boldsymbol{\Psi}; \boldsymbol{y})$ is obtained as the relation (2.15) holds. Technically, the EM algorithm applies iteratively the following two steps for $j \geq 0$.

Steps of the EM-Algorithm:

Expectation (E)-step:
Calculate the posterior probabilities $w_{ik}^{(j)}$ and the expectation of the complete log-likelihood function $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$ for the current estimate $\boldsymbol{\Psi}^{(j)}$.

Maximization (M)-step:
Compute the subsequent update for the unknown parameter vector by

$$\boldsymbol{\Psi}^{(j+1)} = \underset{\boldsymbol{\Psi}\in\Omega}{\arg\max}\, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}).$$

The two steps are successively repeated until the absolute or relative changes of the parameter vector reach a predefined limit of tolerance $\epsilon > 0$, respectively

$$\|\boldsymbol{\Psi}^{(j+1)} - \boldsymbol{\Psi}^{(j)}\| < \epsilon.$$

Another considerations suggest to take the increase in the log-likelihood into account for the stopping criterion. McLachlan and Krishnan (2008, p. 142) address the usual EM algorithm stopping criteria as measures for its performance, but not immediately for its convergence. Due to possible multimodality of the log-likelihood function for FMMs, the convergence of the EM algorithm is, among other things, sensitive to starting values. Converging iterations may indicate a lack in progress, but may not guarantee the achievement of a maximum. Böhning et al. (1994) studied a more significant measure of convergence given by the *Aitken-acceleration based stopping criterion* which can be applied to estimates of the log-likelihood with linear order of convergence according to Definition A.2. Let $\ell^{(j)} := \ell(\boldsymbol{\Psi}^{(j)}; \boldsymbol{y})$ denote the approximated values of the log-likelihood function (2.9) with estimates $\boldsymbol{\Psi}^{(j)}$, $j \geq 0$, for the unknown parameter vector $\boldsymbol{\Psi}$ in the jth iteration of the EM algorithm. Let furthermore $\ell^* := \ell(\boldsymbol{\Psi}^*; \boldsymbol{y})$ denote the log-likelihood in the true parameter vector $\boldsymbol{\Psi}^*$. According to Definition A.2, linear convergence of the sequence $\boldsymbol{\Psi}^{(j)}$ implies the relationship

$$|\ell^* - \ell^{(j+1)}| \leq c|\ell^* - \ell^{(j)}|$$

in each iteration step $j \geq 0$ with $0 < c < 1$. Replacing ℓ^* by its approximation yields under linear convergence the expression

$$|\ell^{(j+1)} - \ell^{(j)}| \approx c|\ell^{(j)} - \ell^{(j-1)}| \approx c^j|\ell^{(1)} - \ell^{(0)}|. \tag{2.29}$$

Böhning et al. (1994) point out the fact that a convergence measure concerning the increase in log-likelihood functions does not necessarily imply that the estimate is close to the log-likelihood evaluated at the real value $\ell^* = \ell(\boldsymbol{\Psi}^*; \boldsymbol{y})$ in the case of $c \ll 1$. The following limit value approximates the true value $\boldsymbol{\Psi}^*$,

$$\ell^* = \lim_{j\to\infty} \ell^{(j)} \approx \ell^{(0)} + \sum_{j=0}^{\infty} c^j(\ell^{(1)} - \ell^{(0)}) = \ell^{(0)} + \frac{\ell^{(1)} - \ell^{(0)}}{1-c}.$$

The constant c can be estimated within each iteration $j \geq 0$ according to (2.29) as

$$c^j = \frac{\ell^{(j+1)} - \ell^{(j)}}{\ell^{(j)} - \ell^{(j-1)}}.$$

The *Aitken-accelerated estimate* for the iteration step $(j+1)$ is denoted as

$$\ell_A^{(j+1)} = \ell^{(j)} + \frac{\ell^{(j+1)} - \ell^{(j)}}{1 - c^j} \tag{2.30}$$

and can be applied to the EM algorithm as a stopping criterion. The criterion (2.30) ensures the monotonicity related to the log-likelihood function $\ell_A^{(j)} \geq \ell^{(j)}$. Böhning et al. (1994) suggest to use (2.30) as the stopping criterion by checking the absolute difference

$$|\ell_A^{(j+1)} - \ell_A^{(j)}| < \epsilon.$$

2.4.6 Geometric Properties and Gradient-Based Methods

Lindsay (1983a) derived central theoretical results on the computation of the MLE for an equivalent geometric problem of the mixture model with pdf (2.1). Therefore, the derivation of the MLE is obtained by the optimization on the convex set of all probability measures $\tilde{\Omega}$. The algorithmic idea is to increase the likelihood function of a current estimate P by the addition of a vertex Q. The addition is obtained by a convex combination of the current estimate and the vertex through a step length factor $\alpha \in [0, 1]$. The vertices are defined as measures with mass at K distinct elements $\theta_1, \ldots, \theta_K$ for the probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$. The directional derivative $\Phi(\cdot)$ is used to find a vertex direction, respectively

$$\Phi(P, Q) = \lim_{\alpha \to 0} \frac{\ell((1-\alpha)P + \alpha Q) - \ell(P)}{\alpha}.$$

In case of a positive directional derivative for a vertex Q, the likelihood function can be increased. Otherwise the current estimate is denoted as Nonparametric Maximum Likelihood Estimator (NPMLE) for a flexible number of components K. The algorithmic procedure from Lindsay (1983a, p. 91) was advanced by Böhning et al. (1992) who introduced the software program C.A.MAN (Computer Assisted Mixture Analysis). Applications on the computation of the NPMLE were carried out to different distributions as the Poisson, exponential or normal distribution. The latter affords the separate specification of the variance parameter as Böhning et al. (1992, p. 297) outline. Böhning et al. (1992, p. 293) emphasize the necessity of an adequate step length choice in order to achieve monotonicity within the procedure. In contrast, the EM algorithm guarantees monotonicity as discussed in Section 2.4.2. For a fixed number of components Lindsay (1983a, p. 92) and Böhning et al. (1992, p. 293) refer to the EM algorithm as an appropriate estimation method. In order to improve convergence results, Böhning (2003, p. 260) discusses the EM algorithm with a gradient function update. This advanced method provides monotonicity and convergence to a stationary point as outlined in Böhning (2003, Theorem 1). Empirical evidence on the improvement was given by the application of mixtures of exponential distributions.

As McLachlan and Peel (2000, p. 24) emphasize, the geometric analysis of mixture models provided by Lindsay (1983a) revealed useful results on the general theory of mixture models where the MLE relates to a convex optimization problem. Nevertheless, the fitting of mixtures of GNMs with pdf (2.1) depends highly on the functional relationship

of the data which necessitates the inclusion of explanatory variables. Furthermore, the related problems require a certain number of components due to particular specifics of the underlying model or population. The identification of the mixture components follows in general a distributional shape which is driven by the nonlinear mean function where particular attention will be given to problem specific variance structures allowing for varying dispersion through the data sample and the components. Therefore, for mixtures of GNMs the present analysis focuses on generalizations of already available mixtures of regression models within the statistical environment R with the package **flexmix** where the EM setup evolved as standard methodology.

2.5 Standard Errors

The previous sections give a detailed derivation of the parameter estimation for mixtures of GNMs. The fitted regression coefficient vector $\hat{\boldsymbol{\beta}}_k$ specifies the component specific mean function within the kth mixture component, whereas the estimate for the dispersion parameter $\hat{\phi}_k$ gives information on the scattering of the data classified to the kth component around the mean function for $k = 1, \ldots, K$. The component weights estimates $\hat{\pi}_k$ summarize the overall proportion of the final allocation of the data to the kth component. Studying the quality of the derived mixture of GNMs implies the knowledge on the accuracy of the derived parameter estimates, given through $\hat{\boldsymbol{\Psi}}$. In order to draw conclusions from the fitted mixture of GNMs and assure reliable results, appropriate quality measures for the parameter estimates are necessary which can be pictured by standard errors. The derivation of standard errors for parameter estimates in FMMs poses a major challenge. One approach to face this task is given by the Fisher information as the originator for standard errors of MLEs in FMMs which will be denoted as $\mathcal{I}(\boldsymbol{\Psi}; \boldsymbol{y})$ in the following. As McLachlan and Peel (2000, S. 42) explain, the asymptotic covariance matrix of the MLE $\hat{\boldsymbol{\Psi}}$ equals to the inverse of the Fisher information, respectively

$$\mathrm{Cov}[\hat{\boldsymbol{\Psi}}] \approx \mathcal{I}(\boldsymbol{\Psi}; \boldsymbol{y})^{-1}.$$

The previous sections already indicate tedious computations and low tractability of the log-likelihood function of mixtures of GNMs due to the complexity of the underlying mixture pdf. As the computation of the Fisher information is generally demanding, $\mathcal{I}(\boldsymbol{\Psi}; \boldsymbol{y})$ is often approximated by the observed information matrix $I(\boldsymbol{\Psi}; \boldsymbol{y})$ where

$$I(\boldsymbol{\Psi}; \boldsymbol{y}) = -H(\boldsymbol{\Psi}; \boldsymbol{y}) = -\frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}^\top}$$

holds. $H(\boldsymbol{\Psi}; \boldsymbol{y})$ denotes the Hessian matrix of the log-likelihood function (2.9) comprising the second order derivatives with respect to the parameter vector $\boldsymbol{\Psi}$. Based on these considerations, the approximation of the asymptotic covariance matrix of the MLE $\hat{\boldsymbol{\Psi}}$ is given by

$$\mathrm{Cov}[\hat{\boldsymbol{\Psi}}] \approx I(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})^{-1}.$$

The asymptotic standard errors will be further denoted as $\mathrm{ASE}(\cdot)$ and can be approxi-

mated through

$$\text{ASE}[\hat{\boldsymbol{\Psi}}_r] \approx \sqrt{I(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})^{-1}_{rr}}, \quad r = 1, \ldots, d, \tag{2.31}$$

where d denotes the number of unknown parameters pooled in $\hat{\boldsymbol{\Psi}}$ according to (2.4). The key to the computation of standard errors of MLEs in FMMs is the computation of the Hessian matrix $H(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})$. The covariance matrix results through its inversion. The subsequent example outlines the Hessian matrix for a two-component mixture model.

Example 2.5 (Hessian Matrix for Two-Component Mixture Model of GNMs)
Consider a two-component mixture model with an underlying nonlinear regression function specified by the regression coefficients β_1 and β_2 with pdf

$$f^M(y_i; \mu_i(\boldsymbol{\beta}), \boldsymbol{\phi}, \boldsymbol{\pi}) = \pi_1 \cdot f(y_i; \mu_i(\beta_1), \phi_1) + \pi_2 \cdot f(y_i; \mu_i(\beta_2), \phi_2), \quad i = 1, \ldots, n.$$

The Hessian matrix for the log-likelihood function $\ell(\boldsymbol{\Psi}; \boldsymbol{y})$ is given by

$$H(\boldsymbol{\Psi}; \boldsymbol{y}) = \begin{pmatrix} \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \pi^2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \pi \partial \beta_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \pi \partial \beta_2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \pi \partial \phi_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \pi \partial \phi_2} \\ \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1 \partial \pi} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1^2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1 \partial \beta_2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1 \partial \phi_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1 \partial \phi_2} \\ \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_2 \partial \pi} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_1 \partial \beta_2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_2^2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_2 \partial \phi_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \beta_2 \partial \phi_2} \\ \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_1 \partial \pi} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_1 \partial \beta_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_1 \partial \beta_2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_1^2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_1 \partial \phi_2} \\ \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_2 \partial \pi} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_2 \partial \beta_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_2 \partial \beta_2} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_2 \partial \phi_1} & \frac{\partial^2 \ell(\boldsymbol{\Psi}; \boldsymbol{y})}{\partial \phi_2^2} \end{pmatrix}.$$

The asymptotic standard errors can be derived by evaluating $H(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})^{-1}$ in the MLE $\hat{\boldsymbol{\Psi}}$ as given in (2.31).

Applications show difficulties in the computation of $H(\boldsymbol{\Psi}; \boldsymbol{y})$ if the log-likelihood $\ell(\boldsymbol{\Psi}; \boldsymbol{y})$ is unbounded. As an example, McLachlan and Peel (2000, S. 94) point out that the likelihood of heteroscedastic normal mixtures tends to infinity if the component specific variance tends to zero in at least one component. This possibility was previously outlined in Example 2.1. Another important aspect, when estimating standard errors within FMMs, is given by the sample size and the component separation. McLachlan and Peel (2000, S. 42) emphasize the importance of large sample sizes in order to obtain valid and accurate standard errors. The discussed method produces reliable standard errors in the case of a good separation of the mixture component means as Basford et al. (1997) outline.

The inverse of the Fisher information matrix $\mathcal{I}(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})^{-1}$ suits as an appropriate method to obtain the asymptotic covariance matrix and thus the standard errors of the MLE $\hat{\boldsymbol{\Psi}}$. Furthermore, the Fisher information is often replaced by the observed information matrix $\text{I}(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})$ which equals the negative of the Hessian matrix. Nevertheless, the direct computation of the second order derivatives of the incomplete log-likelihood $\ell(\boldsymbol{\Psi}; \boldsymbol{y})$ turns out remarkably difficult within FMMs. Applying the EM algorithm shifts these difficulties to the complete-data log-likelihood $\ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)$ and its conditional expectation $\text{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)})$. In addition to these difficulties, the EM algorithm provides in fact no estimates for the covariance matrix within its iteration scheme. McLachlan and Krishnan (2008, p. 105)

take on this major disadvantage and discuss methods to obtain the covariance matrix within the EM algorithm, mostly built on the observed information matrix.

2.5.1 Score Statistics and Missing Information

Let $S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)$ denote the score vector for the complete-data log-likelihood function $\ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)$ with respect to the complete-data vector $\boldsymbol{y}^c$, respectively

$$S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) = \frac{\partial \ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)}{\partial \boldsymbol{\Psi}}.$$

McLachlan and Krishnan (2008, pp. 95-96) highlight the relationship between the incomplete data score vector $S(\boldsymbol{\Psi}; \boldsymbol{y})$ and the complete-data score vector by taking the conditional expectation of the complete-data score as

$$S(\boldsymbol{\Psi}; \boldsymbol{y}) = \mathrm{E}[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)|\boldsymbol{y}, \boldsymbol{\Psi}]. \tag{2.32}$$

The observed information of the complete-data log-likelihood function is given by its negative Hessian matrix, respectively

$$\mathrm{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) = -\frac{\partial^2 \ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}^\top}.$$

Taking up on the relationship of the log-likelihood functions in (2.13), differentiating both sides two times with respect to the parameter vector $\boldsymbol{\Psi}$ yields the representation

$$\mathrm{I}(\boldsymbol{\Psi}; \boldsymbol{y}) = \mathrm{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) - \frac{\partial^2 \log g(\boldsymbol{y}^c|\boldsymbol{y}; \boldsymbol{\Psi})}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}^\top}. \tag{2.33}$$

Taking the conditional expectation of Equation (2.33) results in

$$\mathrm{I}(\boldsymbol{\Psi}; \boldsymbol{y}) = \mathcal{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}) - \mathcal{I}^m(\boldsymbol{\Psi}; \boldsymbol{y}). \tag{2.34}$$

The first expression on the right sight of Equation (2.34), $\mathcal{I}^c(\boldsymbol{\Psi}; \boldsymbol{y})$, corresponds to the conditional expectation of the complete-data observed information given the incomplete-data vector $\boldsymbol{y}$, respectively

$$\mathcal{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}) = \mathrm{E}[\mathrm{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)|\boldsymbol{y}, \boldsymbol{\Psi}]. \tag{2.35}$$

The second term in (2.34) denotes the expected information matrix for $\boldsymbol{\Psi}$ conditional on the missing-data vector,

$$\mathcal{I}^m(\boldsymbol{\Psi}; \boldsymbol{y}) = -\mathrm{E}\left[\frac{\partial^2 \log g(\boldsymbol{y}^c|\boldsymbol{y}; \boldsymbol{\Psi})}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}^\top}\Big|\boldsymbol{y}, \boldsymbol{\Psi}\right].$$

The expected information matrix $\mathcal{I}^m(\boldsymbol{\Psi}; \boldsymbol{y})$ is denoted as *missing information* as McLachlan and Krishnan (2008, p. 96) point out.

2.5.2 Louis' Method for Standard Error Computation

An important result for computing standard errors within the EM framework is given by Louis (1982). Within the scope of his work, Louis (1982) presents a procedure for extracting the observed information matrix $\mathrm{I}(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})$ when using the EM algorithm for ML fitting within missing information problems. The method requires particularly the computation and the gradient or the second order derivatives of the complete-data log-likelihood function. Louis (1982) derives an expression for the missing information $\mathcal{I}^m(\boldsymbol{\Psi}; \boldsymbol{y})$ in terms of the complete-data log likelihood, as

$$\begin{aligned}
\mathcal{I}^m(\boldsymbol{\Psi}; \boldsymbol{y}) &= \mathrm{Cov}(S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)|\boldsymbol{y}, \boldsymbol{\Psi}) \\
&= \mathrm{E}\left[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)^\top \middle| \boldsymbol{y}, \boldsymbol{\Psi}\right] \\
&\qquad -\mathrm{E}[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)|\boldsymbol{y}, \boldsymbol{\Psi}]\mathrm{E}[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)^\top|\boldsymbol{y}, \boldsymbol{\Psi}] \\
&\overset{(2.32)}{=} \mathrm{E}\left[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)^\top \middle| \boldsymbol{y}, \boldsymbol{\Psi}\right] - S(\boldsymbol{\Psi}; \boldsymbol{y}) S(\boldsymbol{\Psi}; \boldsymbol{y})^\top. \qquad (2.36)
\end{aligned}$$

The problem of computing the second order derivative of the incomplete-data problem is reduced to the computation of the first order derivatives of the complete-data log-likelihood function used in the EM algorithm. Furthermore, the information matrix for the missing information is solely based on the complete-data log-likelihood. Reformulating Equation (2.34) leads to

$$\mathrm{I}(\boldsymbol{\Psi}; \boldsymbol{y}) \overset{(2.36)}{=} \mathcal{I}^c(\boldsymbol{\Psi}; \boldsymbol{y}) - \mathrm{E}\left[S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c) S^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)^\top \middle| \boldsymbol{y}, \boldsymbol{\Psi}\right] + S(\boldsymbol{\Psi}; \boldsymbol{y}^c) S(\boldsymbol{\Psi}; \boldsymbol{y})^\top. \quad (2.37)$$

Louis (1982) provides an essential contribution to the calculation of standard errors as he proves that the observed information matrix $\mathrm{I}(\boldsymbol{\Psi}; \boldsymbol{y})$ can be computed by means of the conditional expectation of the first and second order derivatives of the complete-data log-likelihood function.

2.6 Extensions of the EM Algorithm

The EM algorithm represents a popular method for iteratively maximizing complex log-likelihood functions. The reformulation of a computationally tractable complete-data log-likelihood function (2.12) represents its main advantage. Possible modifications of the original EM algorithm aggravate in general its simple form. Additionally, the original EM algorithm still comprises some practical difficulties particularly when dealing with FMMs. Due to the multimodal characteristics of FMMs, the EM algorithm may converge to local solutions and therefore afford further computations. It is recommended to execute the algorithm repeatedly for different starting values in order to improve the chance to obtain a global maximum. The EM algorithm is thus sensitive to the choice of the starting values and may provide slow convergence. Different approaches for speeding up the convergence of the EM algorithm have been introduced, as McLachlan and Krishnan (2008, p. 105) summarize. The following two extensions indicate the speeding up of the EM algorithm. Within this context, the original algorithm is modified which results either in the CEM algorithm or the SEM algorithm.

2.6.1 Classification-Expectation-Maximization (CEM) Algorithm

Celeux and Govaert (1992) introduce a general classification EM algorithm based on a partitioning criterion. Unlike other extensions of the original EM algorithm, the CEM algorithm represents a deterministic version of the EM algorithm. The basic idea builds on finding the optimal partition for each observation y_i where the partition corresponds to the already defined components within FMMs, as outlined in Section 2.1. The observations y_i are deterministically assigned to their optimal partition in every iteration step based on an initial setting of K components $\mathcal{P}^{(0)} = (\mathcal{P}_1^{(0)}, \ldots, \mathcal{P}_K^{(0)})$. The deterministic approach incorporates to replace the computation of the missing information

$$\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iK})^\top$$

through a deterministic assignment of the missing information indicators. Therefore the posterior probabilities $w_{ik}^{(j)}$, resulting from the E-step for a current estimate $\boldsymbol{\Psi}^{(j)}$, are used to classify the observations into K components. Observation y_i is assigned to the respective component providing the maximum posterior weight. The CEM algorithm differs technically from the EM algorithm through a classification step between the already introduced E- and M-steps.

Steps of CEM-Algorithm:

E-step:
Calculate the posterior probabilities $w_{ik}^{(j)}$.

Classification (C)-step:
Assign each observation y_i to that component with the highest posterior probability $w_{ik}^{(j)}$,

$$z_{ik}^{(j+1)} = \underset{l=1,\ldots,K}{\arg\max}\, w_{il}^{(j)} \qquad \forall i = 1, \ldots, n \Longrightarrow y_i \in \mathcal{P}_k$$

For equal values $w_{il}^{(j)} = w_{ih}^{(j)}$, $l \neq h$, the smaller index $\min(h, l)$ is chosen.

M-step:
Compute the subsequent update for the unknown parameter vector by using the classification indicators $z_{ik}^{(j+1)}$ from the C-step by

$$\boldsymbol{\Psi}^{(j+1)} = \underset{\boldsymbol{\Psi} \in \Omega}{\arg\max}\, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}).$$

2.6.2 Stochastic-Expectation-Maximization (SEM) Algorithm

Celeux and Diebolt (1985) present the SEM algorithm as an useful method for speeding up the EM algorithm and handling intractable calculations within the E-step. In doing so they replaced in general the computation of the E-step by a Monte Carlo (MC) simulation. The main idea was to replace the missing data by already observable information and the current estimate for the unknown parameter vector $\boldsymbol{\Psi}$, as McLachlan and Krish-

nan (2008, p. 227) explain. With the focus on the missing information

$$\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iK})^\top$$

associated to each observation y_i, they replaced the computation of the posterior weights (2.16) by a simulation-based computation. The random vectors $\boldsymbol{z}_i$ are assumed to be multinomially distributed with K categories and the mixing proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top$ as success probabilities according to the incomplete-data problem, as postulated in Section 2.4.1. In the SEM algorithm the missing data vector $\boldsymbol{z}_i$ is drawn from the current conditional distribution of $\boldsymbol{z}_i$ in order to be used for the calculation of the posterior probabilities. This step assigns each observation y_i to exactly one component $k = 1, \ldots, K$.

For each $i = 1, \ldots, n$ the parameter $z_{ik}^{(j)}$ is drawn from a multinomial distribution with K categories and success probabilities given by the vector of component weights $\boldsymbol{\pi}$. The simulated values are used for further calculations in the M-step in the same way as the deterministically classified data in the CEM algorithm.

Steps of SEM-Algorithm:

E-step:
Calculate the posterior probabilities $w_{ik}^{(j)}$.

Stochastic (S)-step:
Draw for each observation y_i a multinomially distributed classification vector $\boldsymbol{z}_i^{(j)}$ based on the current estimate of the mixing proportions $\hat{\boldsymbol{\pi}}^{(j)}$.

M-step:
Compute the subsequent update for the unknown parameter vector by using the simulation-based posterior probabilities $z_{ik}^{(j)}$ by

$$\boldsymbol{\Psi}^{(j+1)} = \underset{\boldsymbol{\Psi} \in \Omega}{\arg\max}\, \mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(j)}).$$

2.7 Number of Components

When applying FMMs, the choice of a suitable number of components K arises as a fundamental question since the whole fitting procedure is carried out for a fixed K. In this sense it is desirable to have decision making methods for a proper choice of the right number K. Straightforward analysis suggests the number of modes as an indicator for an appropriate number of components in a FMM but as some modes may not be well separated, this method can distort the true number of components. In specific cases prior knowledge on possible groups may be available and give an indication on a suitable number of components in the underlying FMM. Furthermore, some data sets apply to a known population comprising distinct groups which may correspond to the mixture components. If none of these cases on prior knowledge occurs or the existence of further

components for latent groups is presumed, the final choice of K refers to appropriate tests. Before addressing further discussion on the number of components, the order of a FMM is defined.

Definition 2.1 *Order of a Finite Mixture Model, see McLachlan and Peel (2000, p. 177) The true order K_0 of a FMM with mixture pdf*

$$f^M(y;\boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k f(y;\mu(\boldsymbol{\beta}_k),\phi_k)$$

corresponds to the smallest K such that all mixture components $f(y;\mu(\boldsymbol{\beta}_k),\phi_k)$ differ and have nonzero component weights π_k for $k = 1,\ldots,K$.

Definition 2.1 indicates to assess the number of components K as small as possible but still compatible with the data set.

Approaches for tests on the number of components are based on the likelihood function. A possibility to derive appropriate tests is to penalize the log-likelihood function by the subtraction of a model-dependent penalization term as McLachlan and Peel (2000, p. 184) outline. The penalized log-likelihood can be expressed as

$$-2\ell(\hat{\boldsymbol{\Psi}};\boldsymbol{y}) + 2C. \tag{2.38}$$

The penalty term C measures the model complexity as it comprises often the total number of model parameters. The penalized log-likelihood in (2.38) represent the base for information criteria on assessing the number of components in a FMM.

2.7.1 Model Selection Criteria

Based on the penalized log-likelihood function in (2.38), different model selection criteria have been derived. These are used in order to determine the order in FMMs as well as to choose between different models. Further analysis refers to the following often used information criteria in model selection regarding FMMs. For further discussions reference is made to McLachlan and Peel (2000).

2.7.1.1 Akaike Information Criterion (AIC)

The Akaike Information Criterion sets the penalty term in (2.38) equal to the total number of parameters in the FMM, which is denoted by d. Therefore, it suggests to choose the order K which minimizes

$$\mathrm{AIC} = -2\ell(\hat{\boldsymbol{\Psi}};\boldsymbol{y}) + 2d \tag{2.39}$$

with MLE $\hat{\boldsymbol{\Psi}}$. As McLachlan and Peel (2000, p. 203), outline the AIC is often applied to FMMs although it unfortunately tends to overestimate the total number of components.

2.7.1.2 Bayesian Information Criterion (BIC)

The Bayesian-based information criterion BIC can also be adapted to FMMs. The penalty term in (2.38) equals twice the total number of parameters multiplied by the logarithm of the sample size, that is

$$\mathrm{BIC} = -2\ell(\hat{\boldsymbol{\Psi}}; \boldsymbol{y}) + d \log n. \tag{2.40}$$

Compared to the AIC, the BIC proves as more reliable model selection criteria according to McLachlan and Peel (2000, p. 209) despite some weaknesses.

2.7.1.3 Integrated Classification Likelihood (ICL) Criterion

In the context of the EM algorithm, classification-based information criteria have been adapted for model comparison when dealing with FMMs. As the EM algorithm maximizes the expected complete-data log-likelihood $\ell^c(\boldsymbol{\Psi}; \boldsymbol{y}^c)$, the main idea for classification-based information criteria is to penalize its expectation $\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^*)$ instead of the original log-likelihood function. $\boldsymbol{\Psi}^*$ denotes the MLE derived by the EM algorithm. The penalized complete-data log-likelihood function is then given as

$$\mathrm{ICL} = -2\,\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^*) + 2C. \tag{2.41}$$

with penalization term C. Taking the same penalization term as in the BIC leads to the information criterion

$$\mathrm{ICL} = -2\,\mathrm{Q}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^*) + d \log n. \tag{2.42}$$

which is denoted as ICL-BIC criterion in McLachlan and Peel (2000, p. 217) but will be later referred to as ICL criterion following Grün and Leisch (2008b, p. 6). Empirical comparisons between different information criteria in McLachlan and Peel (2000, p. 220) show a reliable performance in choosing the correct number of components in FMMs for the ICL.

CHAPTER 3

Mixtures of Generalized Nonlinear Models in R

Introduction

The acronym R denotes mainly a programming language for statistical analysis and graphics. It incorporates furthermore a software environment as open source project (GNU) and is therefore freely available. R belongs also to the S language developed by *John Chambers*. Its advanced version S4 plays an important role within this work and will be addressed in more detail in the next section.

R is basically known for its various statistical modeling methods and its graphical exploration tools but can be applied to any kind of data analysis. The overall R design allows for extensions and modifications of already available functionalities. The platform Comprehensive R Archive Network (CRAN) provides a network for storing and exchanging current packages and codes with access to databases containing available packages. CRAN furthermore provides different manuals and documentations for R users. Users contribute to the software system by distributing their own program codes as packages. Main contributors form the international R core team include the initial R developers *Robert Gentleman* and *Ross Ihaka*. The CRAN team maintains the R source code as well as the CRAN platform.

The free software R has established itself to a frequently used programming language within different fields of data analysis. The number of more than 12 900 currently available packages reflects its extent of popularity. The nature of these packages covers various disciplines. Due to its use for big data analysis, R ranks among the top 10 most popular programming languages in recent years evaluated on Spectrum (2018).

3.1 S4 Language in R

According to Venables and Ripley (2000), S declares a functional language with elementary commands consisting of expressions or assignments. The basic elements are *objects*. For instance, objects serve as arguments and return values for functions. This buttress the first observation about the S language according to Venables and Ripley (2000, p. 23):

"Everything in S is an object."

In S, calculations are typically separated from results and outputs. Intermediate results are stored as objects and can be printed as various summaries for the user at the end of the program. The specific appearance and information content of outputs are determined by *classes*. They serve as a framework for the structure of objects to which they are assigned to. This completes the second observation about S:

"Every object has a class."

The S language comprises the two object systems S3 and S4 which are supported by R. While the concept of S3-programming is familiar to the majority of R users because of its simplicity, S4 is less attractive due to a more formal and rigorous programming style. The increasing complexity in S4 is due to a class-oriented programming style, comparable to Object-Oriented Programming (OOP). The use of classes and their definitions are essential in S4. Class definitions contain primary the definition and specification of various slots in objects. Common characteristics of packages written in OOP are complicated hierarchical structures due to a large number of class-linked objects. Hierarchical structures indicate class inheritance. Therefore, inheriting classes take already defined slots from their parent classes and contain information about the inheritance structure. Computations on objects are provided by functions. A generic function represents an overall definition of a function. It allows the selection of different methods according to the class of the objects within function calls. Generic functions allow different computations depending on the transferred function arguments. Therefore methods serve as a link between classes and functions. Generic functions contain by all means the function arguments used by selected methods. The individual methods differ in their *signature* due to a class assignment. The signature specifies the necessary and specific arguments needed for the workflow and execution of the assigned methods. In general methods do not belong to specific classes but to generic functions for which they are defined.

3.2 Generalized Nonlinear Models in R

Nonlinear regression analysis is emphasized by different fields of research as already discussed in Section 1.1. Nevertheless, its realization is unbalanced as applications of nonlinear regression models are often reduced to nonlinear least squares for normally distributed responses given by the RSS in (2.27). For this reason, nonlinear regression analysis has not been widely advanced regarding its implementation as it is the case for linear regression models and its extensions, with special focus on GLMs. The aim of this section is to give a general overview of the available functions and packages in

R for nonlinear regression models based on the theoretical consideration mentioned in Section 1.1.

According to the assumptions made in Section 1.1, nonlinear regression models require a numerical fitting procedure for obtaining estimates for the regression coefficients. Deciding on an appropriate nonlinear regression function implies that no simpler model with less regression coefficients fits the given data better. The different numerical approaches have in general the same problems in common. A proper choice of starting values states often a necessary precondition in order to achieve convergence of the algorithm. In practice, starting values are set by a heuristic choice and possibly close to the true values. The decision may be based on previous graphical exploration of the given data set or simply build on the meaning of the regression coefficients. Advanced methods use precalculations based on grids for the regression coefficients for choosing proper starting values. Taking all these steps into account may still lead to algorithms converging to local optima. The user has still to provide a global optimal solution.

The following description distinguishes between two fitting commands in R. Since the WNLS method has emerged as standard fitting procedure, the corresponding implementation in R will be addressed separately in Section 3.2.1 based on the detailed work of Ritz and Streibig (2008). Section 3.2.2 relates to a fitting procedure based on the theoretical considerations on GNMs which are discussed in Chapter 1. The technical details are provided by Turner and Firth (2018) who developed the respective package.

3.2.1 The Function `nls()` in R

Nonlinear regression analysis is basically provided by the function `nls()` in R. The function `nls()` is available within the standard package **stats** in R, see also R Core Team (2018). Ritz and Streibig (2008) provide a detailed explanation of this function. They furthermore demonstrate accompanying examples to serve for a better explanation of the use of `nls()` for the fitting of nonlinear regression models. While Ritz and Streibig (2008) cover the use of available methods in R, the theoretical concept and background of nonlinear regression can be found in the standard literature like Bates and Watts (1988) and Seber and Wild (2003).

The main command `nls()` for fitting nonlinear regression models comprises the input arguments given in Table 3.1 which represents no closed list due to further optional arguments. The commonly used nonlinear fitting function `nls()` relates the nonlinear regression problem to a LS estimation, also known as the WNLS approach given in Example 2.1. The WNLS algorithm corresponds to the weighted `nls()` procedure with given weights as described in Ritz and Streibig (2008, p. 85). For solving the nonlinear LS problem, `nls()` provides (three) different numerical approaches. The `Gauss-Newton` method represents the default method for minimizing LS problems. Often nonlinear regression models comprise linear regression coefficients, also known as conditionally linear parameters, see Section 1.1 or Bates and Watts (1988, p. 36). `nls()` takes advantage of this fact by fitting the conditionally linear regression coefficients with the well-known function for linear regression models `lm()`, as Ritz and Streibig (2008, p.

Arguments	Explanation/Functionality
`formula`:	generic function for the nonlinear regression model formula
`data`:	data frame containing predictor and response variables
`start`:	list of starting values for the unknown regression coefficient vector
`algorithm`:	default fitting method `"Gauss-Newton"`, optional specifications `"plinear"` and `"port"` available
`control`:	manual settings for controlling estimation method, see Ritz and Streibig (2008, p. 52)
`trace`:	option for displaying intermediate estimates
`weights`:	optional slot for weights, according to estimation problem (2.27)

Table 3.1: Arguments in `nls()`

41) explain. This reduces the numerical effort for fitting the remaining nonlinear regression coefficients. `nls()` contains an optional numerical fitting method based on these conditions. It can be accessed by setting `algorithm="plinear"`. Another optional algorithm, specified by setting `algorithm="port"`, enables the fitting of nonlinear regression parameters with constrained regression coefficients. Ritz and Streibig (2008, p. 38) discuss additional arguments for `nls()`. Its detailed explanation goes beyond the scope of this work. Special focus is placed on fitting methods for further distributional families as the next section shows.

3.2.2 The Package gnm in R

Theoretical considerations emphasize the use of nonlinear regression models for distributional families beyond the normal distribution for the response. The basic model class is given in Chapter 1 with reference to the main literature provided by Wei (1998). Turner and Firth (2018) developed the corresponding package **gnm** in R with the purpose of providing a framework for GNMs. With the appearance of this package, the repertoire of already existing packages and functions in R has been extended by a method for fitting nonlinear regression models with responses stemming from the exponential family.

For linear regression coefficients within nonlinear regression models, Turner and Firth (2018) embedded the well-known procedures `glm()` and `lm()` as underlying fitting procedures. In fact, **gnm** was designed and inspired by the framework provided by `glm()` regarding input arguments, return objects or the compatibility of different accessor functions, see Turner and Firth (2018, p. 17). Therefore it can be viewed as its analogon for fitting nonlinear models. The central fitting procedure is executed by the command `gnm()` requiring the necessary arguments `formula` and `family`. It comprises further optional arguments for controlling the underlying fitting procedure. Some of them are summarized in Table 3.2.

The results based on `gnm()`, obtained by setting the distributional argument to `family="gaussian"`, coincide with those provided by the fitting procedure `nls()` for normally distributed responses which will be illustrated at a later point in this work. Accord-

Arguments	Explanation/Functionality
`formula`:	nonlinear regression model formula
`data`:	predictor and response variables combined in a data frame
`family`:	specification of error distribution and link function
`start`:	initialization of starting values as list for the unknown regression parameters
`tolerance`:	threshold as stopping criterion for numerical procedure, see Turner and Firth (2018, p. 10)
`iterMax`:	maximum number of total iterations
`verbose/trace`:	indicator to print intermediate results during fitting procedure
`trace`:	indicator whether or not to print intermediate results during fitting procedure
`weights`:	optional slot for weights

Table 3.2: Arguments in `gnm()`

ingly, `gnm()` extends the already known method `nls()` by allowing for further distributions specified by the argument `family` in the same way as the fitting procedure `glm()` extends `lm()` for the class of GLMs. The package **gnm** offers a wide range of applications whose detailed explanation exceeds the scope of this work. The following section focuses on the description of the necessary procedures for enabling the implementation of mixtures of GNMs in R.

3.2.2.1 Specification of Nonlinear Functions in gnm

The package **gnm** provides predefined functions for nonlinear terms in order to specify the nonlinear model formula. These functions are compatible with the generic function `formula` in R. Turner and Firth (2018, p. 6) present a list of various mathematical functions expressing simple mathematical relationships as well as a general specification of symbolic functions. The symbolic specification of nonlinear functions affords the `formula` as an object of the internal class `"nonlin"`. The generic function `formula` includes the regression function with the use of symbolic language as parsed arguments. Next to predefined nonlinear functions it is possible to construct individual nonlinear terms as part of the formula argument. Therefore, at least the following parts in Table 3.3 have to be specified. Further arguments are also available but are not of major importance within this work, see Turner and Firth (2018, p. 8) as reference for further details.

Arguments	Explanation/Functionality
`predictors`:	list of (possibly nonlinear) regression coefficients
`variables`:	list comprising the symbolic expression for the explanatory variables
`term`:	parsed form of functional relationship between `predictors` and `variables` as input arguments, RHS of `formula` argument

Table 3.3: Arguments for symbolic functions of class `"nonlin"` in `gnm()`

A further step attaches the three arguments in Table 3.3 as a list and denotes it as a function. The function is assigned as object of class `"nonlin"` which is a well-defined input argument for the fitting procedure `gnm()`. Despite their similarities, `gnm()` differs crucially from `glm()` in the specification of the nonlinear regression function due to its nonlinear terms.

3.2.2.2 Application: The Michaelis-Menten Model (Example 1.1 continuation)

Example 1.1 addresses the Michaelis-Menten model presented by Bates and Watts (1988, p. 33). The Michaelis-Menten model serves as basic regression model in this section with the aim to illustrate the previously discussed nonlinear procedures `nls()` and `gnm()`. According to Example 1.1, the Michaelis-Menten model is linearizable. Therefore a transformation of the originally nonlinear problem yields the well-known linear regression structure and enables the use of the fitting procedure `glm()`. The variable `rate` corresponds to the enzymatic reactions $\boldsymbol{y}$ while variable `conc` relates to the concentration $\boldsymbol{x}$ in Example 1.1.

Applying the nonlinear fitting command `nls()` to the Michaelis-Menten model requires the command in the first line of Listing 3.1 and leads to estimates $\hat{\boldsymbol{\beta}} = (126.03; 17.08)$ in lines 9 and 10.

```
1  > mm.1 = nls(rate ~ a*conc/(b+conc),data=L.minor, start=list(a=120,b=20))
2
3  > summary(mm.1)
4
5  Formula: rate ~ a * conc/(b + conc)
6
7  Parameters:
8    Estimate Std. Error t value Pr(>|t|)
9  a  126.033      7.173  17.570 2.18e-06 ***
10 b   17.079      2.953   5.784  0.00117 **
11 ---
12
13 Residual standard error: 6.25 on 6 degrees of freedom
14
15 Number of iterations to convergence: 7
16 Achieved convergence tolerance: 8.152e-06
```

Listing 3.1: Fitting the Michaelis-Menten model with `nls()`

In the following, the original nonlinear Michalis-Menten model is fitted utilizing the fitting procedure `gnm()` for GNMs. The specification of the nonlinear regression function is given in lines 3 to 10 of Listing 3.2 while it is assigned as object of class `"nonlin"` in line 11. The fitting is provided by performing the function `gnm()` in lines 13 to 15. A short overview on important results, provided by the function `summary`, yields to the same results as in `nls()` for the regression coefficients. `gnm()` reports additional values like the dispersion, deviance and the AIC in lines 33 to 36.

```
1  > library(gnm)
2
3  > mmm = function(x,predictors){
4  +   list(predictors=list(a=1,b=1),
```

```
+         variables = list(substitute(x)),
+         term = function(predictors, variables) {
+           paste(predictors[1],"*",variables[1],"/(",predictors[2],"+",
+                 variables[1],")",sep="")
+         })
+ }
> class(mmm) ="nonlin"

> mm.3 = gnm(formula = rate ~ -1 + mmm(conc), data = L.minor,
+            start = c(a=120,b=20),
+            family  = gaussian(link = "identity"),trace=TRUE)

> summary(mm.3)

Call:
gnm(formula= rate ~ -1 + mmm(conc), family = gaussian(link = "identity"),
    data = L.minor, start = c(a = 120, b = 20), trace = TRUE)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-7.403  -4.663  -2.007   2.741   8.304

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
a  126.033      7.173  17.570 2.18e-06 ***
b   17.079      2.953   5.784  0.00117 **
---

(Dispersion parameter for gaussian family taken to be 39.05885)

Residual deviance: 234.35 on 6 degrees of freedom
AIC: 55.722

Number of iterations: 10
```

Listing 3.2: Fitting the Michaelis-Menten model with `gnm()`

Example 1.1 transforms the original nonlinear model into a linear regression model with modified regression coefficients $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_0, \tilde{\beta}_1) = (\beta_0^{-1}, \beta_1/\beta_0)$. Executing the `glm()` command provides the fitting of the linear regression model. The resulting regression coefficients are given in lines 16 to 17 of Listing 3.3 as $\hat{\tilde{\boldsymbol{\beta}}} = (0.01, 0.14)$.

```
> mm.2 = glm(rate ~ I(1/conc), data=L.minor,
+            family=gaussian(link="inverse"))

> summary(mm.2)

Call:
glm(formula = rate ~ I(1/conc), family = gaussian(link = "inverse"),
    data = L.minor)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-7.403  -4.663  -2.007   2.741   8.304

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0079344  0.0004516  17.570 2.18e-06 ***
I(1/conc)   0.1355123  0.0173574   7.807 0.000233 ***
---
```

```
19
20 (Dispersion parameter for gaussian family taken to be 39.05854)
21
22     Null deviance: 9427.92  on 7  degrees of freedom
23 Residual deviance:  234.35  on 6  degrees of freedom
24 AIC: 55.722
25
26 Number of Fisher Scoring iterations: 7
```

Listing 3.3: Fitting the linearized Michaelis-Menten model with `glm()`

Re-transforming the fitted coefficients obtained by executing `glm()` enables a direct comparison to the regression coefficients fitted by `nls()` and `gnm()`. Therefore the regression coefficients $\tilde{\boldsymbol{\beta}}$ are transformed to their original form $\boldsymbol{\beta} = (\tilde{\beta}_0^{-1}, \tilde{\beta}_1/\tilde{\beta}_0)$ as shown in Listing 3.4. Comparing the regression coefficients fitted by `nls()`, `glm()` and `gnm()` shows non-significant differences after the second decimal point due to different underlying fitting methods. The results for the regression coefficients in lines 3, 5 and 8 of Listing 3.4 equal for all three models as well as the values for the log-likelihood functions and the deviance.

```
1 > coef(mm.1); c(1/coef(mm.2)[1],coef(mm.2)[2]/coef(mm.2)[1]); coef(mm.3)
2         a          b
3  126.03276  17.07899
4 (Intercept)   I(1/conc)
5  126.03276    17.07899
6 Coefficients:
7        a          b
8  126.03286  17.07904
9
10 > logLik(mm.1); logLik(mm.2); logLik(mm.3)
11 'log Lik.' -24.86106 (df=3)
12 'log Lik.' -24.86106 (df=3)
13 'log Lik.' -24.86106 (df=3)
14
15 > deviance(mm.1); deviance(mm.2); deviance(mm.3)
16 [1] 234.3531
17 [1] 234.3531
18 [1] 234.3531
```

Listing 3.4: Comparison of the estimates for the Michaelis-Menten model in R

3.3 Finite Mixture Models in R

When combing through the list of 12 930 available packages in R, mixture models appear regularly. The majority of the underlying packages works with the normal distribution as assumption. Exemplary, the package **mclust** deals with mixture modeling for normally distributed data as well as the package **mixture**. For detailed information on the package **mclust** reference is made to Scrucca et al. (2016) whereas further information on **mixture** is given by Browne et al. (2018), Browne and McNicholas (2014) and Celeux and Govaert (1995). The package **mixdist** allows for fitting multimodal distributions for grouped data, see also Macdonald and with contributions from Juan Du (2018). The package **mixtools** enables analyzing and fitting mixture densities and mixtures of regressions for linear dependence structures as denoted in Benaglia et al. (2009). Further

extensions for the modeling of mixtures of regressions are provided by **mixreg** for linear regression models including Gaussian mixture models as described by Turner (2018) and extensions for censored data provided by **CensMixReg**, see also Sanchez et al. (2018). Some extensions of FMMs already tend to generalize the distribution but with the focus on one specific distribution. Nonlinear regression models can be handled with the package **nlsmsn** for mixtures of skew normal distributions allowing for skewness in the data as Garay et al. (2013) outline. A noticeable group of packages deals with Bayesian approaches and hierarchical structures or is designed for the analysis of gene expression data. A complete list of available packages can be accessed online at the CRAN repository given by CRAN (2018). A very popular package is given by **flexmix** which allows for flexible mixture modeling of GLMs among other models. **flexmix** represents furthermore the main package of interest within this work due to its advantages which will be addressed in the next section.

3.3.1 The Package flexmix in R

Following the idea of extensibility and flexibility, *Bettina Grün* and *Friedrich Leisch* developed the R package **flexmix** for dealing with FMMs. While previous packages mainly covered specific mixture model cases **flexmix** is able to deal with model-based clustering as well as with mixtures of regressions. These represent a key aspect in **flexmix** as it offers for the first time a model framework for mixtures of GLMs.

The following technical (implementation) details serve as explanation of the functionality and the structure of the package. The main explanations of the developers can be found in Leisch (2004b), Grün and Leisch (2007) and Grün and Leisch (2008b).

3.3.1.1 Framework and Basic Commands

The package **flexmix** provides a broad toolkit for fitting mixture models in R. For the last years the functionality has been extended covering efficient estimation techniques for mixture models, model diagnostics and (appropriate) visualization techniques of the results. The package structure is modular as it is based on the S4 class-oriented programming language in R.

The main command for fitting FMMs is `flexmix()`. The fitting procedure is based on a ML estimation through the EM algorithm as described in Section 2.4. `flexmix()` itself contains an overall model formula and function calls specifying the component specific models and the concomitant variable model as arguments as well as a fixed number of components. The separate vector `cluster` represents a placeholder for the initial component membership. If left empty, the default procedure determines a random labeling to the predefined number of components as starting configuration. For different numbers of components the package offers the function `stepFlexmix()` enabling the user to fit mixture models repeatedly with `flexmix()`. This procedure is of interest when comparing fitted mixture models for different numbers of components and in order to prevent choosing a local maximum by comparing different results. The user determines the best result since the fitted models are stored in an object named *models*. The model

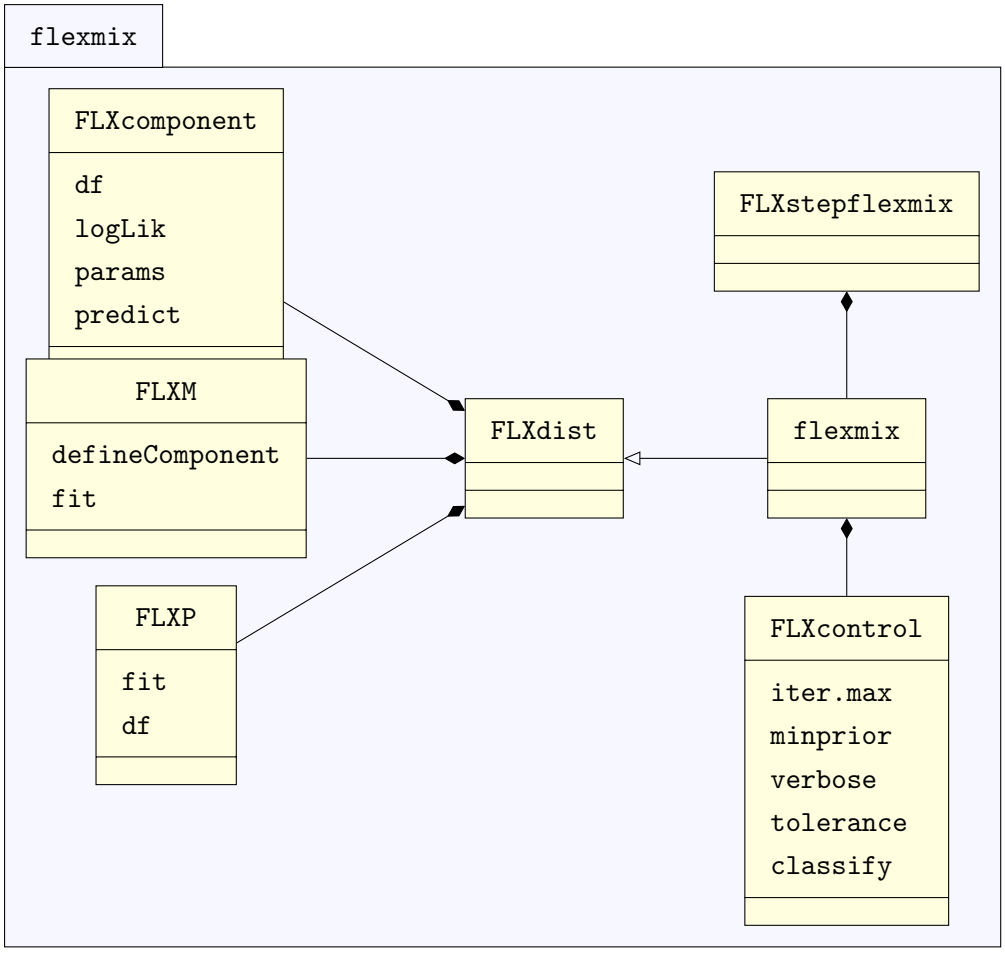

Figure 3.1: UML class diagram for **flexmix**

comparison is possible through model selection criteria given by the AIC, BIC and ICL whose determinations are provided in Section 2.7.1.

The **flexmix** package is based on the class structure given in the UML diagram in Figure 3.1. The return value is an object of a class `"flexmix"` and can be accessed through the function `flexmix()`. As Leisch (2004b, p. 22) explains, the class `"flexmix"` extends the class `"FLXdist"`, also visible in Figure 3.1. It contains the fitted mixture by the EM algorithm which is specified by an object of class `"FLXcontrol"`. The user has impact on the underlying EM algorithm by the class `"FLXcontrol"`. As Figure 3.1 shows, `"FLXcontrol"` defines a list of parameters for the numerical calculation with the help of the slots `iter.max`, `verbose`, `minprior` and `tolerance`. The slot `iter.max` specifies a cap for the number of iterations for the EM algorithm whereas `verbose` serves as a trace method for the results by printing the log-likelihood at every `verbose`-th step. The threshold for removing prior probabilities, as proposed by Leisch (2004b, p. 4), is fixed by `minprior`. The slot `tolerance` controls the increase of the log-likelihood function and the iteration procedure stops if the convergence criterion falls below the given tolerance level. Choosing from available variants of the EM algorithm is enabled by the slot `classify`. Available methods are the original EM, the CEM and the SEM algorithm. The default initialization in `flexmix()` is given by Listing 3.5.

```
> new("FLXcontrol")
An object of class "FLXcontrol"
Slot "iter.max":
[1] 200

Slot "minprior":
[1] 0.05

Slot "tolerance":
[1] 1e-06

Slot "verbose":
[1] 0

Slot "classify":
[1] "auto"

Slot "nrep":
[1] 1
```

Listing 3.5: `"FLXcontrol"` default values

The package also provides functions in order to analyze the results of the fitting procedure and for extracting important values of interest. For an object of the class `"flexmix"` the default plot method `plot()` shows a visualization of the cluster structure in form of a *rootogram*. Rootograms follow the style of histograms with one major difference: they contain the square roots of the counts of posterior probabilities which emphasizes low counts and lessens peak effects in the data. These effects outline the separation of the different clusters. Peaks close to one indicate a clear assignment to the specific component and a good separation between the components. On the other hand, overlapping with other components appears as mass in the center of a rootogram. Counts close to zero or below a previously defined threshold occur frequently when fitting FMMs. **flexmix** neglects them in order to avoid distortions in cluster analysis. The command `posterior()` gives information about the posterior probabilities resulting through the EM algorithm whereas `cluster()` shows the ultimate cluster assignment based on the maximum a posteriori probability for each observation. For any object of the class `"flexmix"` the command `summary()` provides further information about the cluster assignment. A table of the cluster assignment summarizes the number of observations assigned to each component as well as the overall number of observations with posterior probabilities greater than a fixed threshold ϵ for every component. The ratio of these numbers results in a measure for the quality of the cluster separation. Ratios close to one indicate a good separation for a component as the majority of the points with positive a posteriori probability would be finally assigned to the respective component. **flexmix** provides different model selection criteria which are also shown in the `summary()` output including the MLEs, df, AIC, BIC and ICL. The estimated component specific parameters are extracted by the command `parameters()` whereas `summary(refit())` yields the corresponding significance tests. Grün and Leisch (2008b, p. 16) point out that the implemented p-values are approximations without any corrections.

`"FLXM"` represents the virtual parent class for different types of mixture models. This class specifies the models using the slots in Table 3.4 as Leisch (2004b, p. 12) explains.

Arguments	Explanation/Functionality
`fit`:	function in predictor x, response y and component weights w, returns object `FLXcomponent`
`name`:	model identifier, e.g. `"FLXMRnlm"`
`formula`:	model formula, see formula in R
`logLik`:	function in x, y, returning log-likelihood
`predict`:	function in x predicting the mean given x
`df`:	number of estimated parameters defining df
`parameters`:	component specific regression coefficients and dispersion parameter
`defineComponent`:	M-step driver function specifying object `FLXcomponent` with `logLik`, `predict`, `df` and `parameters`
`weighted`:	weight parameters
`control`:	control parameters in `"FLXcontrol"`

Table 3.4: Arguments in `"FLXM"`

`"FLXM"` has the inheriting classes `"FLXMC"` and `"FLXMR"`. `"FLXMC"` covers model-based clustering and contains one additional slot for the distribution. `"FLXMR"` refers to mixtures of regression models and is of main interest in this work. Under the parent class `"FLXMR"`, the package **flexmix** offers already various regression models. The most important ones are GLMs specified by the class `"FLXMRglm"`. While the prefix `"FLXMR"` indicates mixture of regressions, `"FLXMRglm"` links to the clusterwise regression of GLMs. The model specification affords response and predictor variables, the dependency structure and the distribution family. Based on the current package, the implemented distribution families are Gaussian, Gamma, Poisson and binomial. `"FLXMRglm"` allows varying effects between components which affords the component-wise estimation of regression coefficients as well as the dispersion parameters.

3.3.2 The M-Step Driver for Mixtures of Generalized Linear Models

The key step in fitting the component specific parameters is the M-step as part of the EM algorithm. As Leisch (2004b, p. 12) explains, **flexmix** allows the use of self-implemented new driver functions. The denotation of these M-step driver functions is always according to the specific model class. For example, the M-step driver function for mixtures of GLMs is denoted by `"FLXMRglm"`. The following source code shows the basic structure of the M-step driver function `"FLXMRglm"` in **flexmix**.

```
1  FLXMRglm <- function(formula=.~.,
2                       family=c("gaussian", "binomial", "poisson", "Gamma")
                            ,
3                       offset=NULL)
4  {
5    ...
6    z <- new("FLXMRglm", weighted=TRUE, formula=formula,
7             name=paste("FLXMRglm", family, sep=":"), offset = offset,
8             family=family, refit=glmrefit)
9    ...
10
```

```
11   if(family=="gaussian"){
12     z@defineComponent <- function(para) {
13       predict <- function(x, ...) {
14         dotarg = list(...)
15         if("offset" %in% names(dotarg)) offset <- dotarg$offset
16         p <- x %*% para$coef
17         if (!is.null(offset)) p <-  p + offset
18         p
19       }
20
21       logLik <- function(x, y, ...)
22         dnorm(y, mean=predict(x, ...), sd=para$sigma, log=TRUE)
23
24       new("FLXcomponent",
25           parameters=list(coef=para$coef, sigma=para$sigma),
26           logLik=logLik, predict=predict,
27           df=para$df)
28     }
29
30     z@fit <- function(x, y, w, component){
31       fit <- lm.wfit(x, y, w=w, offset=offset)
32       z@defineComponent(para = list(coef = coef(fit), df = ncol(x)+1,
33                                     sigma =  sqrt(sum(fit$weights *
34                                                       fit$residuals^2 /
35                                                       mean(fit$weights)
                                                          )
36                                                   /(nrow(x)-fit$rank))))
37     }
38   }
39  ...
40 }
```

Listing 3.6: M-step driver for mixtures of GLMs in **flexmix** (excerpt)

The functionality of the driver functions is technically based on internal functions and follows a general scheme as the previous code extract on the M-step driver in Listing 3.6 shows. The following simplified explanations refer to the internal functions as well as the sequence of their commands in a typical M-step driver function in **flexmix** with specific reference to `"FLXMRglm"`. A return object `z` is constructed from the specific model class, in the present case an object of class `"FLXMRglm"` for GLMs. Thus, `z` comprises all slots given in Table 3.4 from the parent class `"FLXM"` plus additional slots `offset` (from parent class `"FLXMR"`), `family` and `refit`. In order to enable a clear and structured explanation of the M-step driver functionality, the following description will solely refer to the additional slots related to the fitting procedure, as a full explanation of the underlying classes is beyond the scope of this work. Lines 6 to 8 in Listing 3.6 declare the general slots `formula`, `name` and `family` for a GLM. Starting in line 11 the construction of the remaining distributional based slots is illustrated for the specific case of `family="gaussian"`. The underlying mixture components are assumed to be normal differing only in the distributional parameters as discussed in Section 2.1. The expression `z@defineComponent` assigns a slot containing the framework for the necessary return values. It contains the function specifying the computation of the log-likelihood function, `logLik(x,y)`, as well as the function for the computation of the predicted values given x, denoted as `predict(x)`. The computation of these values requires the regression coefficients available from `parameters` and the variance parameter denoted as `sigma`. The function `logLik(x,y)` returns the value of the log-likelihood function based on the

fitted values evaluated by the function `predict(x)`. The specification of a new object of class `"FLXcomponent"` merges these two functions with placeholders for the model parameters `parameters`, `sigma` and the df. In line 30 a concrete fitting function is assigned to the return object `z` by `z@fit`. The function is defined as `fit(x,y,w,...)` with a vector `w` containing the component specific weight parameters corresponding to the proportions w_{ik} in Section 2.4. The fitting function determines the estimates for the model parameters `parameters`, `sigma` as well as df. These output values serve for the evaluation of the already existing framework `z@defineComponent`. In the case of mixtures of GLMs, the model parameter estimation is provided by the well-known functions `lm.wfit()` for normally distributed components or `glm.fit()` for the binomial, Poisson and gamma distribution. In a final step (line 32) the previously defined expression `z@defineComponent` is evaluated by substituting the placeholders for the model parameters and the df by the numerical values resulting by `z@fit`. An object of the class `"FLXcomponent"` stores `parameters`, `df`, `logLik`, `predict` as final results.

As Leisch (2004b, p. 12) emphasizes, the command function `flexmix()` never operates with model specific parameters as it simply performs the internal functions `logLik(x,y)` and `fit(x,y,w,...)` which have the same form and structure for every available mixture model in `flexmix`.

3.4 Extending flexmix for Mixtures of GNMs

The package **flexmix** provides a broad toolkit for mixtures of GLMs but it is not supporting the fitting of mixtures of nonlinear regressions. The implementation of mixtures of GNMs represents a main achievement of this work. In particular, a new model class covering nonlinear regression models is implemented in the already existing infrastructure of the package **flexmix**. The model specific parameters comprise the regression coefficients and dispersion parameters. Both parameter sets are assumed to vary over all components. Although **flexmix** already covers a huge class of mixture models, these existing models do not cover nonlinear modeling structures. On the other hand **flexmix** already provides a broad toolkit for analyzing mixture models and a framework for fitting mixture models which can be partly adapted to new model classes. For these reasons, the implementation of mixtures of nonlinear regression models in **flexmix** is emphasized to use already existing infrastructure and to extend the model class in a reasonable way. Following are the main details on the extension and implementation of these models which will be bundled in the package **flexmixNL**.

In order to fit nonlinear regression models, a new class `"FLXMRnlm"` is defined. As Figure 3.2 outlines, the new class `"FLXMRnlm"` is an inheriting class of `"FLXMR"`. Therefore `"FLXMRnlm"` inherits all slots from its parent class `"FLXMR"` and specifies additional slots for the model formula, starting values and the distribution. As `"FLXMR"` does not consider model specific details, an additional slot `formula` is defined for the model formula in nonlinear regression models. Furthermore, as discussed in Section 1.1, fitting methods for nonlinear models provide numerical techniques which require the specification of starting values. This information will be covered by the new slot `start`. Similarly to

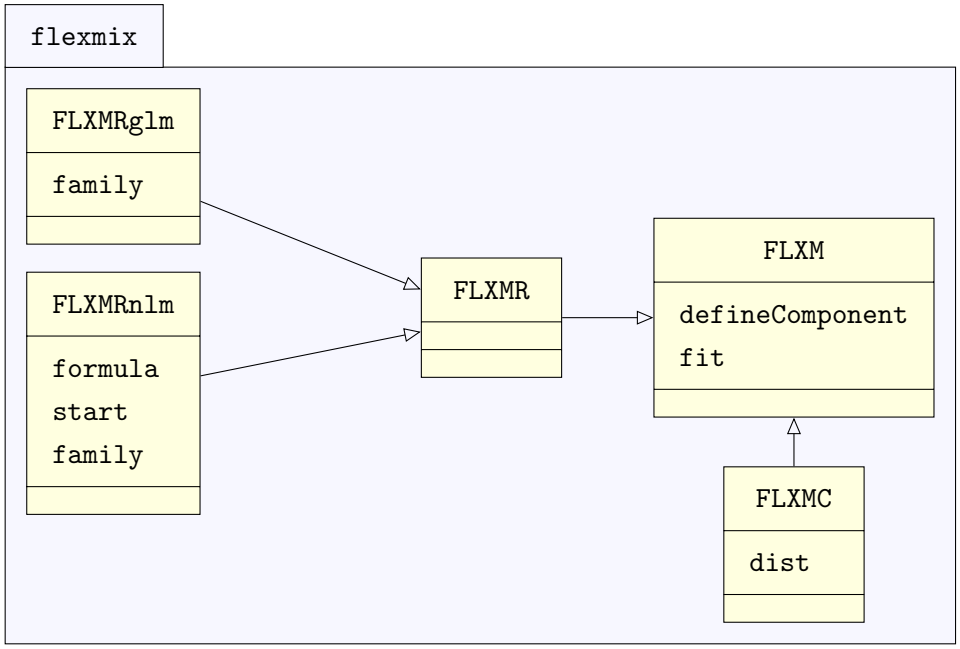

Figure 3.2: UML class diagram for `"FLXM"` in **flexmixNL**

mixtures of GLMs in **flexmix** which are covered by the class `"FLXMRglm"`, the nonlinear regression model will be available for different distributional families. Analogously, this property is denoted by an additional slot `family`.

3.4.1 The M-Step Driver for Mixtures of Generalized Nonlinear Models

Crucial differences to already existing mixture models in **flexmix** emerge with the implementation of the M-step driver function for mixtures of nonlinear regression models. As previously discussed, all M-step drivers are built on the framework of the fitting function `fit(x,y,...)`, the log-likelihood function `logLik(x)` and the function for predicting values, `predict(x)`. In the special case of nonlinear regression models, the parameter estimation is in general provided by numerical techniques which depend on the functional form of the regression function (given by the model formula) and the starting values for the unknown model parameters. These properties lead to complications in the M-step driver as the internal functions have to be adapted to establish the access to the arguments of `formula` and the list of starting values, denoted by `start`. The fitting of mixtures of nonlinear regression models requires model dependent fitting functions in **flexmix**. In consideration of these properties, the internal functions are now embedded in the M-step driver `"FLXMRnlm"` denoted as `fit(x,y,w,formula,start)` and `logLik(x,formula)` compatible with the already existing notation of internal functions in **flexmix** but with further input arguments. To keep a clean interface and structure in the M-step driver function, the parameter estimation in the fitting function `fit(x,y,w,formula,start)` is in general outsourced in **flexmixNL**. In `"FLXMRnlm"` the core fitting function for model parameters of mixtures of nonlinear regression models is outsourced to a function denoted as `nls.wfit` for Gaussian mixture models and `gnm.fit` for general GNMs. The functions return the required estimates of the model parameters

in order to enable further calculations of the final results which are again stored in an object of the class `"FLXcomponent"`.

3.4.1.1 Gaussian Mixtures of Nonlinear Regression Models

The nonlinear regression model with an underlying normal distribution is the first available mixture model in `"FLXMRnlm"`. The model can be accessed by setting the distributional parameter in the model specifications `family="gaussian"` as the following listing shows.

```
flexmix(y ~ x,k = 2,model = list(FLXMRnlm(..., family = "gaussian",...)))
```

Listing 3.7: Function call `flexmix()` for Gaussian mixtures of GNMs

The M-step driver for objects of the class `"FLXMRnlm"`, which is also denoted as `FLXMRnlm`, is given in the next listing for the normal distribution.

```
FLXMRnlm <- function(formula = .~.,
                     family = c("gaussian", "Gamma"),
                     start = list(), offset = NULL)
{
  ...
  z <- new("FLXMRnlm", weighted = TRUE, formula = formula, start = start,
           name = paste("FLXMRnlm", family, sep=":"), offset = offset,
           family = family, refit = refit)

  if(family=="gaussian"){
    z@defineComponent <- function(para){
      predict <- function(x, ...){
        startEnv <- new.env(hash = FALSE, parent = environment(formula))
        for (i in names(para$start)) assign(i, para$coef[[i]],
                                            envir = startEnv)
        p <- eval(formula[[3L]], startEnv)
        p
      }
      logLik <- function(x, y, ...) dnorm(y, mean=predict(x, ...),
                                          sd=para$sigma, log=TRUE)

      new("FLXcomponent",
          parameters=list(coef=para$coef, sigma=para$sigma),
          logLik=logLik, predict=predict,
          df=para$df)
    }
    z@fit <- function(formula, start, x, y, w)
    {
      fit = nls.wfit(formula = formula, start = start,
                     data = data.frame(data,w))
      z@defineComponent(para = list(coef = coef(fit),
                                    start = as.list(start),
                                    df = length(all.vars(formula))-1,
                                    sigma =  sqrt(sum(fit$weights *
                                                      fit$residuals^2 /
                                                      mean(fit$weights)
                                                          )
                                                  /(fit$df.residuals))))
    }
  } ...
}
```

Listing 3.8: M-step driver for Gaussian mixtures of GNMs in **flexmixNL** (excerpt)

In lines 6 to 8 a return object `z` is constructed from the specific model class `"FLXMRnlm"`. The slot `name` is set to `FLXMRnlm:gaussian` as `family="gaussian"` holds. `z` comprises again all slots given in Table 3.4 from the parent class `"FLXM"` plus additional slots `offset` (from parent class `"FLXMR"`) as well as the slots `family`, `formula` and `start`. The last two slots are model-specific as they afford knowledge about the nonlinear model formula. Starting in line 10 the construction of the remaining slots `defineComponent`, `logLik()` and `fit()` is illustrated for the specific case of GNMs. The expression `z@defineComponent` shows already crucial differences compared to its construction for GLMs. The computation of the fitted values, provided by `predict()`, requires the RHS of the nonlinear model formula accessed by `formula[[3L]]`. This is evaluated at the fitted values denoted by `coef` (labelled by `start`). Line 27 reveals another difference as the fitting function comprises further arguments `formula` and `start` given by `fit(formula,start,x,y,w)`. The necessity of these further arguments is explained by the call of the outsourced fitting function in lines 29 and 30.

```
1 nls.wfit = function(formula, start, data = list(), control = list())
2 {
3   fit = nls(formula = formula, start = start, data = data,
4             weights = as.vector(w))
5  ...
6   fit
7 }
```

Listing 3.9: Outsourced fitting function for Gaussian mixtures in **flexmixNL**

For mixtures of nonlinear regression models with normally distributed responses the function `nls()` offers a suitable computational framework by setting the weight components equal to the proportions w_{ik} according to the WNLS method in Example 2.1 (see also Section 2.4). Line 3 in Listing 3.9 performs the fitting procedure returning the results to the control variable `fit`. Well-known accessor functions allow the extraction of these results as, for example, `coef(fit)` for the regression coefficients. For a detailed explanation of further accessor functions reference is made to Ritz and Streibig (2008). Within the scope of `flexmix()` the outsourced fitting function `nls.wfit()` passes the return values provided by `nls()` to the slot `z@fit` within the M-step driver `FLXMRnlm`. These values enable the evaluation of `z@defineComponent` in a last step. The final model parameters are again captioned by an object of the class `"FLXcomponent"`.

3.4.1.2 Mixtures of Generalized Nonlinear Models

In the case of distributional assumptions other than the normal distribution the fitting procedure `nls()` becomes inadequate. Theoretical considerations in Chapter 2 buttress the use of an IWLS procedure for estimating the component specific regression coefficients. The previously discussed package **gnm** provides an appropriate fitting procedure by means of its main function `gnm()`. Within the scope of **flexmixNL**, `gnm()` can be embedded to enable a proper fitting of GNMs in mixture components. Similarly to the previously shown Gaussian mixtures of nonlinear regressions, mixtures of GNMs are accessed by the following call, exemplary for the Gamma distribution.

```
1 flexmix(y ~ x, k = 2, model = list(FLXMRnlm(..., family = "Gamma",...)))
```

Listing 3.10: Function call `flexmix()` for mixtures of GNMs with Gamma responses

The underlying M-step driver `FLXMRnlm` executes the GNM for the Gamma distribution. All cases have a similar structure for the construction of `z@defineComponent` and `z@fit`. They still remain different in the distributional parameters and the underlying fitting procedures. In the case of Gamma distributed responses the outsourced fitting procedure is denoted by `gnm.wfit` emerging in line 3 in the excerpt of the M-step driver `FLXMRnlm` given in the subsequent listing.

```
1 z@fit <- function(formula, start, x, y, w)
2 {
3   fit = gnm.wfit(formula=formula, start = start,data=data.frame(data,w),
4                  family = Gamma(link="identity"))
5   ...
6 }
```

Listing 3.11: Fitting function for Gamma mixtures in **flexmixNL**

The fitting function `gnm.wfit` performs the command `gnm()` for GNMs using the proportions w_{ik} as weights components. The fitting of GNMs in **flexmixNL** is provided under the identity link as specified in the general model framework in Section 1.4. Furthermore the fitting procedure depends on the model formula as well as predefined starting values.

```
1 gnm.wfit = function(formula, start, x, y, w, ...)
2 {
3   fit = gnm(formula = formula,family = Gamma(link = "identity"),
4             data = data, start = unlist(start), weights = as.vector(w),
5             tolerance = 1e-6, verbose = F, trace = F, checkLinear = T)
6  ...
7   fit
8 }
```

Listing 3.12: Outsourced fitting function for Gamma mixtures in **flexmixNL**

Line 3 performs the fitting procedure storing the results again in the control variable denoted by `fit`. The subsequent sequence of steps equals to the already discussed case of Gaussian mixtures of nonlinear regression models. Special attention is given to the computation of specific distributional parameters necessary as arguments for `logLik()` and `predict()` for the evaluation of the expression `z@defineComponent`.

3.5 Standard Errors

Drawing conclusions on the accuracy of the derived parameter estimates, given by the vector $\hat{\boldsymbol{\Psi}}$, requires a statement on their variability. This section aims to present methods in order to compute standard errors for these parameters based on the incomplete-data log-likelihood function (2.9). The main emphasis lies in the introduction of the two computation methods which are being used within the applications of the present work. Information is also given regarding available results on standard errors in **flexmix**.

3.5.1 Standard Errors in flexmix

The package **flexmix** provides the function `refit()` for additional information on the parameter estimates. The output of the function `refit()` summarizes the provided estimates, standard errors and significance tests. As Grün and Leisch (2008b, p. 8) explain, numerical approaches are used in order to obtain an approximation for the standard errors. The implemented function `refit()` involves the optimization function `optim()` which is available in the standard package **stats** in R. Technically, the computation of the Hessian matrix is provided by numerical optimization over the complete log-likelihood function (2.17). The minimization is provided over the MLEs obtained by the EM algorithm as starting values. In order to derive the Hessian matrix numerically, `optim()` offers the optional setting `hessian = TRUE`. The standard errors are derived from the negative of the computed Hessian matrix.

3.5.2 Exact Computation

The exact computation of standard errors for MLE refers to the analytical derivation of the Hessian matrix for the incomplete-data log-likelihood function (2.9) as discussed in Section 2.5. The derived standard errors will be denoted as $\mathrm{SE}^{ex}(\cdot)$ in the following. The standard errors stem from the evaluation of the inverted Hessian matrix in the MLE $\hat{\boldsymbol{\Psi}}$ which was obtained by the EM algorithm. The exact computation of the Hessian matrix and the subsequent computation of the standard errors is difficult to generalize. The form of the second order derivatives of the log-likelihood function (2.9) is highly dependent on the functional form of the mean function. As the mean function stems from nonlinear dependence structures, it is generally complicated to handle due to its low mathematical tractability, in particular for mixtures of GNMs. For FMMs with a predefined mean function and a fixed number of components the second order derivatives can be derived symbolically for a mixture pdf of lower complexity. The statistical software R includes methods for symbolical differentiation, as for example those provided by Clausen and Sokol (2018). Clausen and Sokol (2018) developed a package **Deriv** in order to derive symbolic differentiations from expressions enabled through the operator `D()`. An exemplary approach to the computation of standard errors is given below.

Exact Computation of Standard Errors in R:

1. Define the mixture pdf (2.1) as symbolic function with the command `expression()`.
2. Derive second order derivatives of the incomplete log-likelihood function (2.9) with the help of the differentiation operator `D()`.
3. Evaluate the inverse of the Hessian matrix in the MLE obtained by the EM algorithm and extract the square roots in order to obtain standard errors $\mathrm{SE}^{ex}(\cdot)$.

As $\mathrm{SE}^{ex}(\cdot)$ relies on the exact derivation of the Hessian matrix, complex functional shapes may lead to numerical difficulties in the computation due to collinearities. These

challenges and specific results will be discussed by the means of applications in the subsequent chapters.

3.5.3 Numerical Derivation

As the pdf of GNMs complies generally with an increasing complexity, an exact derivation of the Hessian matrix may be too tedious or mathematically intractable. These cases afford a numerical derivation of the Hessian matrix in order to obtain standard errors. The numerically derived standard errors will be denoted as $\mathrm{SE}^{num}(\cdot)$ in the following analysis. There are different approaches on approximating the Hessian matrix numerically in R. Gilbert and Varadhan (2016) provides the package **numDeriv** with the implemented function `hessian()`. This work takes up on the use of the function `hessian()` which has proven useful for the computation of complex functions. In order to obtain standard errors for $\hat{\boldsymbol{\Psi}}$, the target function is set as the log-likelihood function (2.9). The key step, in order to derive the numerical approximation provided by the command `hessian()`, stems from the implemented function `genD()` in the same package. `genD()` basically approximates the second order derivatives with the central difference quotient. The specific approach on the computation of standard errors $\mathrm{SE}^{num}(\cdot)$ is given below.

Numeric Approximation of Standard Errors in R:

1. Define log-likelihood function (2.9) as function depending on the functional form of the nonlinear mean function.
2. Approximate Hessian matrix with the function `hessian()` initialized with the MLEs obtained by the EM algorithm.
3. Compute standard errors $\mathrm{SE}^{num}(\cdot)$ by the inversion of the Hessian matrix and root extraction.

The application of these methods of standard error computation will be carried out within the subsequent chapters.

CHAPTER 4

Monte Carlo Simulation Study for Two-Component Gamma Mixture Models

Introduction

This section provides a Monte Carlo (MC) simulation study on the performance of the fitting algorithm for mixtures of GNMs. The underlying fitting algorithm is bundled in the package **flexmixNL** consisting of the new model class `"FLMRnlm"`, as discussed in Section 3.4. The main objective of this study is to obtain parameter estimates and adequate standard errors in order to assess the algorithm's performance and deliver estimates with an acceptable precision. To get an impression of the algorithm's functionality and its sensitivity on the given data, the simulation study will be provided for different synthetic data sets with different data sample sizes. Statements on the performance will be derived based on specific measures. The key measures of interest are the obtained parameter estimates and measures describing their variability. The models of interest are restricted to GNMs with underlying Gamma distributions. The distributional specification is motivated by real world applications following in the subsequent chapter. An essential aspect of the models will be given by the fitting of a nonlinear mean function. With regard to a real world application, the nonlinear mean function will be restricted to a sigmoid function of a predefined shape. The complexity and tedious tractability arising with the fitting of GNMs will be intensified by the overlapping of the components and different levels of variability. A key aspect of the simulation study will be the reproducing of an original configuration and the discrepancy of the estimated parameters to the initial values under a randomly generated data sample. Emphasis is placed on the identification of the distinct components referring to two differently parametrized mean functions. Therefore, misclassification rates regarding the true cluster allocation of the sample points are not a focus. As the main focus will be on obtaining appropriate parameter estimates for the components pdf, the framework of the FMM is held preferably at low complexity level. For this reason, the number of components will be restricted to $K = 2$.

Section 4.1 presents the simulation setup for the present study. The MC simulations afford a specification of the underlying GNM and the nonlinear mean function. The data generation will be provided based on the specified presumptions. The subsequent Section 4.2 summarizes all parameters of interest within the overall simulation study. In order to retrace the fitting procedure, Section 4.3 discusses its sequence applied to an exemplary data sample. The final Sections 4.4 and 4.5 discuss the results for two different data sample sizes.

4.1 Simulation Setup

The simulation study is based on a two-component Gamma mixture model where the specifications are given in the subsequent sections.

4.1.1 Model Specification

The two-component mixture distribution follows the pdf

$$f^M(y_i; \mu_i(\boldsymbol{\beta}), \boldsymbol{\phi}, \pi) = \pi \cdot f(y_i; \mu_i(\boldsymbol{\beta}_1), \phi_1) + (1 - \pi) \cdot f(y_i; \mu_i(\boldsymbol{\beta}_2), \phi_2), \tag{4.1}$$

where π denotes the component weight for the first component as discussed in Section 2.1 and $i = 1, \ldots, n$. The specification of the two-component mixture model (4.1) affords the fitting of the first component weight through $\pi_1 = \pi$ as the second results from $\pi_2 = 1 - \pi_1$. In avoidance of problems arising through the interchangeability of components, the two components will be ordered according to the sequence presented in Section 2.2 where the larger prior probability $\max \hat{\pi}_k$, $k = 1, 2$, determines the first component. The parameter vector $\boldsymbol{\phi}$ comprises the component specific dispersion parameters ϕ_1 and ϕ_2. The underlying components are furthermore specified by the means $\mu_i(\boldsymbol{\beta}_1)$ and $\mu_i(\boldsymbol{\beta}_2)$ and follow a Gamma distribution with pdf

$$f(y_i; \mu_i(\boldsymbol{\beta}_k), \nu_k) \quad = \quad \exp\left(-\frac{\nu_k}{\mu_i(\boldsymbol{\beta}_k)} y_i\right) \left(\frac{\nu_k}{\mu_i(\boldsymbol{\beta}_k)}\right)^{\nu_k} {y_i}^{\nu_k - 1} \frac{1}{\Gamma(\nu_k)}, \quad k = 1, 2,$$

where the shape parameters correspond to the reciprocal dispersion parameters through $\nu_k = 1/\phi_k$ and a non-negative mean function $\mu_i(\boldsymbol{\beta}_k) > 0$ for responses $y_i > 0$, $i = 1, \ldots, n$. The specification of the mean function follows in the subsequent section.

4.1.2 Mean Function

The mean function states the central element in the simulation study as it influences the performance of the fitting procedure predominantly. The scope of this analysis addresses a nonlinear regression function which complies with the specification for GNMs as discussed in Chapter 1. The underlying mean function is inspired by a real world application of gas transmission following the research provided by Friedl et al. (2012). The functional structure models a decreasing consumption behavior by a sigmoid curve in dependence of the outside temperature. The functional form is specified by the pa-

rameter vector $\boldsymbol{\beta}_k = (\beta_{k1}, \beta_{k2}, \beta_{k3}, \beta_{k4})^\top$, respectively through the relationship

$$\mu_i(\boldsymbol{\beta}_k) := h(x_i, \boldsymbol{\beta}_k) = \beta_{k4} + \frac{\beta_{k1} - \beta_{k4}}{1 + \left(\frac{\beta_{k2}}{x_i - 40}\right)^{\beta_{k3}}}, \quad k = 1, 2, \tag{4.2}$$

with x_i representing the explanatory variable for $i = 1, \ldots, n$. The regression coefficients β_{k1} and β_{k4} state the upper and lower asymptotes for the kth component where the consumption attains its maximum and minimum level. The remaining regression coefficients β_{k2} and β_{k3} affect the shape and curvature of the resulting sigmoid function for the kth component. The regression coefficient analysis will be intensified at a later time for the estimated values. The construction of the mean function in R follows the discussion on the implementation of GNMs in R in Section 3.2.2 and is given in Listing 4.1.

```
1  > library(gnm)
2  > gas.1 = function(x,predictors){
3  +   list(predictors=list(a=1,b=1,c=1,d=1),
4  +        variables = list(substitute(x)),
5  +        term = function(predictors, variables) {
6  +        paste(predictors[4],"+(",predictors[1],"-",predictors[4],")/(",
7  +                "1+(",predictors[2],"/(",variables,"-40","))^",
8  +                predictors[3],")", sep="")
9  +        })
10 + }
11 > class(gas.1) ="nonlin"
```

Listing 4.1: Specification of mean function with **gnm**

The nonlinear regression function is stored in the argument `gas.1` which is specified in dependence of the variable `x` as a placeholder for the explanatory variables and the vector `predictors` comprising the regression coefficients following the specification in **gnm** by Turner and Firth (2018). The specification of the functional form follows in lines 6 to 8 as sequence of character variables stored in the argument `term`.

4.1.3 Initial Configuration

The initial configuration of the mixture pdf in (4.1) and the mean function (4.2) will serve as starting setup for the further data generation in order to run the simulation study. The configuration of the two components is inspired by a real world situation on gas consumption. The first component will be modeled for the purpose of industrial facilities with a higher consumption rate and a greater component size. The other component is ought to represent consumption with low-level descent behavior generated by private households. Further setting parameters are the shape parameters ν_1 and ν_2 for the Gamma distributed components. Due to the mean dependence of the variance for the Gamma distribution, the variability increases with higher mean values yielding higher variability for the components representing the industrial consumption behavior. The initial configuration of the two-component Gamma mixture model (4.1) is given in the subsequent Table 4.1.

Component	π_k	β_{k1}	β_{k2}	β_{k3}	β_{k4}	ν_k
$k=1$	0.6	58 000	-35.0	10.0	32 000	50.0
$k=2$	0.4	34 000	-35.0	20.0	8 000	50.0

Table 4.1: Initial configuration for two-component Gamma mixture model

The component weights π_k denote the probabilities of a data point y_i, $i = 1, \ldots, n$, lying in the kth component. The parameter vector $\boldsymbol{\beta}_k$ specifies the mean function as given in (4.2). The clearly visible gap between the upper and lower asymptotes enables the modeling of two different consumption levels by means of a mixture distribution. The shaping parameter β_{k3} has significant influence on the decrease of the sigmoid mean function (4.2) for $k = 1, 2$. A central assumption in the construction of the synthetic data set states a continuous consumption level for the production facilities of the industrial component. Therefore the given parameter specification in Table 4.1 allows for a sharper decrease in consumption for the lower component.
The fitted results can be reproduced by setting the random seed equal to `set.seed(17)` and `iter.max = 100`.

4.1.4 Data Generation

This section outlines the framework for the construction of the synthetic data set as basis for the simulation study. The basic idea is to construct a randomly Gamma distributed data vector $\boldsymbol{y} \in \mathbb{R}^n$ following the mean function (4.2) with initial configuration as given in Table 4.1. Predictor variables are given by the vector $\boldsymbol{x} \in \mathbb{R}^n$. The resulting data samples will be denoted as $(\boldsymbol{x}, \boldsymbol{y})^{(j)}$ for the simulation runs $j = 1, \ldots, S$ and comply with the criteria of GNMs where $\boldsymbol{y}$ follows a Gamma mixture model with pdf (4.1). The main objective of this simulation process is to execute the fitting procedure presented in Section 3.4 and to assess the performance. The data samples serve as initial configuration for the fitting procedure which will be initialized by the starting values in Table 4.1. The initial cluster assignment will be set randomly which is a default setting in **flexmix** (see also Section 3.3.1). As previously discussed, the hypothetical data is inspired by a real world application on gas flow. The predictor variable $\boldsymbol{x}$ describes the outside temperature which has a causal relation to the gas flow. For a reasonable assessment the range for the predictor variables is set to the range $\mathbb{R}^{\text{temp}} = [-10, 20]$. In a first step, an equidistant sequence of n numbers $\boldsymbol{x} = x_1, \ldots, x_n$ is chosen as predictor variables. For the purpose of constructing a two-component data sample, the sequence $\boldsymbol{x}$ is split into two sets with samples sizes n_1 and n_2 where $n = n_1 + n_2$ holds. Technically, a sample with length n_1, denoted as $\boldsymbol{x}_{n_1}$, is taken from the sequence $\boldsymbol{x}$ as predictor subset for the first component. The remaining set $\boldsymbol{x}_{n_2}$ represents the predictor variables for the second component with length $n_2 = n - n_1$. In a subsequent step, the component specific mean functions are evaluated in the predictor variables, respectively $\mu(x_i, \boldsymbol{\beta}_1, \nu_1)$ for $x_i \in \boldsymbol{x}_{n_1}$ and $\mu(x_i, \boldsymbol{\beta}_2, \nu_2)$ for $x_i \in \boldsymbol{x}_{n_2}$. Let $G(\nu_1, \lambda_1)$ and $G(\nu_2, \lambda_2)$ denote the underlying Gamma distributions within pdf (4.1). According to Example 1.3 the component rates of the Gamma distributions correspond to $\lambda_1 = \nu_1 / \mu_i(\boldsymbol{\beta}_1)\ \forall x_i \in \boldsymbol{x}_{n_1}$ and $\lambda_2 = \nu_2 / \mu_i(\boldsymbol{\beta}_2),\ \forall x_i \in \boldsymbol{x}_{n_2}$. The vector $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ is generated randomly using the predictor variables $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$,

the shape parameters ν_1 and ν_2 and the evaluated mean functions $\mu_i(\boldsymbol{\beta}_1),\ \forall x_i \in \boldsymbol{x}_{n_1}$, and $\mu_i(\boldsymbol{\beta}_2),\ \forall x_i \in \boldsymbol{x}_{n_2}$, according to the a mixture pdf of Gamma distributions (4.1).

The presented data generation procedure is carried out $S = 1\ 000$ times (simulation runs) in order to provide the MC parameter estimate vectors $\hat{\boldsymbol{\Psi}}^{(j)}$ for $j = 1, \ldots, S$. Within this simulation study the sample sizes will be fixed as $n = 500$ or $n = 1\ 000$ where n_k denotes the number of assigned data points to the kth component on condition that $\sum_k n_k = n$ holds. Figure 4.1 illustrates exemplary data sets for the different sample sizes.

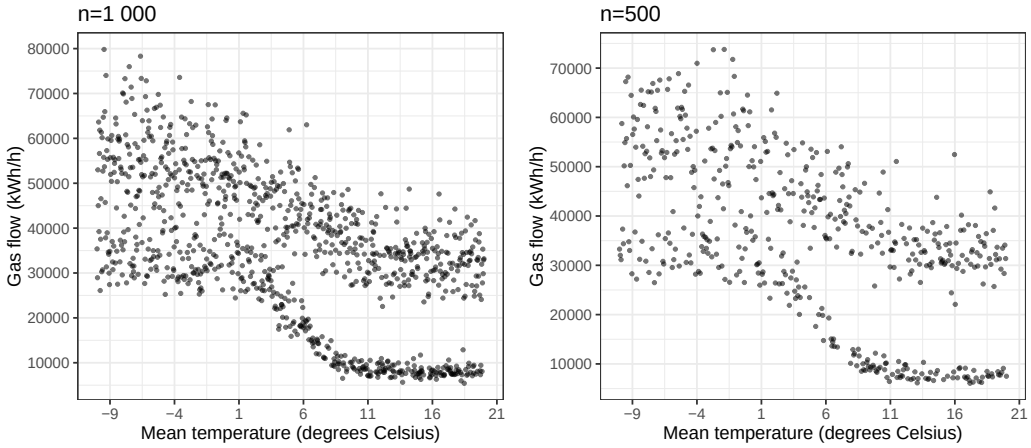

Figure 4.1: Simulated data ($n = 1000$ and $n = 500$)

4.2 Parameters of Interest and Measures of Algorithm Performance

As the exploration of the new fitting procedure in Section 3.4 represents the main aim of this simulation study, adequate criteria for the algorithm's performance have to be determined. The main measures are given by the derived parameter estimates and their variability. In addition, overall performance measures as the number of iterations will be assessed. This section summarizes all performance measures and parameters of interest.

4.2.1 Parameter Estimates

4.2.1.1 Regression Coefficients

The MLE for the regression coefficients, $\hat{\boldsymbol{\beta}}_k = (\hat{\beta}_{k1}, \hat{\beta}_{k2}, \hat{\beta}_{k3}, \hat{\beta}_{k4})^\top$, is derived through the EM algorithm. In a subsequent step the standard errors $\mathrm{SE}^{ex}(\hat{\beta}_{kp})$ and $\mathrm{SE}^{num}(\hat{\beta}_{kp})$ are derived separately through the methods presented in Section 3.5. Following up on the parameter estimation and standard error computation, confidence intervals can be derived as

$$(1-\alpha)\%\ \mathrm{CI}(\hat{\beta}_{kp}) = \left(\hat{\beta}_{kp} \pm z_{1-\alpha/2} \cdot \mathrm{SE}^*(\hat{\beta}_{kp})\right) \tag{4.3}$$

where z_α denotes the α quantile of the standard normal distribution and $\mathrm{SE}^*(\cdot)$ refers to the respective standard error (exact or numerical approach). For the overall MC simu-

lation study the mean values of the fitted parameters and further estimates are derived. The parameters of interest within the MC simulation for the regression coefficients are given below.

MC Mean: Let $\bar{\beta}_{kp}$ denote the mean value over all S simulations for the regression coefficients, respectively given by

$$\bar{\beta}_{kp} = \frac{1}{S}\sum_{j=1}^{S}\hat{\beta}_{kp}^{(j)}, \tag{4.4}$$

for $p = 1, 2, 3, 4$, the components $k = 1, 2$ and $j = 1, \ldots, S$.

MC Standard Deviation: The MC standard deviation is given as

$$\mathrm{SD}(\hat{\beta}_{kp}) = \sqrt{\frac{1}{S-1}\sum_{j=1}^{S}(\hat{\beta}_{kp}^{(j)} - \bar{\beta}_{kp})^2}, \tag{4.5}$$

for $p = 1, 2, 3, 4$, the components $k = 1, 2$ and $j = 1, \ldots, S$.

MC Bias: The MC bias represents the systematic absolute deviation for the MC mean of the true value and is given by

$$\mathrm{BIAS}(\hat{\beta}_{kp}) = \bar{\beta}_{kp} - \beta_{kp}, \tag{4.6}$$

for $p = 1, 2, 3, 4$ and the components $k = 1, 2$.

Asymptotic Confidence Interval (ACI): An asymptotic confidence interval (ACI) will be constructed with the MC mean of the parameter estimates and the respective asymptotic standard error over all MC results. Exemplary, the ACI for the parameter β_{kp} is given by

$$(1-\alpha)\%\ \ \mathrm{ACI}(\beta_{kp}) \approx (\bar{\beta}_{kp} \pm z^*_{1-\alpha/2} \cdot \mathrm{ASE}^*(\beta_{kp})) \tag{4.7}$$

where

$$\mathrm{ASE}^*(\hat{\beta}_{kp}) = \frac{1}{S}\sum_{j=1}^{S}\mathrm{SE}^*(\hat{\beta}_{kp}^{(j)}) \tag{4.8}$$

denotes the MC mean of the asymptotic standard errors of the regression coefficients for $p = 1, 2, 3, 4$ and components $k = 1, 2$ and $j = 1, \ldots, S$.

4.2.1.2 Shape Parameters

The MLE for the shape parameters $\hat{\nu}_k$ will be derived through the estimation given in Example 2.4 for the components $k = 1, 2$. The standard errors $\mathrm{SE}^{ex}(\hat{\nu}_k)$ and $\mathrm{SE}^{num}(\hat{\nu}_k)$ will be derived according to Section 3.5 while the confidence intervals can be computed according to the approach given in (4.3). Let $\hat{\nu}_k^{(j)}$ denote the estimate in the jth simulation step. The computation of the MC mean $\bar{\nu}_k$, deviation $\mathrm{SD}(\hat{\nu}_k)$ and bias $\mathrm{BIAS}(\hat{\nu}_k)$ follows Equations (4.4), (4.5) and (4.6).

4.2.1.3 Component Weights

The estimates for the component weights $\hat{\pi}_1$ and $\hat{\pi}_2 = 1 - \hat{\pi}_1$ will be derived through the relation (2.20) as part of the EM algorithm. In a subsequent step the standard errors $\mathrm{SE}^{ex}(\hat{\pi}_k)$ and $\mathrm{SE}^{num}(\hat{\pi}_k)$ will be derived separately as presented in Section 3.5 for the components $k = 1, 2$. Confidence intervals can be derived according to (4.3). For the overall MC simulation study the mean value of the fitted parameters $\hat{\pi}_k^{(j)}$ and further estimates are derived. Therefore the MC mean $\bar{\pi}_k$, deviation $\mathrm{SD}(\hat{\pi}_k)$ and bias $\mathrm{BIAS}(\hat{\pi}_k)$ follow Equations (4.4), (4.5) and (4.6).

4.2.2 Measures of Algorithm Performance

In addition to the parameter estimates and their variability for the MC simulations further criteria will be derived in order to examine the algorithm performance.

4.2.2.1 Convergence Rate

A general performance criterion is given by the *convergence rate* defined as the proportion of the number of converged simulations to the overall number of simulations of the MC study, respectively

$$\frac{converged\ trials}{number\ of\ trials}.$$

Further distinction will be made between wrongly converged trials and trials converging to the true values. Wrongly converged trials yield usually parameter estimates which are not adequate for the specific problem. A detailed analysis on the handling of these results is given in Section 4.3.1.

4.2.2.2 Coverage Rate

Another criterion for the algorithm performance is given by the *coverage rate* in order to achieve an overall measure of quality for the parameter estimates and their standard errors. The coverage rate is a measure for the degree of coverage of the confidence intervals and the true parameter values. The coverage rate is given by the number of times the respective confidence interval captures the true value given in Table 4.1 for the parameter estimates. It is evident that the construction of the confidence interval is highly depending on the standard errors of the related parameter estimates.

4.3 Fitting Procedure

Given a synthetic data sample set $(\boldsymbol{x}, \boldsymbol{y})^{(j)}$, $j = 1, \ldots, S$, the EM algorithm is applied in order to fit the two-component Gamma mixture model (4.1). This requires the repeated computation of the unknown parameter vector

$$\boldsymbol{\Psi}^{(j)} = (\pi_1^{(j)}, \beta_{11}^{(j)}, \beta_{12}^{(j)}, \beta_{13}^{(j)}, \beta_{14}^{(j)}, \beta_{21}^{(j)}, \beta_{22}^{(j)}, \beta_{23}^{(j)}, \beta_{24}^{(j)}, \nu_1^{(j)}, \nu_2^{(j)})^\top.$$

The fitting of the constructed data is provided in R through the package **flexmixNL** as introduced in Section 3.4. The respective specification of the command `flexmix()` is given in the listing below.

```
> library(flexmixNL)
> model = flexmix(y ~ x, k = 2,
+                 model = FLXMRnlm(formula = formula, family="Gamma",
+                         start = list(list(a=58000,b=-35,c=10,d=32000),
+                                      list(a=34000,b=-35,c=20,d=8000))))
```

Listing 4.2: Function call for nonlinear Gamma mixture model in **flexmixNL**

4.3.1 Assessment of Components

In order to provide the component specific analysis on the parameter estimates, a clear allocation of the given data to the specific components is necessary. The theoretical component specific regression functions for the synthetic data set are visualized in Figure 4.2. In the case of a convergent algorithm, ideally, the component specific regression functions have a shape similar to those shown in Figure 4.2. The simulated data set, as exemplary illustrated in Figure 4.1, shows a significant overlapping for increasing mean values. This effect intends to increase the complexity when fitting the synthetic data set in order to challenge the underlying fitting algorithm. It can be expected that this random variability yields to slightly skewed regression functions in comparison to the initial configuration given in Figure 4.2.

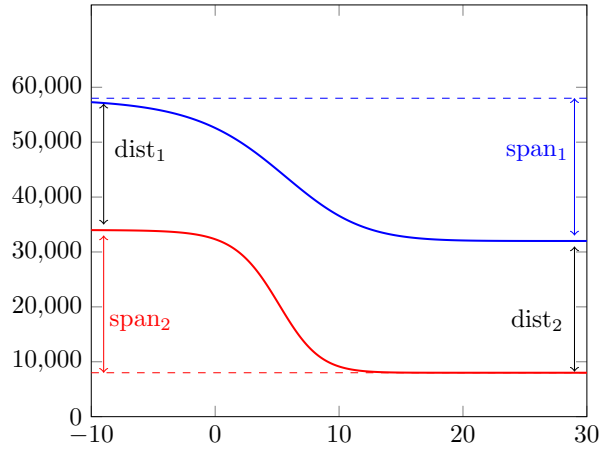

Figure 4.2: Component specific regression functions (initial configuration)

The true component specific regression functions are similar in their shape and differ mainly in their asymptotes. It may be challenging for the underlying EM algorithm to distinguish between the two different components as the shape coefficients of their mean functions β_{k3} differ just slightly, see also Table 4.1. These similarities may lead to unsuitable configurations as, for example, overlapping components or components exhibiting a non-sigmoid shape. Technically, unfavorable results arise from wrongly converging fitting methods which will be referred to as *misfits*. In order to obtain a proper selection of correctly fitted data, misleading results have to be filtered. Therefore, two control variables are defined to deal with misfits: $dist_1$ and $dist_2$ declaring the distance between the two upper asymptotes ($dist_1$) and lower asymptotes ($dist_2$). The span length within components is measured by the control variables $span_1$ and $span_2$ where $span_1$

refers to the distance between the upper and lower asymptote in the upper component while $span_2$ measures the distance of the asymptotes within the second component as sketched in Figure 4.2. The provided simulation study intends to reveal possible misfits for the given constellation of components. The corresponding ranges assess rules for the dropping of wrongly converged results:

$$dist_1, dist_2 \leq 16\,000 \quad \text{or} \quad span_1, span_2 \leq 8\,000$$

Results passing these requirements are considered within the present MC simulation study.

4.3.2 Exemplary Application of the Fitting Procedure

This section briefly sketches the fitting procedure by means of an exemplary data set. The fitting follows the discussion in Section 3.4. For a detailed discussion on available functionalities in `flexmix()` reference is made to Section 3.3.1. The data set is fitted for the nonlinear regression function (4.2) and Gamma mixture model (4.1) with $K = 2$ components and initial configuration as given in Table 4.1. The explicit execution of the fitting procedure is given in Listing 4.2. The corresponding output recalls the fitting function with all input parameters and displays the control variables for the underlying EM algorithm and the final cluster sizes:

```
> model
Call:
flexmix(formula=y~x,data=data,k=2,model=list(FLXMRnlm(formula=y ~
    -1+gas.1(x),family="Gamma",start=list(list(a = 58000, b = -35,
    c = 10, d = 32000),list(a = 34000, b = -35, c = 20, d = 8000)))))

Cluster sizes:
  1   2
546 454

convergence after 26 iterations
```

Listing 4.3: `flexmix()` output for nonlinear two-component Gamma mixture model

The resulting output indicates two fitted component and reveals the associated cluster sizes in relation $n_1 : n_2 = 546 : 454$. The EM algorithm affords 26 iteration steps until reaching an adequate convergence level for the approximated regression functions. A visualization of the synthetic data set with the final assignment to the components is given in Figure 4.3. The allocation of the data to the two components can be distinguished by the different colors. The fitted regression functions are added along the components. The component specific parameter estimates and the corresponding standard errors are displayed in Table 4.2.

The model selection criteria, as explained in Section 2.7.1, can be accessed using the commands in Listing 4.4.

```
> AIC(model)
[1] 20702.97
> BIC(model)
[1] 20756.95
> ICL(model)
```

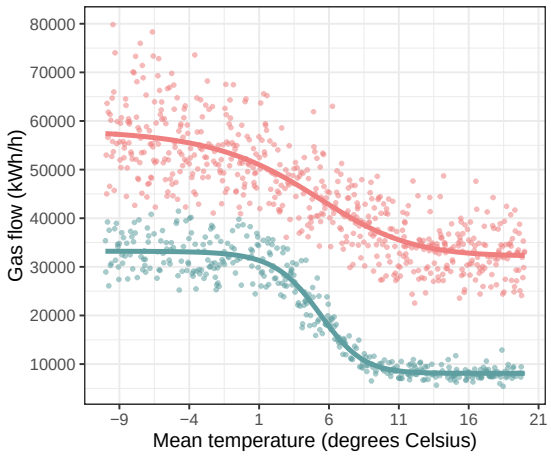

Figure 4.3: Fitted two-component Gamma mixture model ($n = 1000$)

```
[1] 20801.6
```

Listing 4.4: Model selection criteria for exemplary fitting

The package **flexmix** provides the `summary()` command in order to obtain a summarized result as discussed in Section 3.3.1. The output is given in the listing below.

```
> summary(model)
...
        prior size post>0 ratio
Comp.1 0.545  546    741 0.737
Comp.2 0.455  454    626 0.725
...
```

Listing 4.5: `summary()` output in **flexmix** for exemplary fitting

The `summary()` output recalls the function call and displays furthermore the classification of the sample to both components in the column `size`. It reveals that 546 data points were classified to the first component whereas even 741 had a positive posterior probability of lying in this component. The ratio of this relationship results in 0.737 signifying a good separation between both components. The rootogram serves as standard graphical visualization of the fitted model in **flexmix**. The corresponding command is given by `plot(model)` while the graphics is illustrated by Figure 4.4. The rootogram underpins the evident separation of the two components as the plotted histogram shows little mass in the central area.

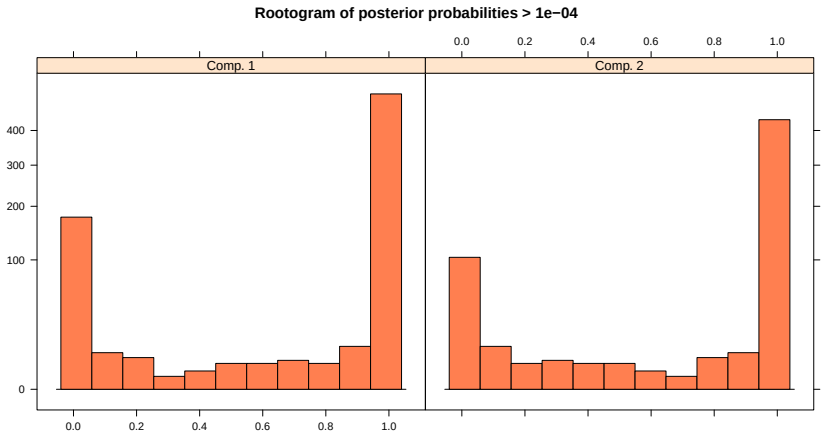

Figure 4.4: Rootogram for two-component Gamma mixture ($n = 1000$)

Parameter estimates and standard errors for exemplary data sample						
	π_k	β_{k1}	β_{k2}	β_{k3}	β_{k4}	ν_k
Component 1 (k=1)						
True values	0.60	58 000.00	−35.00	10.00	32 000.00	50.00
Estimates (EM)	0.58	57 340.01	−35.14	12.82	32 672.65	50.69
$\mathrm{SE}^{ex}(\cdot)$	0.02	1 322.00	0.51	2.10	627.89	4.99
95% CI	(0.53; 0.63)	(54 748.93; 59 931.09)	(−36.14; −34.14)	(8.70; 16.94)	(31 442.01; 33 903.30)	(40.91; 60.46)
$\mathrm{SE}^{num}(\cdot)$	0.02	1 322.06	0.51	2.10	627.89	4.99
95% CI	(0.53; 0.63)	(54 748.82; 59 931.19)	(−36.14; −34.14)	(8.70; 16.94)	(31 442.01; 33 903.30)	(40.91; 60.47)
Component 2 (k=2)						
True values	0.40	34 000.00	−35.00	20.00	8 000.00	50.00
Estimates (EM)	0.42	35 197.82	−35.14	20.07	7 935.89	50.29
$\mathrm{SE}^{ex}(\cdot)$	-	771.90	0.20	1.40	152.52	5.62
95% CI	-	(33 684.93; 36 710.71)	(−35.53; −34.74)	(17.33; 22.81)	(7 636.96; 8 234.82)	(39.28; 61.31)
$\mathrm{SE}^{num}(\cdot)$	-	773.36	0.20	1.40	152.52	5.62
95% CI	-	(33 682.06; 36 713.58)	(−35.53; −34.74)	(17.32; 22.81)	(7 636.95; 8 234.82)	(39.27; 61.32)

Table 4.2: Fitted coefficients, standard errors and confidence intervals ($n = 1000$)

4.3.3 Simulation Procedure

The simulation procedure can be summarized in the following iteration scheme:

Simulation procedure: For each simulation step $j = 1, \ldots, S$

1. Create synthetic data sample (sample size n) $(\boldsymbol{x}, \boldsymbol{y})^{(j)}$ for two-component Gamma mixture with pdf (4.1) and sigmoid mean function (4.2) according to Section 4.1.4.
2. Fit mixture model by estimating the parameter vector
$$\boldsymbol{\Psi}^{(j)} = (\pi_1^{(j)}, \beta_{11}^{(j)}, \beta_{12}^{(j)}, \beta_{13}^{(j)}, \beta_{14}^{(j)}, \beta_{21}^{(j)}, \beta_{22}^{(j)}, \beta_{23}^{(j)}, \beta_{24}^{(j)}, \nu_1^{(j)}, \nu_2^{(j)}).$$
according to Section 4.3.2.
3. Identify misfits as discussed in Section 4.3.1.
4. Compute standard errors
$$\mathrm{SE}^{ex}(\hat{\pi}_1^{(j)}), \mathrm{SE}^{ex}(\hat{\beta}_{1p}^{(j)}), \mathrm{SE}^{ex}(\hat{\beta}_{2p}^{(j)}), \mathrm{SE}^{ex}(\hat{\nu}_1^{(j)}), \mathrm{SE}^{ex}(\hat{\nu}_2^{(j)})$$
and
$$\mathrm{SE}^{num}(\hat{\pi}_1^{(j)}), \mathrm{SE}^{num}(\hat{\beta}_{1p}^{(j)}), \mathrm{SE}^{num}(\hat{\beta}_{2p}^{(j)}), \mathrm{SE}^{num}(\hat{\nu}_1^{(j)}), \mathrm{SE}^{num}(\hat{\nu}_2^{(j)})$$
for fitted parameters in $\hat{\boldsymbol{\Psi}}^{(j)}$ for $p = 1, 2, 3, 4$.
5. Repeat steps 1 to 4 for each simulation step j.
6. Compute MC parameters
$$\bar{\beta}_{kp}, \mathrm{SD}(\hat{\beta}_{kp}), \mathrm{BIAS}(\hat{\beta}_{kp}), \bar{\nu}_k, \mathrm{SD}(\hat{\nu}_k), \mathrm{BIAS}(\hat{\nu}_k), \bar{\pi}_1, \mathrm{SD}(\hat{\pi}_1), \mathrm{BIAS}(\hat{\pi}_1)$$
and measures for algorithm performance as given in Section 4.2.

4.4 Simulation Results (Sample Size $n = 1\ 000$)

The following sections present the results provided by the simulation study for the sample size $n = 1\ 000$ and $S = 1\ 000$ simulation runs. They furthermore give a detailed discussion on the obtained estimates and their quality.

The proportion of the trials converging to the accurate values (convergence rate) corresponds to 97% while 1% of the results were declared as misfits according to Section 4.3.1. Among the trials converging to the true values the number of iterations spans between 18 and 99. The median and the mean number of iteration steps correspond to 22.

4.4.1 Parameter Estimates

The two-component Gamma mixture model (4.1) with nonlinear mean function (4.2) requires the estimation of the parameter vector $\boldsymbol{\Psi}$. The subsequent sections discuss the provided parameter estimates for both components.

4.4.1.1 First Component

The identification of the components follows by decreasing order of mixing proportions $\hat{\pi}_1$ and $\hat{\pi}_2$. The parameter discussion within this subsection refers to the upper component, as illustrated by the highlighted component in Figure 4.5 (blue color). In order to derive a statement on the statistical properties of the estimators the derived parameter estimates and the overall results of the MC study are discussed. Table 4.3 summarizes the MC results for the component specific parameters π_1, ν_1 and regression coefficient vector $\boldsymbol{\beta}_1$ in direct comparison to the true values. The regression coefficients β_{11} and β_{14} show a non-significant bias compared to the true values. Concerning the MC deviation the regression coefficient estimates for β_{13} exhibit the highest deviation taking into account the absolute value of the true parameter $\beta_{13} = 10$. Figure 4.6 visualizes empirical properties of the fitted regression coefficients $\hat{\beta}_{11}^{(j)}$, $\hat{\beta}_{12}^{(j)}$, $\hat{\beta}_{13}^{(j)}$ and $\hat{\beta}_{14}^{(j)}$ over all MC simulations $j = 1, \ldots, S$ by means of histograms and box-plots. The figures yield insight to the variation of the parameter estimates. The solid red line marks the true values while the dotted line gives the MC mean values of the parameter estimates from Table 4.3. As the MC bias already ranges in smaller areas, the true and estimated values almost coincide in the figures. The interquartile range (IQR), comprising the central 50% of the obtained parameter estimates, varies almost symmetrically around the true values for $\hat{\beta}_{14}^{(j)}$ for $j = 1, \ldots, S$. The remaining parameter estimates exhibit a weak skewness which can be attributed to the overlapping constellation of the two components. The parameter β_{12} contributes to the shape of the sigmoid mean function. Large deviations from the true value $\beta_{12} = -35$ may induce significant changes in the regression function. The regression coefficient estimates $\hat{\beta}_{12}^{(j)}$ range between $[-33.6; -36.9]$ with a moderate deviation from the true value. The regression coefficients $\hat{\beta}_{13}^{(j)}$ and $\hat{\beta}_{14}^{(j)}$ show tolerable ranges for the variation over all MC results. The highest MC deviation in absolute values arises for $\hat{\beta}_{11}^{(j)}$ representing the upper asymptote with occurring outliers above $62\ 000$ as visible in Figure 4.6. The MC results tend to increase the span between the two asymptotes in general as $\bar{\beta}_{11}$ exceeds the true value while $\bar{\beta}_{14}$ is slightly decreased compared to the true value. This effect can be attributed to the initiated dense structure of the synthetic data sets.

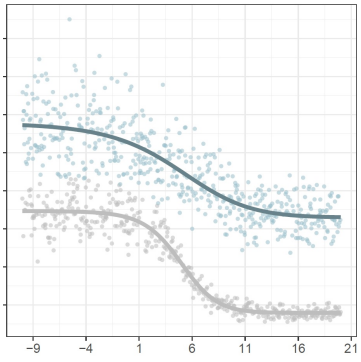

Figure 4.5: Component 1

4.4.1.2 First Component: Regression Coefficients

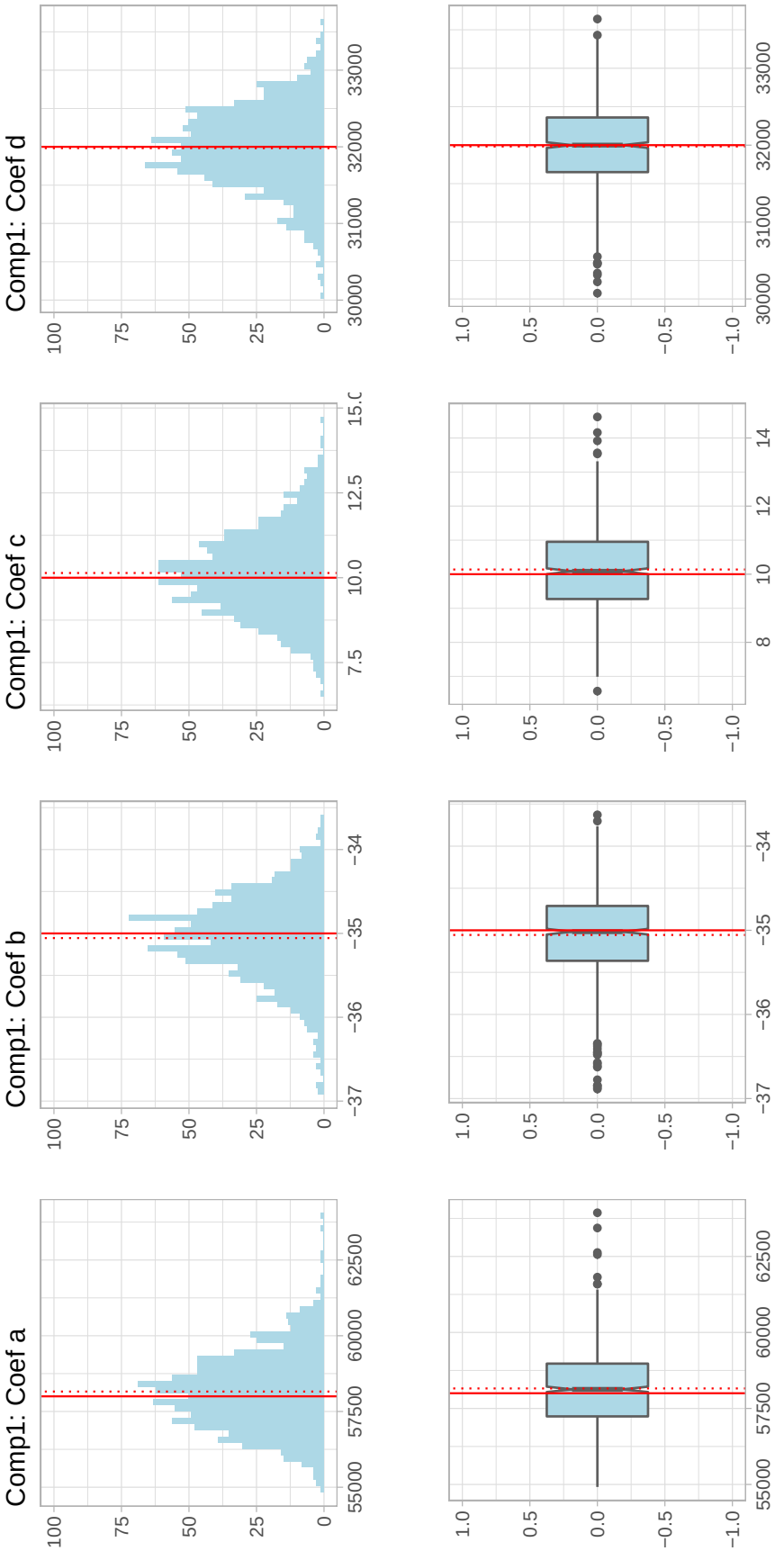

Figure 4.6: Histograms and box-plots for fitted coefficients (component 1, $n = 1000$)

4.4.1.3 Second Component

As the mixing proportions determine the identification of the components by decreasing order, the lower component in the exemplary data sample is classified as the second component with initial mixing proportion $\pi_2 = 0.4$. Figure 4.7 displays the fitted component colored in red.

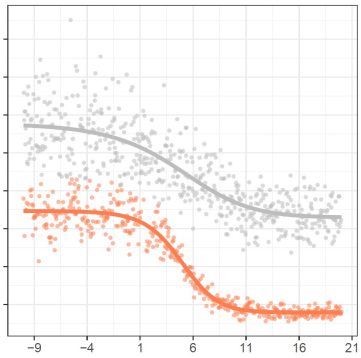

Figure 4.7: Component 2

The fitted parameters comprise estimated values for the mixing proportion π_2, the regression coefficient vector $\boldsymbol{\beta}_2$ and the shape parameter ν_2. Table 4.4 summarizes the results of the overall MC study for the second component. The results indicate smaller deviations and even a smaller bias for the estimated parameters compared to the first component. These effects can be attributed to the smaller true values inducing a decreasing variability compared to the upper component. The MC deviation shows similar results compared to the first component: the highest deviation arises for the regression coefficients β_{21} and β_{24} due to their high absolute values.

Figure 4.8 illustrates the computed regression coefficients for the second component $\hat{\beta}_{21}^{(j)}$, $\hat{\beta}_{22}^{(j)}$, $\hat{\beta}_{23}^{(j)}$ and $\hat{\beta}_{24}^{(j)}$ over all MC simulations $j = 1, \ldots, S$. The smaller deviations for the regression coefficients of the second component are reflected in the smaller ranges as illustrated in the histograms and box-plots of the parameter estimates. The regression coefficients $\hat{\beta}_{21}^{(j)}$, $\hat{\beta}_{22}^{(j)}$ and $\hat{\beta}_{23}^{(j)}$ range almost symmetrically around the true values. The MC means of the regression coefficients match the true values closely in the graphics due to the small deviation which is also reflected in the MC bias given in Table 4.4. The fitted regression coefficients $\hat{\beta}_{24}^{(j)}$ show a slight left-skewness with non-evident influence on the shape of the regression function.

4.4.1.4 Second Component: Regression Coefficients

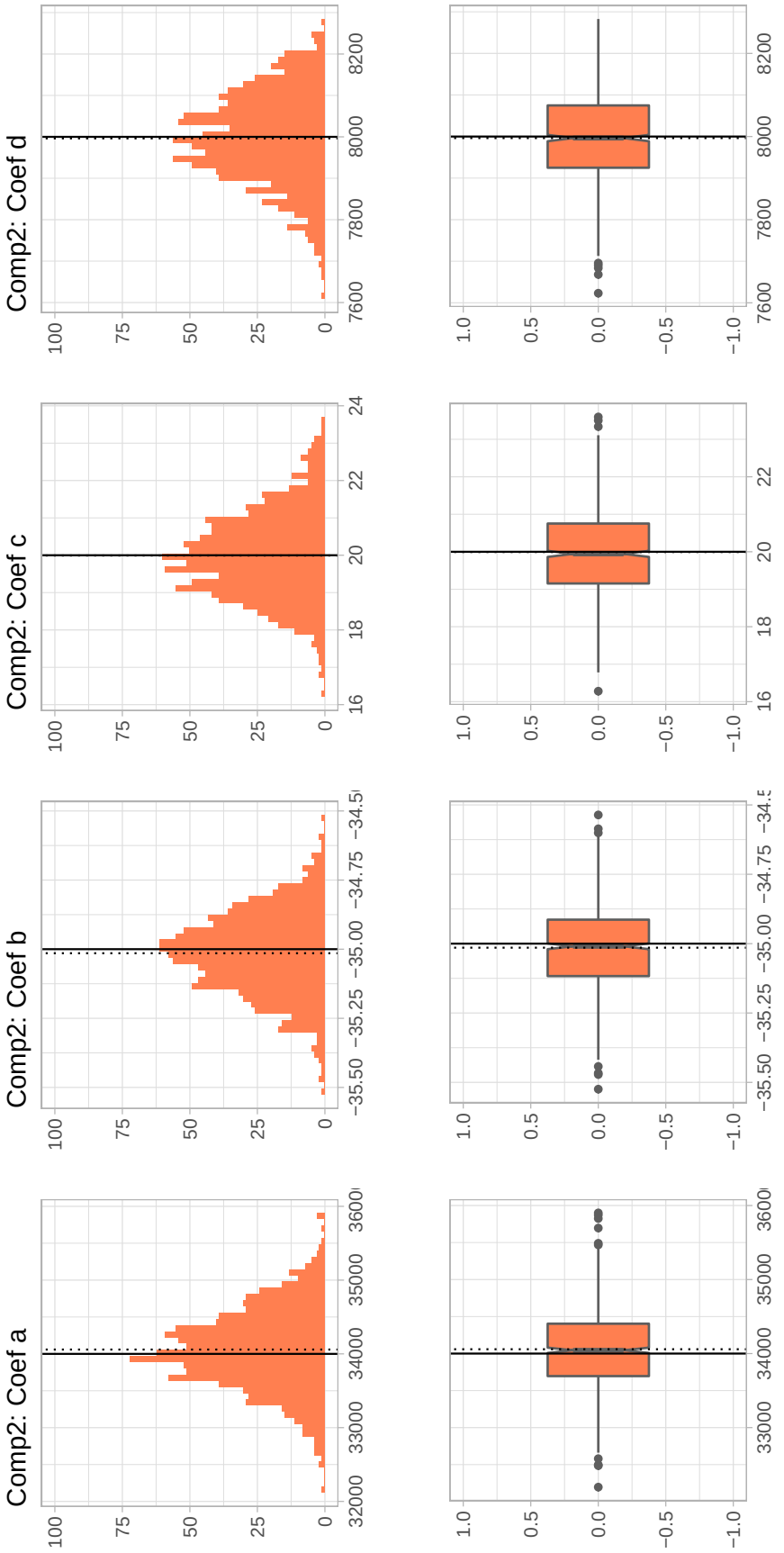

Figure 4.8: Histograms and box-plots for fitted coefficients (component 2, $n = 1000$)

4.4.1.5 Shape Parameters

The obtained estimates for the shape parameters over all MC simulations are illustrated in Figure 4.9 through histograms and box-plots. As the original parameters for both shapes coincide, the fitted values range almost within the same levels. The histogram and box-plot of the fitted shape parameters $\hat{\nu}_1^{(j)}$ for the first component are given by the blue colored graphics while those for the second component $\hat{\nu}_2^{(j)}$ appear in red color.

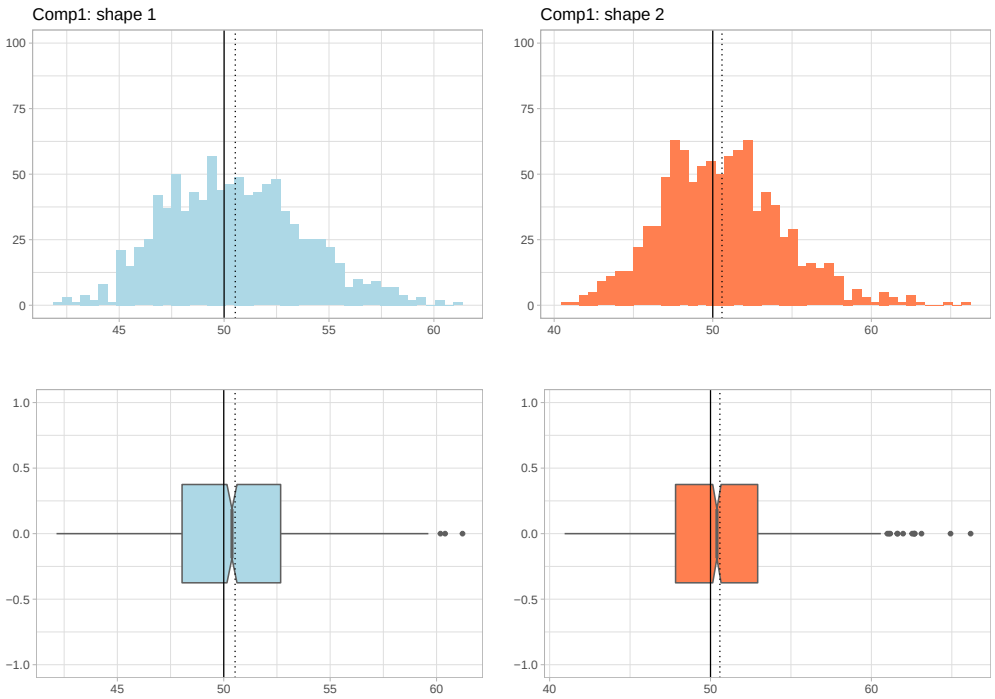

Figure 4.9: Histograms and box-plots for fitted shape parameter estimates ($n = 1000$)

The MC means (dotted line) differ from the true values (full line). Due to the overlapping of the synthetic data samples, as illustrated in Figure 4.1, the initial configuration of the data set prevents an exact assignment to the true components. This effect increases the shape parameters slightly so that the MC means of both components differ about $+0.6$ units from the true value 50.

4.4.1.6 Component Weights

The components weight estimates $\hat{\pi}_1^{(j)}$ and $\hat{\pi}_2^{(j)}$ over all MC simulations are displayed in Figure 4.10. As the results in Tables 4.3 and 4.4 already indicate, the MC mean values $\bar{\pi}_1$ and $\bar{\pi}_2$ match the true values very well. Due to the evident overlapping of the components, data points may be assessed to the other components given a sufficient deviation. This effect is highlighted in Figure 4.1 and pointed out in the rootogram for the exemplary fitting of a data set in Figure 4.4. The variation of the computed component weights is non-evident in the numerical values as the MC bias and deviation equal almost zero for both component weight estimates in Tables 4.3 and 4.4.

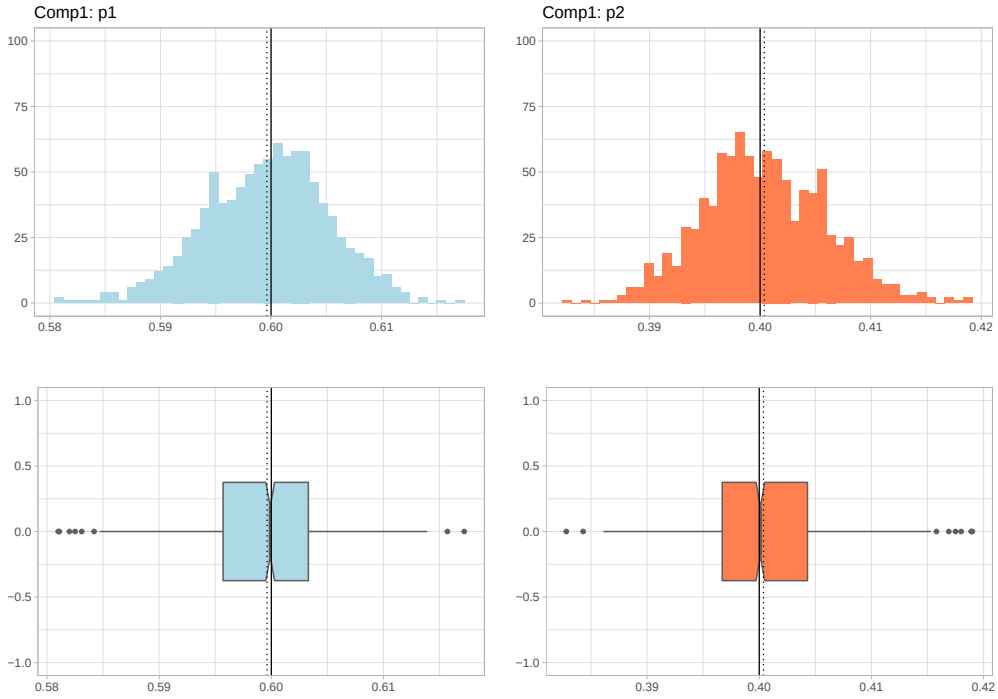

Figure 4.10: Histograms and box-plots for fitted prior weight estimates ($n = 1000$)

4.4.2 Standard Errors and Confidence Intervals

As the previous results on the parameter estimates already point out, the MC mean values for the parameter estimates fit on average. These results are underpinned with graphical illustrations for all simulated values. In order to achieve an impression on the variability of the estimated parameters, the (asymptotic) standard errors as presented in Section 2.5 are computed for all parameter estimates over all MC simulations. The MC mean of those errors qualifies a statement on the variability of the final parameter estimates. The computation of asymptotic confidence intervals as given in Equation (4.7) enables the estimation of standard errors for the MC means. A further measure of algorithm performance is provided by the coverage rate which indicates the rate of the true value lying in the computed asymptotic 95% or 99% confidence interval. A preferable value is given by the level of the confidence interval.

4.4.2.1 First Component

Table 4.3 summarizes the mean errors, confidence intervals and coverage rates for the parameter estimates of the upper component, also denoted as first component. The first two rows *True values* and *MC mean* enable the comparison of the true values to the results of the MC study. Table 4.3 comprises furthermore the MC results for standard errors derived by exact computation, denoted as $\mathrm{ASE}^{ex}(\cdot)$, and numerically computed standard errors, denoted as $\mathrm{ASE}^{num}(\cdot)$. The MC means of the standard errors indicate a tolerable variability for both methods. The outlined confidence intervals are computed on the MC mean of the parameter values and standard errors. The coverage rate results over all MC simulations by checking if the asymptotic confidence intervals contain the

true parameter value from Table 4.1. The results indicate satisfiable coverage rates over all simulations.

4.4.2.2 Second Component

Table 4.4 summarizes the mean errors, confidence intervals and coverage rates for the parameter estimates of the lower component (second component). Due to the nature of the absolute values, the MC means of the standard errors indicate a smaller variability compared to the first component. This can be attributed to evidently smaller mean values for the lower component. The outlined confidence intervals are computed on the MC mean of the parameter values and the asymptotic standard errors. The coverage rates indicate satisfiable results for the second component.

4.4.2.3 First Component: MC Results, Standard Errors and Coverage Rates

True Parameters and MC Results						
	π_1	β_{11}	β_{12}	β_{13}	β_{14}	ν_1
True values	0.60	58 000.00	-35.00	10.00	32 000.00	50.00
MC mean	0.60	58 158.38	-35.06	10.14	31 982.78	50.53
MC bias	0.00	158.38	-0.06	0.14	-17.22	0.53
MC deviation	0.01	1 277.91	0.52	1.25	534.23	3.29
Asymptotic Standard Errors						
	$\text{ASE}^{ex}(\hat{\pi}_1)$	$\text{ASE}^{ex}(\hat{\beta}_{11})$	$\text{ASE}^{ex}(\hat{\beta}_{12})$	$\text{ASE}^{ex}(\hat{\beta}_{13})$	$\text{ASE}^{ex}(\hat{\beta}_{14})$	$\text{ASE}^{ex}(\hat{\nu}_1)$
MC mean	0.02	904.42	0.33	1.21	326.34	3.63
95% ACI	$(0.55; 0.65)$	$(55\,672.06; 60\,644.43)$	$(-36.07; -34.04)$	$(7.64; 12.64)$	$(30\,933.65; 33\,034.76)$	$(44.04; 56.93)$
Coverage Rate	100.00%	96.34%	94.84%	96.67%	95.59%	96.56%
99% ACI	$(0.54; 0.66)$	$(54\,890.85; 61\,425.64)$	$(-36.39; -33.72)$	$(6.85; 13.43)$	$(30\,603.54; 33\,364.87)$	$(42.02; 58.95)$
Coverage Rate	100.00%	98.60%	99.25%	99.89%	98.71%	98.82%
Asymptotic Standard Errors						
	$\text{ASE}^{num}(\hat{\pi}_1)$	$\text{ASE}^{num}(\hat{\beta}_{11})$	$\text{ASE}^{num}(\hat{\beta}_{12})$	$\text{ASE}^{num}(\hat{\beta}_{13})$	$\text{ASE}^{num}(\hat{\beta}_{14})$	$\text{ASE}^{num}(\hat{\nu}_1)$
MC mean	0.02	903.60	0.33	1.21	326.00	3.66
95% ACI	$(0.55; 0.65)$	$(55\,671.19; 60\,645.63)$	$(-36.07; -34.04)$	$(7.64; 12.64)$	$(30\,931.98; 33\,033.34)$	$(44.03; 57.04)$
Coverage Rate	100.00%	96.17%	95.03%	96.37%	95.23%	95.65%
99% ACI	$(0.54; 0.66)$	$(54\,889.65; 61\,427.17)$	$(-36.39; -33.72)$	$(6.85; 13.43)$	$(30\,601.83; 33\,363.49)$	$(42.02; 58.95)$
Coverage Rate	100.00%	98.55%	99.28%	99.79%	98.76%	99.37%

Table 4.3: MC results, standard errors and coverage rates (component 1, $n = 1000$)

4.4.2.4 Second Component: MC Results, Standard Errors and Coverage Rates

True parameters and MC Results						
	π_2	β_{21}	β_{22}	β_{23}	β_{24}	ν_2
True values	0.40	34 000.00	-35.00	20.00	8 000.00	50.00
MC mean	0.40	34 058.64	-35.02	19.99	7 996.32	50.58
MC bias	0.00	58.64	-0.02	0.01	3.69	0.58
MC deviation	0.01	549.04	0.15	1.16	115.76	3.86
Asymptotic Standard Errors						
	$\text{ASE}^{ex}(\hat{\pi}_2)$	$\text{ASE}^{ex}(\hat{\beta}_{21})$	$\text{ASE}^{ex}(\hat{\beta}_{22})$	$\text{ASE}^{ex}(\hat{\beta}_{23})$	$\text{ASE}^{ex}(\hat{\beta}_{24})$	$\text{ASE}^{ex}(\hat{\nu}_2)$
MC mean	0.02	905.70	0.33	1.20	319.46	3.66
95% ACI	(0.36; 0.44)	(32 999.59; 35 122.77)	$(-35.31; -34.73)$	(17.78; 22.21)	(7 781.60; 8 211.97)	(42.46; 58.13)
Coverage rate	100.00%	94.52%	95.16%	94.95%	95.16%	97.85%
99% ACI	(0.35; 0.45)	(32 666.01; 35 456.35)	$(-35.40; -34.63)$	(17.09; 22.91)	(7 713.98; 8 279.59)	(40.00; 60.59)
Coverage rate	100.00%	98.27%	99.03%	98.92%	99.36%	99.68%
Asymptotic Standard Errors						
	$\text{ASE}^{num}(\hat{\pi}_2)$	$\text{ASE}^{num}(\hat{\beta}_{21})$	$\text{ASE}^{num}(\hat{\beta}_{22})$	$\text{ASE}^{num}(\hat{\beta}_{23})$	$\text{ASE}^{num}(\hat{\beta}_{24})$	$\text{ASE}^{num}(\hat{\nu}_2)$
MC mean	0.02	905.29	0.33	1.19	319.59	3.69
95% ACI	(0.36; 0.44)	(32 999.79; 35 116.06)	$(-35.30; -34.73)$	(17.78; 22.20)	(7 781.82; 8 211.15)	(42.69; 58.48)
Coverage rate	100.00%	94.51%	95.13%	94.61%	94.92%	95.65%
99% ACI	(0.35; 0.45)	(32 667.30; 35 448.55)	$(-35.40; -34.64)$	(17.09; 22.90)	(7 714.37; 8 278.60)	(40.21; 60.96)
Coverage rate	100.00%	98.24%	98.86%	98.76%	99.28%	99.59%

Table 4.4: MC results, standard errors and coverage rates (component 2, $n = 1000$)

4.5 Simulation Results (Sample Size $n = 500$)

The following sections present the results provided by the simulation study for the sample size $n = 500$ and $S = 1\ 000$ simulation runs. The parameter estimates exhibit a higher variability, as expected for decreasing sample sizes. A smaller data sample size aggravates the fitting procedure as the specific nonlinear functional structure may not be clearly defined and outlier become crucial in the fitting process. The algorithm faces therefore higher demands concerning accuracy. In order to improve the fitting procedure, the shape parameter for the first component is increased to $\nu_1 = 55$ within the subsequent simulations. The remaining parameters from the initial configuration in Table 4.1 stay unchanged as well as the random initial cluster assignment of the sample points. The increase in complexity due to the challenging setup is reflected by a convergence rate (converging trials) corresponding to 84%. The accurately converged trials were identified according to the restrictions in Section 4.3.1 which indicated an exclusion of 2% (misfits) of the fitted results. Among the accurately converged trials the number of iterations spans between 16 and 44. The median and the mean number of iteration steps correspond to 22. The following sections give an analogous discussion on the provided estimates and their quality in order to address the sensitivity of the algorithm's performance to a smaller sample size.

4.5.1 Parameter Estimates

The subsequent sections discuss the provided parameter estimates and their quality for the two-component Gamma mixture model (4.1) with nonlinear mean function (4.2) and sample size $n = 500$.

4.5.1.1 First Component

The identification of the components follows by decreasing order of mixing proportions $\hat{\pi}_1$ and $\hat{\pi}_2$ according to the specifications in Section 2.2. The parameter discussion within this subsection refers to the upper component, as illustrated in Figure 4.11 (blue color). Table 4.3 summarizes the results of the MC study for the component specific parameters π_1, $\boldsymbol{\beta}_1$ and ν_1 and their true values. The results correspond in general to those of the upper component for the large sample size whereas the regression coefficients β_{11} and β_{14} show a higher bias. Figure 4.12 visualizes the fitted regression coefficients $\hat{\beta}_{11}^{(j)}$, $\hat{\beta}_{12}^{(j)}$, $\hat{\beta}_{13}^{(j)}$ and $\hat{\beta}_{14}^{(j)}$ over all MC simulations $j = 1, \ldots, S$ by means of histograms and box-plots. Deviations from the previous results are given by slightly broader ranges which can be attributed to the greater variability induced by a smaller sample size.

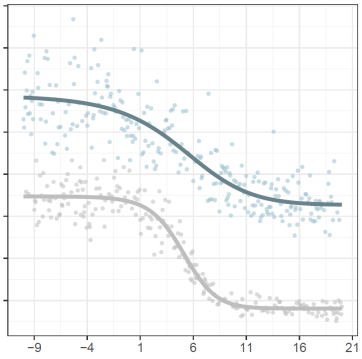

Figure 4.11: Component 1

4.5.1.2 First Component: Regression Coefficients

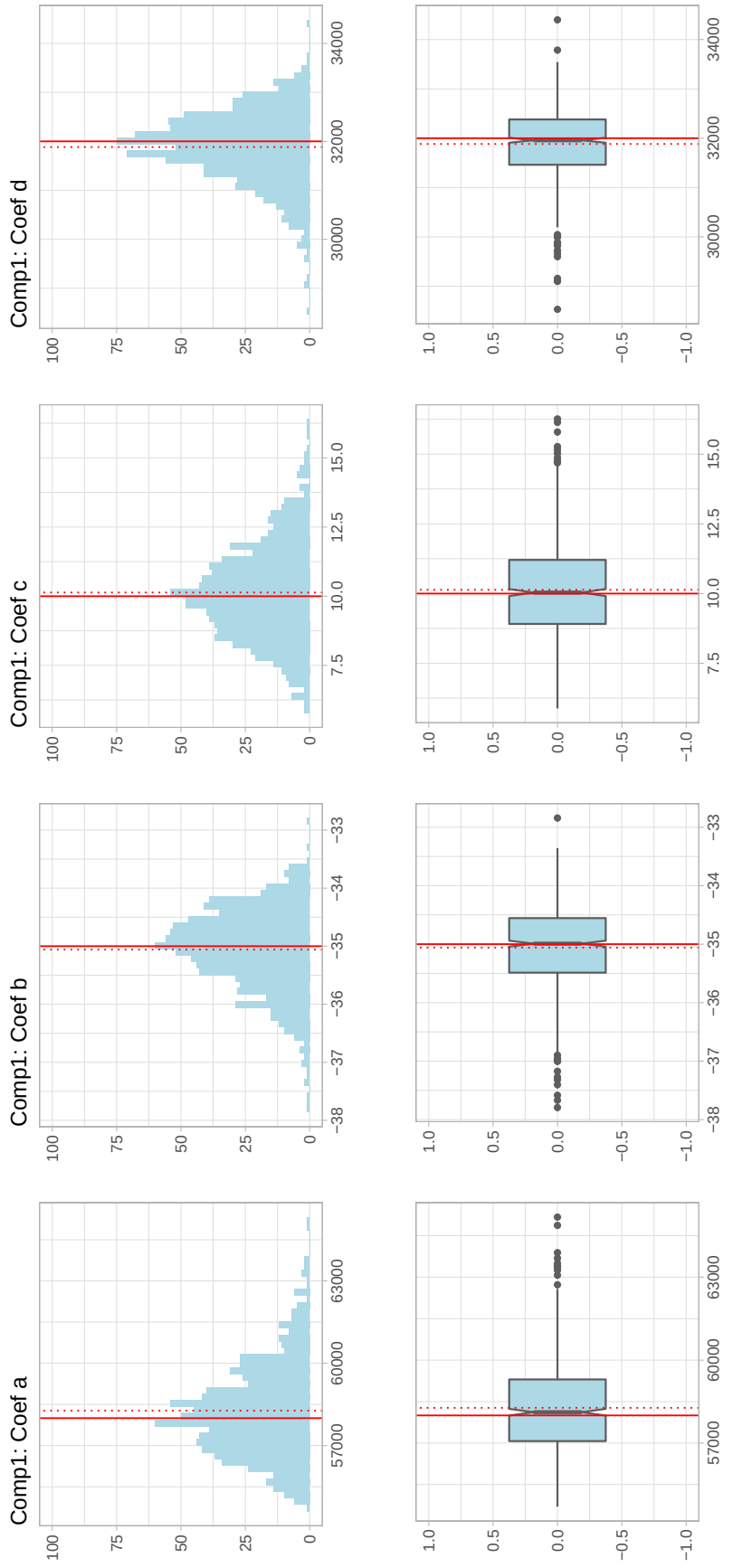

Figure 4.12: Histogram and box-plots for fitted coefficients (component 1, $n = 500$)

4.5.1.3 Second Component

The second and lower component arises through the initial mixing proportion $\pi_2 = 0.4$. An illustration of the fitted component is emphasized by the colored sample points given in Figure 4.13. The estimated parameters comprise the mixing proportion π_2, the regression coefficient vector $\boldsymbol{\beta}_2$ and the shape parameter ν_2 and are given in Table 4.6. Figure 4.14 illustrates the obtained regression coefficients for the second component $\hat{\beta}_{21}^{(j)}$, $\hat{\beta}_{22}^{(j)}$, $\hat{\beta}_{23}^{(j)}$ and $\hat{\beta}_{24}^{(j)}$ over all MC simulations $j = 1, \ldots, S$. The typical smaller ranges for the second component are evident in the histograms and box-plots in Figure 4.14. The MC means of the regression coefficients coincide graphically with the true values due to the small deviation which is also reflected in the MC bias of the estimates in Table 4.6. The MC means and graphics show coherent results compared to the sample size $n = 1\,000$.

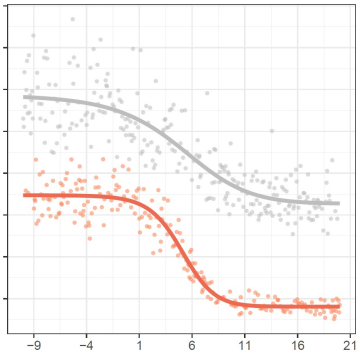

Figure 4.13: Component 2

4.5.1.4 Second Component: Regression Coefficients

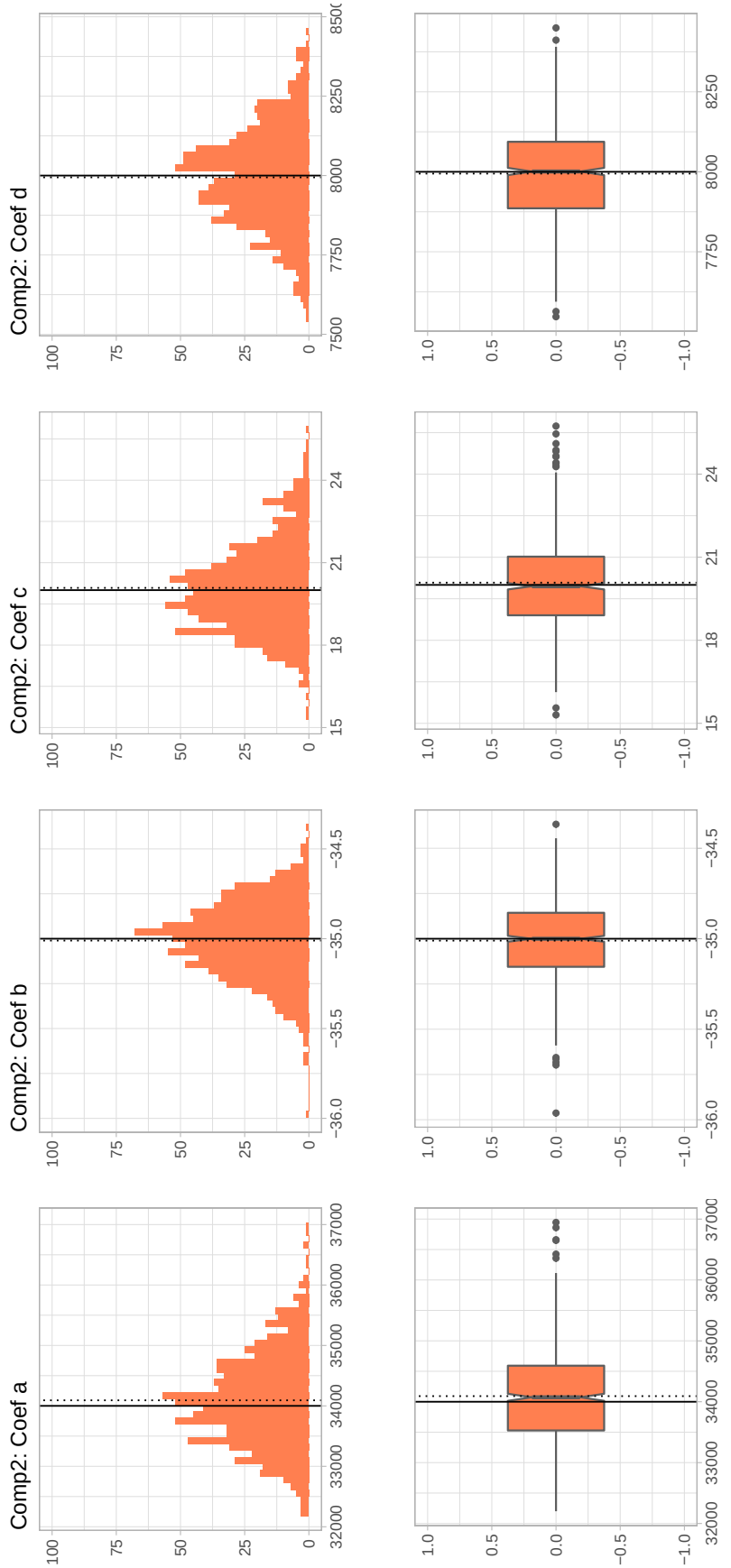

Figure 4.14: Histogram and box-plots for fitted coefficients (component 2, $n = 500$)

4.5.1.5 Shape Parameters

The estimated shape parameters differ from the previously discussed sample as the MC deviation and bias have increased. This is reflected in the Figure 4.15 where the shape parameters for both components over all MC simulations span a greater range. The estimates $\hat{\nu}_1^{(j)}$ are given through the blue-colored graphics for the first component while the shape parameter estimates for the second component $\hat{\nu}_2^{(j)}$ are red-colored for $j = 1, \ldots, S$.

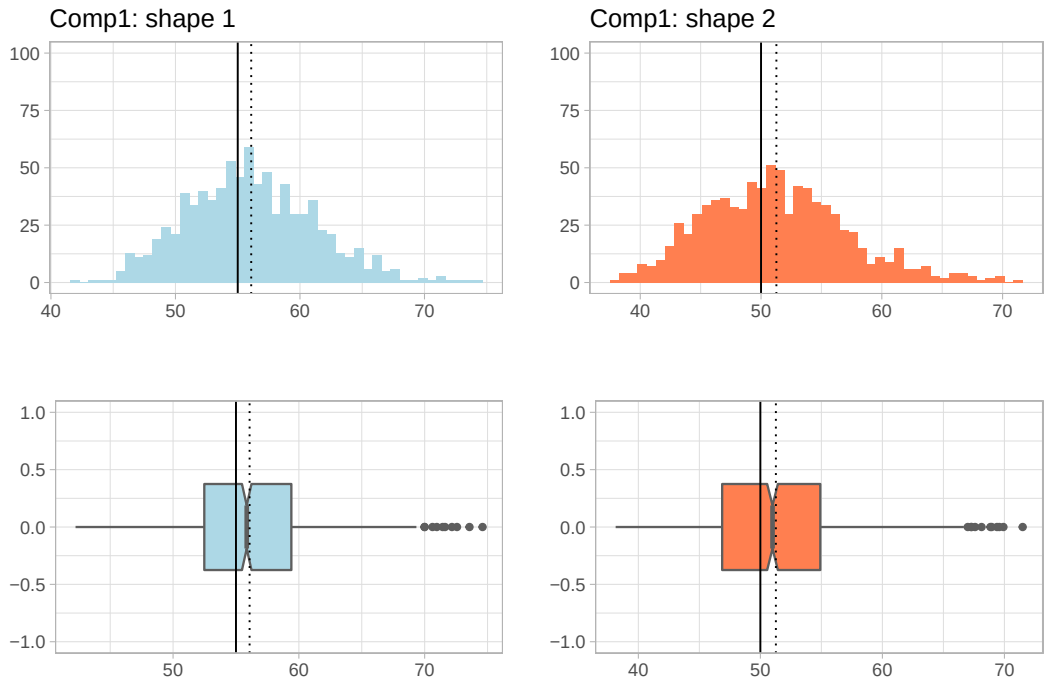

Figure 4.15: Histograms and box-plots for fitted shape parameter estimates ($n = 500$)

Due to the overlapping of the synthetic data samples, the MC means (dotted line) differ from the true values (full line). This effect indicates a smaller dispersion for both components and a slight right-skewness.

4.5.1.6 Component Weights

Figure 4.16 illustrates the fitted component weights of the provided MC simulations graphically. As the results in Tables 4.5 and 4.6 already indicate, the MC mean values $\bar{\pi}_1$ and $\bar{\pi}_2$ differ slightly from the true values. The upper component shows a slightly smaller MC mean value compared to the true value which can be attributed to the overlapping of the respective components.

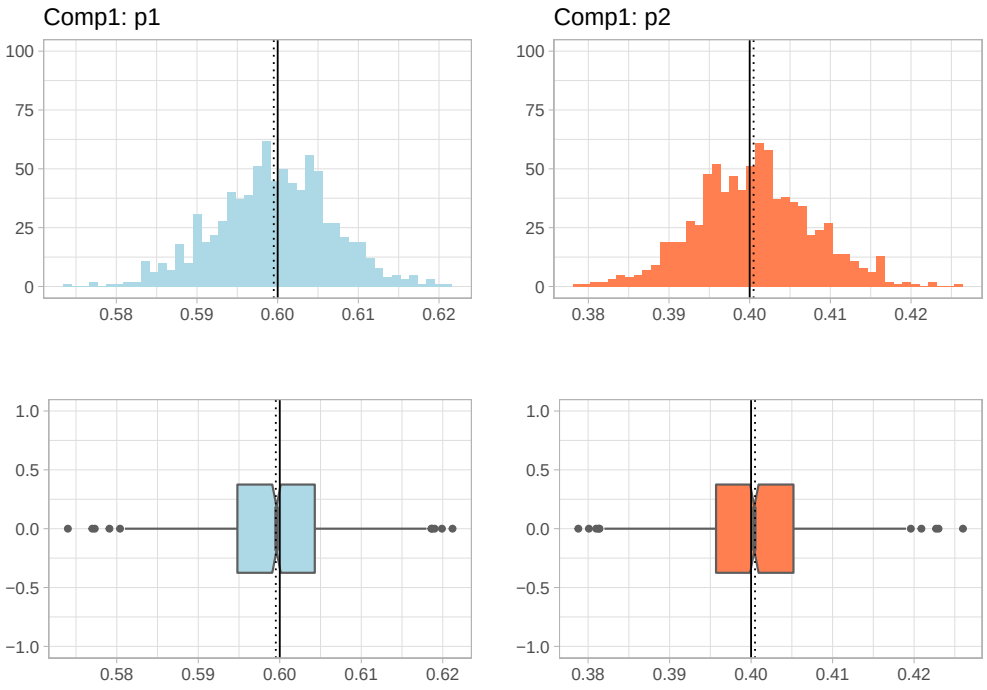

Figure 4.16: Histograms and box-plots for fitted prior weight estimates ($n = 500$)

4.5.2 Standard Errors and Confidence Intervals

The MC estimates for the sample size $n = 500$ fit again on average. These results are underpinned with graphical illustrations for all fitted values. In order to express the variability of the estimated parameters, the standard errors as presented in Section 2.5 are computed for all parameter estimates over all MC simulations. In contrast to the previous discussion, the smaller sample size produces numerical difficulties computing the standard errors through the exact derivation $\mathrm{SE}^{ex}(\cdot)$ of the Hessian matrix as given in Section 2.5. Therefore the results are shown for the numerically derived standard errors $\mathrm{SE}^{num}(\cdot)$ as discussed in Section 3.5. The MC mean of the standard errors states again the asymptotic standard errors. A further measure of algorithm performance is given again by the coverage rate. Tables 4.5 and 4.6 summarize the mean errors, confidence intervals and coverage rates for the parameter estimates of the two components. The MC means of the standard errors indicate a tolerable variability for the parameter estimates and enable the computation of the confidence intervals. The computed coverage rates show satisfying results.

4.5.2.1 First Component: MC Results, Standard Errors and Coverage Rates

True Parameters and MC Results						
	π_1	β_{11}	β_{12}	β_{13}	β_{14}	ν_1
True values	0.60	58 000.00	−35.00	10.00	32 000.00	55.00
MC mean	0.60	58 274.96	−35.06	10.14	31 883.45	56.08
MC bias	0.00	274.96	−0.06	0.14	−116.55	6.08
MC deviation	0.01	1 697.33	0.72	1.74	740.80	5.19
Asymptotic Standard Errors						
	$\text{ASE}^{num}(\hat{\pi}_1)$	$\text{ASE}^{num}(\hat{\beta}_{11})$	$\text{ASE}^{num}(\hat{\beta}_{12})$	$\text{ASE}^{num}(\hat{\beta}_{13})$	$\text{ASE}^{num}(\hat{\beta}_{14})$	$\text{ASE}^{num}(\hat{\nu}_1)$
MC mean	0.03	1 253.58	0.46	1.67	444.69	5.45
95% ACI	(0.53; 0.67)	(54 804.57; 61 745.35)	(−36.47; −33.65)	(6.75; 13.52)	(30 425.83; 33 341.06)	(45.95; 66.22)
Coverage rate	100.00%	95.84%	95.60%	95.01%	96.43%	95.60%
99% ACI	(0.51; 0.69)	(53 714.10; 62 835.82)	(−36.91; −33.21)	(5.69; 14.58)	(29 967.81; 33 799.08)	(42.77; 69.40)
Coverage rate	100.00%	99.17%	98.69%	98.93%	99.41%	99.05%

Table 4.5: MC results, standard errors and coverage rates (component 1, $n = 500$)

4.5.2.2 Second Component: MC Results, Standard Errors and Coverage Rates

True Parameters and MC Results						
	π_2	β_{21}	β_{22}	β_{23}	β_{24}	ν_2
True values	0.40	34 000.00	-35.00	20.00	8 000.00	50.00
MC Mean	0.40	34 092.15	-35.01	20.08	7 994.38	51.27
MC bias	0.00	92.15	-0.01	0.08	-5.62	1.27
MC deviation	0.01	780.07	0.21	1.64	156.95	5.90
Asymptotic Standard Errors						
	$\text{ASE}^{num}(\hat{\pi}_2)$	$\text{ASE}^{num}(\hat{\beta}_{21})$	$\text{ASE}^{num}(\hat{\beta}_{22})$	$\text{ASE}^{num}(\hat{\beta}_{23})$	$\text{ASE}^{num}(\hat{\beta}_{24})$	$\text{ASE}^{num}(\hat{\nu}_2)$
MC Mean s.e.	0.02	1 274.87	759.42	1.78	147.73	5.66
95% ACI	$(0.34; 0.46)$	$(32\,606.88; 35\,577.43)$	$(-35.42; -34.60)$	$(16.93; 23.28)$	$(7\,692.15; 8\,296.61)$	$(39.98; 62.56)$
Coverage Rate	100.00%	94.89%	95.72%	95.00%	94.41%	94.65%
99% ACI	$(0.33; 0.58)$	$(32\,140.18; 36\,044.13)$	$(-35.55; -34.48)$	$(15.93; 24.23)$	$(7\,597.18; 8\,391.58)$	$(36.44; 66.10)$
Coverage Rate	100.00%	99.17%	99.29%	99.29%	98.93%	99.52%

Table 4.6: MC results, standard errors and coverage rates (component 2, $n = 500$)

4.6 Conclusions

The present MC simulation study was performed in order to achieve an impression on the functionality and performance of the new model class `"FLXMRnlm"` for fitting mixtures of GNMs. For this purpose, a two-component Gamma mixture model was considered where the mean followed a sigmoid regression function. Based on synthetic data sets, the fitting algorithm was executed for different sample sizes in order to reveal sensitivities of the underlying methods. The complexity increased with the large scattering of one component and the random setting of the initial classification of the sample points. The derived results for the component specific parameters are satisfactory as they fit on average and show a manageable variability. The latter was assessed through the computation of standard errors based on the marginal log-likelihood function. Graphical visualizations of the fitted values revealed also manageable deviations around the true values.

The simulation study revealed also technical limits and numeric obstacles which can be related to the specifics of the nonlinearity of the underlying regression function and the application of the EM algorithm. For one thing, the fitting of different sample sizes came along with fitting problems. The fitting of mixtures of GNMs requires obviously at least a clear shape of the functional structure in order to identify components along these functions. The application of the Gamma distribution in combination with a sigmoid regression function came along with colinearity when deriving the Hessian matrix of the marginal log-likelihood function. Therefore, one method for computing standard errors experienced numerical difficulties for the smaller data set. The application of the EM algorithm for mixtures of GNM showed also the possibility of misfits by means of results converging to wrong solutions which were not suitable in order to describe the nature of the data set. A sensible initialization of the algorithm and the analysis of the derived results remains a problem specific task when fitting mixtures of GNMs. Nonetheless, the derived parameter estimates proved to be reliable due to satisfactory coverage rates over all simulations runs and coherent MC simulation results for both sample sizes.

CHAPTER 5

Modeling Gas Flow on Exits of Gas Transmission Networks

Introduction

The application of mixtures of GNMs is naturally related to real world applications based on cross-sectional regression where the natural shape of the response exhibits a nonlinear functional structure and heterogeneity due to latent groups. An appropriate application is given by the modeling of gas flow which typically exhibits a nonlinear decreasing consumption pattern subject to the outside temperature. Heterogeneity may appear due to the non-constant variability of the pattern and possibly differing consumption levels. The natural nonlinear shape of gas flow motivates the use of a sigmoid regression function as basic framework for standard loading profiles. The latter enables the prediction of gas consumption in gas transmission networks. Detailed research on their functional structure is given by Hellwig (2003), BDEW et al. (1990) and Koch et al. (2015). The current framework agreement on standard loading profiles can be accessed on the web page of the *Federal Association of the German Energy and Water Industries (BDEW)* through BDEW (2018). The application of the sigmoid function was also studied for the Austrian market provided by the study *"Lastprofile nicht-leistungsgemessener Kunden"* by Almbauer (2008) and can be accessed on the web page of the *AGCS Gas Clearing and Settlement AG* through AGCS (2018). The modeling of gas flow presents a suitable application of the new model class of mixtures of GNMs. The main focus lies on the application of the sigmoid mean function within mixture models which was primary motivated by Friedl et al. (2012, p. 31).

Friedl et al. (2012) studied historical data of gas consumption in the gas supplier's interest with the aim to improve the gas supply and reduce operational costs. They included daily mean temperatures and the information of working days and holidays as explanatory variables within their statistical analysis. The relationship between gas flow data and the mean temperature appeared nonlinear in the specific form of sigmoid growth models. Presuming an underlying growth function as smooth curve for the given data Friedl et al. (2012) studied two different types of regression models: parametric sigmoid

regression and a semi-parametric approach provided by penalized splines (P-splines). A crucial criterion for a suitable model was the flawless gas supply prediction of extraordinary negative temperatures denoted as *design temperatures*. Friedl et al. (2012) chose the P-spline approach due to its flexibility and predictions of design temperatures. This thesis presents new approaches in order to predict the demanded gas supply for design temperatures. Therefore this research takes on the parametric nonlinear models used in Friedl et al. (2012) which are capable of advancement in two ways. Since the authors concentrated on the normal distribution for their statistical models, this distributional assumption can be extended. Friedl et al. (2012, p. 31) furthermore emphasize the use of **flexmix** for mixture models and presume improving results for mixtures of sigmoid models. The presented extensions in Section 3.4 enable the fitting of gas flow data through mixtures of sigmoid regression models for the first time.

The objective of this research is to present mixtures of GNMs as a suitable semi-parametric approach for modeling and predicting gas flow. The subsequent applications deal with various gas flow patterns and present the functionality of the new fitting procedure for mixtures of GNMs. The reliability of the results is underpinned by the computation of standard errors. Special attention is given to predictions of gas flow for low temperatures. These are outlined with the corresponding prediction intervals in order to picture the inherent variability. Section 5.1 presents the model framework for the subsequent applications comprising the sigmoid mean functions and the mixture models for the normal and Gamma distribution. It furthermore outlines the corresponding fitting commands in R and the specification of predictions within mixture models. Section 5.2 deals with the work by Friedl et al. (2012) and extends the results by further models. A second gas flow pattern is discussed in Section 5.3 demonstrating the advantages of the Gamma distribution within the identification of evident consumer specific subgroups. Section 5.4 analyzes a gas flow data set with similar appearance to the synthetic data set used within the simulation study in Chapter 4. The real world data represent a suitable complement to the provided simulation study proving the adequacy of the new fitting method to real data.

5.1 Model Framework

As Friedl et al. (2012) point out gas suppliers agree to use a nonlinear regression model, particularly a sigmoid growth curve, as statistical model for gas flow and its predictions for design temperatures. In the following research two models will be picked out of their study in order to provide further analysis.

5.1.1 Sigmoid Mean Functions

The first regression model comprises four model parameters given by $\boldsymbol{\beta} := (\beta_1, \beta_2, \beta_3, \beta_4)$ corresponding to the regression coefficients. The nonlinear regression function is given through the sigmoid function

$$\mu_i(\boldsymbol{\beta}) = \mu(x_i, \boldsymbol{\beta}) = \beta_4 + \frac{\beta_1 - \beta_4}{1 + \left(\frac{\beta_2}{x_i - 40}\right)^{\beta_3}}, \quad i = 1, \ldots, n, \tag{5.1}$$

where the regression coefficients incorporate a physical meaning. Coefficients β_1 and β_4 describe upper and lower horizontal asymptotes of the sigmoid curve while β_2 and β_3 affect the shape and decrease of the curve with increasing temperature values. These parameters can be interpreted according to the energy industry where the lower bound β_4 incorporates a constant share of energy (warm water supply or share energy). The difference $\beta_1 - \beta_4$ indicates the decrease in gas consumption on cold days. The coefficient β_2 measures the change in gas consumption due to cold periods while β_3 refers to the dependence on the heating period. For a detailed explanation of the coefficients' physical meaning reference is made to the original paper by Friedl et al. (2012). The predictor variables within the Model (5.1) are denoted as x_i and consist within this thesis of maximum daily temperatures influencing the gas consumption. This thesis gives particular attention to the distributional properties of the maximum gas flow data.

Gas consumption raises the question of the statistical significance of working days and holidays as, for example, industrial consumers may exhibit a heterogeneous consumption behavior depending on their working times. The available gas consumption data allow for a classification between working days and holidays. Let the dummy variable d_i denote whether or not observation x_i relates to a working day or a holiday, respectively

$$d_i = \begin{cases} 1, & \text{holiday} \\ 0, & \text{working day.} \end{cases}$$

Holidays refer in general to non-working days which comprise Saturdays and Sundays and, if available, public holidays. Including the working day component to the scope of the sigmoid regression function (5.1) motivates the extension to the functional form

$$\mu_i(\boldsymbol{\beta}) = \mu(x_i, \boldsymbol{\beta}) = \beta_4 + \frac{\beta_1 - \beta_4}{1 + \left(\frac{\beta_2}{x_i - 40} + \beta_5 \cdot d_i\right)^{\beta_3}}, \quad i = 1, \ldots, n, \tag{5.2}$$

where the regression coefficient vector is now given by $\boldsymbol{\beta} := (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$. The properties of the first four regression coefficients stay unchanged regarding their physical meanings while β_5 relates to an additional predictor variable within Model (5.2). The nonlinear regression models (5.1) and (5.2) will also be denoted as *model 1* and *model 2* in the following.

5.1.1.1 Application Environment in R

Normal Distribution

The fitting of Model (5.1) is provided according to the procedures from Section 3.2.1. An exemplary execution and the standard summary output in R is given in the listings below.

```
> m1 = nls(max.flow ~ d + I((a-d)/(1+(b/(temp-40))^(c))), data=data,
+          start=list(a=..,b=..,c=..,d=..))
```

Listing 5.1: Fitting command with `nls()` for Model (5.1)

The response `max.flow` is fitted according to the given expression on the RHS in the formula which corresponds to Model (5.1). The coefficients `a,b,c,d` correspond to the regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)$ while `temp` denotes the predictor x_i. The data frame `data` comprises the response and predictor values. The parameter `start` necessitates a list of starting values for the coefficients `a,b,c,d`. The fitted output provided by `nls()` is stored in `m1`. The fitted parameters and the corresponding standard errors can be accessed through the command `summary(nls())`.

The fitting of Model (5.2) in R can be performed similar to the previous procedure for Model (5.1). Including a daily component affords primary a change in the formula of the fitting function `nls()`.

```
> m2 = nls(max.flow ~ d + I((a-d)/(1+(b/(temp-40)+dd*(day == 1))^(c))),
+           data=data, start=list(a=..,b=-..,c=..,d=..,dd=..))
```

Listing 5.2: Fitting command with `nls()` for Model (5.2)

When using `nls()` the holiday indicator variable can be included as an additional coefficient given by `dd` to address the daily component as given on the RHS of the formula in Listing 5.2. The remaining arguments stay the same as in the previous fitting of Model (5.1). The output is stored in object `m2`. The fitted regression coefficients, their standard errors and significance within the fitted model can be accessed using `summary(nls())`.

Gamma Distribution

The fitting of models (5.1) and (5.2) under a Gamma distribution can be provided by the package **gnm** which was discussed in Section 3.2.2 in detail. The specification and execution is given in the following listing.

```
> m3 = gnm(formula = max.flow ~ -1 + gas.1(temp), data= data,
+          start = c(a=..,b=..,c=..,d=..), family = Gamma(link="identity")
    )
```

Listing 5.3: Fitting command with `gnm()` for Model (5.1)

The argument `gas.1(temp)` on the RHS of the model formula represents the mean function and is equivalent to the functional form already specified in Listing 4.1 corresponding to Model (5.1). The mean function is specified for the predictor `temp`. The regression coefficients $\beta_1, \ldots, \beta_4$ are included in the specification by the variables `a,b,c,d`. The R command `gnm()` requires starting values which are specified in the argument `start` as vector. The underlying Gamma distribution is specified through setting the argument `family=Gamma()` where the identity link is chosen as a general approach within nonlinear regression. The fitted values can be accessed by executing the command `summary(gnm())`.

The function `gnm()` provides the possibility of fitting Model (5.2) with a holiday indicator. For this purpose the functional form of `gas.1(temp)` has to be modified including the vector with the dummy variables. The new mean function is stored in `gas.2(temp,day)`.

The specification is given in the following listing.

```
1  > gas.2 = function(temp,day,predictors){
2  +   list(predictors=list(a=1,b=1,c=1,d=1,dd=1),
3  +         variables = list(substitute(temp),substitute(day)),
4  +         term = function(predictors, variables) {
5  +         paste(predictors[4],"+(",predictors[1],"-",predictors[4],")/(",
6  +                 "1+(",predictors[2],"/(",variables[1],"-40",")+",
7  +                 variables[2],"*", predictors[5],")^", predictors[3],")",
8  +                 sep="")
9  +         })
10 + }
11 > class(gas.2) ="nonlin"
```

Listing 5.4: Specification of Model (5.2) with **gnm**

The function `gas.2()` is specified in dependence of two variables which are placeholders for the predictors `temp` and `day`. The vector `predictors` (line 2) comprises the regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$. The specification of the functional form follows in lines 4 to 8 as sequence of character variables stored in the argument `term`. The fitting call is provided again by executing the command `gnm()`.

```
1 > m4 = gnm(formula = max.flow ~ -1+gas.2(temp,day),
2 +      start=c(a=..,b=..,c=..,d=..,dd=..),family=Gamma(link="identity"))
```

Listing 5.5: Fitting command with `gnm()` for Model (5.2)

Analogous to the fitting of Model (5.1) the response `max.flow` is fitted according to the mean function `gas.2()` on the RHS of the formula argument with predictors `temp` and `day`. The required starting values are passed by a vector to the argument `start` while the Gamma distribution and the identity link are specified by the argument `family`. The coefficients can be accessed again by `summary(gnm())`.

5.1.2 Mixture Distributions

The subsequent applications relate to two-component and three-component mixture models with the normal and Gamma as possible underlying component specific distributions. The specification of the mixture model requires the fitting of the related unknown parameters by the EM algorithm.

5.1.2.1 Two-Component Mixture Model

Presuming a two-component mixture model yields the mixture density

$$f^M(y_i; \mu_i(\boldsymbol{\beta}), \boldsymbol{\phi}, \boldsymbol{\pi}) = \pi_1 f(y_i; \mu_i(\boldsymbol{\beta}_1), \phi_1) + (1 - \pi_1) f(y_i; \mu_i(\boldsymbol{\beta}_2), \phi_2), \tag{5.3}$$

where $f(\cdot)$ represents the univariate pdf and π_1 the prior probability for the first mixture component. The parameter vector $\boldsymbol{\Psi}$ summarizes all unknown parameters as

$$\boldsymbol{\Psi} = (\pi_1, \boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \phi_1, \phi_2)^\top,$$

where the dispersion parameters correspond to the variances as $\phi_1 = \sigma_1^2$ and $\phi_2 = \sigma_2^2$ in the case of a normal distribution. In the case of an underlying Gamma distribution

Mod.	Distr.	Comp.	$\boldsymbol{\pi}$	$\boldsymbol{\beta}_k$	ϕ_k	#Par.
Model (5.1)	Normal	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14})$	$\phi_1 = {\sigma_1}^2$	11
		k=2		$\boldsymbol{\beta}_2 = (\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24})$	$\phi_2 = {\sigma_2}^2$	
	Gamma	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14})$	$\phi_1 = {\nu_1}^{-1}$	11
		k=2		$\boldsymbol{\beta}_2 = (\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24})$	$\phi_2 = {\nu_2}^{-1}$	
Model (5.2)	Normal	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15})$	$\phi_1 = {\sigma_1}^2$	13
		k=2		$\boldsymbol{\beta}_2 = (\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25})$	$\phi_2 = {\sigma_2}^2$	
	Gamma	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15})$	$\phi_1 = {\nu_1}^{-1}$	13
		k=2		$\boldsymbol{\beta}_2 = (\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25})$	$\phi_2 = {\nu_2}^{-1}$	

Table 5.1: Parameters of interest for two-component mixture models

the dispersion parameters correspond to the reciprocal shape parameters through the relation $\phi_1 = \nu_1^{-1}$ and $\phi_2 = \nu_2^{-1}$. Table 5.1 outlines all unknown parameters within a two-component mixture model of GNMs (5.3) for the mean functions (5.1) and (5.2).

5.1.2.2 Application Environment in R

All applications are carried out with the new model class `"FLXMRnlm"` wrapped up in the package **flexmixNL**. The specific commands and the access of results in R will be outlined for demonstrative purposes within the present section. For detailed explanations regarding the functionality of **flexmix** and **flexmixNL** reference is made to Chapter 3. The reproducibility of the results is enabled through predefined starting values and a fixed seed for the random number generation. These settings are outlined in the following application specific sections.

The command for fitting a two-component mixture of GNMs with normally distributed responses is given in Listing 5.6.

```
1 > library(flexmixNL)
2 > formula = max.flow ~ d + I((a-d)/(1+(b/(temp-40))^(c)))
3 > m1 = flexmix(max.flow ~ temp, k = 2, data = data,
4 +      model = list(FLXMRnlm(formula = formula, family="gaussian",
5 +                   start = list(list(a=..,b=..,c=..,d=..),..))))
```

Listing 5.6: Fitting command for two-component normal mixtures with `flexmix()`

The first line loads the package **flexmixNL** to enable the fitting of nonlinear mixture models in R. The function `flexmix()` contains arguments designed for a formula expression, the number of components and the specific mixture model. The first formula `max.flow ~ temp` serves as a placeholder which does not enter the computation of GNMs in **flexmixNL**. The number of components `k = 2` in the third line determines a two-component mixture framework. Starting with the fourth code line the distributional and functional framework of the two-component mixture model is specified. The

slot `formula` determines the mean function corresponding to the functional relationship of Model (5.1) similar to the specification of single nonlinear regression. The output is stored in `m1`. In the present setup the EM algorithm, as discussed in Section 2.4, requires appropriate starting values for the regression coefficient vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)$. The starting values are in general of considerable importance in order to achieve convergence within the EM algorithm for mixture models. Arbitrary starting values may not ensure convergence for the provided method which buttress the specification of starting values sufficiently close to the true values. Within the new package **flexmixNL** for mixtures of GNMs starting values have to be chosen for every component and every regression coefficient separately. Technically, the starting values are included as input data in the command `flexmix()` through the list `start` within the model specification `FLXMRnlm(..., start=list(...),...)` for each component. In the particular case of nonlinear regression models it is of special importance that starting values reflect at least the rough shape of the real results. The fitting of mixture models with **flexmix** affords furthermore an initial cluster assignment. The default assignment is randomly set.

The fitting of a two-component Gamma mixture model is executed through the command given in the following listing:

```
1 > formula = max.flow ~ -1 + gas.1(temp)
2 > m2 = flexmix(max.flow ~ temp, data = data, k = 2,
3 +       model = list(FLXMRnlm(formula = formula, family="Gamma",
4 +                start = list(list(a=..,b=..,c=..,d=..),..))))
```

Listing 5.7: Fitting command for two-component Gamma mixtures with `flexmix()`

The distributional specification is given by the argument `family="Gamma"`. The functional relationship is specified through the argument `formula` according to the predefined framework within the package **gnm**.

Another important aspect within the EM approach is an appropriate choice of the number of components for the mixture of GNMs. The assumed number of components is supposed to be reasonable for the given data. Special emphasis is given to identification problems arising due to empty components or equally parametrized components as discussed in Section 2.2. Nevertheless, an increasing number of components correlates with an increasing fitting effort due to additional parameters. Weighing up the additional benefit of further components with the simultaneously increasing complexity of the mixture model remains a problem specific task. Therefore regardless of the remaining model specification the choice of the number of components represents a crucial step when fitting mixture models. The specification of the number of components aims at an explainable choice for the given data set and an adequate choice for the fitting procedure. Statistically, the choice is underpinned by appropriate model selection criteria such as those presented in Section 2.7.1 which can be accessed through the function calls `AIC()`, `BIC()` and `ICL()`.

5.1.2.3 Three-Component Mixture Model

Presuming three mixture components yields the mixture density

$$f^M(y_i;\mu_i(\boldsymbol{\beta}),\boldsymbol{\phi},\boldsymbol{\pi}) = \pi_1 f(y_i;\mu_i(\boldsymbol{\beta}_1),\phi_1)+\pi_2 f(y_i;\mu_i(\boldsymbol{\beta}_2),\phi_2)+(1-\pi_1-\pi_2)f(y_i;\mu_i(\boldsymbol{\beta}_3),\phi_3).$$

The unknown parameter vector $\boldsymbol{\Psi}$ comprises therefore the following parameters,

$$\boldsymbol{\Psi} = (\pi_1,\pi_2,\boldsymbol{\beta}_1^\top,\boldsymbol{\beta}_2^\top,\boldsymbol{\beta}_3^\top,\phi_1,\phi_2,\phi_3)^\top.$$

The increase in model complexity is reflected in the summary of all parameters of interest when fitting a three-component mixture model which are listed in Table 5.2 for mean functions (5.1) and (5.2) and the normal and Gamma distribution.

Mod.	Distr.	Comp.	$\boldsymbol{\pi}$	$\boldsymbol{\beta}_k$	ϕ_k	#Par.
Model (5.1)	Normal	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11},\beta_{12},\beta_{13},\beta_{14})$	$\phi_1 = {\sigma_1}^2$	17
		k=2	π_2	$\boldsymbol{\beta}_2 = (\beta_{21},\beta_{22},\beta_{23},\beta_{24})$	$\phi_2 = {\sigma_2}^2$	
		k=3		$\boldsymbol{\beta}_3 = (\beta_{31},\beta_{32},\beta_{33},\beta_{34})$	$\phi_3 = {\sigma_3}^2$	
	Gamma	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11},\beta_{12},\beta_{13},\beta_{14})$	$\phi_1 = {\nu_1}^{-1}$	17
		k=2	π_2	$\boldsymbol{\beta}_2 = (\beta_{21},\beta_{22},\beta_{23},\beta_{24})$	$\phi_2 = {\nu_2}^{-1}$	
		k=3		$\boldsymbol{\beta}_3 = (\beta_{31},\beta_{32},\beta_{33},\beta_{34})$	$\phi_3 = {\nu_3}^{-1}$	
Model (5.2)	Normal	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11},\beta_{12},\beta_{13},\beta_{14},\beta_{15})$	$\phi_1 = {\sigma_1}^2$	20
		k=2	π_2	$\boldsymbol{\beta}_2 = (\beta_{21},\beta_{22},\beta_{23},\beta_{24},\beta_{25})$	$\phi_2 = {\sigma_2}^2$	
		k=3		$\boldsymbol{\beta}_3 = (\beta_{31},\beta_{32},\beta_{33},\beta_{34},\beta_{35})$	$\phi_3 = {\sigma_3}^2$	
	Gamma	k=1	π_1	$\boldsymbol{\beta}_1 = (\beta_{11},\beta_{12},\beta_{13},\beta_{14},\beta_{15})$	$\phi_1 = {\nu_1}^{-1}$	20
		k=2	π_2	$\boldsymbol{\beta}_2 = (\beta_{21},\beta_{22},\beta_{23},\beta_{24},\beta_{25})$	$\phi_2 = {\nu_2}^{-1}$	
		k=3		$\boldsymbol{\beta}_3 = (\beta_{31},\beta_{32},\beta_{33},\beta_{34},\beta_{35})$	$\phi_3 = {\nu_3}^{-1}$	

Table 5.2: Parameters of interest for three-component mixture models

5.1.3 Predictions

A key aspect of the subsequent applications emerges by the uncertainty of gas flow for low temperatures. The observations exhibit typically sparse data for low temperatures whereas very low temperatures may occur even beyond the usual observation range for outside temperatures of a specific time horizon. The use of mixtures of GNMs enables the prediction of gas flow comprising individual differences in consumption levels within the identified components. For this purpose, the expectation of the gas flow for a temperature value x_i conditional on the observed variables and the parameter vector $\boldsymbol{\Psi}$ is

used, respectively

$$\mu_i^M(\boldsymbol{\beta}) := \mathrm{E}[y_i|x_i, \boldsymbol{\Psi}] = \sum_{k=1}^{K} \pi_k \mu_i(\boldsymbol{\beta}_k), \quad i = 1, \ldots, n. \tag{5.4}$$

Equation (5.4) expresses the forecast of gas flow by means of the K-component mixture of mean functions $\mu(\cdot)$ evaluated at the component specific distribution parameters $\boldsymbol{\beta}_k$ and weighted by the prior probabilities π_k for $k = 1, \ldots, K$. The predictions are subject to a specific level of uncertainty. Informative indications related to the accuracy of the predictions can be expressed by confidence intervals. In order to assess the variability of the mean predictions in subsequent applications, the corresponding confidence intervals are constructed by the use of the *Delta method*. Thus, the variance of the mean function $\mu^M(\hat{\boldsymbol{\beta}})$ can be approximated by its gradient and the variance-covariance matrix of the MLE $\hat{\boldsymbol{\beta}}$. The latter was discussed in Section 2.5 and can be approximated by the following expression

$$\mathrm{Var}(\mu_i^M(\hat{\boldsymbol{\beta}})) \approx \nabla(\mu_i^M(\hat{\boldsymbol{\beta}}))^\top \, \mathrm{Cov}[\hat{\boldsymbol{\beta}}] \nabla(\mu_i^M(\hat{\boldsymbol{\beta}})),$$

where the gradient $\nabla(\mu_i^M(\hat{\boldsymbol{\beta}})) \in \mathbb{R}^{KP}$ containing the derivatives is given by

$$\nabla(\mu_i^M(\boldsymbol{\beta})) = \left(\frac{\partial \mu_i^M(\boldsymbol{\beta})}{\partial \beta_{kp}} \right)_{k=1,\ldots,K; p=1,\ldots,P}.$$

The corresponding confidence interval for $\mu^M(\hat{\boldsymbol{\beta}})$ for the confidence level $(1-\alpha)$ can be derived as

$$(1-\alpha)\% \ \mathrm{CI}(\mu_i^M(\boldsymbol{\beta})) \approx \left(\mu_i^M(\hat{\boldsymbol{\beta}}) \pm z_{1-\alpha/2} \cdot \sqrt{\mathrm{Var}(\mu_i^M(\hat{\boldsymbol{\beta}}))} \right) \tag{5.5}$$

where $\mu_i^M(\hat{\boldsymbol{\beta}})$ corresponds to the predicted value for temperature x_i. Equations (5.4) and (5.5) will be applied in the subsequent applications in order to predict gas flow for low temperatures.

5.2 Gas Flow Data 1

The given data relates to measured values from stations within a gas pipeline network and stems from the study by Friedl et al. (2012). The data set consists of hourly gas flow data for the period from January 2004 to June 2009 in degrees Celsius or centigrade (°C). The gas flow is measured in kilowatt hours per hour (kWh/h). Additional information is given by mean daily temperatures from corresponding weather stations as well as a working day indicator. The objective of the work by Friedl et al. (2012) was to study the gas flow in dependence on the given air temperature. With a view to maximize the transportation capacity through the pipelines Friedl et al. (2012) concentrated on the standardized daily maximum flows as responses within the used models. An important aspect of their study was the analysis of the *design temperature*. The design temperature refers to the lowest temperature at which the gas supplier has the obligation to supply gas without failure. In the case of the present data set design temperatures range between -12 °C and -16 °C. As these temperatures rarely occur, less observations for them are available and gas suppliers rely on predictions which emphasizes the importance of accurate statistical models. Figure 5.1 displays the data with the typical decreasing pattern of gas flow in relation to the outdoor temperature.

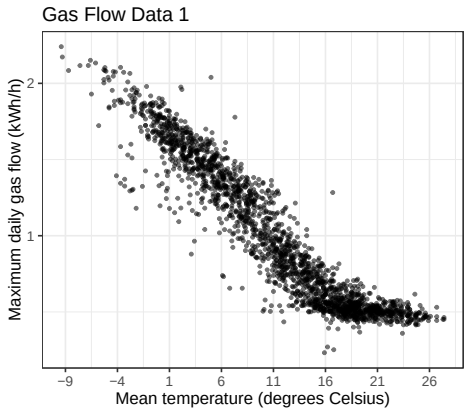

Figure 5.1: Gas flow data 1

Figure 5.1 shows a clear nonlinear dependence structure between the gas flow and the outdoor temperature with considerable variation of the gas flow data. Furthermore, the data appear to converge to a lower limit for temperature values exceeding 16 °C. This effect seems plausible as a minimum gas flow level can be assumed for warm temperatures. At the negative end of the temperature scale the gas flow exhibits a visible variation with evident outliers for below-average gas flow as well as high level of consumptions. The present data sample exhibits sparse data for low temperatures. Considering the gas flow prediction for these low temperatures it is therefore important to find a suitable nonlinear functional form which enables an adequate fitting of the gas flow. The data raise furthermore the question of a suitable distribution comprising the variation structure. The distinction between working days and holidays reveals two different subsets with a similar pattern.

5.2.1 Nonlinear Regression

The aim of this section is to explore the given gas flow data by nonlinear regression analysis and to present new approaches and new results for the problem of modeling gas flow in transmission networks. The upcoming analysis draws on the selected parametric models (5.1) and (5.2). For comparative purposes to the original study the daily maximum gas flow values will be standardized by the empirical mean of the maximal daily gas flows. Due to the characteristics of the error term distribution of maximum values, the existing research will be extended by the fitting of models (5.1) and (5.2) to the Gamma distribution. The measure for the quality of the derived regression coefficients will be given by the simultaneously computed standard errors and additional graphical visualizations of the fitted values. Model selection criteria are given by the information-based criteria AIC and BIC. The key objective of this analysis is to explore the aforementioned models and distributions in order to achieve the best model fit. Given these findings the appropriate model will serve as base for the subsequent fitting with mixtures of GNMs.

5.2.1.1 Gaussian Distribution

Friedl et al. (2012) analyze the available data set on gas flow for different parametric models with an underlying normal distribution. Therefore the mean function $\boldsymbol{\mu}(\boldsymbol{\beta})$ and the variance σ^2 have to be estimated which affords the fitting of the regression coefficient vector $\boldsymbol{\beta}$. The fitting of the models (5.1) and (5.2) was carried out with the help of the function `nls()`. A discussion of the fitted parameters follows in the next subsections. The fitted parameters are displayed in Table 5.3.

Model 1
The mean function attains its upper limit at $\hat{\beta}_1 = 2.04$ and the lower limit at $\hat{\beta}_4 = 0.45$. The mean function decreases with shape parameters $\hat{\beta}_2 = -32.8$ and $\hat{\beta}_3 = 6.6$. The corresponding standard errors in Table 5.3 show a small variability of the regression coefficients proportionate to their sizes. Information on the statistical significance of the coefficients is often given by the p-values which relate the standard errors to the fitted coefficients and test the hypothesis that the regression parameter is zero. The present results yield low p-values (<0.05) indicating statistical significance. As Ritz and Streibig (2008) point out, the hypothesis tests may be of limited relevance for nonlinear regression models.

Model 2
Including the working day indicator moves the upper limit of the mean function towards $\hat{\beta}_1 = 2.05$ while the lower limit is attained at $\hat{\beta}_4 = 0.45$. The mean function displays furthermore a stronger decrease given by the shape parameters $\hat{\beta}_2 = -34.1$ and $\hat{\beta}_3 = 6.3$. The regression coefficient β_5 for the working day indicator results in $\hat{\beta}_5 = -0.05$. The standard errors show a similar relationship in terms of the size ratio to the fitted coefficients compared to the results for model (5.1). Within model (5.2) all five regression coefficients yield again low p-values (<0.05) indicating statistical significance.

5.2.1.2 Gamma Distribution

Friedl et al. (2012, p. 38) acknowledge the careful choice of the Gaussian error distribution when modeling maximal daily gas flow data. The error structure of maximum values exhibits usually heavy-tails. The daily maximum gas flow requires an adequate distribution assumption taking into account for the heavy tail characteristics. Figure 5.2 displays the Q-Q plot for the gas flow data compared to a theoretical normal distribution. The figure shows an evident discrepancy between the shape of a normal distribution and the empirical shape of the given data set. The present data sample exhibits evidently heavier tails as the empirical quantiles spread significantly wider from the compared (theoretical) normal distribution in the tails.

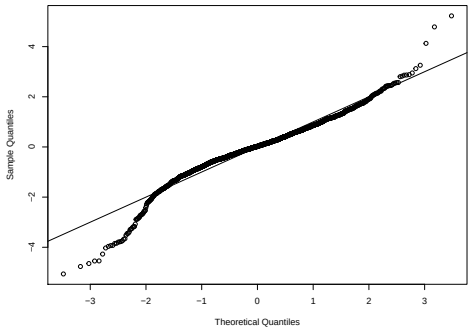

Figure 5.2: Q-Q plot of the residuals for the fitted nonlinear model

In order to complement the graphical results the *Jarque-Bera* test provides a further statement on the comparison to a potentially normal distribution. The Jarque-Bera test tests the discrepancy of the underlying empirical distribution to a normal distribution using the kurtosis and skewness. The null hypothesis states that the skewness and excess kurtosis correspond each to zero as it is the case within a normal distribution. The *Shapiro-Wilk* test serves as another normal distributional test. It states the null hypothesis that the given sample stems from a normal distribution. The *Shapiro-Wilk* as well as the *Jarque-Bera* test both result in a p-value substantially smaller than 0.01 which leads to a rejection of an underlying normal distribution for the residuals. The present graphical and inductive results encourage the use of a heavy-tail distribution for the modeling of maximum gas flow. Based on these conclusions, the Gamma distribution will be used for the fitting of the gas flow in the remaining section. For this purpose the mean function $\boldsymbol{\mu}(\boldsymbol{\beta})$, including the regression coefficients $\boldsymbol{\beta}$, and the shape parameter ν have to be estimated. A special focus will be given to the prediction on design temperatures in Section 5.2.4.

Model 1

Table 5.3 summarizes the fitted regression coefficients for both distributions. The results for the Gamma distribution are shown on the right side. Differences to the normal distribution appear primary in the upper asymptotes. The graphical visualizations for model (5.1) in Figure 5.3 show a clear discrepancy between the modeling of the data for very

low temperatures. While the normal distribution (blue color) yields an upper asymptote $\hat{\beta}_1 = 2.0$ the Gamma distribution (red color) results in a significantly lower value of $\hat{\beta}_1 = 1.9$. The coefficients $\hat{\beta}_2 = -32$ and $\hat{\beta}_3 = 8$ indicate only low differences in the shapes of the both curves which is confirmed by the visualization of the results in Figure 5.3. The lower asymptotes of model (5.1) match very closely under the two distributions. Based on previously made distributional considerations it can be assumed that the Gamma distribution yields more accurate results for the maximum values compared to the normal distribution. Within the present example the use of the normal distribution yields particularly to an over-estimation of the gas flow for design temperatures. Comparing the fitted log-likelihood functions and the thereof deduced information criteria AIC and BIC assign the Gamma distribution a better model fit. Both information criteria show lower values for the fitted model under a Gamma distribution as summarized in Table 5.3. The decrease in β_1 under a Gamma distribution appears as significant difference compared to the results from the original study by Friedl et al. (2012, p. 27).

Model 2
The fitted coefficient for the included working day indicator results in $\hat{\beta}_5 = -0.05$ and resembles the result under a normal distribution. The coefficients influencing the decreasing behavior $\hat{\beta}_2$ and $\hat{\beta}_3$ show minor differences compared to those provided by fitting model (5.1) under the same distribution. Again, all five coefficients are declared as significant. Figure 5.3 outlines that the inclusion of a working day factor tends to shift the upper asymptotes slightly higher for the normal distribution. The results indicate a better fit of the Gamma distribution as the derived AIC and BIC result in lower values. The inclusion of a working day indicator increases the model accuracy for both distributions.

Figure 5.3 displays the fitted mean functions for models (5.1) (left figure) and (5.2) (right figure). The blue colored lines refer to results under a normal distribution while red colored lines match fitted results under a Gamma distribution. The left figure shows in general a high matching of the fitted mean function (5.1) under both distributions except for the upper asymptote which is decreasing under a Gamma distribution. The right figure vizualizes the fitted mean functions under model (5.2). The dotted lines refer to the mean function arising from the gas flow on holidays while the solid lines relate to working days. As expected, the gas flow exhibits a lower level on holidays.
Table 5.3 gives a tabular representation of the fitted regression coefficients for both models and both distributions with the corresponding standard errors. It furthermore displays the variance parameters, the log-Likelihood, the AIC and BIC for model comparative purposes.

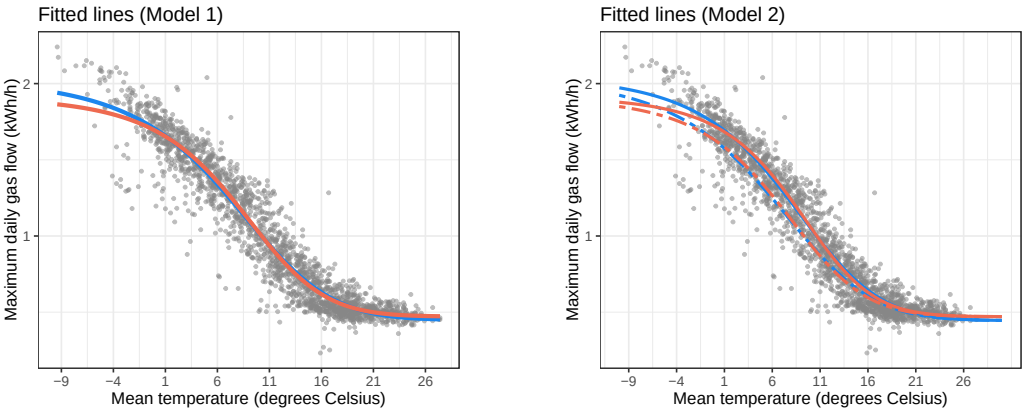

Figure 5.3: Fitted mean function for gas flow data 1 and models (5.1) and (5.2)

Distribution:	**Normal**		**Gamma**	
Model:	Model (5.1)	Model (5.2)	Model (5.1)	Model (5.2)
$\hat{\beta}_1$	2.04	2.05	1.92	1.92
$s.e.(\hat{\beta}_1)$	0.03	0.03	0.03	0.03
$\hat{\beta}_2$	-32.78	-34.05	-31.99	-33.08
$s.e.(\hat{\beta}_2)$	0.23	0.24	0.23	0.24
$\hat{\beta}_3$	6.61	6.28	7.59	7.34
$s.e.(\hat{\beta}_3)$	0.22	0.20	0.23	0.21
$\hat{\beta}_4$	0.45	0.45	0.47	0.47
$s.e.(\hat{\beta}_4)$	0.01	0.01	0.01	0.01
$\hat{\beta}_5$	-	-0.05	-	-0.05
$s.e.(\hat{\beta}_5)$	-	0.00	-	0.00
$\hat{\sigma}$	0.13	0.12	$\hat{\nu}$ 55.87	59.68
AIC	$-2\,470$	$-2\,673$	$-2\,734$	$-2\,868$
BIC	$-2\,442$	$-2\,639$	$-2\,706$	$-2\,834$

Table 5.3: Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 1 (nonlinear regression)

Friedl et al. (2012) emphasize the use of **flexmix** for generalized mixtures of sigmoid models. The extension of **flexmix** as introduced in Section 3.4 enables the fitting of mixtures of nonlinear models. It will be applied to the gas flow data 1 within the following section.

5.2.2 Two-Component Mixtures of GNMs

The aim of this section is to apply mixtures of GNMs to gas flow data 1. The evident heterogeneity for low temperatures in Figure 5.1 motivates the fitting of the data by multiple regression functions. Building on the results from single nonlinear regression in Section 5.2.1, a sigmoid function appears useful for modeling gas flow in general. Therefore two-component mixture models with sigmoid functions (5.1) and (5.2) will be applied to gas flow data 1. In order to explore the quality and performance of **flexmixNL**, several simulation fittings will be performed for different starting configurations.

Section 5.2.2.1 presents the simulation setup while subsequent sections discuss the fitted results for two-component mixture models in detail for the normal and Gamma distribution. Section 5.2.3 draws comparisons between the fitted models. Section 5.2.4 outlines the predicted gas flow for design temperatures. Final conclusions summarize the main results in Section 5.2.5.

5.2.2.1 Initial Configuration and the Number of Components

In the given gas flow data higher variability for lower temperatures motivates the use of two components. Due to the decreasing pattern of gas flow with increasing temperatures, the observations concentrate around a central curve with rare outliers. Therefore, the following models presume two components which provide a similar global shape except for the modeling of low temperatures where different predictions are expected. Corresponding to the original paper Friedl et al. (2012), the normal distribution will be used to fit the *two-component Gaussian mixture model*. The nonlinear regression analysis results in the previous section buttress the use of the Gamma distribution. Therefore also a *two-component Gamma mixture model* will be presented as a suitable extension of the present research. In order to explore the accuracy and performance of the fitting algorithm provided by **flexmixNL**, the given data set will be fitted several times with randomly chosen starting values. The ranges for the starting values for the two-component mixture models are given in Table 5.4. The initial starting values are selected uniformly from the respective intervals.

Parameter	β_{k1}	β_{k2}	β_{k3}	β_{k4}	β_{k5}
Range	$[1.5; 2.5]$	$[-45; -30]$	$[6; 9]$	$[0.4; 0.55]$	$[-0.5; 0]$

Table 5.4: Ranges for starting values for gas flow data 1

The regression coefficient representing the upper asymptote β_{k1} exhibits the highest variability stemming from the largest interval in Table 5.4. Varying the upper asymptote influences the height as well as the shape of the mean functions for $k = 1, 2$. As both coefficients stem from the same ranges, the starting configuration for different components may coincide or overlap. Figure 5.4 illustrates possible starting values for both components graphically. Taking into account the very similar shapes of the regression lines in Figure 5.4, the EM algorithm builds on a difficult starting position considering the dense structure of the given data set. The fitted results can be reproduced by setting

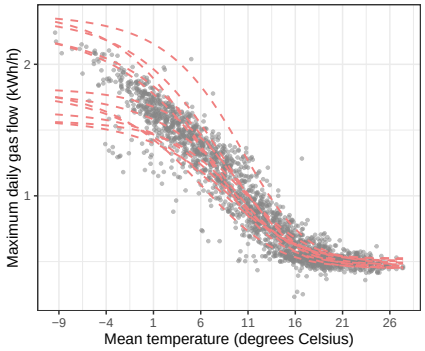

Figure 5.4: Starting configuration for gas flow data 1

the random seed equal to `set.seed(2357)` and `tolerance=1e-05` for Gamma mixture models. The present data set will be fitted 50 times for each model with randomly selected starting values and the default initial cluster assignment in order to obtain an overview on the quality of the results. Section 5.2.2.2 briefly summarizes the respective results.

5.2.2.2 Simulation results

Applying the simulation setup in Section 5.2.2.1 provides estimates for all parameters as listed in Table 5.4. Table 5.5 outlines the mean values of the point estimates and dispersion parameters for the normal and Gamma distribution. Both distributions were fitted 50 times for the two models (5.1) and (5.2) with randomly chosen starting values. The fitted components are ordered decreasing in terms of the prior weights $\hat{\pi}_k$ for $k = 1, 2$. The column `mean` displays the average of the fitted coefficients for each component whereas column `sd` shows the standard deviations over the simulation results. The results indicate stable fitting results as they point out minor deviations for both distributions. The simulation results in Table 5.6 for Model (5.2) show a similar positive performance compared to the previous results due to minor deviations. The subsequent sections display the fitting of each model under the two distributions in more detail.

		2-Component Gaussian			2-Component Gamma	
		mean	sd		mean	sd
Component 1	$\hat{\pi}_1$	0.79	0.00		0.81	0.04
	$\hat{\beta}_{11}$	2.19	0.00		1.96	0.01
	$\hat{\beta}_{12}$	-33.73	0.00		-32.07	0.11
	$\hat{\beta}_{13}$	6.52	0.00		7.61	0.22
	$\hat{\beta}_{14}$	0.45	0.00		0.47	0.00
	$\hat{\sigma}_1$	0.09	0.00	$\hat{\nu}_1$	98.32	2.34
Component 2	$\hat{\beta}_{21}$	1.51	0.00		1.66	0.02
	$\hat{\beta}_{22}$	-29.09	0.00		-31.00	0.30
	$\hat{\beta}_{23}$	9.62	0.00		7.88	0.50
	$\hat{\beta}_{24}$	0.49	0.00		0.49	0.00
	$\hat{\sigma}_2$	0.20	0.00	$\hat{\nu}_2$	19.11	6.04

Table 5.5: Simulation results for two-component mixtures with Model (5.1)

		2-Component Gaussian			2-Component Gamma	
		mean	sd		mean	sd
Component 1	$\hat{\pi}_1$	0.81	0.00		0.82	0.00
	$\hat{\beta}_{11}$	2.20	0.00		1.96	0.00
	$\hat{\beta}_{12}$	-34.72	0.00		-33.02	0.00
	$\hat{\beta}_{13}$	6.23	0.00		7.28	0.00
	$\hat{\beta}_{14}$	0.45	0.00		0.47	0.00
	$\hat{\beta}_{15}$	-0.04	0.00		-0.04	0.00
	$\hat{\sigma}_1$	0.09	0.00	$\hat{\nu}_1$	103.94	0.06
Component 2	$\hat{\beta}_{21}$	1.53	0.00		1.66	0.00
	$\hat{\beta}_{22}$	-30.76	0.00		-32.87	0.00
	$\hat{\beta}_{23}$	8.83	0.00		7.89	0.00
	$\hat{\beta}_{24}$	0.48	0.00		0.49	0.00
	$\hat{\beta}_{25}$	-0.08	0.00		-0.09	0.00
	$\hat{\sigma}_2$	0.19	0.00	$\hat{\nu}_2$	20.69	0.02

Table 5.6: Simulation results for two-component mixtures with Model (5.2)

5.2.2.3 Model 1

Table 5.7 summarizes the fitted coefficients for Model (5.1) for the normal and Gamma distribution. Both mixture models identify a major component yielding a prior probability about 0.8. The estimates for the upper asymptotes of the two components $\hat{\beta}_{11}$ and $\hat{\beta}_{21}$ on the other hand show an evident difference when comparing the two different distributions as they differ about 0.2 units. Considered individually, the mixture models exhibit the following effect: the coefficients with influence to the shapes of the both decreasing curves are given by $\hat{\beta}_{12}$, $\hat{\beta}_{13}$ and $\hat{\beta}_{22}$, $\hat{\beta}_{23}$. As $\hat{\beta}_{13}$ and $\hat{\beta}_{23}$ do not differ considerably for the two components of the Gamma mixture models the shape of the curves is very similar. The normal mixture model exhibits a slightly varying shape in the mean functions. For the normal distribution the component specific variability is given by the deviations σ_1 and σ_2. The smaller deviation in the first component $\hat{\sigma}_1 = 0.09$ indicates a concentration within a central component while the second component covers the remaining observations with greater variability as $\hat{\sigma}_2 = 0.20$. The variability of the fitted values for the Gamma mixture model is given by the estimate shape parameters $\hat{\nu}_1$ and $\hat{\nu}_2$ indicating also a significantly higher variability in the second component. Considering the distinction between working days and holidays indicates a larger gas flow on working days for both distributions.

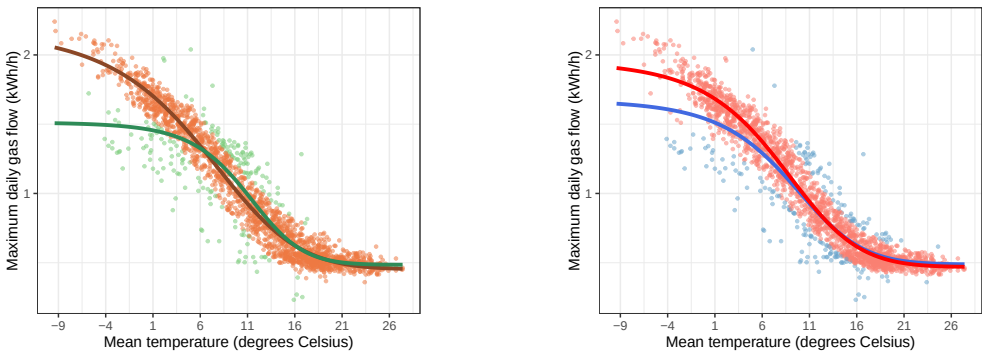

Figure 5.5: Fitted two-component mixtures of normal (left) and Gamma (right) for Model (5.1)

The fitted regression functions and assigned components are displayed in Figure 5.5. As the fitted regression functions clearly show, the fitted mean at low temperatures differs significantly for the two distributions due to the discrepancy of the estimates for the upper asymptote $\hat{\beta}_{11}$ and $\hat{\beta}_{21}$. The differences in the fitted values for the dispersion parameters (as evident in Table 5.7) indicate one central component comprising the majority of the observations while the remaining component covers the observations with higher variability. For Gamma distributed responses this effect is even intensified due to the mean-dependent variance. The resulting nonlinear regression lines show significant differences in the modeling of gas flow for low temperatures below 0 °C due to the increasing variability. With increasing temperatures the gas flow and its variability decreases indicating similar shapes for the two components. The corresponding rootogram in Figure 5.6 outlines the visualization of the component assignments for the normal distribution. The rootograms indicate a moderate separation due to the dense data structure. The rootogram in Figure 5.7 for the Gamma distribution shows a simi-

lar pattern as for the mixture of normal distributions. An improvement in separation is visible for the first component due to more mass near 1. The quality of the cluster assignment can be derived from the information provided by the `summary()` function. Listings 5.8 and 5.9 are also informative on the component separation for the two-component mixture models. They show four columns containing the component specific posterior probabilities, the final cluster sizes and the overall number of observations assigned to the specific component with a posterior probability greater than a predefined threshold ϵ. The component sizes will be referred to as $n_{\cdot}^{normal}$ for the normal distribution and $n_{\cdot}^{Gamma}$ for the Gamma distribution. The results show a similar assignment to the two components comparing the normal and Gamma distribution. Small differences arise through the slightly smaller first component for a normal distribution with component size $n_1^{normal} = 193$ whereas the Gamma distribution yields $n_1^{Gamma} = 200$. The column `post>0` shows the number of observations with a positive posterior probability of lying in the component. The results reveal a complete overlapping of the two components as all 2008 observations appear in the first component with a posterior probability greater zero and $n_2^{normal} = 1980$ observations or rather $n_2^{Gamma} = 1994$ in the second component. In the case of a two-component mixture of Gaussian models both components exhibit a moderate separation. Missing peaks at 1 and the visible mass in the center of the rootogram in Figure 5.6 underpin this finding.

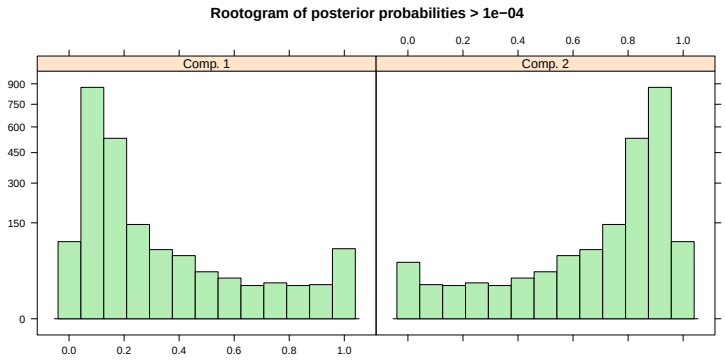

Figure 5.6: Rootogram for two-component normal mixtures for Model (5.1)

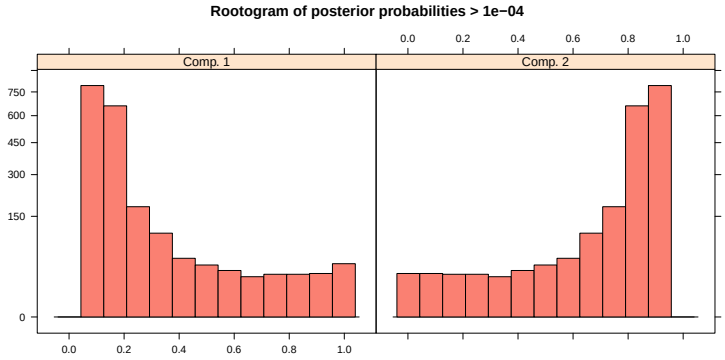

Figure 5.7: Rootogram for two-component Gamma mixtures for Model (5.1)

```
> summary(mod1.n)
Call:
flexmix(...)
       prior size post>0  ratio
Comp.1 0.219  193   2008 0.0961
Comp.2 0.781 1815   1980 0.9167
```

Listing 5.8: `summary()` output for two-component Gaussian mixtures for Model (5.1)

```
> summary(mod1.g)
Call:
flexmix(...)
      prior size post>0  ratio
Comp.1 0.227  200   2008 0.0996
Comp.2 0.773 1808   1994 0.9067
```

Listing 5.9: `summary()` for two-component Gamma mixtures for Model (5.1)

5.2.2.4 Model 2

Including the working day effect for the fitting of gas flow data 1 requires the use of Model (5.2) where β_{k5} represents the additional coefficient. Table 5.7 summarizes the fitted coefficients for Model (5.2) for both distributions. The third column shows the results for an underlying normal distribution. The inclusion of a working day effect reduces the variability in the second component for an underlying normal distribution. The ranges of the sigmoid mean functions increase slightly as the upper asymptotes increase for the normal distribution while the lower bounds decrease. The shape coefficients β_{k2} and β_{k3}, $k = 1, 2$, do not exhibit significant changes compared to the model without a working day indicator. The fourth column in Table 5.7 displays the fitted coefficients for an underlying Gamma distribution. In contrast to the normal distribution, adding a working day indicator to the Gamma components does not affect the dispersion significantly. The upper and lower asymptotes exhibit a similar but weaker behavior compared to the normal distribution in terms of an increased range for the sigmoid function. The fitted coefficients influencing the shape exhibit no significant changes. The quality of the final cluster assignment is provided again by executing the `summary()` function and rootograms for the fitted mixture models with mean function (5.2).

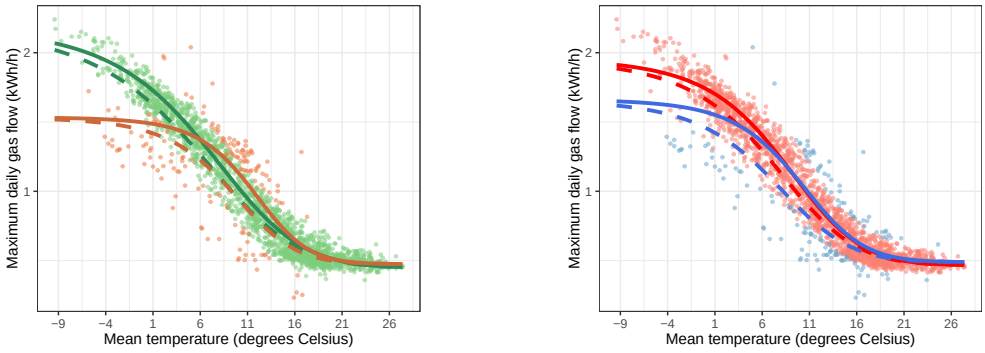

Figure 5.8: Fitted two-component mixtures of normal (left) and Gamma (right) for Model (5.2)

The rootograms coincide with Figures 5.6 and 5.7. Listing 5.10 displays the posterior probabilities for an underlying normal distribution. The greater component with posterior 0.798 realizes in a final cluster size of $n_2^{normal} = 1\,828$ while in total $1\,982$ observations show a posterior probability greater than a threshold ϵ. The smaller component with cluster size $n_1^{normal} = 180$ covers the whole data set as the complete sample shows

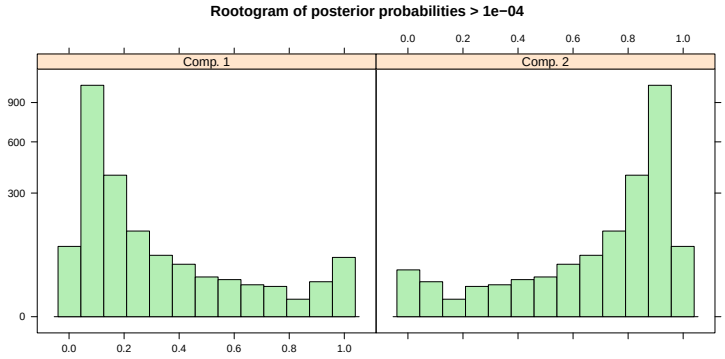

Figure 5.9: Rootogram for two-component normal mixture for Model (5.2)

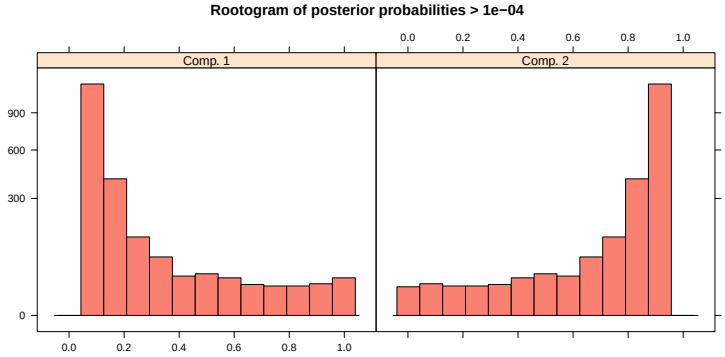

Figure 5.10: Rootogram for two-component Gamma mixture for Model (5.2)

a positive posterior probability. Listing 5.11 and the rootogram display the results on the cluster assignment for an underlying Gamma distribution where the same deductions as in the previous analysis can be made.

```
> summary(mod2.n)
Call:
flexmix(...)
      prior size post>0  ratio
Comp.1 0.202  180   2008 0.0896
Comp.2 0.798 1828   1982 0.9223
```

Listing 5.10: `summary()` for two-component Gaussian mixtures for Model (5.2)

```
> summary(mod2.g)
Call:
flexmix(...)
      prior size post>0  ratio
Comp.1 0.192  157   2008 0.0782
Comp.2 0.808 1851   1995 0.9278
```

Listing 5.11: `summary()` for two-component Gamma mixtures for Model (5.2)

Mean function:		Model (5.1)						Model (5.2)					
Mixture model:		2-Component Gaussian			2-Component Gamma			2-Component Gaussian			2-Component Gamma		
	Results:	Mean	std.err.		Mean	std.err.		Mean	std.err.		Mean	std.err.	
Component 1	$\hat{\pi}_1$	0.79	0.03		0.78	0.07		0.81	0.04		0.82	0.07	
	$\hat{\beta}_{11}$	2.19	0.04		1.96	0.04		2.20	0.03		1.96	0.04	
	$\hat{\beta}_{12}$	-33.73	0.25		-32.11	0.27		-34.72	0.26		-33.02	0.29	
	$\hat{\beta}_{13}$	6.52	0.19		7.60	0.27		6.23	0.17		7.28	0.25	
	$\hat{\beta}_{14}$	0.45	0.01		0.47	0.01		0.45	0.01		0.48	.01	
	$\hat{\beta}_{15}$	-	-		-	-		-0.04	0.00		-0.04	0.01	
	$\hat{\sigma}_1$	0.09	0.00	$\hat{\nu}_1$	98.34	11.33	$\hat{\sigma}_1$	0.09	0.0	$\hat{\nu}_1$	103.96	14.06	
Component 2	$\hat{\beta}_{21}$	1.51	0.05		1.68	0.14		1.53	0.06		1.66	0.13	
	$\hat{\beta}_{22}$	-29.09	0.47		-31.01	1.08		-30.76	0.69		-32.87	1.35	
	$\hat{\beta}_{23}$	9.62	1.41		7.96	1.35		8.83	1.38		7.89	1.43	
	$\hat{\beta}_{24}$	0.49	0.04		0.49	0.02		0.48	0.04		0.49	0.02	
	$\hat{\beta}_{25}$	-	-		-	-		-0.08	0.02		-0.09	0.03	
	$\hat{\sigma}_2$	0.20	0.01	$\hat{\nu}_2$	19.11	3.36	$\hat{\sigma}_2$	0.19	0.01	$\hat{\nu}_2$	20.70	4.33	
	AIC	$-2\,845$			$-2\,923$			$-3\,045$			$-3\,059$		
	BIC	$-2\,783$			$-2\,861$			$-2\,972$			$-2\,986$		
	ICL	$-2\,105$			$-2\,090$			$-2\,343$			$-2\,329$		

Table 5.7: Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 1 (two components)

5.2.3 Model Comparison

As discussed in Section 2.7.1, the ICL has proven as a suitable criterion for the adequate choice of the number of components within mixture models. Therefore the comparison of the fitted mixture models through the well-known criteria AIC and BIC will be complemented by means of the ICL for the underlying data in Figure 5.1. Figure 5.2.3 visualizes the different outcomes for the fitted Models (5.1) and (5.2). Model (5.1) is referred to as *M1* while *M2* corresponds to the mean function (5.2).

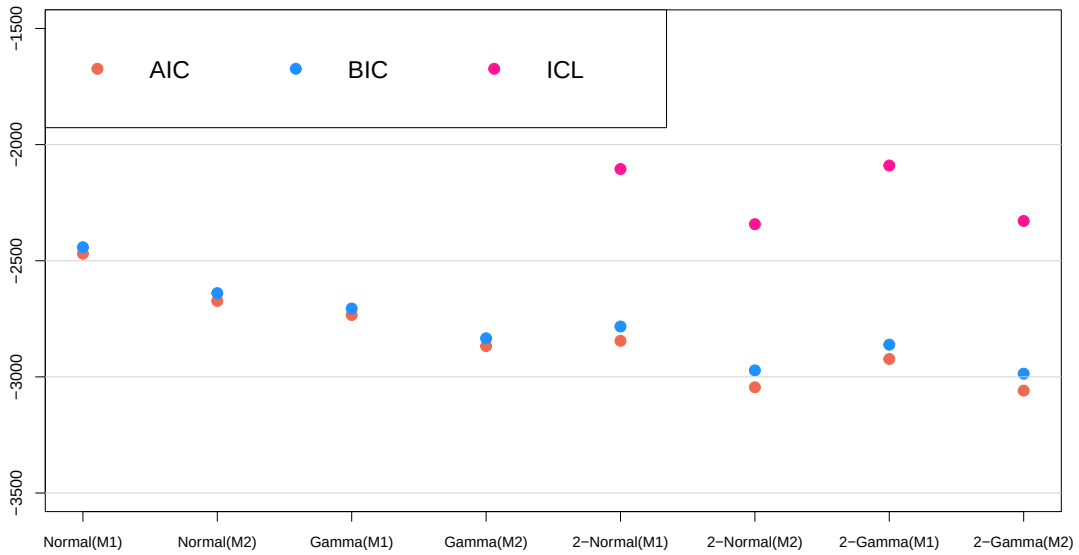

Figure 5.11: AIC,BIC and ICL for fitted models for gas flow data 1

The direct comparison of the single nonlinear regression and the two-component mixture models demonstrates an improvement in the model fit using the second mean function (5.2). That conclusion is underpinned by the already outlined statistical significance of the working day indicator within the nonlinear regression model in Section 5.2.1. The use of the Gamma distribution does not indicate a decisive improvement regarding the mixture models while the single nonlinear regression results exhibit an evident improvement. As the use of the Gamma distribution is highly motivated due to the nature of the present data it can be stated as the better statistical fit due to general considerations. Drawing on the previous discussion, in the fitted Gamma mixture model the upper asymptote is attained at a lower level preventing an over-estimation of the gas flow for lower temperatures. A direct comparison of the predicted values for low temperatures is given in the subsequent section.

5.2.4 Predictions of Gas Flow on Design Temperatures

An important aspect of the study provided by Friedl et al. (2012) was the accurate prediction of gas flow for design temperatures ranging between -16 °C and -12 °C. In

order to assess the necessary maximum gas demand even for design temperatures the gas operator relies on predictions based on the available models. Within this subsection special attention is given to the prediction of gas flow for design temperature for the previously fitted Models (5.1) and (5.2). For this purpose discrepancies between the normal and Gamma assumptions will be outlined. Table 5.8 displays the predicted gas flow for the exemplary set of design temperature values -12 °C, -14 °C and -16 °C for the fitted models. The gas flow for design temperature values within mixture models can be predicted for the underlying mixture models as the overall mean over all components according to Equation (2.3). The corresponding 95% prediction intervals are outlined below the predicted values. The construction of the confidence interval follows Equation (5.5).

The derived results in Table 5.8 give a clear evidence on the discrepancies between the two distributions for all models. The predicted values are highly driven by the coefficients β_1 representing the value where the fitted mean function attains its maximum. The predicted gas flow for design temperatures under the fitted Gamma models is systematically lower compared to the normal distribution. This effect can be attributed to the lower asymptotes under a Gamma distribution and matches the results on the coefficients in Table 5.7. As a consequence and due to the shape of the fitted mean functions, the predicted values under the Gamma distribution stabilize faster for decreasing temperatures. Model (5.2) enables the distinction between working days and holidays. Therefore under both distributions the predicted gas flow on holidays tends to decrease compared to working days. Based on the computed predictions in Table 5.8 the lowest gas flow is being predicted under the Model (5.2) for holidays under the Gamma distribution. The prediction of the highest gas flow is given with Model (5.2) restricted to working days under the normal distribution. The use of two-component Gamma mixture models decreases in general the predicted consumption level in direct comparison to the predicted values for the nonlinear regression.

The particular modeling of heterogeneity outlined different consumer groups which contribute to the general mean in accordance with their final mixing proportion. As the variability, particularly for lower temperatures, motivated the use of mixture models their use can be seen as more appropriate. The predicted values for -12 °C range (after re-scaling) between 34 MWh/h (Gamma mixture model on holidays) and 36 MWh/h (normal mixture model on working days). In direct comparison to the original paper by Friedl et al. (2012, p. 35), where the predicted values for -12 °C ranged between 39 MWh/h and 43 MWh/h, the application of mixture models decreases the predicted gas flow.

Design Temperatures:			**−12 °C**	**−14 °C**	**−16 °C**
Nonlinear regression	Model (5.1)	Normal	1.97	1.98	1.99
			(1.92, 2.01)	(1.94, 2.03)	(1.94, 2.04)
		Gamma	1.88	1.89	1.90
			(1.83, 1.93)	(1.84, 1.94)	(1.84, 1.95)
	Model (5.2) (holiday)	Normal	1.95	1.97	1.99
			(1.91, 1.99)	(1.93, 2.01)	(1.95, 2.03)
		Gamma	1.87	1.88	1.89
			(1.82, 1.91)	(1.83, 1.93)	(1.84, 1.94)
	Model (5.2) (working day)	Normal	1.99	2.01	2.02
			(1.95, 2.04)	(1.96, 2.05)	(1.97, 2.07)
		Gamma	1.89	1.90	1.90
			(1.84, 1.94)	(1.84, 1.95)	(1.85, 1.96)
2-Component mixture models	Model (5.1)	Normal	1.97	1.98	2.00
			(1.93, 2.01)	(1.94, 2.03)	(1.95, 2.04)
		Gamma	1.86	1.87	1.87
			(1.81, 1.92)	(1.81, 1.93)	(1.82, 1.94)
	Model (5.2) (holiday)	Normal	1.96	1.98	2.00
			(1.93, 2.00)	(1.94, 2.02)	(1.96, 2.04)
		Gamma	1.86	1.87	1.88
			(1.81, 1.91)	(1.82, 1.92)	(1.83, 1.93)
	Model (5.2) (working day)	Normal	1.99	2.01	2.02
			(1.95, 2.03)	(1.97, 2.05)	(1.98, 2.07)
		Gamma	1.88	1.89	1.89
			(1.83, 1.93)	(1.83, 1.94)	(1.84, 1.95)

Table 5.8: Prediction of gas flow for design temperatures for gas flow data 1

5.2.5 Final Remarks

Friedl et al. (2012) modeled the gas flow for predefined statistical models in order to obtain reliable predictions concerning the gas flow for design temperatures. For the choice of two parametric models (5.1) and (5.2), Section 5.2.1 performed the refitting of the nonlinear regression model for an underlying normal distribution and provided a further model extension by applying a heavy-tail distribution. The presented analysis was able to provide improving results for both models with an underlying Gamma distribution compared to the already available results under a normal distribution. With a specific focus on very low temperatures (design temperatures) gas owners refer to statistical models in order to predict the necessary gas demand. Therefore the normal distribution may lead to an over-fitting for the prediction of gas flow as it does not match the present data appropriately. The Gamma distribution yields to significantly lower predictions for the gas flow when focusing on design temperatures. Both models showed an improvement in statistical significance when adding a working day factor to the models under

both distributions while Model (5.2) under a Gamma distribution proved to be the best fit for the present data set.

As Friedl et al. (2012) emphasized the use of **flexmix** for generalized mixtures of sigmoid models, the appropriate extension was introduced in Chapter 3.4. Mixtures of GNMs were applied to the specific gas flow data 1 in Section 5.2.2. Within this context, the present results were extended through the application of mixtures of GNMs where both sigmoid regression functions (5.1) and (5.2) were taken into account. The data motivates the use of mixture models as it exhibits evident heterogeneity for lower temperatures. The present work applied two-component mixture models for the normal and Gamma distribution. All models were successfully fitted by the underlying methodology from Section 3.4. According to the ICL as suitable model selection criterion, the two-component mixture models comprising the working day indicator proved as the most appropriate with the highest model accuracy. The two-component Gamma mixture model comprising the working day indicator can be highlighted in particular considering the available results and the specifics of the problem. The present work outlines a decrease in the predicted values with the use of mixture models. As mixture models allow for the necessary flexibility to model heterogeneity in data, the decrease in predicted values are driven by the specifics of the data. The presented results state a substantive extension of the work in Friedl et al. (2012) proving the appropriateness of mixture models for the present data.

5.3 Gas Flow Data 2

The second data set refers to the gas flow of the market area *West Austria* (Tyrol and Vorarlberg in Austria). The data is available on the website of the Austrian balance group coordinators *AGCS Gas Clearing and Settlement AG* and can be accessed through `www.energymonitor.at`. Considering gas flow data in general, AGCS provides gas flow data for West and East Austria on quarter-hourly basis. The present analysis considers the time horizon between January 2013 and December 2017 which corresponds to $1\,553$ observation days. In order to achieve a general conclusion on the maximum transportation capacity for the specific period, the available data is reduced to the maximum daily gas flow. The derived maxima are merged with the mean temperatures from a measuring station in Vienna (Austria) which can be accessed on the web page of the *Zentralanstalt für Meteorologie und Geodynamik (ZAMG)* through `https://www.zamg.ac.at/cms/de/klima/klimauebersichten/jahrbuch`. The final data set is visualized in Figure 5.12. The time series of the daily gas flow data is displayed on the left, while its relationship to the outside temperature is given in the scatterplot on the right.

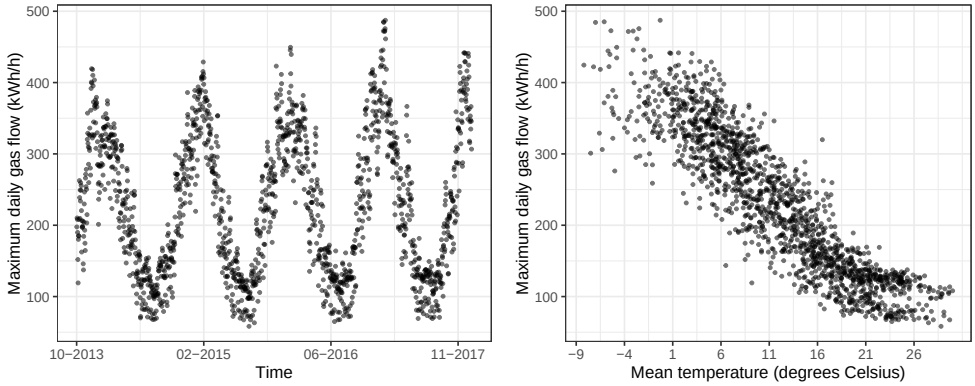

Figure 5.12: Gas flow data 2

The figure outlines a clear nonlinear functional relationship between the maximum daily gas flow and the outside temperature. The present data exhibit the typical pattern for gas flow by means of increasing variability for an increase in gas flow applicable to low temperatures. This effect can be attributed to the natural shape of gas flow data. The data range between 50 and 500 kWh/h. The graphical exploration of the data indicates the use of an adequate distribution outlining the non-constant variability of the data. In contrast to other gas flow patterns, this data set shows a heterogeneous consumption for high temperatures which is usually explainable as a minimum constant energy share. The present data reveal a clear separation in gas flow for high temperatures which indicates two different consumption levels. A visualization of the data in Figure 5.13 displays the consumption levels with distinction between holidays and working days in Austria which can be considered as the main driver for the heterogeneity for high temperatures. Figure 5.13 clearly outlines a lower consumption level on holidays (455 data points) on the right with smaller variability compared to the pattern on working days. Based on this

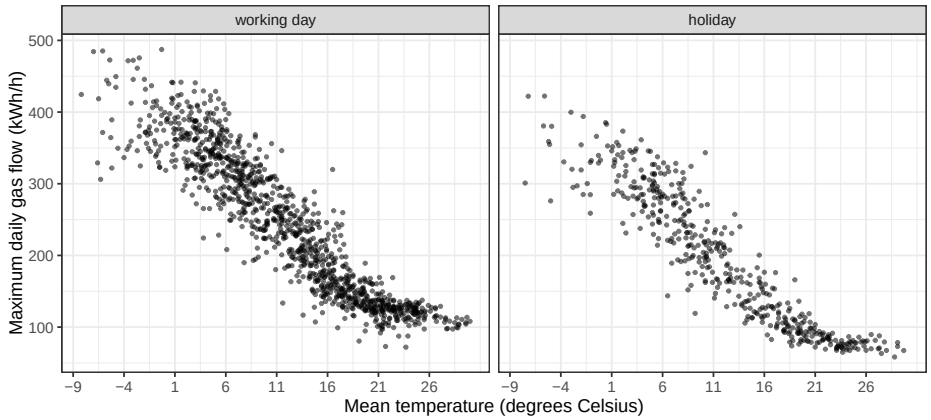

Figure 5.13: Gas flow data 2 with distinction working days (right) and holidays (left)

visualization, a reduced gas consumption on holidays can be concluded. The remaining data can be attributed to the gas consumption on working days ($1\,098$ data points) which induces the variability and the higher constant consumption level. The higher component containing the majority of the data points exhibits therefore a higher scattering and a clearly higher variability in gas consumption for low temperatures below zero. Both components comprise a significant variability which increases with decreasing outside temperatures. Given these conclusions, the Gamma distribution appears as an adequate choice for modeling the present mean dependent variation structure.

The main purpose of the subsequent analysis is to prove the adequacy of mixtures of GNMs as a suitable statistical model for the present data set. Drawing on the fitted models, special focus will be given to the prediction of gas consumption for low temperatures exceeding the available observed data. Representative design temperatures will be given below -12 °C which is considered as convenient for the climate region. Technical and methodological assumptions are presented in the subsequent section.

5.3.1 Two-Component Gamma Mixture Models

Due to the nature of the maximum daily gas flow and the increasing variability in dependence of the outside temperature, the subsequent fitting of mixtures of GNMs will focus on the Gamma distribution for the components. The fitting was also provided with the use of a normal distribution for comparative analysis. The application of normal components failed in the detection of two different minimum consumption levels due to the underlying constant variance structure within the components. A brief discussion on misfits and possible extensions is outsourced to Section 5.3.5. In order to achieve conclusive predictions for design temperatures below -12 °C, a mixture of GNMs will be applied to the data set. The derived model serves as a basic model class for the prediction of gas consumption for design temperatures where the sigmoid mean function (5.1) will be applied. As the graphical visualization in Figure 5.13 reveals, different consumption levels are being observed on working days and holidays. Therefore, the present work will address the adequacy of the sigmoid mean function including a holiday indicator by

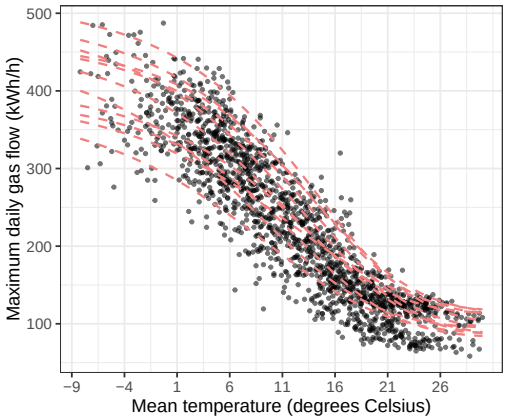

Figure 5.14: Starting configuration for gas flow data 2

means of Model (5.2).

5.3.1.1 Initial Configuration and Number of Components

In order to achieve reliable results, the data set will be fitted 50 times with randomly chosen starting values. The starting values are uniformly selected from the ranges as displayed in Table 5.9. Possible starting configurations are displayed in Figure 5.14. Due to the various configurations and the strong overlapping of the two components, the performance of the underlying EM algorithm is challenged. The initial cluster assignment is randomly set (default in `flexmix()`). The simulation procedure is repeatable by setting `set.seed(2357)` and `max.iter=100` as control variables. Due to the two evident minimum gas flow levels, the present study will focus in the beginning on two-component mixture models. The number of components will be increased in a subsequent step.

Parameter	β_{k1}	β_{k2}	β_{k3}	β_{k4}	β_{k5}
Range	$[390; 530]$	$[-35; -28]$	$[4; 6]$	$[65; 120]$	$[0; 0.5]$

Table 5.9: Ranges for starting values for gas flow data 2

The subsequent sections discuss the results of the application of two-component Gamma mixture models to the present data set as displayed in Figure 5.12. The underlying regression functions are given by Models (5.1) and (5.2).

5.3.1.2 Simulation Results

Table 5.10 summarizes the results of the fitting, based on randomly chosen starting values as defined in Table 5.9. The results are given by the mean values of the derived point estimates and the corresponding standard deviation. The results refer to both sigmoid regression functions through *Model (5.1)* and *Model (5.2)*. The overall results indicate reliable fitted values due to the manageable standard deviations. Considering the application of Model (5.1), one misfit was excluded in order to avoid a bias in the final

		Model (5.1)		Model (5.2)	
		mean	sd	mean	sd
Component 1	$\hat{\pi}_1$	0.66	0.00	0.52	0.00
	$\hat{\beta}_{11}$	396.93	0.00	423.65	0.62
	$\hat{\beta}_{12}$	-28.49	0.00	-30.22	0.04
	$\hat{\beta}_{13}$	6.13	0.00	6.81	0.03
	$\hat{\beta}_{14}$	110.72	0.00	116.50	0.07
	$\hat{\beta}_{15}$	-	-	0.08	0.00
	$\hat{\nu}_1$	63.36	0.02	93.40	1.68
Component 2	$\hat{\beta}_{21}$	522.91	0.08	393.71	1.24
	$\hat{\beta}_{22}$	-35.13	0.00	-26.31	0.08
	$\hat{\beta}_{23}$	4.84	0.00	5.39	0.03
	$\hat{\beta}_{24}$	69.04	0.00	68.44	0.02
	$\hat{\beta}_{25}$	-	-	0.22	0.00
	$\hat{\nu}_2$	42.49	0.02	35.92	0.49

Table 5.10: Simulation results for two-component Gamma mixtures with Models (5.1) and (5.2)

results. The number of iterations ranges between 55 and 84 when fitting Model (5.1) and between 19 and 40 for the second model including a working day indicator.

5.3.1.3 Model 1

Considering Model (5.1), the fitting procedure succeeds to reveal two components corresponding to the evidently different consumption levels for higher temperatures. The two fitted components appear moderately separated due to the dense data structure which is underpinned by the rootogram in Figure 5.16. The centered mass in the rootogram indicates overlapping observations for the two components. The higher component exhibits a greater variability and comprises the majority of the data points which reflects the original data structure as displayed in Figure 5.13. The results show that the gas consumption on working days tends to exceed those on holidays for outside temperatures above zero degrees Celsius. This relation twists for outside temperatures below zero degrees Celsius where the consumption on working days stabilizes at 397 kWh/h, while those on holidays increases to 523 kWh/h. This effect is clearly evident in the visualization of lower temperatures in Figure 5.15 (on the right). It furthermore reveals that most of the data points lying in the lower temperature region are allocated to the upper component with higher consumption levels and greater variability. Outliers above 450 kWh/h are allocated to the second component indicating the sharp increase to the upper asymptote at 523 kWh/h.

5.3.1.4 Model 2

The two-component Gamma mixture with Model (5.2) includes the information on working days and holidays for the gas flow data 2. The fitted components and mean functions are displayed in Figure 5.17 where the dashed lines refer to the gas flow on holidays. Similar to the application of Model (5.1), the fitted components succeed to

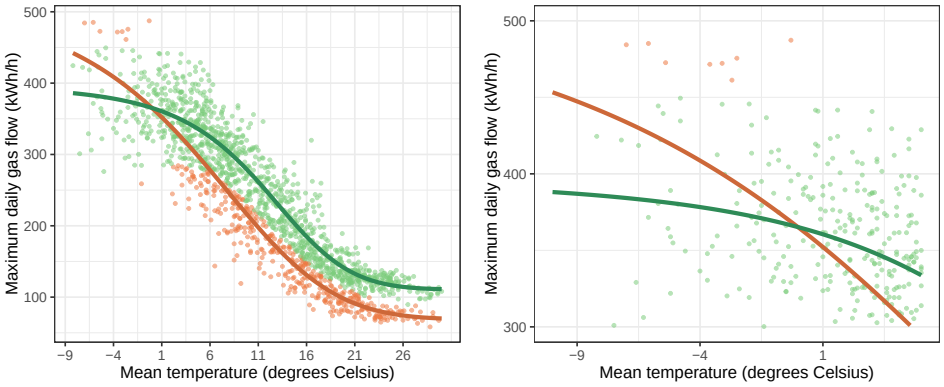

Figure 5.15: Fitted two-component Gamma mixture for Model (5.1)

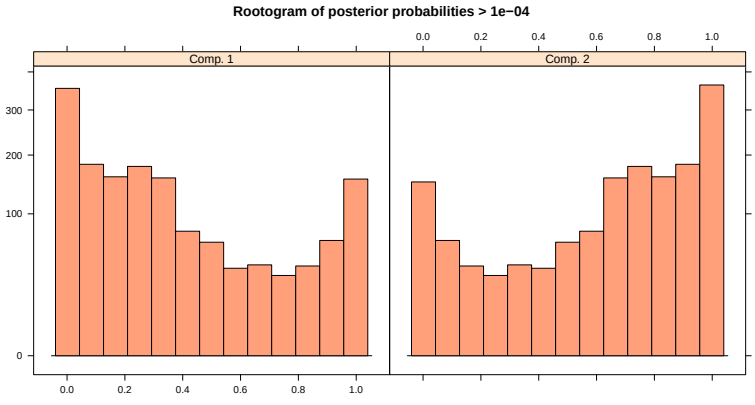

Figure 5.16: Rootogram for two-component Gamma mixture for Model (5.1)

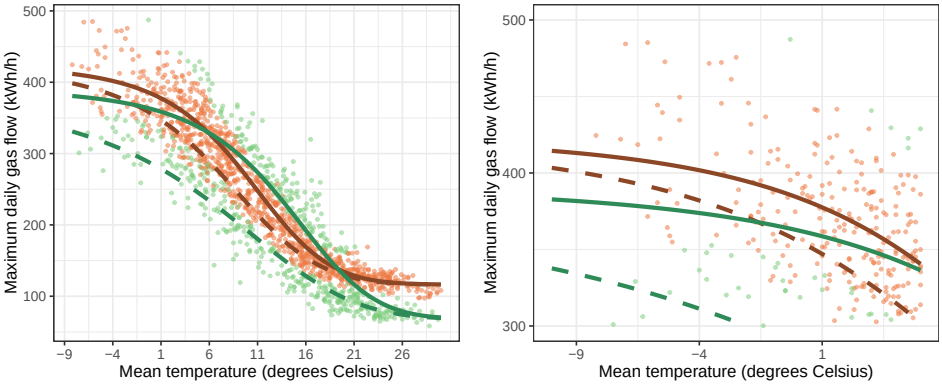

Figure 5.17: Fitted two-component Gamma mixture for Model (5.2)

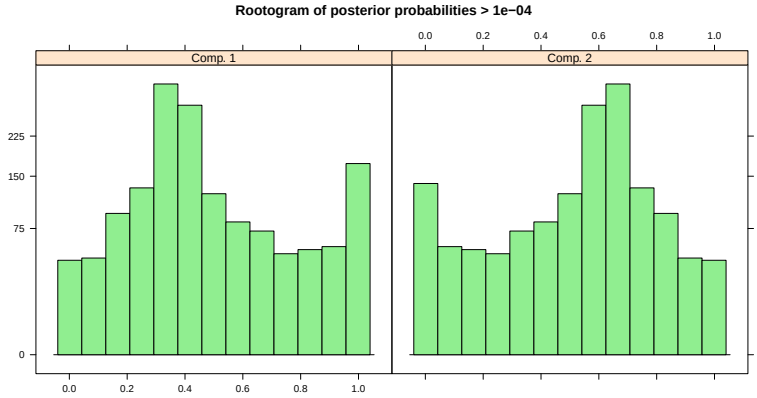

Figure 5.18: Rootogram for two-component Gamma mixture and Model (5.2)

model the two different minimum gas flow levels. The fitted mean functions show intersections which can be attributed to the dense structure of the data and the different shapes of the fitted curves. While the smaller component exhibits a more bellied shape with smaller variability, it is also embedded in the greater component yielding a considerable overlapping of the components. The rootogram in Figure 5.18 outlines a moderate separation due to the evident overlapping. Nevertheless, the different component allocation for the use of Model (5.2) results in a decrease of the upper asymptotes within the mixture model. As a consequence, the predicted gas flow for design temperatures is assumed to realize in smaller values compared to the use of Model (5.1).

		Model (5.1)		Model (5.2)	
		mean	s.e.	mean	s.e.
Component 1	$\hat{\pi}_1$	0.66	0.04	0.52	0.04
	$\hat{\beta}_{11}$	396.93	11.12	424.36	15.34
	$\hat{\beta}_{12}$	−28.49	0.44	−30.28	0.60
	$\hat{\beta}_{13}$	6.13	0.34	6.79	0.50
	$\hat{\beta}_{14}$	110.72	2.13	116.43	1.82
	$\hat{\beta}_{15}$	-	-	0.08	0.01
	$\hat{\nu}_1$	63.34	4.55	91.29	12.72
Component 2	$\hat{\beta}_{21}$	522.96	58.45	392.15	17.99
	$\hat{\beta}_{22}$	−35.13	1.85	−26.21	0.75
	$\hat{\beta}_{23}$	4.84	0.39	5.43	0.43
	$\hat{\beta}_{24}$	69.04	2.97	68.45	3.16
	$\hat{\beta}_{25}$	-	-	0.22	0.02
	$\hat{\nu}_2$	42.50	4.26	36.50	3.53
	AIC	15 628		16 326	
	BIC	15 687		15 245	
	ICL	16 300		16 326	

Table 5.11: Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 2 (two components)

5.3.2 Three-Component Gamma Mixture Models

The two-component Gamma mixture models indicate already a good fit as they succeed to identify the two heterogeneous minimum consumption levels. They furthermore reveal two components for the gas flow for low temperatures. As the first component comprises the majority of the observations and exhibits still a considerable scattering, the present section discusses the possible benefit of adding a third component in order to deal with the remaining variability.

5.3.2.1 Simulation results

Table 5.12 displays the results for 49 simulation runs for the three-component Gamma mixture model with Model (5.1). Within this context, the gas flow data 2 was fitted 50 times by uniformly chosen starting values stemming from the ranges in Table 5.9. One fitting result was excluded due to noticeable deviations in the log-likelihood and the ICL. The respective mixture model produced a splitting of the lower component whereas the upper component is considered to comprise inherent variability. Applying Model (5.1) yields satisfiable and stable results with major deviations in the upper and lower asymptotes. All simulation runs identified three distinct components. The control variables remain unchanged compared to the fitting of the two-component mixture model for reasons of comparability.
The results for the application of the three-component Gamma mixture models to Model (5.2) is displayed in Table 5.12 on the right side. The results in Table 5.12 comprise 29 fitted mixture models with three identified components. The algorithm drops components below the minimum prior threshold of 0.05 by default resulting in 12 fitted

		Model (5.1)		Model (5.2)	
		mean	sd	mean	sd
Component 1	$\hat{\pi}_1$	0.47	0.02	0.55	0.02
	$\hat{\beta}_{11}$	415.95	2.22	438.39	4.37
	$\hat{\beta}_{12}$	-29.73	0.11	-30.85	0.35
	$\hat{\beta}_{13}$	7.22	0.05	6.59	0.05
	$\hat{\beta}_{14}$	116.29	0.25	115.69	0.47
	$\hat{\beta}_{15}$	-	-	0.07	0.00
	$\hat{\nu}_1$	110.13	3.85	98.44	2.36
Component 2	$\hat{\pi}_2$	0.33	0.00	0.32	0.03
	$\hat{\beta}_{21}$	522.36	2.06	364.50	4.43
	$\hat{\beta}_{22}$	-35.53	0.06	-24.44	0.50
	$\hat{\beta}_{23}$	4.89	0.01	6.80	0.07
	$\hat{\beta}_{24}$	69.77	0.03	70.55	0.64
	$\hat{\beta}_{25}$	-	-	0.25	0.01
	$\hat{\nu}_2$	48.84	0.19	98.45	7.07
Component 3	$\hat{\beta}_{31}$	340.83	3.47	552.77	36.90
	$\hat{\beta}_{32}$	-24.43	0.26	-34.78	1.35
	$\hat{\beta}_{33}$	6.40	0.08	3.64	0.81
	$\hat{\beta}_{34}$	105.20	0.04	70.94	3.32
	$\hat{\beta}_{35}$	-	-	0.17	0.02
	$\hat{\nu}_3$	78.33	1.64	22.64	10.94

Table 5.12: Simulation results for three-component Gamma mixtures with Models (5.1) and (5.2)

two-component mixture models. The remaining models reached the maximum number of iteration steps 200 which terminated the fitting procedure without convergence. Compared to the three-component mixture models with Model (5.1), the inclusion of a working day indicator allocates more sample points to the first component. Model (5.2) aims in general to identify components with the information on working days and holidays. On that account, the identification of three components appears challenging. The smaller component exhibits higher deviations for the upper asymptote which can be attributed to the remaining variability of the data. The results can be considered as satisfiable as the point estimates in Table 5.12 show minor deviations.

5.3.2.2 Model 1

Figure 5.19 displays the gas flow data 2 colored by the component classification of the fitted three-component Gamma mixture model with Model (5.1). The major advancement, compared to the fitted components of the two-component mixture model, appears by the splitting of the upper component into two subgroups. The latter identify the minimum consumption level stemming from gas flow on working days. Both components diverge for temperatures below 20 °C where one component exhibits a more bellied shape up to an intersection at 7 °C. The further shapes show an evident discrepancy where the two components attain their upper asymptotes at 418 kWh/h and 520 kWh/h. The lowest

asymptote exhibits the same pattern as before within the two-component model where it coincides roughly with the gas flow on holidays.

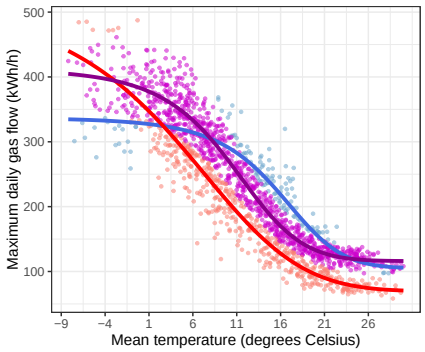

Figure 5.19: Fitted three-component Gamma mixture for Model (5.1)

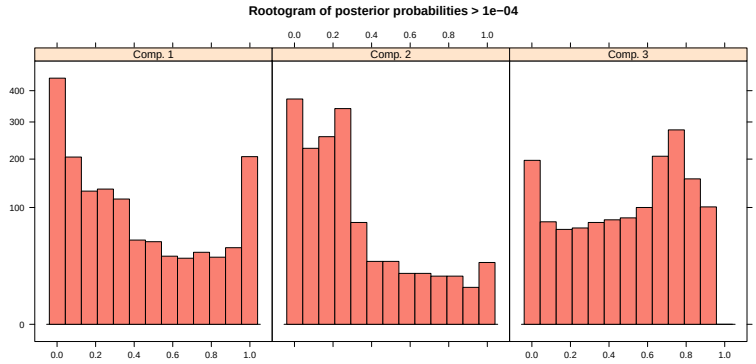

Figure 5.20: Rootogram for three-component Gamma mixture and Model (5.1)

The fitted regression coefficients for the three-component mixture model for both mean functions are shown in column `mean` in Table 5.13. The standard errors are given in the columns `std.err`. For reasons of comparability, the model selection criteria AIC, BIC and ICL are outlined for each model. A comparison of all fitted models is discussed in the subsequent section.

5.3.2.3 Model 2

The fitted components and mean functions of the gas flow data 2 by a three-component Gamma mixture and Model (5.2) are displayed in Figure 5.21. The addition of a third component yields a decrease in variability of the already identified two-component mixture model. The third component exhibits a linear pattern and high variability comprising outliers, particularly evident for gas flow observations ranging between 1 °C and 22 °C. The rootogram in Figure 5.22 and Listings 5.12 and 5.13 indicate a moderate separation.

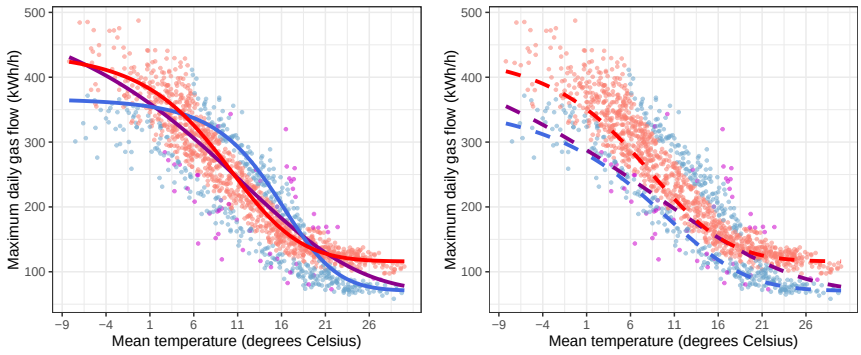

Figure 5.21: Fitted three-component Gamma mixture with Model (5.2) for working days (left) and holidays (right)

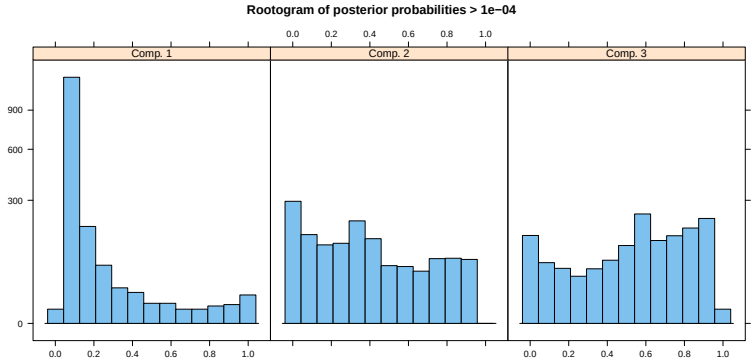

Figure 5.22: Rootogram for three-component Gamma mixture with Model (5.2)

```
> summary(m3.g)
        prior size post>0  ratio
Comp.1 0.325  472   1517 0.3111
Comp.2 0.489  951   1487 0.6395
Comp.3 0.186  130   1442 0.0902
```

Listing 5.12: `summary()` output for three-component mixture model and Model (5.1)

```
> summary(m3.gd)
        prior size post>0  ratio
Comp.1 0.134   53   1553 0.0341
Comp.2 0.537 1054   1515 0.6957
Comp.3 0.330  446   1484 0.3005
```

Listing 5.13: `summary()` output for three-component mixture model and Model (5.2)

5.3.3 Model Comparison

The present data set exhibits heterogeneous consumption structures for higher outside temperatures where both mixture models motivate the use of two different components. Simultaneously, the variability of the remaining observations initiates different components for the gas flow above the minimum consumption levels. The statistical model accuracy of the fitted mixture models can be compared by the AIC, BIC and ICL. Figure 5.23 displays the corresponding values for the fitted models.

		Model (5.1)		Model (5.2)	
		`mean`	`std.err`	`mean`	`std.err`
Component 1	$\hat{\pi}_1$	0.46	0.07	0.53	0.04
	$\hat{\beta}_{11}$	417.65	16.02	440.42	13.12
	$\hat{\beta}_{12}$	-29.80	0.63	-31.03	0.48
	$\hat{\beta}_{13}$	7.22	0.54	6.62	0.36
	$\hat{\beta}_{14}$	116.34	2.18	115.95	1.64
	$\hat{\beta}_{15}$	-	-	0.07	0.01
	$\hat{\nu}_1$	111.58	16.21	100.25	10.11
Component 2	$\hat{\pi}_2$	0.33	0.03	0.33	0.04
	$\hat{\beta}_{21}$	519.51	63.11	367.18	9.88
	$\hat{\beta}_{22}$	-35.47	1.95	-24.72	0.41
	$\hat{\beta}_{23}$	4.90	0.40	6.84	0.37
	$\hat{\beta}_{24}$	69.81	2.73	70.88	1.74
	$\hat{\beta}_{25}$	-	-	0.24	0.01
	$\hat{\nu}_2$	48.96	5.26	103.66	15.59
Component 3	$\hat{\beta}_{31}$	342.72	16.21	544.30	228.57
	$\hat{\beta}_{32}$	-24.56	0.96	-34.11	9.87
	$\hat{\beta}_{33}$	6.35	0.74	3.32	1.17
	$\hat{\beta}_{34}$	105.25	5.10	70.21	20.57
	$\hat{\beta}_{35}$	-	-	0.18	0.07
	$\hat{\nu}_3$	77.98	13.55	17.89	3.27
	AIC	15 568		15 109	
	BIC	15 659		15 216	
	ICL	16 736		16 277	

Table 5.13: Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 2 (three components)

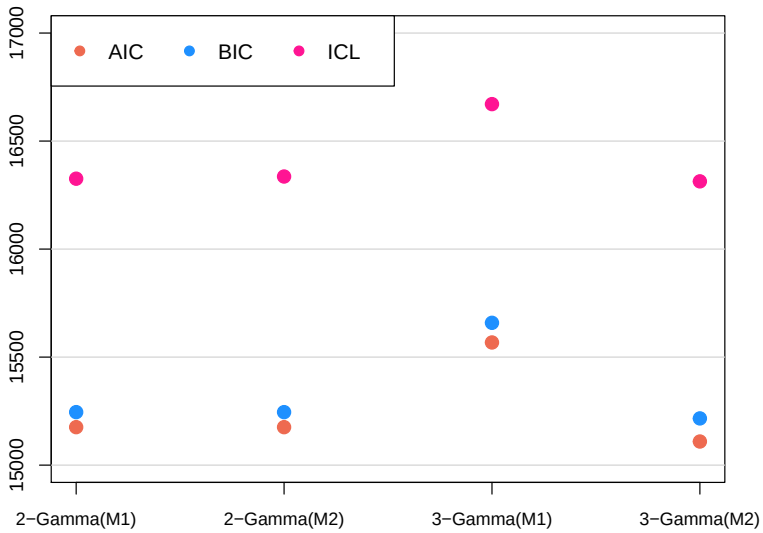

Figure 5.23: AIC, BIC and ICL for fitted models for gas flow data 2

The two-component mixture models result in similar values for the AIC, BIC and ICL. The addition of a third component increases the model complexity through additional distributional parameters while it yields a decrease in model accuracy considering the model selection criteria. The three-component mixture for Model (5.2) results in a lower ICL compared to Model (5.1) indicating a gain in model accuracy. Taking into consideration the linear structure and high variability of the third component (Figure 5.21), the latter comprises mainly outliers and does not exhibit the typical nonlinear shape for gas flow. The same mixture with Model (5.1) identifies otherwise a third gas flow component exhibiting a sigmoid shape. Therefore, the latter can be assumed to fit the shape of the present data set in a more reasonable way.

5.3.4 Predictions of Gas Flow on Design Temperatures

A key aspect of modeling gas consumption in dependence of outside temperature is given by the accurate prediction of gas flow for low temperatures. Within the present data sample, as displayed in Figure 5.12, the gas flow attains temperatures up to a level of about -10 °C. The fitted mean functions enable the prediction of gas flow for design temperatures for low temperatures below the observed values. The predicted values for the temperatures -12 °C, -14 °C and -16 °C are displayed in Table 5.14. In order to assess the variability of the predicted values, the 95% confidence intervals are displayed as additional information.

Design Temperatures:			**−12 °C**	**−14 °C**	**−16 °C**
2 Components	Model (5.1)	Gamma	414.9 (396.4, 433.5)	418.9 (398.5, 439.2)	422.1 (400.1, 444.0)
	Model (5.2) (holiday)	Gamma	377.0 (366.7, 387.9)	381.3 (369.9, 392.8)	385.0 (373.0, 396.9)
	Model (5.2) (working day)	Gamma	401.4 (387.6, 415.2)	403.0 (388.6, 417.3)	404.2 (389.3, 419.0)
3 Components	Model (5.1)	Gamma	412.5 (440.6, 384.5)	416.1 (386.3, 445.9)	419.0 (387.7, 450.4)
	Model (5.2) (holiday)	Gamma	387.5 (365.4, 409.6)	392.3 (369.4, 415.3)	396.4 (372.6, 420.2)
	Model (5.2) (working day)	Gamma	411.6 (382.8, 440.4)	414.1 (383.5, 444.7)	416.2 (383.8, 448.7)

Table 5.14: Prediction of gas flow for design temperatures for gas flow data 2

The predicted values exhibit differences up to 36 kWh/h. Large deviations in the predicted values arise in direct comparison between the application of Models (5.1) and (5.2), restricted to holidays. While Model (5.1) yields predictions for gas flow about 422 kWh/h, Model (5.2) predicts a flow about 404 kWh/h on working days and 385 kWh/h on holidays for a daily mean temperature -16 °C. As previously outlined, Model (5.2) attributes less importance to the outliers at the upper gas flow range. This effect is reflected in the predicted values through significantly higher predictions with Model (5.1). The addition of a third component yields opposite effects for the two different mean functions. The three-component model yields a decrease in predicted gas flow compared to the two-component model for mean function given by Model (5.1). The same approach increases the predicted values for Model (5.2).

5.3.5 Final Remarks

The specific gas flow data 2 in Figure 5.12 exhibits the typical gas flow pattern by means of an increasing variability for decreasing outside temperatures. It shows furthermore the specifics of two minimum gas flow levels. The presented applications indicate that the two-component Gamma mixture model succeeds to identify the evident heterogeneous structure. Alternative approaches comprise the application of the normal distribution for the components. The application of the two-component normal mixture model fails to identify the two different levels of minimum gas flow for higher temperatures. This misfitting can be attributed to the distributional property of constant variances within normally distributed components. The Gamma distribution succeeds to model the increasing variability in the present data set which proves as a suitable application for the present model class. Building on the presented results, the two-component Gamma mixture models show a better model accuracy compared to the three-component Gamma mixture models for the gas flow data 2.

5.4 Gas Flow Data 3

The third data set stems from an European gas distribution point available on the transparency platform for European gas providers `https://transparency.entsog.eu/` under the Energy Identification Code (EIC) 27ZG-UVAL-CZ-PLZ. The website provides a transparency platform for gas transmission and is supplied by the European Network of Transmission System Operators for Gas (ENTSOG). The considered time span ranges from September 2016 to September 2018. The corresponding data sample comprises gas flow data on hourly basis which was reduced to the maximum daily gas flow for the present analysis (757 points). The gas flow is measured in kilowatt hours per hour (kWh/h). The available gas flow data was merged with the mean temperatures from ZAMG (2018) from Austria in degrees Celsius (°C) which is considered as representative for the climate region. The resulting daily gas flow data 3 is displayed in Figure 5.24 by means of the time series and the scatterplot of the gas flow in dependence to the outside temperature.

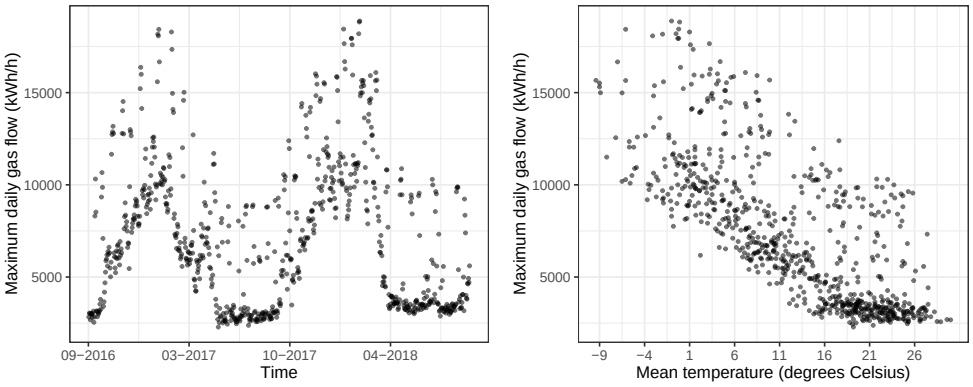

Figure 5.24: Gas flow data 3

The scatterplot in Figure 5.24 (on the right) reveals two evident components in the present data set. The main component, comprising the majority of the data points, exhibits a nonlinear consumption level from $3\,000$ kWh/h to a maximum of $15\,000$ kWh/h. Simultaneously, the data shows a second component exhibiting a higher consumption level ranging from $7\,000$ kWh/h to $20\,000$ kWh/h. The two evident components appear as parallel bands with a low level of overlapping between them. The present data structure resembles the synthetic data set in the simulation study in Chapter 4. The latter aimed to simulate two different sized components representing heterogeneous consumption levels, as for example, between private households and industrial customers. As the fitting procedure obtained already satisfiable results for the synthetic data set, the application to gas flow data 3 appears even more insightful on the performance of the new algorithm. As supplementary information, the data set is visualized in dependence of working days and holidays in Figure 5.25.
Therefore, the higher consumption level in Figure 5.24 may be attributed to the industrial sector as it appears on working days (defined from Monday to Friday). As both components share the same level of consumption given by the lower component, the daily

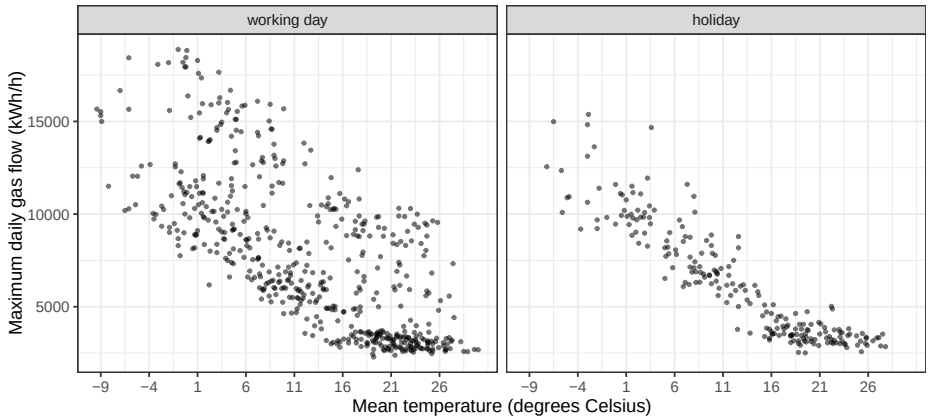

Figure 5.25: Gas flow data 3 with distinction working days (right) and holidays (left)

indicator is not considered as a decisive variable in the mixture modeling. Exemplary, the reduction of the data to working days reveals no structural change in the data which exhibits still the same heterogeneity as visible in the original data set. The present data are furthermore assumed to consist of two heterogeneous consumption groups where the daily indicator does not provide a key information. Given the present heterogeneity in the data, the use of a mixtures with Model (5.1) as mean function appears useful. Therefore, the two components are fitted simultaneously ensuring component specific consumption modeling given a joint mixture distribution.

5.4.1 Two-Component Mixtures of GNMs

The present data exhibit a considerable scattering even for very high temperatures where other gas flow data converge to a so-called minimum consumption level. This effect distorts the typical increase in variability for low temperatures and generates a band-like structure in the present data. Based on these considerations, the subsequent analysis will focus on two-component mixture models with components based on the normal and the Gamma distribution and Model (5.1).

5.4.1.1 Initial Configuration and Number of Components

In order to achieve reliable results, the present data set will be fitted 50 times with randomly selected starting values. The starting values stem from the ranges as displayed in Table 5.9. Possible starting configurations are displayed in Figure 5.26. Due to the various initial configurations, the performance of the underlying EM algorithm is challenged. The initial cluster assignment of the sample points is randomly set (default in `flexmix()`). The simulation procedure is repeatable by setting `set.seed(2357)`.

Parameter	β_{k1}	β_{k2}	β_{k3}	β_{k4}
Range	$[10\,000; 20\,000]$	$[-35; -28]$	$[4; 10]$	$[2\,000; 8\,000]$

Table 5.15: Ranges for starting values for gas flow data 3

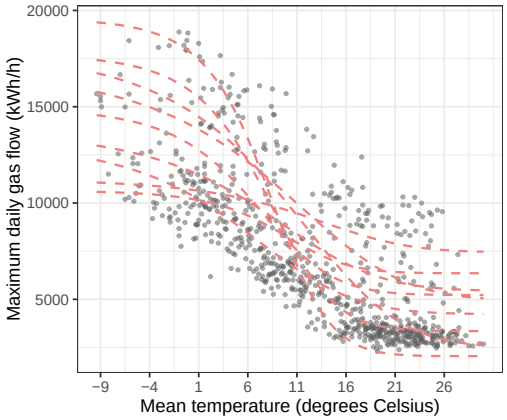

Figure 5.26: Starting configuration for gas flow data 3

The subsequent sections discuss the fitted two-component mixture models for the gas flow data 3 as displayed in Figure 5.24. The underlying distributions are given by the normal and Gamma distribution while Model (5.1) specifies the mean function.

5.4.1.2 Simulation Results

Table 5.16 displays the results for the simulation runs through the mean values and standard deviations of the fitted point estimates. The fitted components are ordered in decreasing order with respect to the prior weights. The results for the normal and Gamma distribution indicate similar results. The majority of the data points is attributed to the lower component which exhibits a smaller variability compared to the upper component. The upper component differs slightly in the shape of the mean function between the two distributions. The Gamma distribution produces even broader ranges for the upper component. Considering the diverse starting configurations in Figure 5.26, the results are satisfactory and reliable due to the manageable standard deviations over all fitted coefficients. The number of iterations ranges between 60 and 70 for the normal distribution and between 30 and 39 for the Gamma distribution.

5.4.1.3 Two-Component Normal Mixture Model

The two-component mixture model with an underlying normal distribution identifies two band-like components along Model (5.1) as mean function. The shapes of the two mean functions exhibit a strong similarity while the upper component is still steeper and slightly bellied. Additionally, the upper component exhibits a greater variability which is underpinned by the differences in $\hat{\sigma}_1$ and $\hat{\sigma}_2$ in Table 5.17. Therefore, the upper component comprises those values with higher gas flow levels as well as outliers compared to the main (lower) component comprising the majority of the data points. The fitted components are displayed in Figure 5.27 with the corresponding mean functions. The observations are colored according to the final component allocation.
The fitted components reveal two different consumption levels for the overall temperature range. They are well-separated as the data exhibit a band-like structure and a low

		2-Component Gaussian			2-Component Gamma	
		mean	sd		mean	sd
Component 1	$\hat{\pi}_1$	0.75	0.00		0.74	0.00
	$\hat{\beta}_{11}$	11 963.84	0.23		12 176.25	0.06
	$\hat{\beta}_{12}$	−32.23	0.00		−32.30	0.00
	$\hat{\beta}_{13}$	6.35	0.00		6.54	0.00
	$\hat{\beta}_{14}$	3 020.60	0.10		2 982.68	0.03
	$\hat{\sigma}_1$	916.97	0.23	$\hat{\nu}_1$	42.88	0.01
Component 2	$\hat{\beta}_{21}$	16 723.85	1.29		17 095.62	1.20
	$\hat{\beta}_{22}$	−30.98	0.00		−31.30	0.00
	$\hat{\beta}_{23}$	7.37	0.00		6.33	0.00
	$\hat{\beta}_{24}$	7 883.43	0.86		7 511.06	0.10
	$\hat{\sigma}_2$	1 852.22	1.22	$\hat{\nu}_2$	26.12	0.03

Table 5.16: Simulation results for two-component mixtures with Model (5.1)

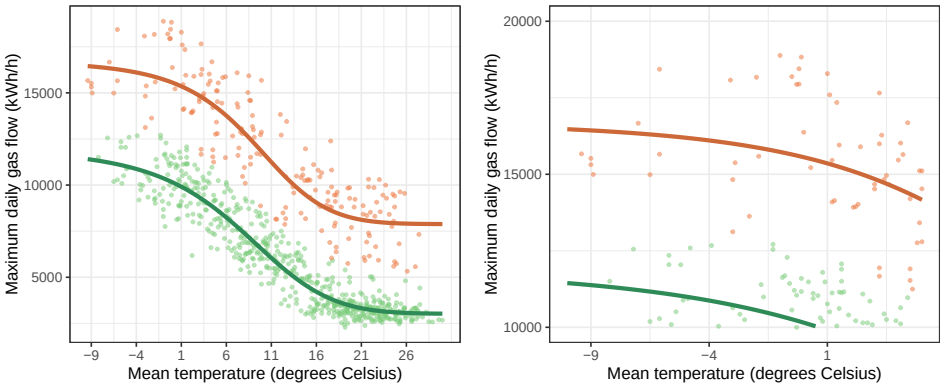

Figure 5.27: Fitted two-component normal mixture for Model (5.1)

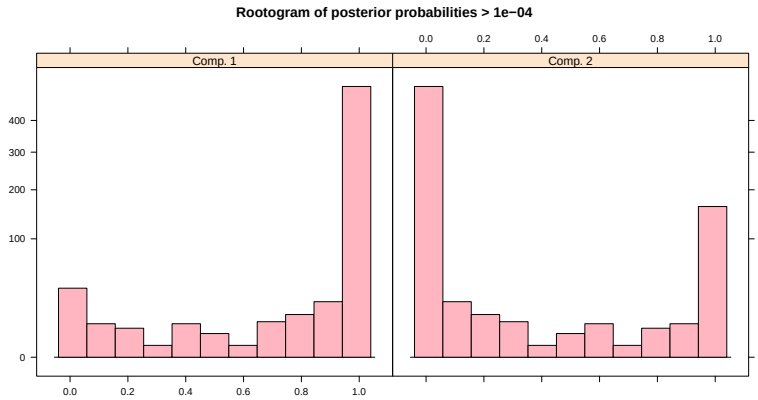

Figure 5.28: Rootogram for fitted two-component normal mixture for Model (5.1)

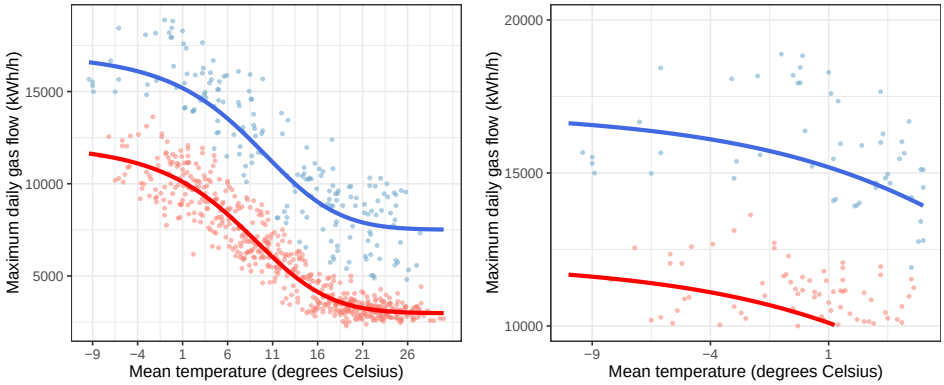

Figure 5.29: Fitted two-component Gamma mixture for Model (5.1)

overlapping between the two components. The rootogram in Figure 5.28 underpins the conclusion on the separation where little mass is concentrated in the center of the scale. It actually reveals a clear allocation to the components by means of concentrated mass at zero and one. As the modeling of gas flow for design temperatures is of particular interest, the respective temperatures are displayed separately in Figure 5.27 in the right graphics. The predicted gas flow depends on the fitted mean functions and the prior weights of the fitted components. A discussion on the prediction of gas flow for design temperatures for gas flow data 3 is given in Section 5.4.3.

5.4.1.4 Two-Component Gamma Mixture Model

The two-component Gamma mixture model reveals two well separated components for Model (5.1). The fitted components and mean functions are displayed in Figure 5.29 (colored according to final component allocation) and resemble those of the two-component normal mixture model. The lower component comprises again the majority of the data points and exhibits a lower variability compared to the upper component. This conclusion is reflected by the fitted shape parameters $\hat{\nu}_1$ and $\hat{\nu}_2$ in Table 5.17. The fitted components are well-separated and the rootogram corresponds in general to those given by the two-component normal mixture model. The fitted coefficients and their standard errors are displayed in Table 5.17.

5.4.2 Model Comparison

This section briefly summarizes the specifics of the fitted models for the gas flow data 3 as displayed in Figure 5.24. The fitted coefficients in Table 5.17 indicate very similar results for the use of the normal and the Gamma distribution due to the band-like structure of the data. The fitted components for an underlying Gamma distribution reveal noticeable greater ranges as the fitted coefficients for the upper asymptotes exceed those of the normal distribution for both components. Taking into account the model selection criteria, the latter attribute the two-component Gamma mixture model a significant gain in accuracy as, for example, the ICL drops to $13\,499$ compared to the normal distribution (ICL$=13\,607$). Based on these conclusions, the two-component Gamma mixture model

		2-Component Gaussian			2-Component Gamma	
		mean	s.e.		mean	s.e.
Component 1	$\hat{\pi}_1$	0.75	0.04		0.74	0.03
	$\hat{\beta}_{11}$	11 964.18	397.34		12 176.12	616.80
	$\hat{\beta}_{12}$	−32.23	0.54		−32.30	0.80
	$\hat{\beta}_{13}$	6.35	0.48		6.54	0.49
	$\hat{\beta}_{14}$	3 020.74	99.94		2 982.61	57.41
	$\hat{\sigma}_1$	917.31	45.85	$\hat{\nu}_1$	42.91	3.44
Component 2	$\hat{\beta}_{21}$	16 725.70	764.06		17 093.18	1 598.78
	$\hat{\beta}_{22}$	−30.98	1.09		−31.30	2.02
	$\hat{\beta}_{23}$	7.37	1.77		6.33	1.73
	$\hat{\beta}_{24}$	7 884.68	449.15		7 510.86	383.63
	$\hat{\sigma}_2$	1 850.46	209.02	$\hat{\nu}_2$	26.07	4.46
	AIC	13 511			13 386	
	BIC	13 562			13 437	
	ICL	13 607			13 499	

Table 5.17: Fitted coefficients, standard errors, information criteria and dispersion parameters for gas flow data 3 (two components)

can be considered as the more appropriate model for the present gas flow data 3.

5.4.3 Predictions of Gas Flow on Design Temperatures

The previously discussed models succeed to describe the heterogeneous band-like structure of the present data set. The data was fitted for the normal and Gamma distribution for Model (5.1). Both mixture models yield similar results for the fitted mean functions whereas the two-component Gamma mixture model indicates a better statistical significance. The two-component mixture models are suitable to compute predictions for gas flow for low temperatures beyond the observed temperatures. The predicted values are displayed in Table 5.18 and yield similar results for both distributions. As the Gamma distribution attains slightly higher upper asymptotes for both components, the predicted values exceed those for the normal distribution. Exemplary, the predicted gas flow for -16 °C estimates about $13\,215$ kWh/h for the Gamma distribution whereas the normal distribution yields a gas flow prediction about $12\,952$ kWh/h. The 95% confidence intervals are furthermore wider for the Gamma distribution which can be attributed to the higher standard errors for the respective coefficients.

5.4.4 Final Remarks

The present gas flow data 3, as displayed in Figure 5.24, exhibits the specific gas flow pattern with two band-like structured components. The upper component includes less data points while it exhibits a greater scattering compared to the lower component. A similar gas flow pattern was considered in order to trial the performance of the new fitting method for mixtures of GNMs by means of a simulation study in Chapter 4. Therefore, synthetic data sets represented two consumption groups with different consump-

Design Temperatures:			**-12 °C**	**-14 °C**	**-16 °C**
2-Component	Model (5.1)	Normal	12 820.4 (12 220.6, 13 420.2)	12 894.7 (12 266.3, 13 523.2)	12 951.9 (12 299.2, 13 604.6)
	Model (5.1)	Gamma	13 071.8 (12 112.6, 14 031.0)	13 152.8 (12 137.22, 14 168.3)	13 215.1 (12 153.0, 14 277.2)

Table 5.18: Prediction of gas flow for design temperatures for gas flow data 3

tion levels and sizes. The present gas flow data 3 correspond to real data which were fitted by the sigmoid mean function given by Model (5.1) for the normal and Gamma distribution. The evident band-like structure motivates the use of a two-component mixture model in order to comprise the two different consumptions levels. Both mixture models yield similar results regarding the fitted mean functions and the predicted gas flow for design-temperatures. Considering an underlying Gamma distribution, the upper asymptotes of the components attain slightly higher values compared to those of a normal distribution. The two-component Gamma mixture model proves furthermore as the statistically more significant model due to well-known model selection criteria.

5.5 Conclusions

The present Chapter discussed advanced applications for the modeling of gas flow by means of mixtures of GNMs. Mixtures of GNMs add a certain degree of flexibility in the modeling as they provide additional model parameters as well as an advanced distributional setting. Based on the work of Friedl et al. (2012), the nonlinear regression model (5.1) with Gamma distributed responses appeared as an insightful extension of the already available results. The results indicate a suitable approach for the given data set and the used sigmoid growth curve despite the dense structure of the data. Further computations were made with an additional predictor variable denoting working days and holidays which increased the model accuracy in general. The subsequent application of the new class of mixtures of GNMs focused on the fitting of gas flow for two-component mixture models. The two-component Gamma mixture model proved as a suitable model for gas flow data. The application of mixtures of GNMs yields decreasing predicted values for design temperatures compared to the single nonlinear regression models. The methodology was applied to several data sets exhibiting challenging component configurations. Considering gas flow data 2, the algorithm enables to detect two different components stemming from heterogeneous gas flow levels. Therefore, the Gamma distribution proved throughout as the more suitable distribution considering the natural variability of gas flow. Within this context, Gamma mixtures succeed to identify even distinct minimum consumption levels while the application of normal mixtures failed in their identification. The methodology proved robust by the detection of three components preserving the data structure of minimum consumption levels. Following the pattern of the synthetic data set in the simulation study, gas flow data 3 exhibit a band-like structure motivating the use of a two-component mixture model. Therefore, mixtures of GNM proved as an appropriate method in order to detect the distinct gas flow levels. The applications proved throughout robust to the challenge by randomly chosen starting values. Nevertheless, special attention has to be drawn to a proper choice of the starting values for the mixture components as they highly influence the convergence and the results of the underlying EM algorithm.

CHAPTER 6

Modeling Multiple Regimes in Economic Growth

Introduction

The present chapter gives a new approach to the well-known *Solow model*. *Robert Merton Solow* und *Trevor Swan* developed the *Solow model* to describe the economic growth of an economy by means of the production output as presented in Solow (1956) and Swan (1956). The *Cobb-Douglas* function represents a key property and relates the production output to predefined economic factors. An economy's production output is typically measured through the Gross Domestic Product (GDP). For reasons of comparability, the GDP is divided by an economy's population expressing the GDP per capita (per person). The Solow model is often applied due to the possibility of adding arbitrary factors in order to improve the model quality. Due to the mathematical tractability, the Solow model has been widely studied and advanced. Extensions comprise advanced models with an extended number on production factors. The related economic factors are often given by the capital stock or the investment share of the economy, which can be considered as important drivers of the GDP development. Typical production factors comprise also the workshare performed by the related population of an economy. An obvious application is given by the comparison of country specific development of the GDP in order to asses the statistical value of predefined economic factors. The economic factors within the Solow model may often differ for different countries producing an evident heterogeneity in cross-sectional comparisons of the GDP. This is especially the case when comparing global data where, for example, developing countries exhibit exceptional patterns in comparison to industrial countries. Another example for heterogeneity may be given by exceptional drivers of the economic growth due to country-specific dominant industries as it is the case for the so-called oil countries. Due to differences in the economic conditions, the *technological knowledge* represents an economic factor varying across different (national) economies. This effect is driven by differences in resource endowments, institutions or climate conditions as Mankiw et al. (1990, p. 410-411) point out. These circumstances aggravate the modeling of the development of the GDP through one coherent Solow model. Durlauf and Johnson (1995) follow therefore the approach

of modeling the national GDP by means of regression trees enabling the modeling of heterogeneous subgroups. The separate modeling of subgroups represents a *multiple regime* model for economic growth. Alfo et al. (2008) consider a mixture model approach by modeling the economic growth through mixtures of linear regressions. In the following, the standard approach for the Solow model will be discussed and applied to mixtures of GNMs. While the research of Alfo et al. (2008) follows the approach to cluster the present data into subgroups and apply different linear models, the present work aims to model the original nonlinear function with mixtures of GNMs based on probabilistic clustering through the EM algorithm (Section 2.4). The main focus of the present analysis lies in the application of the new model class of mixtures of GNMs to the country-specific economic growth model as an approach to deal with heterogeneous data.

The underlying regression function will be outlined in the subsequent Section 6.1. Section 6.2 gives an overview on the underlying data set and underpins the use of mixture models by discussing graphical visualizations of the data. The reliability of the fitted results will be challenged through different starting configurations where possible values are discussed in Section 6.3. For comparative purposes, the data will be fitted by simple nonlinear regression. The underlying approach is briefly outlined in Section 6.4. Section 6.5 discusses the central application by a two-component Gaussian mixture model and highlights key results. Section 6.6 concludes the chapter with final remarks.

6.1 Solow Model with Human Capital Accumulation

The central measure of interest is the production or output of an economy when drawing comparisons on cross-national levels. The underlying factor, measuring the country specific economic growth, will be given by the logarithmic return of the GDP growth for the period $[0, t]$ which will be denoted as Y_t. The standard *Cobbs-Douglas* production function models the production output through the following relationship

$$Y_t = K_t^{\alpha} H_t^{\gamma} (A_t L_t)^{1-\alpha-\gamma}, \tag{6.1}$$

where K_t represents the capital, A_t the level of technology, L_t the labor and H_t the stock of human capital, while $0 < \alpha, \gamma < 1$ holds. The labor and technology components are assumed to follow an exponential growth process with exogenous rates n and g as given in Mankiw et al. (1990, p. 409), respectively

$$L_t = L_0 \exp nt \quad \text{and} \quad A_t = A_0 \exp gt. \tag{6.2}$$

Therefore the number of effective units of labor $A_t L_t$ grows at the rate $n+g$. All variables are assumed to evolve in continuous time according to Durlauf and Johnson (1995).

Another important assumption stated within the Solow model is the convergence to the *steady state*. This assumption is supported by diminishing returns inducing the economic long-term growth rate to reach a steady state. As a necessary condition, $\alpha + \gamma < 1$ needs to hold. Within the steady state, the production output per worker can be derived as

$$\log\left(\frac{Y_t}{L_t}\right) - \log\left(\frac{Y_0}{L_0}\right) = gt + (1 - \exp(-\lambda t)) \cdot \ldots$$
$$\left(\Theta + \frac{\alpha}{1-\alpha-\gamma}\log\left(\frac{s_K}{n+g+\delta}\right) + \frac{\gamma}{1-\alpha-\gamma}\log\left(\frac{s_H}{n+g+\delta}\right) - \log\left(\frac{Y_0}{L_0}\right)\right), \quad (6.3)$$

where $\Theta = 1/(1-\alpha-\gamma)\log(\phi) - \log A_0 - gt$ holds and $\lambda = (1-\alpha-\gamma)(n+g+\delta)$ represents the country specific *convergence rate* towards the steady state model. According to Mankiw et al. (1990, p. 410), g will be assumed as a constant value representing the technological advancement. The parameter δ represents the depreciation rate and is also assumed as a constant for all national economies. In contrast to these assumptions, the parameter A_0 represents a country specific technological endowment. The parameters s_K and s_H denote the saving rates for the physical and the human capital, compliant to Mankiw et al. (1990). Within the steady state, the GDP is influenced by the technological endowment. The aim of the present work is to examine if the present data obey multiple regimes corresponding to Durlauf and Johnson (1995, p. 368) by means of distinct components following the nonlinear regression function (6.3). For this purpose, the subsequent analysis considers a two-component mixture model of GNMs. The technical derivation of the steady state function (6.3) is given in Appendix B. The subsequent sections comprise the application of mixtures of GNMs with nonlinear regression function (6.3) and discuss the fitted results. The fitting is carried out with the new package **flexmixNL**.

6.2 Country Data Set

In order to achieve comparability to the original study by Durlauf and Johnson (1995), the present applications consider the original data set provided by Summers and Heston (1988). This data set consists of economic variables for 121 countries for the period from 1960 to 1985. As the authors expected the former oil countries not to exhibit the standard growth process considering the GDP, these countries were excluded from the statistical analysis. The authors furthermore omitted countries with populations less than a million within specific applications in order to reduce measurement errors. Applying these restrictions, reduces the country sample to 75 countries. Table 6.1 summarizes the specific regression variables for the mean function (6.3).

Variable	Explanation
`log25`:	log-return of GDP per working member of population aged 15 - 64
`IONY`:	GDP share devoted to investments s_K (annual averages)
`POPGRO`:	growth rate n of working age population (annual averages)
`SCHOOL`:	working age population in secondary school s_H (annual averages)

Table 6.1: Cross-country regression variables for time period 1960 - 1985

As Mankiw et al. (1990, p. 413) point out, the depreciation rate δ and advancement in knowledge g is not expected to vary across countries. In order to match the original

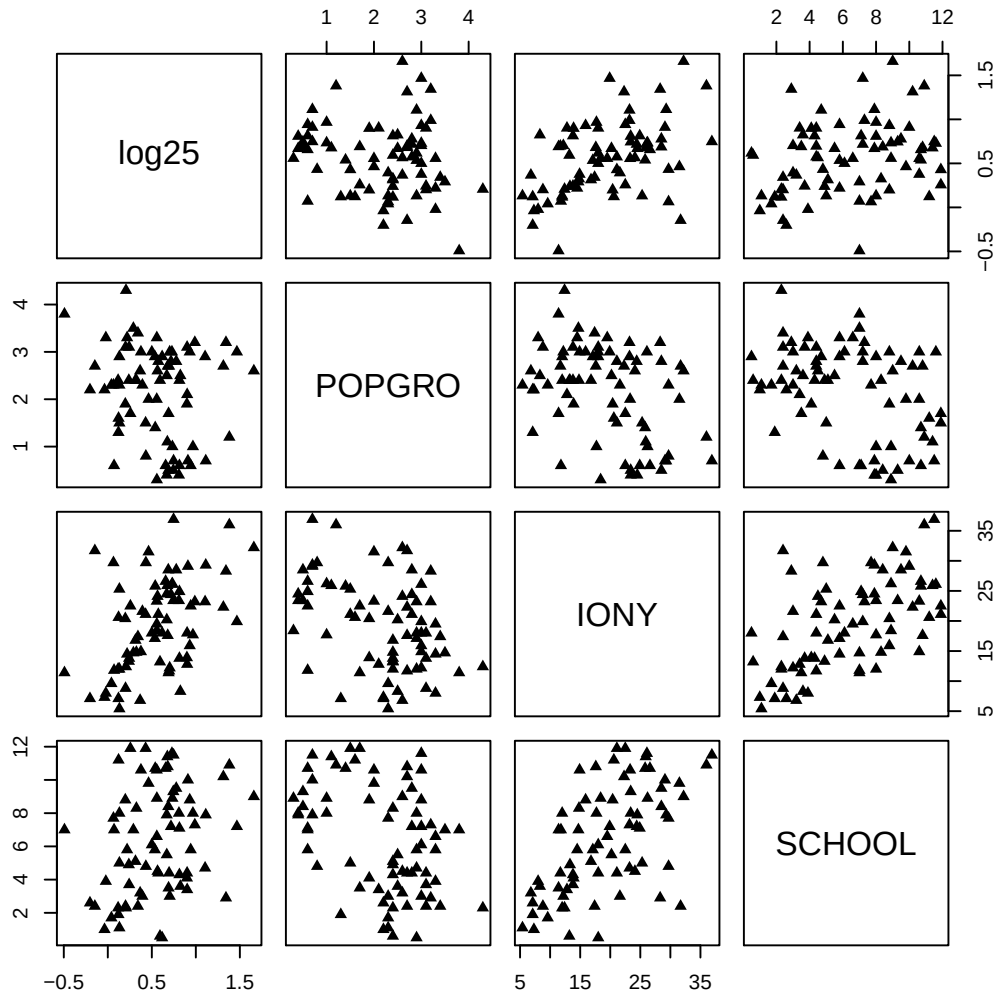

Figure 6.1: Cross-correlation for GDP growth and production factors

data, the value $g + \delta$ is chosen as 0.05 (derived from the United States economy as representative values). Apart from that, the knowledge A_0 is assumed to be unobservable and can vary between different national economies. The economic factors in Table 6.1 will be considered within the present application in order to model the GDP growth. The cross-correlation between the relevant economic factors is visualized in Figure 6.1. The pairwise scatterplots in Figure 6.1 reveal an evident positive correlation between the log-return of the GDP growth and the average logarithmic investment share. On the other hand, the GDP growth is moderately negatively correlated with the population growth of a national economy. The GDP growth exhibits also a moderate positive correlation with the education level of the population. The scatterplot of the production factors reveals also a positive correlation between the investment share and the education level of the population. These effects can be explained in a plausible way: increasing investments have a positive effect on the production output. Furthermore, the GDP per capita grows in the case of a decreasing population growth rate. Consequently, an increase in eco-

nomic growth can be achieved by decreasing the population growth within the present model. Increasing population growth decreases in general the remaining economic factors. This effect can be attributed to the distribution of the same countries' resources to an increased population as Mankiw et al. (1990, p. 418) outline. The opposite effect occurs for a decreasing population. The education level of the population influences the production output directly in a positive way as it increases simultaneously the technical advancement. These effects yield a positive cross-correlation between the log-return of the GDP growth and the considered education level in Figure 6.1.

As the application of mixture models considers the underlying distribution of the response `log25`, Figure 6.2 displays the empirical pdf of the modeled log-return of the economic growth (GDP). The visualization shows an evident multimodal structure driven by peaks at 0.2 and 0.7 and several countries exhibiting a higher GDP growth ranging around 1.4. The subsequent analysis points out the fitting of the response by means of a two-component Gaussian mixture model. A main focus within the mixture model will be the dealing with the multimodality in dependence of the underlying regression function (6.3). The distributional assumption will focus on the normal distribution following the original analysis by Durlauf and Johnson (1995).

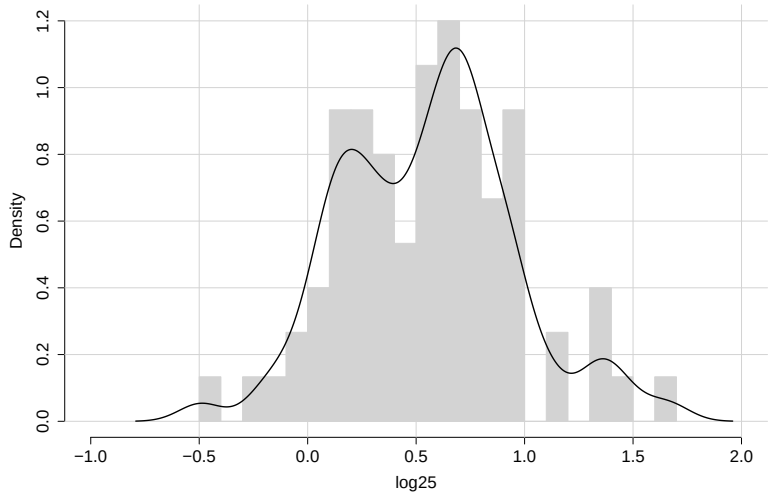

Figure 6.2: Density function of GDP growth

6.3 Starting Configuration and Simulation

The response `log25`, as displayed in Figure 6.2, will be fitted according to the nonlinear regression function (6.3). The explanatory variables comprise the countries' investment share (`IONY`), the population growth (`POPGRO`) and the education level (`SCHOOL`) as specified in Table 6.1. The regression function will be fitted through simple nonlinear regression and a two-component mixture of GNMs. The specification in R is given in Listing 6.1 where the functional structure is stored in the term `eco.fct()`.

```
> eco.fct = function(alpha,gamma,theta,POPGRO,IONY,SCHOOL){
+   0.5 + (1-exp(-(1-alpha-gamma)*((POPGRO + 5)/100)*25))*
+     (theta + log((IONY/(POPGRO + 5))/100) * alpha/(1-alpha-gamma) +
+      log((SCHOOL/(POPGRO + 5))/100)*gamma/(1-alpha-gamma)-log(GDP60))}
```

Listing 6.1: Regression function for GDP growth model in R

The present data set, as displayed through the cross-relationships of the production factors in Figure 6.1, is challenging for the mixture model due to two aspects: The functional relationship is based on multiple nonlinear dependency structures and it can be assumed that the derived components exhibit a strong overlapping. In order to explore the accuracy and performance of the fitting method provided by **flexmixNL**, the given data set will be fitted several times with randomly chosen starting values. The ranges for possible values were chosen to allow for a sufficient flexibility in modeling. They comprise values around the fitted coefficients obtained by the single nonlinear regression assuming a coherent single steady state. Exemplary, the coefficients α and γ are chosen to range between 0.01 and 0.6. According to Mankiw et al. (1990, p. 416), the assumption $\alpha + \gamma < 1$ is required in order to achieve convergence to steady state. The ranges for the starting values for the two-component mixture models are given in Table 6.2. The variability of the derived coefficients will be addressed by the computation of standard errors $\mathrm{SE}^{num}(\cdot)$ following Section 3.5.

Parameter	Θ	α	γ
Range	[10; 20]	[0.01; 0.60]	[0.01; 0.60]

Table 6.2: Ranges for starting values

The starting values, stemming from the ranges in Table 6.2, will be applied to the simple nonlinear regression model and to a two-component Gaussian mixture model. Fitting the simple nonlinear regression model and the two-component mixture model 50 times with randomly chosen starting values achieves the results as summarized in Table 6.3. The fitting was controlled by setting a fixed seed `set.seed(2357)` and predefined classification vector comprising an alternating allocation to the two components for the data sample.

6.4 Nonlinear Regression

The nonlinear mean function for modeling the production output is given by Equation (6.3) compliant to the original problem from Mankiw et al. (1990). The fitting in R can be provided with the function `nls()` as discussed in Section 3.2.1. The corresponding command and output is given in Listing 6.2.

```
> nls1 <- nls(log(GDP85/GDP60) ~
+               eco.fct(alpha,gamma,theta,POPGRO,IONY,SCHOOL),
+             data = durlauf,
+             start = list(theta=runif(1,10,20),
+                          alpha= runif(1,0.01,0.6),
+                          gamma = runif(1,0.01,0.6)))
```

```
> nls1
Nonlinear regression model
  model: log(GDP85/GDP60)~eco.fct(alpha,gamma,theta,POPGRO,IONY,SCHOOL)
   data: durlauf
  theta   alpha   gamma
16.1161  0.4113  0.2507
 residual sum-of-squares: 6.794

Number of iterations to convergence: 5
Achieved convergence tolerance: 9.186e-08

> summary(nls1)
...
Parameters:
      Estimate Std. Error t value Pr(>|t|)
theta 16.11607    0.76712  21.009  < 2e-16 ***
alpha  0.41125    0.05852   7.028 9.82e-10 ***
gamma  0.25074    0.04762   5.265 1.39e-06 ***
```

Listing 6.2: Nonlinear regression output for GDP growth model

Simple nonlinear regression yields the coefficient $\hat{\alpha} = 0.41$. This equals the capital share in income or investment share of the GDP. The coefficient $\hat{\gamma} = 0.25$ relates the production output to the education level of the population or human capital. Both coefficients are slightly different from those in the original study. This effect may be addressed to the different underlying fitting methods where Mankiw et al. (1990) applied a restricted regression. The corresponding standard errors are smaller compared to those in the original study, which indicates a gain in accuracy for the derived results with nonlinear regression analysis. All coefficients are classified as highly significant due to their low p-values (<0.05). The subsequent section applies the regression model to a two-component mixture model.

6.5 Two-Component Mixtures of GNMs

Due to the complex nonlinear functional relationship in Equation (6.3), a multiple cluster assignment may be possible and plausible for specific countries. Considering the sample size and in order to limit the complexity, the mixture modeling will build on two components. Comparisons will be made to the simple nonlinear regression model. Presuming two underlying distribution components yields the mixture density

$$f^M(y_i; \mu_i(\Theta, \alpha, \gamma), \boldsymbol{\phi}, \boldsymbol{\pi}) = \pi_1 f(y_i; \mu_i(\Theta_1, \alpha_1, \gamma_1), \phi_1) + (1 - \pi_1) f(y_i; \mu_i(\Theta_2, \alpha_2, \gamma_2), \phi_2) \tag{6.4}$$

where $f(\cdot)$ represents the pdf and π_1 the probability for the first mixture component with $i = 1, \ldots, n$. The parameter vector $\boldsymbol{\Psi}$ summarizes all unknown parameters as

$$\boldsymbol{\Psi} = (\pi_1, \Theta_1, \alpha_1, \gamma_1, \Theta_2, \alpha_2, \gamma_2, \phi_1, \phi_2)^\top,$$

where the dispersion parameters correspond to the variances through $\phi_1 = \sigma_1^2$ and $\phi_2 = \sigma_2^2$. The application of a two-component mixture of GNMs can be provided by the command `flexmix()` with the package **flexmixNL**. The corresponding command and output in R is given in Listing 6.3.

```
> formula <- log25 ~ eco.fct(alpha,gamma,theta,POPGRO,IONY,SCHOOL)
> flexfit <- flexmix(log25 ~ ..., k = 2,  data = data,
+                    model = list(FLXMRnlm(formula = formula,
+                                          family="gaussian",
+                                          start = ...)))

> flexfit
...
Cluster sizes:
 1  2
67  8

convergence after 117 iterations
```

Listing 6.3: Fitting command and output for two-component normal mixture with Model (6.3)

The two-component mixture model succeeds to uniquely determine two components. These results were underpinned by a repeated refitting of the data with randomly chosen starting values (50 times) as outlined in Section 6.3. The R output (line 11) outlines the separation of a smaller component (8 countries) from a central component containing the majority of the countries (67 countries). The rootogram in Figure 6.3 and the component ratios in Listing 6.4 indicate a moderate separation of the two components.

```
> summary(flexfit)
...
       prior size post>0 ratio
Comp.1 0.774   67     74 0.905
Comp.2 0.226    8     75 0.107

'log Lik.' -11.41643 (df=9)
AIC: 40.83285   BIC: 61.69025
```

Listing 6.4: `summary()` output for for two-component normal mixture and Model (6.3)

Figure 6.2 displays the multimodal structure of the kernel density of the response. A graphical comparison to the density shapes of the two fitted components can be made in the following way: Figure 6.4 visualizes the normal pdfs of the two fitted components given by two subsets comprising 8 and 67 countries in direct comparison to the kernel density of the data. For comparative purposes, the component specific pdfs are scaled by their prior probabilities $\hat{\pi}_1$ and $\hat{\pi}_2$. Comparing the original kernel density to the fitted two-component mixture distribution shows the following effect: The second component comprises economies driving the multimodal structure with modes at 0.2 and 1.4. Taking into consideration the visualizations in Figures 6.5, 6.6 and 6.7, the second component shows a concentration in lower GDP returns, building on the first peak near 0.2, but comprising also the two highest values given by Singapore and Hong Kong. The latter can be considered as drivers of the second peak around 1.4 as displayed within the density function in Figure 6.4. Withdrawing these countries from the original data lowers the multimodal structure and the heterogeneity which is clearly evident within the data. Therefore, the first and central component represents an *adjusted* set of the original data as the multimodal structure diminishes.

The fitted coefficients and their standard errors are outlined in Table 6.3 summarizing the results of the simple nonlinear regression and the two-component mixture model. The present results outline the increasing importance of investment share within the first component due to the increase in $\hat{\alpha}$ compared to the simple nonlinear regression model. The population growth shows the opposite effect as $\hat{\gamma}$ decreases even further compared to the simple nonlinear regression. The second and smaller component exhibits the opposite effect. Due to its size, the second component is very sensitive to patterns of single countries. Therefore, the coefficient $\hat{\alpha}$ relating the output to its average investment share decreases more than a half while the coefficient for population growth $\hat{\gamma}$ nearly doubles. The decrease in investment share can be attributed to two effects: regarding the investment share, Morocco is lagging behind the remaining countries within the second component. As Figure 6.6 displays, the members of the second component range at the upper end concerning the population growth. Due to the functional relationship in Equation (6.3), the capital share is allocated to the country's population where the second component evidently ranges at the head of the data. The level of education states a level for improvement in direct comparison to the other countries. Therefore, the importance of the related coefficient $\hat{\gamma}$ increases when regarding the country's economic growth within the second component.

Within the two-component mixture model, the coefficient Θ, representing the technological advancement, increases slightly for the first component while it decreases for the second component compared to the results obtained by nonlinear regression. Country members of the first component exhibit a decrease in the production output related to the education level ($\hat{\gamma}$) in addition to an increase in investment shares ($\hat{\alpha}$). Due to these effects, it can be assumed that the theoretical convergence rate to steady state changes for the two-component mixture model in direct comparison to the nonlinear regression model applied to Equation (6.3). The functional relationship in Equation (6.3) enables the computation of the convergence rate for each country's economy for the two distinct components. Table 6.4 displays the convergence rates to steady state for the first component, whereas Table 6.5 shows the analogous values for the smaller second component. The convergence rates resulting from the nonlinear regression model are denoted as λ^{nls} while λ^{mix} refers to the respective component of the mixture model. The results give a clear impression on the change in convergence rates. The significant decrease in capital share through $\hat{\alpha}$ for the second component indicates a decrease in convergence to the steady state. The first component shows the opposite effect. Due to the disappearance of the countries allocated to the second component, the coefficient relating the GDP to investments gains in value while the coefficient corresponding to the average population growth decreases. As a result the country specific convergence rates accelerate.

A direct comparison between the nonlinear regression model and the two-component mixture model is enabled through the model selection criteria AIC, BIC and ICL, as displayed in Table 6.3. The AIC values show a minor difference for the models. As the higher number of parameters (9 instead of 4) increases the model complexity for the two-component mixture model, an even higher log-likelihood yields a similar AIC due to the penalization term (see Section 2.7.1). The BIC outlines a larger difference between

		2-Component Gaussian			Nonlinear Regression	
		mean	s.e.		mean	s.e.
Comp. 1	$\hat{\pi}_1$	0.77	0.16			
	$\hat{\Theta}_1$	16.59	0.79	$\hat{\Theta}$	16.12	0.77
	$\hat{\alpha}_1$	0.49	0.06	$\hat{\alpha}$	0.41	0.06
	$\hat{\gamma}_1$	0.19	0.05	$\hat{\gamma}$	0.25	0.05
	$\hat{\sigma}_1$	0.21	0.04	$\hat{\sigma}$	0.31	-
Comp. 2	$\hat{\Theta}_2$	15.34	2.87			
	$\hat{\alpha}_2$	0.15	0.27			
	$\hat{\gamma}_2$	0.47	0.24			
	$\hat{\sigma}_2$	0.47	0.12			
	AIC	40.8		AIC	40.7	
	BIC	61.7		BIC	50.0	
	ICL	88.2				

Table 6.3: Regression coefficients for economic growth model (6.3)

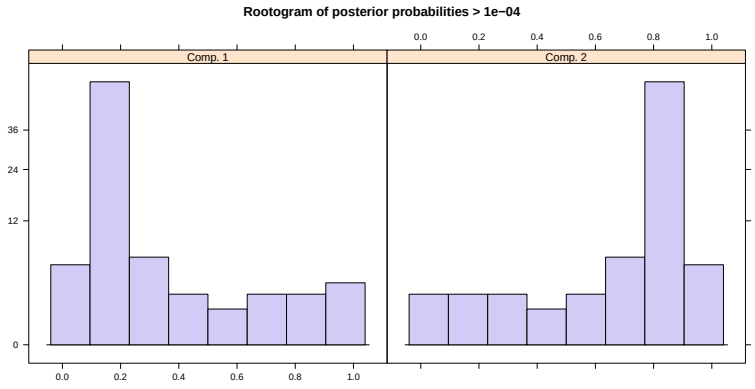

Figure 6.3: Rootogram for two-component normal mixture model

the two models due to the consideration of the sample size within the penalization. Drawing on the previous analysis, the two fitted components exhibit different drivers in their economies' growth. The direct comparison to the fitted nonlinear regression model shows a different convergence behavior by means of an acceleration for the first component and a deceleration for the economies of those countries classified to the second component. Based on these considerations, it can be assumed that the mixture model is more suitable in order to detect similar patterns in the economic growth development through different economies.

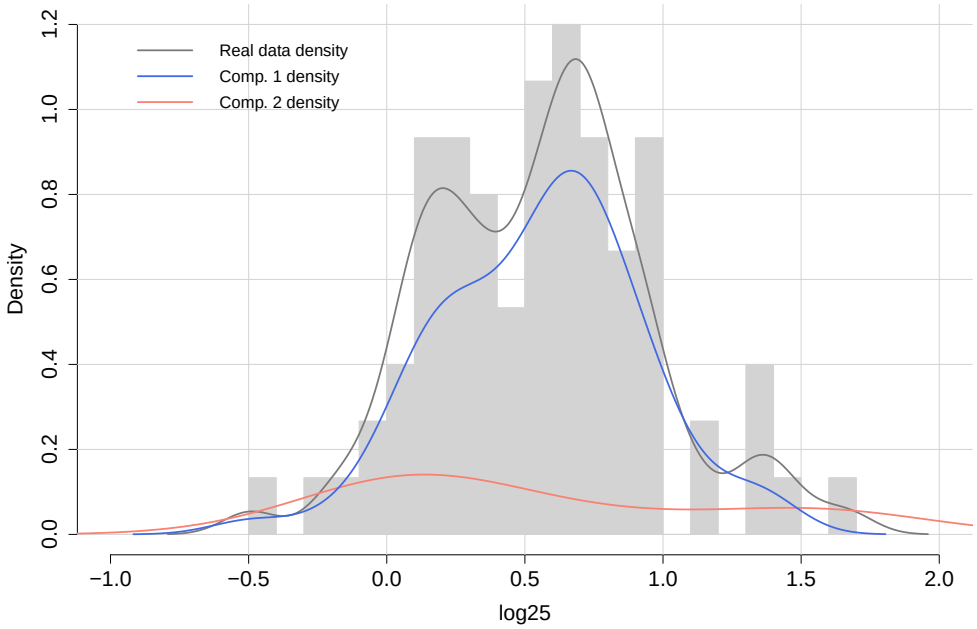

Figure 6.4: Fitted two-component mixture distribution for the GDP growth

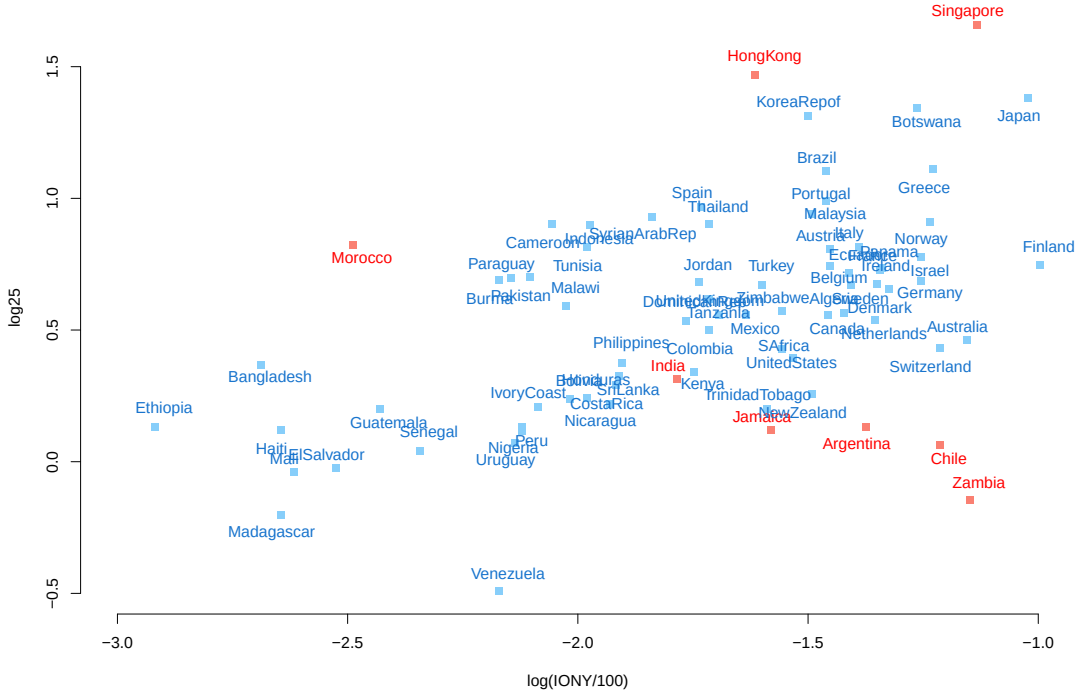

Figure 6.5: Cross-correlation GDP growth with investment share

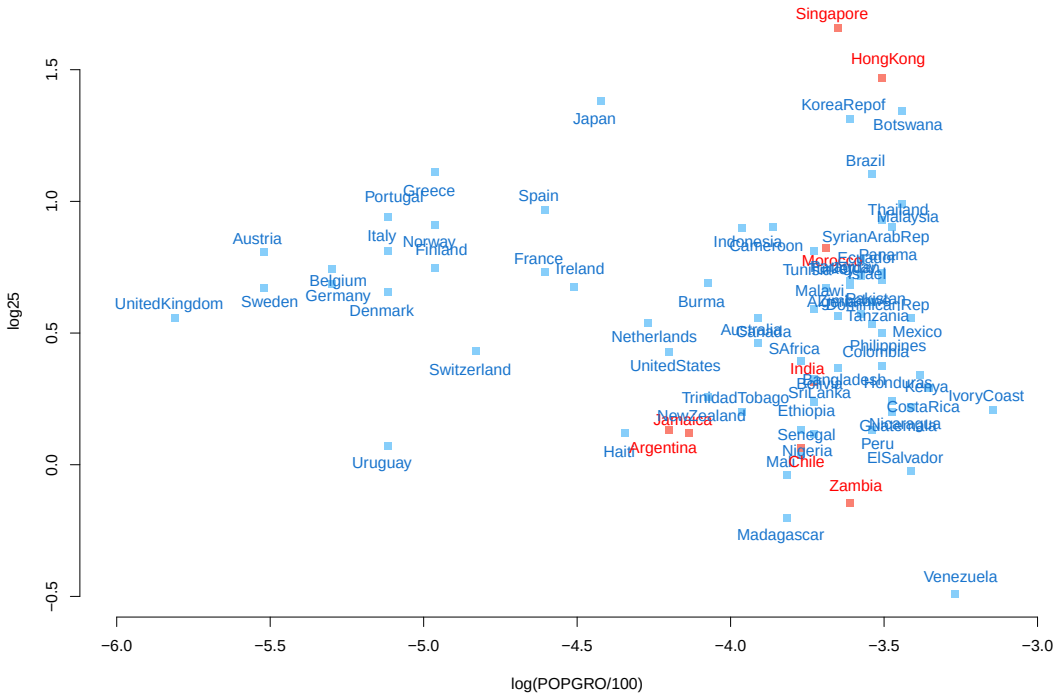

Figure 6.6: Cross-correlation GDP growth with country population growth

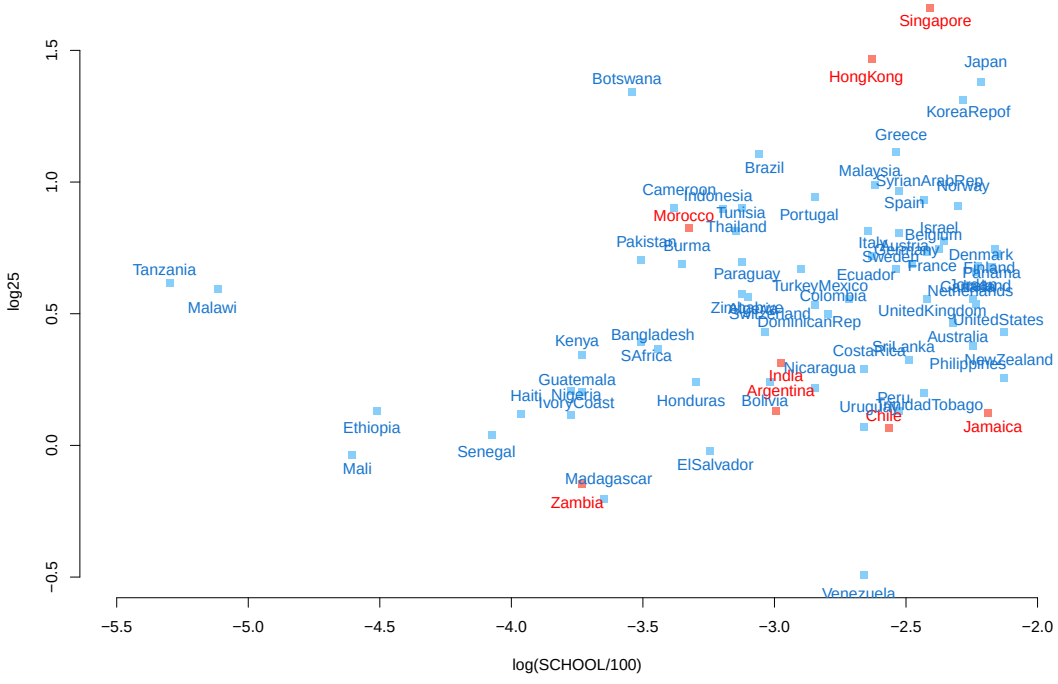

Figure 6.7: Cross-correlation GDP growth with education level

	Country	λ^{nls}	λ^{mix}
1	Algeria	0.225	0.237
2	Botswana	0.243	0.256
3	Cameroon	0.210	0.222
4	Ethiopia	0.216	0.228
5	Ivory Coast	0.275	0.290
6	Kenya	0.249	0.262
7	Madagascar	0.213	0.225
8	Malawi	0.219	0.231
9	Mali	0.213	0.225
10	Nigeria	0.219	0.231
11	Senegal	0.216	0.228
12	South Africa	0.216	0.228
13	Tanzania	0.234	0.247
14	Tunisia	0.219	0.231
15	Zimbabwe	0.231	0.244
16	Bangladesh	0.225	0.237
17	Burma	0.198	0.209
18	Israel	0.231	0.244
19	Japan	0.183	0.194
20	Jordan	0.228	0.240
21	Rep. of Korea	0.228	0.240
22	Malaysia	0.243	0.256
23	Pakistan	0.237	0.250
24	Philippines	0.237	0.250
25	Sri Lanka	0.219	0.231
26	Syrian Arab. Rep.	0.237	0.250
27	Thailand	0.240	0.253
28	Austria	0.160	0.169
29	Belgium	0.163	0.172
30	Denmark	0.166	0.175
31	Finland	0.169	0.178
32	France	0.178	0.187
33	Germany	0.163	0.172
34	Greece	0.169	0.178
35	Ireland	0.181	0.191
36	Italy	0.166	0.175
37	Netherlands	0.189	0.200
38	Norway	0.169	0.178
39	Portugal	0.166	0.175
40	Spain	0.178	0.187
41	Sweden	0.160	0.169
42	Switzerland	0.172	0.181
43	Turkey	0.222	0.234
44	United Kingdom	0.157	0.166
45	Canada	0.207	0.219
46	Costa Rica	0.252	0.265
47	Dominican Rep.	0.234	0.247
48	El Salvador	0.246	0.259
49	Guatemala	0.240	0.253
50	Haiti	0.186	0.197
51	Honduras	0.240	0.253
52	Mexico	0.246	0.259
53	Nicaragua	0.246	0.259
54	Panama	0.237	0.250
55	Trinidad Tobago	0.204	0.215
56	United States	0.192	0.203
57	Bolivia	0.219	0.231
58	Brazil	0.234	0.247
59	Colombia	0.237	0.250
60	Ecuador	0.231	0.244
61	Paraguay	0.228	0.240
62	Peru	0.234	0.247
63	Uruguay	0.166	0.175
64	Venezuela	0.260	0.275
65	Australia	0.207	0.219
66	Indonesia	0.204	0.215
67	New Zealand	0.198	0.209

Table 6.4: Convergence rates towards steady state (component 1)

	Country	λ^{nls}	λ^{mix}
1	Morocco	0.222	0.197
2	Zambia	0.228	0.202
3	Hong Kong	0.237	0.210
4	India	0.219	0.194
5	Singapore	0.225	0.199
6	Jamaica	0.195	0.173
7	Argentina	0.192	0.171
8	Chile	0.216	0.192

Table 6.5: Convergence rates towards steady state (component 2)

6.6 Conclusions

The present application gives a possible approach to the dealing with heterogeneity in economic growth models by means of mixtures of GNMs. The well-known Solow model allows to derive a nonlinear regression model for the modeling of the production output (GDP) in functional dependence to economic variables like labor rates, the education level of the population or the investment shares as main drivers. The nonlinear functional structure arising therefrom states the underlying mean function, whereas the separate mixture components succeed to deal with occurring heterogeneity in supranational data. As approaches for mixture models within economic growth applications have been applied to mixtures of linear regression models, the new model class within **flexmixNL** allows for the fitting of the underlying nonlinear regression model. Therefore, the original nonlinear functional structure can be maintained. This has the advantage that the original regression coefficients which comprise a specific econometric meaning within the Solow model can be directly fitted. Therefore, the new model class represents a direct approach for dealing with heterogeneity within economic growth models. The main idea of modeling different subgroups within economic data (multiple regime) can be realized by the use of mixture models. Applying the new model class to the original data revealed two different components which differ in the main drivers of their economic growth and subsequently in their convergence rates to steady state. An open question remains the choice of the number of components which is strongly problem specific. A direct comparison is enabled through model selection criteria. In the present analysis, the fitting of two components showed an improvement in information regarding the countries' economies whereas the AIC values differed slightly between the two models while the BIC value increases for the mixture model due to the increasing number of parameters.

The derived results indicate the sensible use of mixture distributions in order to asses different economic subgroups. For the original data sample from Mankiw et al. (1990) the application of two Gaussian components succeeds to manage the present heterogeneity in a better way than the simple nonlinear regression, as it allows to distinguish between country-specific economic growth patterns. As a smaller second component separated countries with a lower GDP investment share average and an increase in GDP related

to the education level of the population, the convergence rates towards the steady state improved for the first component. The specific approach buttress the modeling of subgroups within the theory of steady state according to Solow (1956). Due to the sample size, a further increase in the number of components was not prosecuted but may be possible for similar data.

Final Remarks

Major results of the present work comprise the introduction of the new model class of mixtures of GNMs. As mixture models have already proven as an appropriate solution to address heterogeneity in data, the extension to nonlinear models increases their flexibility even further. The present work outlined that the new model class given by mixtures of GNMs handles specific problems in a direct manner as it does not require adjustments like transformations regarding the mean functions. It enables furthermore the fitting of arbitrary nonlinear regression functions where the component pdfs stem from the exponential family. The original functional dependency structure can be applied where specific parameters are of considerable importance. This may be motivated by their meaning or due to further interpretation of the problem. Numerical problems represent an obstacle within the application of nonlinear regression which requires often problem-specific knowledge. Possible solutions are therefore in general problem-specific in an individual manner. The construction of mixtures of nonlinear models highlighted the increasing complexity which comes along the embedding of nonlinear regression within the framework of mixture models. In order to provide an opportunity to derive a standardized fitting procedure, the overall problem was carried over to the EM algorithm. The fitting of mixtures of GNMs was successfully implemented building on the package **flexmix** in R. The implementation procedure was outlined in detail with particular emphasis on GNM specific modifications. Thereby the new fitting procedure takes advantage of an efficient and well-established fitting methodology. Its value is reflected in the currently broad number on available models in the repertoire of **flexmix** which is now extended by the possibility of modeling nonlinear functional structures.

Dealing with heterogeneous data, where prior knowledge motivates the use if nonlinear functional structures, is enabled through the new model class wrapped up in the package **flexmixNL**. The new functionality is easy to apply and consistent to already existing methods for nonlinear regression problems in R. The extension to the Gamma distribution represents a key feature in the application of mixtures of GNMs. Embedding the new model class in R enables to run applications with evident nonlinear dependency structures. It allows currently for the fitting of normal and Gamma distributed variables which enables the distinction between light- and heavy-tailed distributional patterns.

The performance of the new fitting method was successfully challenged by an extensive simulation study which underpinned the reliability of the derived values and revealed also numerical limits. Within this context, divergent configurations have also been subject to a detailed analysis as well as the requirement of appropriate starting values in order to achieve convergence and accurate results.

The mixtures of GNMs were applied to real data where the modeling of nonlinear functional structures was motivated by prior knowledge. Particular emphasis was given to the derived results and their interpretation within the context of the specific application. Applying the methods to real data enabled the handling of heterogeneous subgroups or components within different data structures, allowing to derive particular statements on component specific characteristics. The applications produced positive and reliable outcomes.

The present work introduced mixtures of GNMs as an advanced method for modeling heterogeneity for different subgroups. Mixtures of GNMs have proven as a reliable method to comprise variability or diversity in data with nonlinear functional patterns.

Appendices

A Definitions

Definition A.1 *Generalized Inverse Matrix (Penrose (1955))*
For any matrix $A \in \mathbb{R}^{n \times m}$ an unique matrix $A^+ \in \mathbb{R}^{m \times n}$ exists satisfying the four conditions:

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^\top = AA^+$
4. $(A^+A)^\top = A^+A$

For a regular matrix A the general inverse matrix satisfies the previous conditions and the relationship $A^+ = A^{-1}$ holds due to the uniqueness of the generalized inverse matrix A^+.

Definition A.2 *Order of Convergence (Atkinson (1989, p. 56))*
A sequence of iterates x_i with $i \geq 0$ is said to converge to x^* with order $p \geq 1$ if

$$|x^* - x_{i+1}| \leq c|x^* - x_i|^p \quad \forall\, i \geq 0, c > 0.$$

If $p = 1$ the sequence x_i is said to converge linearly *to x^*. The constant c is referred to as* rate of convergence *and satisfies the condition $0 < c < 1$.*

B Equilibrium or Steady State in the Solow Model

As outlined in Chapter 6, the Solow model represents an economic growth model by means of the Gross Domestic Product (GDP). The latter will be denoted as Y_t. The driving factors of the economic growth are given by the capital K_t, the level of technology A_t, the labor L_t and the stock of human capital H_t. Let the following be the analogous quantities per effective unit of labor corresponding to Section 6.1,

$$y_t := \frac{Y_t}{A_t L_t}, \quad k_t := \frac{K_t}{A_t L_t}, \quad h_t := \frac{H_t}{A_t L_t}.$$

Substituting the equations in (6.1) yields the corresponding production output per effective unit of labor, respectively

$$y_t = {k_t}^{\alpha} {h_t}^{\gamma}. \tag{B.1}$$

According to Mankiw et al. (1990, p. 410), the Solow model assumes that a constant fraction s of the output y_t is invested. Under the assumption of investments taken in the amount of s_K for physical capital and s_H for human capital, with respect to $s = s_K + s_H$, the evolution of the economy follows therefore

$$\begin{aligned} \frac{\partial k_t}{\partial t} &= s_K y_t - (n+g+\delta)k = s_K k^{\alpha} h^{\gamma} - (n+g+\delta)k \\ \frac{\partial h_t}{\partial t} &= s_H y_t - (n+g+\delta)h = s_H k^{\alpha} h^{\gamma} - (n+g+\delta)h. \end{aligned}$$

taking into account the growth rate of labour units $n+g$ and the depreciation rate of the capital δ. The evolution of k_t and h_t is assumed to converge to a *steady state* where the Solow model reaches its *equilibrium*. According to Mankiw et al. (1990, p. 416), the steady state equilibrium satisfies $\frac{\partial k}{\partial t} = 0 = \frac{\partial h}{\partial t}$, yielding the steady state levels

$$k^* = \left(\frac{s_K^{1-\gamma} s_H^{\gamma}}{n+g+\delta} \right)^{1/(1-\alpha-\gamma)} \quad \text{and} \quad h^* = \left(\frac{s_H^{1-\alpha} s_K^{\alpha}}{n+g+\delta} \right)^{1/(1-\alpha-\gamma)}. \tag{B.2}$$

The *steady state level of income per worker* follows from (B.1) through

$$y^* \overset{(B.2)}{=} {k^*}^{\alpha} {h^*}^{\gamma} = \left(\frac{s_K^{1-\gamma} s_H^{\gamma}}{n+g+\delta} \right)^{\alpha/(1-\alpha-\gamma)} \left(\frac{s_H^{1-\alpha} s_K^{\alpha}}{n+g+\delta} \right)^{\gamma/(1-\alpha-\gamma)}.$$

The derivation of an explicit function for the production output per capita results by substituting the variables representing the evolution of the economy in the original process yielding

$$\begin{aligned} \frac{Y_t}{L_t} &= A_t k_t^{\alpha} h_t^{\gamma} \\ &\overset{(6.2)+(B.2)}{=} A_0 \exp(gt) \left(\frac{s_K^{1-\gamma} s_H^{\gamma}}{n+g+\delta} \right)^{\alpha/(1-\alpha-\gamma)} \left(\frac{s_H^{1-\alpha} s_K^{\alpha}}{n+g+\delta} \right)^{\gamma/(1-\alpha-\gamma)}. \end{aligned}$$

Taking the logarithm and rearranging the formula yields the expression

$$\begin{aligned}\log\left(\frac{Y_t}{L_t}\right) &= \log A_0 + gt + \frac{\alpha}{1-\alpha-\gamma}\left((1-\gamma)\log s_K + \gamma \log s_H + \log(n+g+\delta)\right) + \\ &\quad \ldots \frac{\gamma}{1-\alpha-\gamma}\left((1-\alpha)\log s_H + \alpha \log s_K + \log(n+g+\delta)\right) \\ &= \log A_0 + gt + \frac{\alpha+\gamma}{1-\alpha-\gamma}\log(n+g+\delta) + \frac{\alpha}{1-\alpha-\gamma}\log s_K + \\ &\quad \ldots \frac{\gamma}{1-\alpha-\gamma}\log s_H.\end{aligned}$$

The Solow model allows for a statement on the speed of convergence to the steady state. For the steady state level y^* in the *Solow model*, the speed of convergence is denoted as λ and the growth dynamics follows

$$\frac{\partial y_t}{\partial t} = \lambda(\log y^* - \log y_t) \tag{B.3}$$

with $\lambda = (n+g+\delta)(1-\alpha-\gamma)$ according to Mankiw et al. (1990, p. 422). The solution of (B.3) is given by

$$\begin{aligned}\log y_t &= \log y_0 \exp(-\lambda t) + \log y^*(1-\exp(-\lambda t)) \\ \log y_t - \log y_0 &= (1-\exp(-\lambda t))(\log y^* - \log y_0).\end{aligned} \tag{B.4}$$

Reformulating the expression on the left hand side in (B.4) yields

$$\begin{aligned}\log y_t - \log y_0 &= \log\left(\frac{Y_t}{L_t}\right) - \log\left(\frac{Y_0}{L_0}\right) - \log A_t + \log A_0 \\ &= \log\left(\frac{Y_t}{L_t}\right) - \log\left(\frac{Y_0}{L_0}\right) - \log A_0 - gt + \log A_0 \\ &= \log\left(\frac{Y_t}{L_t}\right) - \log\left(\frac{Y_0}{L_0}\right) - gt\end{aligned} \tag{B.5}$$

and substituting (B.5) in (B.4) yields after rearrangement the economic growth function

$$\begin{aligned}\log\left(\frac{Y_t}{L_t}\right) - \log\left(\frac{Y_0}{L_0}\right) &= gt + (1-\exp(-\lambda t))(\log y^* - \log y_0) \\ &= gt + (1-\exp(-\lambda t)) \cdot \\ &\quad \left(-\log A_0 + \frac{\alpha+\gamma}{1-\alpha-\gamma}\log(n+g+\delta) + \frac{\alpha}{1-\alpha-\gamma}\log s_K\right. \\ &\quad \left. \ldots + \frac{\gamma}{1-\alpha-\gamma}\log s_H - \log\left(\frac{Y_0}{L_0}\right)\right).\end{aligned}$$

For further details on the Solow model reference is made to Mankiw et al. (1990) and Solow (1956).

C Packages in R

The following packages were used within the present work:

- **Deriv** (Version 3.8.5)
- **doParallel** (Version 1.0.11)
- **flexmix** (Version 2.3.14)
- **foreach** (Version 1.4.4)
- **ggplot2** (Version 2.2.1)
- **gnm** (Version 1.0.8)
- **MASS** (Version 7.3.49)
- **minpack.lm** (Version 1.2.1)
- **numDeriv** (Version 2016.8.1)
- **stringr** (Version 1.3.0)

Bibliography

AGCS (2018). Loading profiles. https://www.agcs.at/de/clearing/technisches/lastprofile/lastprofile_ab_01.04.2009, [Accessed: 05-11-2018]. AGCS Gas Clearing and Settlement AG.

Aitkin, M., Francis, B., and Hinde, J. (2005). *Statistical Modelling in GLIM 4*. Oxford University Press, second edition.

Alfo, M., Trovato, G., and Waldmann, R. J. (2008). Testing for country heterogeneity in growth models using a finite mixture approach. *Journal of Applied Econometrics*, 23:487–514.

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132.

Almbauer, R. (2008). Lastprofile nicht-leistungsgemessener Kunden. https://www.agcs.at/agcs/clearing/lastprofile/lp_studie2008.pdf, [Accessed: 05-11-2018]. University of Technology Graz and Association of Gas- and District Heating Supply Companies (FGW).

Atkinson, K. (1989). *An Introduction to Numerical Analysis*. Wiley.

Basford, K. E., Greenway, D. R., Mclachlan, G. J., and Peel, D. (1997). Standard errors of fitted component means of normal mixtures. *Computational Statistics*, 12:1–17.

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons.

BDEW (2018). Kooperationsvereinbarung Gas. https://www.bdew.de/service/standardvertraege/kooperationsvereinbarung-gas/, [Accessed: 01-11-2018]. Bundesverband der Energie und Wasserwirtschaft e. V.

BDEW, VKU, and GEODE (1990). BDEW/VKU/GEODE-Leitfaden Abwicklung von Standardlastprofilen Gas. Leitfaden 3541, BDEW, VKU and GEODE.

Ben-Israel, A. (1965). A modified Newton-Raphson method for the solution of systems of equations. *Israel Journal of Mathematics*, 3(2):94–98.

Ben-Israel, A. (1966). A Newton-Raphson method for the solution of systems of equations. *Journal of Mathematical Analysis and Applications*, 15(2):243–252.

Benaglia, T., Chauveau, D., Hunter, D. R., and Young, D. (2009). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29.

Böhning, D. (1995). A review of reliable maximum likelihood algorithms for semiparametric mixture models. *Journal of Statistical Planning and Inference*, 47:5–28.

Böhning, D. (1999). *Computer-Assisted Analysis of Mixtures and Applications*.

Böhning, D. (2003). The em algorithm with gradient function update for discrete mixtures with known (fixed) number of components. *Statistics and Computing*, 13:257–265.

Böhning, D., Bruce, P. S., and Lindsay (1992). Computer-assisted analysis of mixtures (c.a.man): Statistical algorithms. *Biometrics*, 48:283–303.

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. G. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, 46(2):373–388.

Böhning, D., Dietz, E., and Schlattmann, P. (1998). Recent developments in computer-assisted analysis of mixtures. *Biometrics*, 54:525–36.

Browne, R. P., ElSherbiny, A., and McNicholas, P. D. (2018). *mixture: Mixture Models for Clustering and Classification*. R package version 1.5.

Browne, R. P. and McNicholas, P. D. (2014). Estimating common principal components in high dimensions. *Advances in Data Analysis and Classification*, 8(2):217–226.

Casella, G. and Berger, R. L. (2001). *Statistical Inference*. Duxbury Press.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

Chambers, J. M. (2008). *Software for Data Analysis Programming with R*. Statistics and Computing. Springer.

Charnes, A., Frome, E. L., and Yu, P. L. (1976). The equivalence of generalized least squares and maximum likelihood estimates in the exponential family. *Journal of the American Statistical Association*, 71(353):169–171.

CISMO (2018). Clearing Integrated Services and Market Operations GmbH. `https://www.energymonitor.at/en/open-data/timeseries-download`. [Accessed: 05-11-2018].

Clausen, A. and Sokol, S. (2018). *Deriv: R-based Symbolic Differentiation*. Deriv package version 3.8.

CRAN (2018). Table of available packages in CRAN.

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

Durlauf, S. N. and Johnson, P. A. (1995). Multiple regimes and cross-country growth behaviour. *Journal of Applied Econometrics*, 10(4):365–384.

Efron, B. (1979). Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 7(1):1–26.

ENTSOG (2018). European Network of Transmission System Operators for Gas. `https://transparency.entsog.eu/`. [Accessed: 05-11-2018].

Faria, S. and Soromenho, G. (2010). Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):210–225.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Friedl, H., Mirkov, R., and Steinkamp, A. (2012). Modelling and forecasting gas flow on exits of gas transmission networks. *International Statistical Review*, 80(1):24–39.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer.

Garay, A., Prates, M., and Lachos, V. (2013). *nlsmsn: Fitting nonlinear models with scale mixture of skew-normal distributions.* R package version 0.0-4.

Geweke, J. (2005). *Contemporary Bayesian Econometrics and Statistics*. John Wiley & Sons.

Gilbert, P. and Varadhan, R. (2016). *numDeriv: Accurate Numerical Derivatives*. numDeriv package version 2016.8-1.

Green, P. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society*, 46(2):149–192.

Greene, W. H. (2012). *Econometric Analysis*. Prentice Hall, seventh edition.

Grün, B. (2006). *Identification and Estimation of Finite Mixture Models*. PhD thesis, University of Technology Vienna.

Grün, B. and Leisch, F. (2004). Bootstrapping finite mixture models. COMPSTAT'2004 SYMPOSIUM.

Grün, B. and Leisch, F. (2006). Fitting finite mixtures of linear regression models with varying and fixed effects in R. In *Proceedings in Computational Statistics*, pages 853–860.

Grün, B. and Leisch, F. (2007). Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics & Data Analysis*, 51(11):5247–5252.

Grün, B. and Leisch, F. (2008a). Finite mixtures of generalized linear regression models. *Recent Advances in Linear Models and Related Areas*.

Grün, B. and Leisch, F. (2008b). FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4):1–35.

Grün, B. and Leisch, F. (2008c). Identifiability of finite mixtures of multinomial logit models with varying and fixed effects. *Journal of Classification*, 25:225–247.

Grün, B. and Leisch, F. (2019). CRAN task view: Cluster analysis and finite mixture models. `https://cran.r-project.org/view=Cluster`, [Accessed: 02-02-2019].

Hellwig, M. (2003). *Entwicklung und Anwendung parametrisierter Standard-Lastprofile*. PhD thesis, Technischen Universität München.

Hennig, C. (2000). Identifiability of models for clusterwise linear regression. *Journal of Classification*, 17:273–296.

James, M. (1978). The generalised inverse. *The Mathematical Gazette*, 62(420):109–114.

Kim, D. and Lindsay, B. (2015). Empirical Identifiability in Finite Mixture Models. *Annals of the Institute of Statistical Mathematics*, 67(4):745–772.

Koch, T., Hiller, B., Pfetsch, M., and Schewe, L. (2015). *Evaluating Gas Network Capacities*. Society for Industrial and Applied Mathematics.

Leisch, F. (2004a). Exploring the structure of mixture model components. *Physica-Verlag/Springer*. COMPSTAT 2004 Symposium.

Leisch, F. (2004b). FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8):1–18.

Lindsay, B. (1983a). The Geometry of Mixture Likelihoods: A General Theory. *The Annals of Statistics*, 11:86–94.

Lindsay, B. (1983b). The Geometry of Mixture Likelihoods, part II: The exponential family. *The Annals of Statistics*, 11:783–792.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society*, 44(2):226–233.

Macdonald, P. and with contributions from Juan Du (2018). *mixdist: Finite Mixture Distribution Models*. R package version 0.5-5.

Mankiw, N. G., Romer, D., and Weil, D. N. (1990). A contribution to the empirics of economic growth. Working Paper 3541, National Bureau of Economic Research.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, second edition.

McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions*. Wiley series in probability and statistics. Applied probability and statistics section. John Wiley & Sons, second edition.

McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*, volume 2 of *Wiley series in probability and statistics. Applied probability and statistics section*. John Wiley & Sons.

Meilijson, I. (1989). A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society. Series B*, 51(1):127–138.

Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135(3):370–384.

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London A*, 185:71–110.

Penrose, R. (1955). A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*, pages 406–413.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ritz, C. and Streibig, J. C. (2008). *Nonlinear Regression with R*. Springer.

Rothenberg, T. J. (1971). Identification in parametric models. *Econometrica*, 39(3):577–91.

Sanchez, L. B., Lachos, V. H., and Moreno, E. J. L. (2018). *CensMixReg: Censored Linear Mixture Regression Models*. R package version 3.1.

Schlattmann, P. (2009). *Medical Applications of Finite Mixture Models*. Springer-Verlag Berlin Heidelberg, first edition.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):205–233.

Seber, G. A. F. and Wild, C. J. (2003). *Nonlinear Regression*. John Wiley & Sons.

Smyth, G. K. (2003). *Pearson's goodness of fit statistic as a score test statistic*, volume 40 of *Lecture Notes–Monograph Series*, pages 115–126. Institute of Mathematical Statistics.

Solow, R. (1956). A contribution to the theory of economic growth. *The Quarterly Journal of Economics*, 70(1):65–94.

Spectrum, I. (2018). Institute of Electrical and Electronics Engineers, incorporated, IEEE Spectrum. `https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages`. [Accessed: 19-08-2018].

Summers, R. and Heston, A. (1988). A new set of international comparisons of real product and price levels estimates for 130 countries, 1950–1985. *Review of Income and Wealth*, 34(1):1–25.

Swan, T. W. (1956). Economic growth and capital accumulation. *The Economic Record*, 32(2):334–361.

Teicher, H. (1961). Identifiability of mixtures. *The Annals of Mathematical Statistics*, 32(1):244–248.

Teicher, H. (1963). Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 34:1265–1269.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley.

Turner, H. and Firth, D. (2018). *Generalized nonlinear models in R: An overview of the gnm package*. R package version 1.1-0.

Turner, R. (2018). *mixreg: Functions to Fit Mixtures of Regressions*. R package version 0.0-6.

van Leeuwen, M. (2014). Estimating standard errors of parameters obtained by the EM-algorithm.

Veaux, R. D. D. (1989). Mixtures of linear regressions. *Computational Statistics and Data Analysis*, 8(3):227–245.

Venables, W. and Ripley, B. (2000). *S Programming*. Springer-Verlag New York.

Wang, P., Cockerburn, I. M., and Puterman, M. L. (1998). Analysis of patent data: A mixed-poisson-regression-model approach. *Journal of Business and Economic Statistics*, 16(1):27–41.

Wei, B.-C. (1998). *Exponential Family Nonlinear Models*. Springer-Verlag Singapore Pte. Ltd.

Yakowitz, S. J. and Spragins, J. S. (1968). On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39:209–214.

ZAMG (2018). Zentralanstalt für Meteorologie und Geodynamik. `https://www.zamg.ac.at/cms/de/klima/klimauebersichten/jahrbuch`. [Accessed: 05-11-2018].