Chapter 6
Bayesian Methods, Regularization and Expectation-Maximization

The previous chapter has been focusing on MLE of regression parameters within GLMs. Alternatively, we could address the parameter estimation problem within a Bayesian setting. The purpose of this chapter is to discuss the Bayesian estimation approach. This leads us to the notion of regularization within GLMs. Bayesian methods are also used in the Expectation-Maximization (EM) algorithm for MLE in the case of incomplete data. For literature on Bayesian theory we recommend Gelman et al. [157], Congdon [79], Robert [319], Bühlmann–Gisler [58] and Gilks et al. [158]. A nice historical (non-mathematical) review of Bayesian methods is presented in McGrayne [266]. Regularization is discussed in the book of Hastie et al. [184], and a good reference for the EM algorithm is McLachlan–Krishnan [267].

6.1 Bayesian Parameter Estimation

The Bayesian estimator has been introduced in Definition 3.6. Assume that the observation $\boldsymbol{Y}$ has independent components Y_i that can be described by a GLM with link function g and regression parameter $\boldsymbol{\beta} \in \mathbb{R}^{q+1}$, i.e., the random variables Y_i have densities

$$Y_i \overset{\text{ind.}}{\sim} f(y; \boldsymbol{\beta}, \boldsymbol{x}_i, v_i/\varphi) = \exp\left\{ \frac{y (h \circ g^{-1})\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle - (\kappa \circ h \circ g^{-1})\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle}{\varphi / v_i} + a(y; v_i/\varphi) \right\},$$

with canonical link $h = (\kappa')^{-1}$. In a Bayesian approach one models the regression parameter $\boldsymbol{\beta}$ with a prior distribution[1] $\pi(\boldsymbol{\beta})$ on the parameter space $\mathbb{R}^{q+1}$, and the independence assumption between the components of $\boldsymbol{Y}$ needs to be understood

[1] Often, in Bayesian arguing, distribution and density is used in an interchangeable (and not fully precise) way, and it is left to the reader to give the right meaning to π.

M. V. Wüthrich, M. Merz, *Statistical Foundations of Actuarial Learning and its Applications*, Springer Actuarial, https://doi.org/10.1007/978-3-031-12409-9_6

conditionally, given the regression parameter $\boldsymbol{\beta}$. In other words, all observations Y_i share the same regression parameter $\boldsymbol{\beta}$, which itself is modeled by a prior distribution π.

The joint density of $\boldsymbol{Y}$ and $\boldsymbol{\beta}$ is given by

$$p(\boldsymbol{y}, \boldsymbol{\beta}) = \left(\prod_{i=1}^{n} f(y_i; \boldsymbol{\beta}, \boldsymbol{x}_i, v_i/\varphi) \right) \pi(\boldsymbol{\beta}) = \exp\left\{ \ell_{\boldsymbol{Y}=\boldsymbol{y}}(\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta}) \right\}. \tag{6.1}$$

For the given observation $\boldsymbol{Y}$, this allows us to calculate the posterior density of $\boldsymbol{\beta}$ using Bayes' rule

$$\pi(\boldsymbol{\beta}|\boldsymbol{Y}) \; = \; \frac{p(\boldsymbol{Y}, \boldsymbol{\beta})}{\int p(\boldsymbol{Y}, \widetilde{\boldsymbol{\beta}}) d\widetilde{\boldsymbol{\beta}}} \; \propto \; \left(\prod_{i=1}^{n} f(Y_i; \boldsymbol{\beta}, \boldsymbol{x}_i, v_i/\varphi) \right) \pi(\boldsymbol{\beta}), \tag{6.2}$$

where the proportionality sign $\propto$ indicates that we have dropped the terms that do not depend on $\boldsymbol{\beta}$. Thus, the functional form in $\boldsymbol{\beta}$ of the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ is fully determined by the joint density $p(\boldsymbol{Y}, \boldsymbol{\beta})$, and the remaining term is a normalization to obtain a proper probability distribution. In many situations, the knowledge of the functional form of the posterior density in $\boldsymbol{\beta}$ is sufficient to perform Bayesian parameter estimation, at least, numerically. We will give some references, below.

The Bayesian estimator for $\boldsymbol{\beta}$ is given by the posterior mean (supposed it exists)

$$\widehat{\boldsymbol{\beta}}^{\text{Bayes}} = \mathbb{E}_{\pi}\left[\boldsymbol{\beta} \middle| \boldsymbol{Y}\right] = \int \boldsymbol{\beta} \, \pi(\boldsymbol{\beta}|\boldsymbol{Y}) d\nu(\boldsymbol{\beta}).$$

If we want to calculate the expectation of a new random variable Y_{n+1} that is conditionally, given $\boldsymbol{\beta}$, independent of $\boldsymbol{Y}$ and follows the same GLM as $\boldsymbol{Y}$, we can directly calculate, using the tower property and conditional independence,[2]

$$\begin{aligned} \mathbb{E}_{\pi}\left[Y_{n+1} \middle| \boldsymbol{Y}\right] = \mathbb{E}_{\pi}\left[\mathbb{E}\left[Y_{n+1} \middle| \boldsymbol{\beta}, \boldsymbol{Y}\right] \middle| \boldsymbol{Y}\right] \; &= \; \mathbb{E}_{\pi}\left[\mathbb{E}\left[Y_{n+1} \middle| \boldsymbol{\beta}\right] \middle| \boldsymbol{Y}\right] \\ = \mathbb{E}_{\pi}\left[g^{-1}\langle \boldsymbol{\beta}, \boldsymbol{x}_{n+1} \rangle \middle| \boldsymbol{Y} \right] \; &= \; \int g^{-1}\langle \boldsymbol{\beta}, \boldsymbol{x}_{n+1} \rangle \, \pi(\boldsymbol{\beta}|\boldsymbol{Y}) d\nu(\boldsymbol{\beta}), \end{aligned}$$

supposed that this first moment exists and that $\boldsymbol{x}_{n+1}$ is the feature of Y_{n+1}. We see that it all boils down to have sufficiently explicit knowledge about the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ given in (6.2).

Remark 6.1 (Conditional MSEP) Based on the assumption that the posterior distribution $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ can be determined, we can analyze the GL. In a Bayesian setup one

[2] Note that we identify probabilities $\mathbb{P}_{\boldsymbol{\beta}}[\cdot] = \mathbb{P}[\cdot|\boldsymbol{\beta}]$ for given $\boldsymbol{\beta}$.

usually does not calculate the MSEP as described in Theorem 4.1, but one rather studies the conditional MSEP, conditioned exactly on the collected information $\boldsymbol{Y}$. That is,

$$\begin{aligned}
\mathbb{E}_\pi\left[\left(Y_{n+1}-\mathbb{E}_\pi\left[\left.Y_{n+1}\right|\boldsymbol{Y}\right]\right)^2\Big|\,\boldsymbol{Y}\right] &= \operatorname{Var}_\pi\left(\left.Y_{n+1}\right|\boldsymbol{Y}\right)\\
&= \operatorname{Var}_\pi\left(\mathbb{E}\left[\left.Y_{n+1}\right|\boldsymbol{\beta},\boldsymbol{Y}\right]\right|\boldsymbol{Y}) + \mathbb{E}_\pi\left[\operatorname{Var}\left(\left.Y_{n+1}\right|\boldsymbol{\beta},\boldsymbol{Y}\right)\right|\boldsymbol{Y}]\\
&= \operatorname{Var}_\pi\left(g^{-1}\langle\boldsymbol{\beta},\boldsymbol{x}_{n+1}\rangle\Big|\,\boldsymbol{Y}\right)+\frac{\varphi}{v_{n+1}}\mathbb{E}_\pi\left[(\kappa''\circ h\circ g^{-1})\langle\boldsymbol{\beta},\boldsymbol{x}_{n+1}\rangle\Big|\,\boldsymbol{Y}\right]\\
&= \operatorname{Var}_\pi\left(g^{-1}\langle\boldsymbol{\beta},\boldsymbol{x}_{n+1}\rangle\Big|\,\boldsymbol{Y}\right)+\frac{\varphi}{v_{n+1}}\mathbb{E}_\pi\left[V(g^{-1}\langle\boldsymbol{\beta},\boldsymbol{x}_{n+1}\rangle)\Big|\,\boldsymbol{Y}\right],
\end{aligned}$$

where we need to assume existence of second moments. Similar to Theorem 4.1, the first term is the estimation variance (in a Bayesian setting) and the second term is the average process variance (using the EDF variance function $\mu \mapsto V(\mu)$).

The remaining difficulty is the calculation of the posterior expectation of functions of $\boldsymbol{\beta}$, based on posterior density (6.2). In very well-designed experiments the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ can be determined explicitly, for instance, in the homogeneous EDF case with so-called conjugate priors, see Chapter 2 in Bühlmann–Gisler [58]. But in most cases, there is no closed from solution for the posterior distribution. Major progress in Bayesian modeling has been made with the emergence of computational methods like the Markov chain Monte Carlo (MCMC) method, Gibbs sampling, the Metropolis–Hastings (MH) algorithm [185, 274], sequential Monte Carlo (SMC) sampling, non-linear particle filters, and the Hamilton Monte Carlo (HMC) algorithm. These methods help us to empirically approximate the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ in different modeling setups. These methods have in common that the explicit knowledge of the normalizing constant in (6.2) is not necessary, but it suffices to know the functional form in $\boldsymbol{\beta}$ of the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$.

For a detailed description of MCMC methods in general, which includes Gibbs sampling and MH algorithms, we refer to Gilks et al. [158], Green [169, 170], Johansen et al. [199]; SMC sampling and non-linear particle filters are explained in Del Moral et al. [92, 93], Johansen–Evers [199], Doucet–Johansen [111], Creal [85] and Wüthrich [389]; the HMC algorithm is described in Neal [281]. We do not present these algorithms here, but for the description of the most popular algorithms we refer to Section 4.4 in Wüthrich–Buser [392]. The reason for not presenting these algorithms here is that they still face the curse of dimensionality, which makes it difficult to use Bayesian methods for high-dimensional data sets in large models; we provide another short discussion in Sect. 11.6.3, below.

6.2 Regularization

6.2.1 *Maximal a Posterior Estimator*

In the previous section we have proposed to approximate the posterior density $\pi(\boldsymbol{\beta}|\boldsymbol{Y})$ of the regression parameter $\boldsymbol{\beta}$, given $\boldsymbol{Y}$, using MCMC methods. The posterior log-likelihood in the Bayesian GLM is given by, see (6.2),

$$\begin{aligned}\log \pi(\boldsymbol{\beta}|\boldsymbol{Y}) &\propto \ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta}) \\ &\propto \sum_{i=1}^{n} \frac{Y_i (h \circ g^{-1})\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle - (\kappa \circ h \circ g^{-1})\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle}{\varphi / v_i} + \log \pi(\boldsymbol{\beta}).\end{aligned}$$

Compared to the classical log-likelihood function $\ell_{\boldsymbol{Y}}(\boldsymbol{\beta})$ for MLE, there is an additional log-density term $\log \pi(\boldsymbol{\beta})$ that comes from the prior distribution of $\boldsymbol{\beta}$. Thus, the posterior log-likelihood is a balanced version of the log-likelihood $\ell_{\boldsymbol{Y}}(\boldsymbol{\beta})$ of the data $\boldsymbol{Y}$ and the prior log-density $\log \pi(\boldsymbol{\beta})$ of the regression parameter $\boldsymbol{\beta}$. We interpret this as *regularization* because the prior π smooths extremes in the log-likelihood of the observation $\boldsymbol{Y}$. This gives rise to estimate the regression parameter $\boldsymbol{\beta}$ by the so-called maximal a posterior (MAP) estimator

$$\widehat{\boldsymbol{\beta}}^{\mathrm{MAP}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{q+1}}{\arg\max} \; \log \pi(\boldsymbol{\beta}|\boldsymbol{Y}) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{q+1}}{\arg\max} \; \ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta}). \tag{6.3}$$

This π-regularized (MAP) parameter estimation has gained much popularity because it is a useful tool to prevent the model from over-fitting under suitable prior choices. Moreover, under specific choices, it allows for parameter selection. This is especially useful in high-dimensional problems; for a reference we refer to Hastie et al. [184].

Popular choices for π are prior densities coming from L^p-norms for some $p \geq 1$, that is, $\pi(\boldsymbol{\beta}) \propto \exp\{-\lambda \|\boldsymbol{\beta}\|_p^p\}$ for $\lambda > 0$. Optimization problem (6.3) then becomes

$$\widehat{\boldsymbol{\beta}}^{\mathrm{MAP}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{q+1}}{\arg\max} \; \ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_p^p,$$

for a fixed *regularization parameter* $\lambda > 0$ (also called tuning parameter). In practical applications we should exclude the intercept parameter $\beta_0 \in \mathbb{R}$ from regularization: if we work with the canonical link within the GLM framework we have the balance property which implies unbiasedness, see Corollary 5.7. This property gets lost if β_0 is included in the regularization term. For this reason, we set $\boldsymbol{\beta}_- = (\beta_1, \ldots, \beta_q)^\top \in \mathbb{R}^q$ and we let regularization only act on these components

$$\widehat{\boldsymbol{\beta}}^{\rm MAP} = \widehat{\boldsymbol{\beta}}^{\rm MAP}(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\|\boldsymbol{\beta}_-\|_p^p, \tag{6.4}$$

we also scale with the sample size n to make the units of the tuning parameter λ independent of the sample size n.

Remarks 6.2

- The regularization term $\lambda\|\boldsymbol{\beta}_-\|_p^p$ keeps the components of the regression parameter $\boldsymbol{\beta}_-$ close to zero, thus, it prevents from over-fitting by letting parameters only take moderate values. The magnitudes of the parameter values are controlled by the regularization parameter $\lambda > 0$ which acts as a hyper-parameter. Optimal hyper-parameters are determined by cross-validation.
- In (6.4) all components of $\boldsymbol{\beta}_-$ are treated equally. This may not be appropriate if the feature components of $\boldsymbol{x}$ live on different scales. This problem of different scales can be solved by either scaling the components of $\boldsymbol{x}$ to a unit scale, or by introducing a diagonal importance matrix $T = \mathrm{diag}(t_1, \ldots, t_q)$ with $t_j > 0$ that describes the scales of the components of $\boldsymbol{x}$. This allows us to regularize $\|T^{-1}\boldsymbol{\beta}_-\|_p^p$ instead of $\|\boldsymbol{\beta}_-\|_p^p$. Thus, in this latter case we replace (6.4) by the weighted version

$$\widehat{\boldsymbol{\beta}}^{\rm MAP} = \underset{\boldsymbol{\beta}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\sum_{j=1}^{q} t_j^{-p}|\beta_j|^p.$$

- Often, the features have a natural group structure $\boldsymbol{x} = (x_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$, for instance, $\boldsymbol{x}_k \in \{0,1\}^{q_k}$ may represent dummy coding of a categorical feature component with $q_k + 1$ levels. In that case regularization should equally act on all components of $\boldsymbol{\beta}_k \in \mathbb{R}^{q_k}$ (that correspond to $\boldsymbol{x}_k$) because these components describe the same systematic effect. Yuan–Lin [398] proposed for this problem grouped penalties of the form

$$\widehat{\boldsymbol{\beta}}^{\rm MAP} = \underset{\boldsymbol{\beta}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2. \tag{6.5}$$

  This proposal leads to sparsity, i.e., for large regularization parameters λ the entire $\boldsymbol{\beta}_k$ may be shrunk (exactly) to zero; this is discussed in Sect. 6.2.5, below. We also refer to Section 4.3 in Hastie et al. [184], and Devriendt et al. [104] proposed this approach in the actuarial literature.
- There are more versions of regularization, e.g., in the fused LASSO approach we ensure that the first differences $\beta_j - \beta_{j-1}$ remain small.

Our motivation for considering regularization has been inspired by Bayesian theory, but we can also come from a completely different angle, namely, we can consider a constraint optimization problem with a given budget constraint $c > 0$. That is, we can consider

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) \qquad \text{subject to } \|\boldsymbol{\beta}_-\|_p^p \le c. \tag{6.6}$$

This optimization problem can be tackled by the method of Karush, Kuhn and Tucker (KKT) [208, 228]. Optimization problem (6.4) corresponds by Lagrangian duality to the constraint optimization problem (6.6). For every c for which the budget constraint in (6.6) is binding $\|\boldsymbol{\beta}_-\|_p^p = c$, there is a corresponding regularization parameter $\lambda = \lambda(c)$, and, conversely, the solution of (6.4) solves (6.6) with $c = \|\widehat{\boldsymbol{\beta}}_-^{\text{MAP}}(\lambda)\|_p^p$.

6.2.2 Ridge vs. LASSO Regularization

We compare the two special cases of $p = 1, 2$ in this section, and in the subsequent Sects. 6.2.3 and 6.2.4 we discuss how these two cases can be solved numerically.

Ridge Regularization $p = 2$ For $p = 2$, the prior distribution π in (6.4) is a centered Gaussian distribution. This L^2-regularization is called *ridge regularization* or Tikhonov regularization [353], and we have

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \widehat{\boldsymbol{\beta}}^{\text{ridge}}(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\sum_{j=1}^{q}\beta_j^2. \tag{6.7}$$

LASSO Regularization $p = 1$ For $p = 1$, the prior distribution π in (6.4) is a Laplace distribution. This L^1-regularization is called *LASSO regularization* (least absolute shrinkage and selection operator), see Tibshirani [352], and we have

$$\widehat{\boldsymbol{\beta}}^{\text{LASSO}} = \widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\sum_{j=1}^{q}|\beta_j|. \tag{6.8}$$

LASSO regularization has the advantage that it shrinks (unimportant) regression components to exactly zero, i.e., LASSO regularization can be used for parameter elimination and model reduction. This is discussed in the next paragraphs.

Ridge vs. LASSO Regularization Ridge ($p = 2$) and LASSO ($p = 1$) regularization behave rather differently. This can be understood best by using the budget constraint (6.6) interpretation which gives us a nice geometric illustration. The crucial part is that the side constraint gives us either a budget constraint $\|\boldsymbol{\beta}_-\|_2^2 = \sum_{j=1}^q \beta_j^2 \leq c$ (squared Euclidean norm) or $\|\boldsymbol{\beta}_-\|_1 = \sum_{j=1}^q |\beta_j| \leq c$ (Manhattan norm). In Fig. 6.1 we illustrate these two cases, the left-hand side shows the Euclidean ball in blue color (in two dimensions) and the right-hand side shows the corresponding Manhattan square in blue color; this figure is similar to Figure 2.2 in Hastie et al. [184].

The (unconstraint) MLE $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ is illustrated by the red dot in Fig. 6.1. If the red dot would lie within the blue area, the budget constraint would not be binding. In Fig. 6.1 the red dot (MLE) does not lie within the blue budget constraint, and we need to compromise in the optimality of the MLE. Assume that the log-likelihood $\boldsymbol{\beta} \mapsto \ell_{\boldsymbol{Y}}(\boldsymbol{\beta})$ is a concave function in $\boldsymbol{\beta}$, then we receive convex level sets $\{\boldsymbol{\beta}; \ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) \geq \gamma_0\}$ around the MLE $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$. The critical constant γ_0 for which this level set is tangential to the blue budget constraint exactly gives us the solution to (6.6); this solution corresponds to the yellow dots in Fig. 6.1. The crucial difference between ridge and LASSO regularization is that in the latter case the yellow dot will eventually be in the corner of the Manhattan square if we shrink the budget constraint c to zero. Or in other words, some of the components of $\boldsymbol{\beta}$ are set exactly equal to zero for small c or large λ, respectively; in Fig. 6.1 (rhs) this happens to the first component of $\widehat{\boldsymbol{\beta}}^{\mathrm{LASSO}}$ (under the given budget constraint c). In

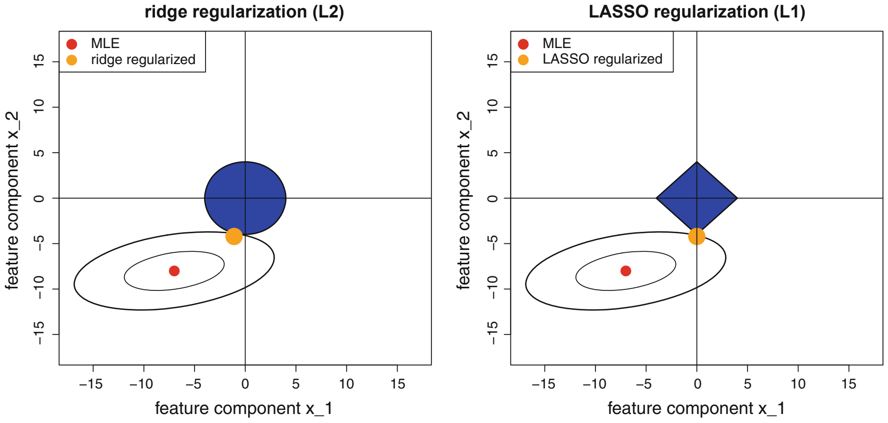
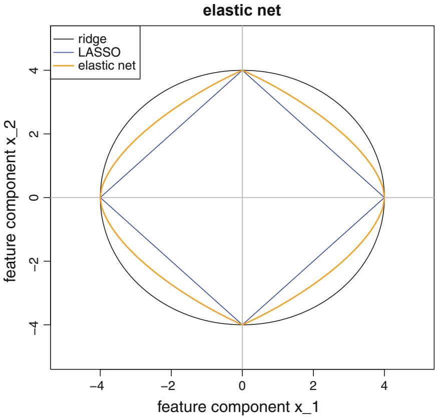


Fig. 6.1 Illustration of optimization problem (6.6) under a budget constraint (lhs) for $p = 2$ (Euclidean norm) and (rhs) $p = 1$ (Manhattan norm)

Fig. 6.2 Elastic net regularization



ridge regularization this is not the case, except for special situations concerning the position of the red MLE. Thus, ridge regression makes components of parameter estimates generally smaller, whereas LASSO shrinks some of these components exactly to zero (this also explains the name LASSO).

Remark 6.3 (Elastic Net) LASSO regularization faces difficulties with collinearity in feature components. In particular, if we have a group of highly correlated feature components, LASSO fails to do a grouped selection, but it selects one component and ignores the other ones. On the other hand, ridge regularization can deal with this issue. For this reason, Zou–Hastie [409] proposed the *elastic net regularization*, which uses a combined regularization term

$$\widehat{\boldsymbol{\beta}}^{\text{elastic net}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\max}\ \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda\left[(1-\alpha)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1\right],$$

for some $\alpha \in (0,1)$. The L^1-term gives sparsity and the quadratic term removes the limitation on the number of selected variables, providing a grouped selection. In Fig. 6.2 we compare the elastic net regularization (orange color) to ridge and LASSO regularization (black and blue color). Ridge regularization provides a smooth strictly convex boundary (black), whereas LASSO provides a boundary that is non-differentiable in the corners (blue). The elastic net is still non-differentiable in the corners, this is needed for variable selection, and at the same time it is strictly convex between the corners which is needed for grouping.

6.2.3 Ridge Regression

In this section we consider ridge regression ($p = 2$) in more detail and we provide an example. The ridge estimator $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ in (6.7) is found by solving the score equations

$$\widetilde{s}(\boldsymbol{\beta}, \boldsymbol{Y}) = \nabla_{\boldsymbol{\beta}} \left(\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - n\lambda \|\boldsymbol{\beta}_-\|_2^2\right) = \mathfrak{X}^\top W(\boldsymbol{\beta}) \boldsymbol{R}(\boldsymbol{Y}, \boldsymbol{\beta}) - 2n\lambda \boldsymbol{\beta}_- = 0, \tag{6.9}$$

note that we exclude the intercept β_0 from regularization (we use a slight abuse of notation, here), and we also refer to Proposition 5.1. The negative expected Hessian of this optimization problem is given by

$$\mathcal{J}(\boldsymbol{\beta}) = -\mathbb{E}_{\boldsymbol{\beta}} \left[\nabla^2_{\boldsymbol{\beta}} \left(\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - n\lambda \|\boldsymbol{\beta}_-\|_2^2\right)\right] = \mathcal{I}(\boldsymbol{\beta}) + 2n\lambda \text{diag}(0, 1, \ldots, 1) \in \mathbb{R}^{(q+1)\times(q+1)},$$

where $\mathcal{I}(\boldsymbol{\beta}) = \mathfrak{X}^\top W(\boldsymbol{\beta}) \mathfrak{X}$ is Fisher's information matrix of the unconstraint MLE problem. This provides us with Fisher's scoring updates for $t \geq 0$, see (5.13),

$$\widehat{\boldsymbol{\beta}}^{(t)} \mapsto \widehat{\boldsymbol{\beta}}^{(t+1)} = \widehat{\boldsymbol{\beta}}^{(t)} + \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1}\, \widetilde{s}(\widehat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{Y}). \tag{6.10}$$

Lemma 6.4 *Fisher's scoring update* (6.10) *can be rewritten as follows*

$$\widehat{\boldsymbol{\beta}}^{(t)} \mapsto \widehat{\boldsymbol{\beta}}^{(t+1)} = \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1} \mathfrak{X}^\top W(\widehat{\boldsymbol{\beta}}^{(t)}) \left(\mathfrak{X}\widehat{\boldsymbol{\beta}}^{(t)} + \boldsymbol{R}(\boldsymbol{Y}, \widehat{\boldsymbol{\beta}}^{(t)})\right).$$

Proof A straightforward calculation shows

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{(t+1)} &= \widehat{\boldsymbol{\beta}}^{(t)} + \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1}\, \widetilde{s}(\widehat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{Y}) \\
&= \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1} \left(\mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})\widehat{\boldsymbol{\beta}}^{(t)} + \mathfrak{X}^\top W(\widehat{\boldsymbol{\beta}}^{(t)}) \boldsymbol{R}(\boldsymbol{Y}, \widehat{\boldsymbol{\beta}}^{(t)}) - 2n\lambda \widehat{\boldsymbol{\beta}}_-^{(t)}\right) \\
&= \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1} \left(\mathcal{I}(\widehat{\boldsymbol{\beta}}^{(t)})\widehat{\boldsymbol{\beta}}^{(t)} + \mathfrak{X}^\top W(\widehat{\boldsymbol{\beta}}^{(t)}) \boldsymbol{R}(\boldsymbol{Y}, \widehat{\boldsymbol{\beta}}^{(t)})\right) \\
&= \mathcal{J}(\widehat{\boldsymbol{\beta}}^{(t)})^{-1} \mathfrak{X}^\top W(\widehat{\boldsymbol{\beta}}^{(t)}) \left(\mathfrak{X}\widehat{\boldsymbol{\beta}}^{(t)} + \boldsymbol{R}(\boldsymbol{Y}, \widehat{\boldsymbol{\beta}}^{(t)})\right).
\end{aligned}$$

This proves the claim. □

Lemma 6.4 allows us to fit a ridge regularized GLM. To determine an optimal regularization parameter $\lambda \geq 0$ one uses cross-validation, in particular, generalized cross-validation is used to receive an efficient cross-validation method, see (5.67).

Example 6.5 (Ridge Regression) We revisit the gamma claim size example of Sect. 5.3.7, and we choose model Gamma GLM1, see Listing 5.11. This example does not consider any categorical features, but only continuous ones. We directly

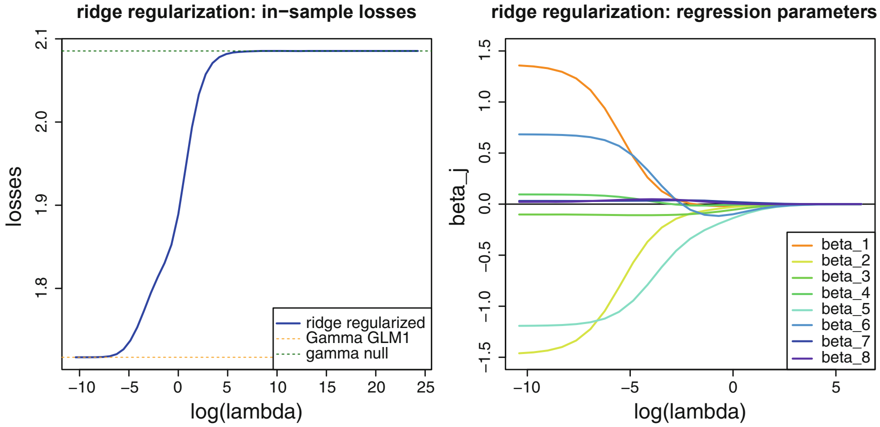


Fig. 6.3 Ridge regularized MLEs in model Gamma GLM1: (lhs) in-sample deviance losses as a function of the regularization parameter $\lambda > 0$, (rhs) resulting $\widehat{\beta}_j^{\text{ridge}}(\lambda)$ for $1 \le j \le q = 8$

apply Fisher's scoring updates (6.10).[3] For this analysis we center and normalize (to unit variance) the columns of the design matrix (except for the initial column of $\mathfrak{X}$ encoding the intercept).

Figure 6.3 (lhs) shows the resulting in-sample deviance losses as a function of $\lambda > 0$. Regularization parameter λ allows us to continuously connect the in-sample deviance losses of the null model (2.085) and model Gamma GLM1 (1.717), see Table 5.13. Figure 6.3 (rhs) shows the regression parameter estimates $\widehat{\beta}_j^{\text{ridge}}(\lambda)$, $1 \le j \le q = 8$, as a function of $\lambda > 0$. Overall they decrease because the budget constraint gets more tight for increasing λ, however, the individual parameters do not need to be monotone, since one parameter may (better) compensate a decrease of another (through correlations in feature components).

Finally, we need to choose the optimal regularization parameter $\lambda > 0$. This is done by cross-validation. We exploit the generalized cross-validation loss, see (5.67), and the hat matrix in this ridge regularized case is given by

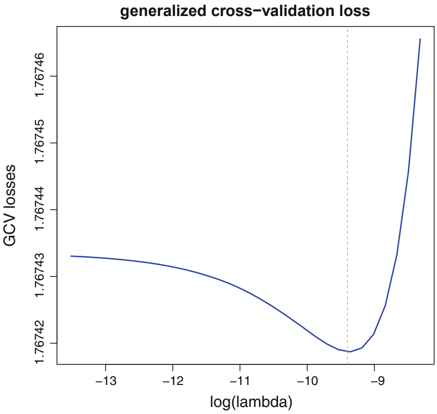
$$H_\lambda = W(\widehat{\boldsymbol{\beta}}^{\text{ridge}})^{1/2} \mathfrak{X}\, \mathcal{J}(\widehat{\boldsymbol{\beta}}^{\text{ridge}})^{-1}\, \mathfrak{X}^\top W(\widehat{\boldsymbol{\beta}}^{\text{ridge}})^{1/2}.$$

In contrast to (5.66), this hat matrix H_λ is not a projection but we would need to work in an augmented model to receive the projection property (accounting for the regularization part).

Figure 6.4 plots the generalized cross-validation loss as a function of $\lambda > 0$. We observe the minimum in parameter $\lambda = e^{-9.4}$. The resulting generalized cross-validation loss is 1.76742. This is bigger than the one received in model Gamma

[3] The R command `glmnet` [142] allows for regularized MLE, however, the current version does not include the gamma distribution. Therefore, we have implemented our own routine.

Fig. 6.4 Generalized cross-validation loss $\widehat{\mathfrak{D}}^{\mathrm{GCV}}(\lambda)$ as a function of $\lambda > 0$



GLM2, see Table 5.16, thus, we still prefer model Gamma GLM2 over the optimally ridge regularized model GLM1. Note that for model Gamma GLM2 we did variable selection, whereas ridge regression just generally shrinks regression parameters. For more interpretation we refer to Example 6.8, below, which considers LASSO regularization. ■

6.2.4 LASSO Regularization

In this section we consider LASSO regularization ($p = 1$). This is more challenging than ridge regularization because of the non-differentiability of the budget constraint, see Fig. 6.1 (rhs). This section follows Chapters 2 and 5 of Hastie et al. [184] and Parikh–Boyd [292].

Gaussian Case

We start with the homoskedastic Gaussian model having unit variance $\sigma^2 = 1$. In a first step, the regression model only involves one feature component $q = 1$. Thus, we aim at solving LASSO optimization

$$\widehat{\boldsymbol{\beta}}^{\mathrm{LASSO}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^2}{\arg\max} \; -\frac{1}{2n}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2 - \lambda|\beta_1|.$$

We standardize the observations and features $(Y_i, x_i)_{1\le i\le n}$ such that we have $\sum_{i=1}^n Y_i = 0$, $\sum_{i=1}^n x_i = 0$ and $n^{-1}\sum_{i=1}^n x_i^2 = 1$. This implies that we can omit the intercept parameter β_0, as the optimal intercept satisfies for this standardized data (and any $\beta_1 \in \mathbb{R}$)

$$\widehat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} Y_i - \beta_1 x_i = 0. \tag{6.11}$$

Thus, w.l.o.g., we assume to work with standardized data in this section, this gives us the optimization problem (we drop the lower index in β_1 because we only have one component)

$$\widehat{\beta}^{\text{LASSO}} = \widehat{\beta}^{\text{LASSO}}(\lambda) = \underset{\beta\in\mathbb{R}}{\arg\max} \; -\frac{1}{2n}\sum_{i=1}^{n}(Y_i - \beta x_i)^2 - \lambda|\beta|. \tag{6.12}$$

The difficulty is that the regularization term is not differentiable in zero. Since this term is convex we can express its derivative in terms of a sub-gradient $\mathfrak{s}$. This provides score

$$\frac{\partial}{\partial\beta}\left(-\frac{1}{2n}\sum_{i=1}^{n}(Y_i-\beta x_i)^2 - \lambda|\beta|\right) = \frac{1}{n}\sum_{i=1}^{n}(Y_i-\beta x_i)\,x_i - \lambda\mathfrak{s} = \frac{1}{n}\langle \boldsymbol{Y},\boldsymbol{x}\rangle - \beta - \lambda\mathfrak{s},$$

where we use standardization $n^{-1}\sum_{i=1}^{n} x_i^2 = 1$ in the second step, $\langle \boldsymbol{Y},\boldsymbol{x}\rangle$ is the scalar product of $\boldsymbol{Y}, \boldsymbol{x} = (x_1,\ldots,x_n)^\top \in \mathbb{R}^n$, and where we consider the sub-gradient

$$\mathfrak{s} = \mathfrak{s}(\beta) = \begin{cases} +1 & \text{if } \beta > 0,\\ -1 & \text{if } \beta < 0,\\ \in [-1,1] & \text{otherwise.}\end{cases}$$

Henceforth, we receive the score equation for $\beta \neq 0$

$$n^{-1}\langle \boldsymbol{Y},\boldsymbol{x}\rangle - \beta - \lambda\mathfrak{s} = n^{-1}\langle \boldsymbol{Y},\boldsymbol{x}\rangle - \beta - \text{sign}(\beta)\lambda \overset{!}{=} 0.$$

This score equation has a proper solution $\widehat{\beta} > 0$ if $n^{-1}\langle \boldsymbol{Y},\boldsymbol{x}\rangle > \lambda$, and it has a proper solution $\widehat{\beta} < 0$ if $n^{-1}\langle \boldsymbol{Y},\boldsymbol{x}\rangle < -\lambda$. In any other case we have a boundary solution $\widehat{\beta} = 0$ for our maximization problem (6.12).

This solution can be written in terms of the following *soft-thresholding operator* for $\lambda \geq 0$

$$\widehat{\beta}^{\text{LASSO}} = \mathcal{S}_\lambda\left(n^{-1}\langle \boldsymbol{Y},\boldsymbol{x}\rangle\right) \qquad \text{with} \qquad \mathcal{S}_\lambda(x) = \text{sign}(x)(|x|-\lambda)_+. \tag{6.13}$$

This soft-thresholding operator is illustrated in Fig. 6.5 for $\lambda = 4$.

This approach can be generalized to multiple feature components $\boldsymbol{x} \in \mathbb{R}^q$. We standardize the observations and features $\sum_{i=1}^{n} Y_i = 0$, $\sum_{i=1}^{n} x_{i,j} = 0$ and

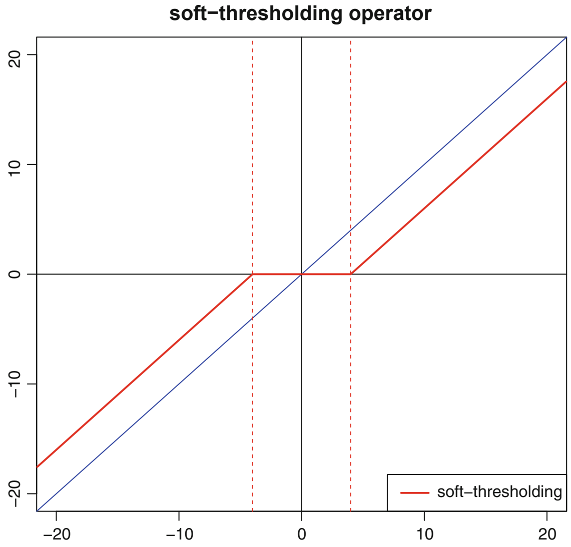


Fig. 6.5 Soft-thresholding operator $x \mapsto \mathcal{S}_\lambda(x)$ for $\lambda = 4$ (red dotted lines)

$n^{-1}\sum_{i=1}^n x_{i,j}^2 = 1$ for all $1 \le j \le q$. This allows us again to drop the intercept term and to directly consider

$$\widehat{\boldsymbol{\beta}}^{\text{LASSO}} = \widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^q}{\arg\max} \; -\frac{1}{2n}\sum_{i=1}^n \left(Y_i - \sum_{j=1}^q \beta_j x_{i,j}\right)^2 - \lambda\|\boldsymbol{\beta}\|_1.$$

Since this is a concave (quadratic) maximization problem with a separable (convex) penalty term, we can apply a *cycle coordinate descent method* that iterates a cyclic coordinate-wise maximization until convergence. Thus, if we want to maximize in the t-th iteration the j-th coordinate of the regression parameter we consider recursively

$$\widehat{\beta}_j^{(t)} = \underset{\beta_j\in\mathbb{R}}{\arg\max} \; -\frac{1}{2n}\sum_{i=1}^n \left(Y_i - \sum_{l=1}^{j-1}\beta_l^{(t)}x_{i,l} - \sum_{l=j+1}^{q}\beta_l^{(t-1)}x_{i,l} - \beta_j x_{i,j}\right)^2 - \lambda|\beta_j|.$$

Using the soft-thresholding operator (6.13) we find the optimal solution

$$\widehat{\beta}_j^{(t)} = \mathcal{S}_\lambda\left(n^{-1}\left\langle \boldsymbol{Y} - \sum_{l=1}^{j-1}\beta_l^{(t)}\boldsymbol{x}_l - \sum_{l=j+1}^{q}\beta_l^{(t-1)}\boldsymbol{x}_l,\; \boldsymbol{x}_j\right\rangle\right),$$

with vectors $\boldsymbol{x}_l = (x_{1,l}, \ldots, x_{n,l})^\top \in \mathbb{R}^n$ for $1 \le l \le q$. Iteration until convergence provides the LASSO regularized estimator $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda)$ for given regularization parameter $\lambda > 0$.

Typically, we want to explore $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda)$ for multiple λ's. For this, one runs a *pathwise cyclic coordinate descent method*. We start with a large value for λ, namely, we define

$$\lambda^{\max} = \max_{1 \le j \le q} n^{-1} \left| \langle \boldsymbol{Y}, \boldsymbol{x}_j \rangle \right|.$$

For $\lambda \ge \lambda^{\max}$, we have $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda) = 0$, i.e., we have the null model. Pathwise cycle coordinate descent starts with this solution for $\lambda_0 = \lambda^{\max}$. In a next step, one slightly decreases λ_0 and runs the cyclic coordinate descent algorithm until convergence for this slightly smaller $\lambda_1 < \lambda_0$, and with starting value $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda_0)$. This is then iterated for $\lambda_{t+1} < \lambda_t$, $t \ge 0$, which provides a sequence of LASSO regularized estimators $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda_t)$ along the path $(\lambda_t)_{t \ge 0}$.

For further remarks we refer to Section 2.6 in Hastie et al. [184]. This concerns statements about uniqueness for general design matrices, also in the set-up where $q > n$, i.e., where we have more parameters than observations. Moreover, references to convergence results are given in Section 2.7 of Hastie et al. [184]. This closes the Gaussian case.

Gradient Descent Algorithm for LASSO Regularization

In Sect. 7.2.3 we will discuss gradient descent methods for network fitting. In this section we provide preliminary considerations on gradient descent methods because these are also useful to fit LASSO regularized parameters within GLMs (different from Gaussian GLMs). Remark that we do a sign switch in what follows, and we aim at minimizing an objective function g.

Choose a convex and differentiable function $g : \mathbb{R}^{q+1} \to \mathbb{R}$. Assuming that the global minimum of g is achieved, a necessary and sufficient condition for the optimality of $\boldsymbol{\beta}^* \in \mathbb{R}^{q+1}$ in this convex setting is $\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\boldsymbol{\beta}^*} = 0$. *Gradient descent algorithms* find this optimal point by iterating for $t \ge 0$

$$\boldsymbol{\beta}^{(t)} \mapsto \boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \varrho_{t+1} \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)}), \tag{6.14}$$

for tempered *learning rates* $\varrho_{t+1} > 0$. This algorithm is motivated by a first order Taylor expansion that determines the direction of the maximal local decrease of the objective function g supposed we are in position $\boldsymbol{\beta}$, i.e.,

$$g(\widetilde{\boldsymbol{\beta}}) = g(\boldsymbol{\beta}) + \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta})^\top \left(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) + o\left(\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2\right) \qquad \text{as } \|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 \to 0.$$

The gradient descent algorithm (6.14) leads to the (unconstraint) minimum of the objective function g at convergence. A budget constraint like (6.6) leads to a convex constraint $\boldsymbol{\beta} \in \mathcal{C} \subset \mathbb{R}^{q+1}$. Consideration of such a convex constraint requires that we reformulate the gradient descent algorithm (6.14). The gradient descent step (6.14) can also be found, for given learning rate ϱ_{t+1}, by solving the following

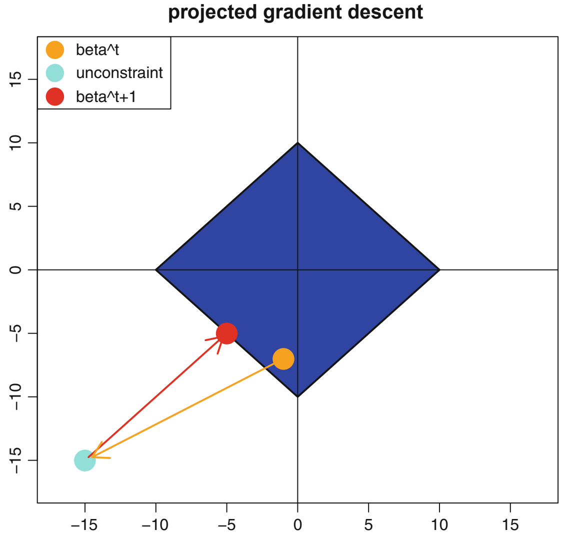


Fig. 6.6 Projected gradient descent step, first, mapping $\boldsymbol{\beta}^{(t)}$ to the unconstraint solution $\boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})$ of (6.15) and, second, projecting this unconstraint solution back to the convex set $\mathcal{C}$ giving $\boldsymbol{\beta}^{(t+1)}$; see also Figure 5.5 in Hastie et al. [184]

linearized problem for g with the Euclidean square distance penalty term (ridge regularization) for too big gradient descent steps

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\min}\left\{ g(\boldsymbol{\beta}^{(t)}) + \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})^\top \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\right) + \frac{1}{2\varrho_{t+1}} \|\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\|_2^2 \right\}. \tag{6.15}$$

The solution to this optimization problem exactly gives the gradient descent step (6.14). This is now adapted to a constraint gradient descent update for convex constraint $\mathcal{C}$:

$$\boldsymbol{\beta}^{(t+1)} = \underset{\boldsymbol{\beta}\in\mathcal{C}}{\arg\min}\left\{ g(\boldsymbol{\beta}^{(t)}) + \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})^\top \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\right) + \frac{1}{2\varrho_{t+1}} \|\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\|_2^2 \right\}. \tag{6.16}$$

The solution to this constraint convex optimization problem is obtained by, first, taking an unconstraint gradient descent step $\boldsymbol{\beta}^{(t)} \mapsto \boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})$, and, second, if this step is not within the convex set $\mathcal{C}$, it is projected back to $\mathcal{C}$; this is illustrated in Fig. 6.6, and it is called *projected gradient descent step* (justification is given in Lemma 6.6 below). Thus, the only difficulty in applying this projected gradient descent step is to find an efficient method of projecting the unconstraint solution (6.14)–(6.15) back to the convex constraint set $\mathcal{C}$.

Assume that the convex constraint set $\mathcal{C}$ is expressed by a convex function h (not necessarily being differentiable). To solve (6.16) and to motivate the projected gradient descent step, we use the *proximal gradient method* discussed in Section 5.3.3 of Hastie et al. [184]. The proximal gradient method helps us to do the projection in the projected gradient descent step. We introduce the *generalized* *projection operator*, for $z \in \mathbb{R}^{q+1}$

$$\mathrm{prox}_h(z) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\min} \left\{ \frac{1}{2} \|z - \boldsymbol{\beta}\|_2^2 + h(\boldsymbol{\beta}) \right\}. \tag{6.17}$$

This generalized projection operator should be interpreted as a square minimization problem $\|z - \boldsymbol{\beta}\|_2^2/2$ on a convex set $\mathcal{C}$ being expressed by its dual Lagrangian formulation described by the regularization term $h(\boldsymbol{\beta})$. The following lemma shows that the generalized projection operator solves the Lagrangian form of (6.16).

Lemma 6.6 *Assume the convex constraint $\mathcal{C}$ is expressed by the convex function h. The generalized projection operator solves*

$$\begin{aligned}\boldsymbol{\beta}^{(t+1)} &= \mathrm{prox}_{\varrho_{t+1}h}\left(\boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})\right) \\ &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\min}\left\{ g(\boldsymbol{\beta}^{(t)}) + \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})^\top \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\right) + \frac{1}{2\varrho_{t+1}}\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\|_2^2 + h(\boldsymbol{\beta})\right\}.\end{aligned} \tag{6.18}$$

Proof of Lemma 6.6 It suffices to consider the following calculation

$$\begin{aligned}&\frac{1}{2}\left\|\boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)}) - \boldsymbol{\beta}\right\|_2^2 + \varrho_{t+1}h(\boldsymbol{\beta}) \\ &= \frac{1}{2}\varrho_{t+1}^2\left\|\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})\right\|_2^2 - \varrho_{t+1}\left\langle\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)}), \boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}\right\rangle + \frac{1}{2}\left\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}\right\|_2^2 + \varrho_{t+1}h(\boldsymbol{\beta}) \\ &= \frac{1}{2}\varrho_{t+1}^2\left\|\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})\right\|_2^2 + \varrho_{t+1}\left(\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})^\top\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}\right) + \frac{1}{2\varrho_{t+1}}\left\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}\right\|_2^2 + h(\boldsymbol{\beta})\right).\end{aligned}$$

This is exactly the right objective function (in the round brackets) if we ignore all terms that are independent of $\boldsymbol{\beta}$. This proves the lemma. □

Thus, to solve the constraint optimization problem (6.16) we bring it into its dual Lagrangian form (6.18). Then we apply the generalized projection operator to the unconstraint solution to find the constraint solution, see Lemma 6.6. This approach will be successful if we can explicitly compute the generalized projection operator $\mathrm{prox}_h(\cdot)$.

Lemma 6.7 *The generalized projection operator* (6.17) *satisfies for LASSO constraint $h(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}_-\|_1$*

$$\mathrm{prox}_h(z) \;=\; \mathcal{S}_\lambda^{\mathrm{LASSO}}(z) \;\overset{\text{def.}}{=}\; \left(z_0, \mathrm{sign}(z_1)(|z_1| - \lambda)_+, \ldots, \mathrm{sign}(z_q)(|z_q| - \lambda)_+\right)^\top,$$

for $z \in \mathbb{R}^{q+1}$.

Proof of Lemma 6.7 We need to solve for function $\boldsymbol{\beta} \mapsto h(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}_-\|_1$

$$\mathrm{prox}_{\lambda\|(\cdot)_-\|_1}(z) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\min} \left\{ \frac{1}{2}\|z - \boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}_-\|_1 \right\} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{q+1}}{\arg\min} \left\{ \frac{1}{2}\sum_{j=0}^{q}(z_j - \beta_j)^2 + \lambda \sum_{j=1}^{q} |\beta_j| \right\}.$$

This decouples into $q + 1$ independent optimization problems. The first one is solved by $\beta_0 = z_0$ and the remaining ones are solved by the soft-thresholding operator (6.13). This finishes the proof. □

We conclude that the constraint optimization problem (6.16) for the (convex) LASSO constraint $\mathcal{C} = \{\boldsymbol{\beta};\ \|\boldsymbol{\beta}_-\|_1 \le c\}$ is brought into its dual Lagrangian form (6.18) of Lemma 6.6 with $h(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}_-\|_1$ for suitable $\lambda = \lambda(c)$. The LASSO regularized parameter estimation is then solved by first performing an unconstraint gradient descent step $\boldsymbol{\beta}^{(t)} \mapsto \boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)})$, and this updated parameter is projected back to $\mathcal{C}$ using the generalized projection operator of Lemma 6.7 with $h(\boldsymbol{\beta}) = \varrho_{t+1}\lambda\|\boldsymbol{\beta}_-\|_1$.

Proximal gradient descent algorithm for LASSO

1. Make the gradient descent step for a suitable learning rate $\varrho_{t+1} > 0$

$$\boldsymbol{\beta}^{(t)} \mapsto \widetilde{\boldsymbol{\beta}}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \varrho_{t+1}\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)}).$$

2. Perform soft-thresholding of the gradient descent solution

$$\widetilde{\boldsymbol{\beta}}^{(t+1)} \mapsto \boldsymbol{\beta}^{(t+1)} = \mathcal{S}^{\mathrm{LASSO}}_{\varrho_{t+1}\lambda}\left(\widetilde{\boldsymbol{\beta}}^{(t+1)}\right),$$

where the latter soft-thresholding function is defined in Lemma 6.7.

3. Iterate these two steps until a stopping criterion is met.

If the gradient $\nabla_{\boldsymbol{\beta}} g(\cdot)$ is Lipschitz continuous with Lipschitz constant $L > 0$, the proximal gradient descent algorithm will converge at rate $O(1/t)$ for a fixed step size $0 < \varrho = \varrho_{t+1} \le L$, see Section 4.2 in Parikh–Boyd [292].

Example 6.8 (LASSO Regression) We revisit Example 6.5 which considers claim size modeling using model Gamma GLM1. In order to apply the proximal gradient descent algorithm for LASSO regularization we need to calculate the gradient of the negative log-likelihood. In the gamma case with log-link, it is given by, see Example 5.5,

$$\begin{aligned} -\nabla_{\boldsymbol{\beta}}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) &= -\mathfrak{X}^\top W(\boldsymbol{\beta})\boldsymbol{R}(\boldsymbol{Y},\boldsymbol{\beta}) \\ &= -\mathfrak{X}^\top \mathrm{diag}\left(\frac{n_1}{\varphi}, \ldots, \frac{n_m}{\varphi}\right)\left(\frac{Y_1}{\mu_1} - 1, \ldots, \frac{Y_m}{\mu_m} - 1\right)^\top, \end{aligned}$$

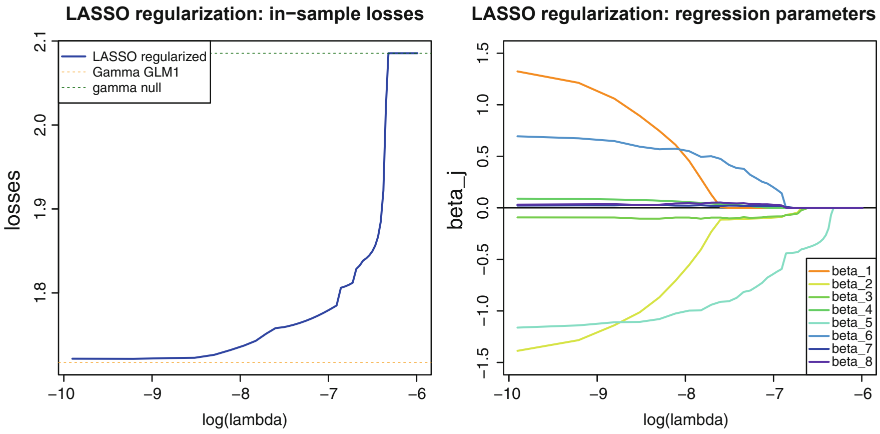


Fig. 6.7 LASSO regularized MLEs in model Gamma GLM1: (lhs) in-sample losses as a function of the regularization parameter $\lambda > 0$, (rhs) resulting $\widehat{\beta}_j^{\text{LASSO}}(\lambda)$ for $1 \leq j \leq q$

where $m \in \mathbb{N}$ is the number of policies with claims, and $\mu_i = \mu_i(\boldsymbol{\beta}) = \exp\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle$. We set $\varphi = 1$ as this constant can be integrated into the learning rates ϱ_{t+1}.

We have implemented the proximal gradient descent algorithm ourselves using an equidistant grid for the regularization parameter $\lambda > 0$, a fixed learning rate $\varrho_{t+1} = 0.05$ and normalized features. Since this has been done rather brute force, the results presented in Fig. 6.7 look a bit wiggly. These results should be compared to Fig. 6.3. We see that, in contrast to ridge regularization, less important regression parameters are shrunk exactly to zero in LASSO regularization. We give the order in which the parameters are shrunk to zero: β_1 (OwnerAge), β_4 (RiskClass), β_6 (VehAge^2), β_8 (BonusClass), β_7 (GenderMale), β_2 (OwnerAge^2), β_3 (AreaGLM) and β_5 (VehAge). In view of Listing 5.11 this order seems a bit surprising. The reason for this surprising order is that we have grouped features here, and, obviously, these should be considered jointly. In particular, we first drop OwnerAge because this can also be partially explained by OwnerAge^2, therefore, we should not treat these two variables individually. Having this weakness in mind supports the conclusions drawn from the Wald tests in Listing 5.11, and we come back to this in Example 6.10, below.

■

Oracle Property

An interesting question is whether the chosen regularization fulfills the so-called oracle property. For simplicity, we assume to work in the normalized Gaussian case that allows us to exclude the intercept β_0, see (6.11). Thus, we work with a regression parameter $\boldsymbol{\beta} \in \mathbb{R}^q$. Assume that there is a true data model that can be described by the (true) regression parameter $\boldsymbol{\beta}^* \in \mathbb{R}^q$. Denote by $\mathcal{A}^* = \{j \in \{1, \ldots, q\};\ \beta_j^* \neq 0\}$ the set of feature components of $\boldsymbol{x} \in \mathbb{R}^q$ that determine the true regression function, and we assume $|\mathcal{A}^*| < q$. Denote by $\widehat{\boldsymbol{\beta}}_n(\lambda)$ the parameter estimate that has been received by the regularized MAP estimation for a given regularization parameter $\lambda \geq 0$ and based on i.i.d. data of sample size n. We say that $(\widehat{\boldsymbol{\beta}}_n(\lambda_n))_{n\in\mathbb{N}}$ fulfills the *oracle property* if there exists a sequence $(\lambda_n)_{n\in\mathbb{N}}$ of regularization parameters $\lambda_n \geq 0$ such that

$$\lim_{n\to\infty} \mathbb{P}[\widehat{\mathcal{A}}_n = \mathcal{A}^*] = 1, \tag{6.19}$$

$$\sqrt{n}\left(\widehat{\boldsymbol{\beta}}_{n,\mathcal{A}^*}(\lambda_n) - \boldsymbol{\beta}^*_{\mathcal{A}^*}\right) \Rightarrow \mathcal{N}\left(0, \mathcal{I}^{-1}_{\mathcal{A}^*}\right) \qquad \text{as } n \to \infty, \tag{6.20}$$

where $\widehat{\mathcal{A}}_n = \{j \in \{1, \ldots, q\};\ (\widehat{\boldsymbol{\beta}}_n(\lambda_n))_j \neq 0\}$, $\boldsymbol{\beta}_{\mathcal{A}}$ only considers the components in $\mathcal{A} \subset \{1, \ldots, q\}$, and $\mathcal{I}_{\mathcal{A}^*}$ is Fisher's information matrix on the true feature components. The first oracle property (6.19) tells us that asymptotically we choose the right feature components, and the second oracle property (6.20) tells us that we have asymptotic normality and, in particular, consistency on the right feature components.

Zou [408] states that LASSO regularization, in general, does not satisfy the oracle property. LASSO regularization can perform variable selection, however, as Zou [408] argues, there are situations where consistency is violated and, therefore, the oracle property cannot hold in general. Zou [408] therefore proposes an adaptive LASSO regularization method. Alternatively, Fan–Li [124] introduced smoothly clipped absolute deviation (SCAD) regularization which is a non-convex regularization that possesses the oracle property. SCAD regularization of $\boldsymbol{\beta}$ is obtained by penalizing

$$J_\lambda(\boldsymbol{\beta}) = \sum_{j=1}^{q} \lambda|\beta_j|\mathbb{1}_{\{|\beta_j|\leq\lambda\}} - \frac{|\beta_j|^2 - 2a\lambda|\beta_j| + \lambda^2}{2(a-1)}\mathbb{1}_{\{\lambda<|\beta_j|\leq a\lambda\}} + \frac{(a+1)\lambda^2}{2}\mathbb{1}_{\{|\beta_j|>a\lambda\}},$$

for a hyperparameter $a > 2$. This function is continuous and differentiable except in $\beta_j = 0$ with partial derivatives for $\beta > 0$

$$\lambda\left(\mathbb{1}_{\{\beta\leq\lambda\}} + \frac{(a\lambda-\beta)_+}{\lambda(a-1)}\mathbb{1}_{\{\beta>\lambda\}}\right).$$

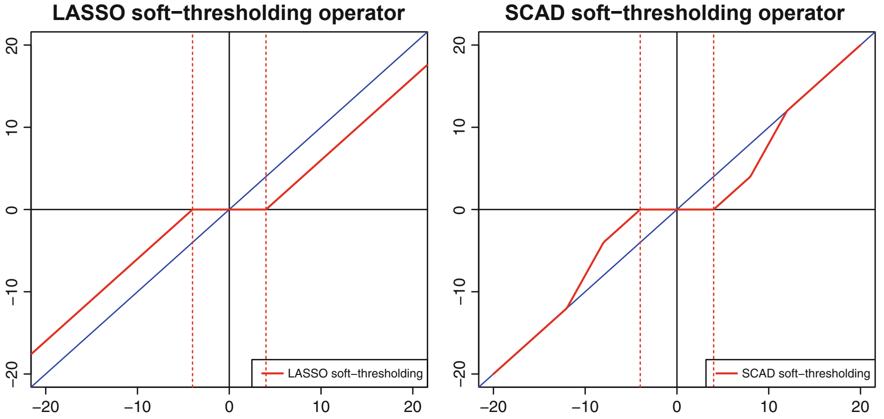


Fig. 6.8 (lhs) LASSO soft-thresholding operator $x \mapsto \mathcal{S}_\lambda(x)$ for $\lambda = 4$ (red dotted lines), (rhs) SCAD thresholding operator $x \mapsto \mathcal{S}_\lambda^{\text{SCAD}}(x)$ for $\lambda = 4$ and $a = 3$

Thus, we have a constant LASSO-like slope $\lambda > 0$ for $0 < \beta \leq \lambda$, shrinking some components exactly to zero. For $\beta > a\lambda$ the slope is 0, removing regularization, and it is concatenated between the two scenarios. The thresholding operator for SCAD regularization is given by, see Fan–Li [124],

$$\mathcal{S}_\lambda^{\text{SCAD}}(x) = \begin{cases} \text{sign}(x)(|x| - \lambda)_+ & \text{for } |x| \leq 2\lambda, \\ \frac{(a-1)x - \text{sign}(x)a\lambda}{a-2} & \text{for } 2\lambda < |x| \leq a\lambda, \\ x & \text{for } |x| > a\lambda. \end{cases}$$

Figure 6.8 compares the two thresholding operators of LASSO and SCAD.

Alternatively, we propose to do variable selection with LASSO regularization in a first step. Since the resulting LASSO regularized estimator may not be consistent, one should explore a second regression step where one uses an un-penalized regression model on the LASSO selected components, we also refer to Lee et al. [237].

6.2.5 Group LASSO Regularization

In Example 6.8 we have seen that if there are natural groups within the feature components they should be treated simultaneously. Assume we have a group structure $\boldsymbol{x} = (x_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$ with groups $\boldsymbol{x}_k \in \mathbb{R}^{q_k}$ that should be treated simultaneously. This motivates the grouped penalties proposed by Yuan–Lin [398], see (6.5),

$$\widehat{\boldsymbol{\beta}}^{\text{group}} = \widehat{\boldsymbol{\beta}}^{\text{group}}(\lambda) = \underset{\boldsymbol{\beta}=(\beta_0,\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_K)}{\arg\max} \frac{1}{n}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) - \lambda \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|_2, \tag{6.21}$$

where we assume a group structure in the linear predictor providing

$$\boldsymbol{x} \mapsto \eta(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle = \beta_0 + \sum_{k=1}^{K} \langle \boldsymbol{\beta}_k, \boldsymbol{x}_k \rangle.$$

LASSO regularization is a special case of this grouped regularization, namely, if all groups $1 \leq k \leq K$ only contain one single component, i.e., $K = q$, we have $\widehat{\boldsymbol{\beta}}^{\text{group}} = \widehat{\boldsymbol{\beta}}^{\text{LASSO}}$.

The side constraint in (6.21) is convex, and the optimization problem (6.21) can again be solved by the proximal gradient descent algorithm. That is, in view of Lemma 6.6, the only difficulty is the calculation of the generalized projection operator for regularization term $h(\boldsymbol{\beta}) = \lambda \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|_2$. We therefore need to solve for $\boldsymbol{z} = (z_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K)$, $\boldsymbol{z}_k \in \mathbb{R}^{q_k}$,

$$\begin{aligned} \text{prox}_h(\boldsymbol{z}) &= \underset{\boldsymbol{\beta}=(\beta_0,\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_K)}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{\beta}\|_2^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|_2 \right\} \\ &= \left(z_0, \left(\underset{\boldsymbol{\beta}_k \in \mathbb{R}^{q_k}}{\arg\min} \left\{ \frac{1}{2} \left\|\boldsymbol{z}_k - \boldsymbol{\beta}_k\right\|_2^2 + \lambda \|\boldsymbol{\beta}_k\|_2 \right\} \right)_{1 \leq k \leq K} \right). \end{aligned}$$

The latter highlights that the problem decouples into K independent problems. Thus, we need to solve for all $1 \leq k \leq K$ the optimization problems

$$\underset{\boldsymbol{\beta}_k \in \mathbb{R}^{q_k}}{\arg\min} \left\{ \frac{1}{2} \left\|\boldsymbol{z}_k - \boldsymbol{\beta}_k\right\|_2^2 + \lambda \|\boldsymbol{\beta}_k\|_2 \right\}.$$

Lemma 6.9 *The group LASSO generalized soft-thresholding operator satisfies for* $z_k \in \mathbb{R}^{q_k}$

$$\mathcal{S}_\lambda^{q_k}(z_k) = \underset{\boldsymbol{\beta}_k \in \mathbb{R}^{q_k}}{\arg\min} \left\{ \frac{1}{2} \left\| z_k - \boldsymbol{\beta}_k \right\|_2^2 + \lambda \|\boldsymbol{\beta}_k\|_2 \right\} = z_k \left(1 - \frac{\lambda}{\|z_k\|_2} \right)_+ \in \mathbb{R}^{q_k},$$

and for the generalized projection operator for $h(\boldsymbol{\beta}) = \lambda \sum_{k=1}^K \|\boldsymbol{\beta}_k\|_2$ *we have*

$$\mathrm{prox}_h(z) = \mathcal{S}_\lambda^{\mathrm{group}}(z) \overset{\text{def.}}{=} \left(z_0, \mathcal{S}_\lambda^{q_1}(z_1), \ldots, \mathcal{S}_\lambda^{q_K}(z_K) \right),$$

for $z = (z_0, z_1, \ldots, z_K)$ *with* $z_k \in \mathbb{R}^{q_k}$.

Proof We prove this lemma. In a first step we have

$$\underset{\boldsymbol{\beta}_k}{\arg\min} \left\{ \frac{1}{2} \left\| z_k - \boldsymbol{\beta}_k \right\|_2^2 + \lambda \|\boldsymbol{\beta}_k\|_2 \right\} = \underset{\boldsymbol{\beta}_k = \varrho z_k / \|z_k\|_2,\ \varrho \ge 0}{\arg\min} \left\{ \frac{1}{2} \|z_k\|_2^2 \left(1 - \frac{\varrho}{\|z_k\|_2} \right)^2 + \lambda \varrho \right\},$$

this follows because the square distance $\left\| z_k - \boldsymbol{\beta}_k \right\|_2^2 = \|z_k\|_2^2 - 2\langle z_k, \boldsymbol{\beta}_k \rangle + \left\| \boldsymbol{\beta}_k \right\|_2^2$ is minimized if z_k and $\boldsymbol{\beta}_k$ point into the same direction. Thus, there remains the minimization of the objective function in $\varrho \ge 0$. The first derivative is given by

$$\frac{\partial}{\partial \varrho} \left(\frac{1}{2} \|z_k\|_2^2 \left(1 - \frac{\varrho}{\|z_k\|_2} \right)^2 + \lambda \varrho \right) = - \|z_k\|_2 \left(1 - \frac{\varrho}{\|z_k\|_2} \right) + \lambda = \lambda - \|z_k\|_2 + \varrho.$$

If $\|z_k\|_2 > \lambda$ we have $\varrho = \|z_k\|_2 - \lambda > 0$, and otherwise we need to set $\varrho = 0$. This implies

$$\mathcal{S}_\lambda^{q_k}(z_k) = \left(\|z_k\|_2 - \lambda \right)_+ z_k / \|z_k\|_2.$$

This completes the proof. □

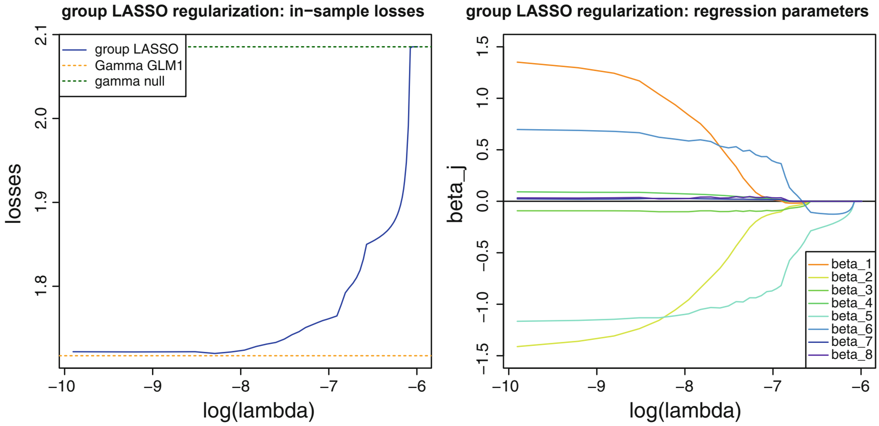


Fig. 6.9 Group LASSO regularized MLEs in model Gamma GLM1: (lhs) in-sample losses as a function of the regularization parameter $\lambda > 0$, (rhs) resulting $\widehat{\beta}_j^{\text{group}}(\lambda)$ for $1 \leq j \leq q$

Proximal gradient descent algorithm for group LASSO

---

1. Make the gradient descent step for a suitable learning rate $\varrho_{t+1} > 0$

$$\boldsymbol{\beta}^{(t)} \mapsto \widetilde{\boldsymbol{\beta}}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \varrho_{t+1} \nabla_{\boldsymbol{\beta}} g(\boldsymbol{\beta}^{(t)}).$$

2. Perform soft-thresholding of the gradient descent solution

$$\widetilde{\boldsymbol{\beta}}^{(t+1)} \mapsto \boldsymbol{\beta}^{(t+1)} = \mathcal{S}^{\text{group}}_{\varrho_{t+1}\lambda}\left(\widetilde{\boldsymbol{\beta}}^{(t+1)}\right),$$

where the latter soft-thresholding function is defined in Lemma 6.9.

3. Iterate these two steps until a stopping criterion is met.

Example 6.10 (Group LASSO Regression) We revisit Example 6.8 which considers claim size modeling using model Gamma GLM1. This time we group the variables `OwnerAge` and `OwnerAge`^{2} (β_1, β_2) as well as `VehAge` and `VehAge`^{2} (β_5, β_6). The results are shown in Fig. 6.9.

The order in which the parameters are regularized to zero is: β_4 (`RiskClass`), β_8 (`BonusClass`), β_7 (`GenderMale`), (β_1, β_2) (`OwnerAge`, `OwnerAge`^{2}), β_3 (`AreaGLM`) and (β_5, β_6) (`VehAge`, `VehAge`^{2}). This order now reflects more the variable importance as received from the Wald statistics of Listing 5.11, and it shows that grouped features should be regularized jointly in order to determine their importance. ■

6.3 Expectation-Maximization Algorithm

6.3.1 Mixture Distributions

In many applied problems there does not exist a simple off-the-shelf distribution that is suitable to model the whole range of observations. We think of claim size modeling which may range from small to very large claims; the main body of the data may look like, say, gamma distributed, but the tail of the data being regularly varying. Another related problem is that claims may come from different insurance policy modules. For instance, in property insurance, one can insure water damage, fire, glass and theft claims on the same insurance policy, and feature information about the claim type may not always be available. In such cases, it looks attractive to choose a mixture or a composition of different distributions. In this section we focus on mixtures.

Choose a fixed integer K bigger than 1 and define the $(K-1)$-unit simplex excluding the edges by

$$\Delta_K = \left\{ \boldsymbol{p} \in (0,1)^K;\ \sum_{k=1}^K p_k = 1 \right\}. \tag{6.22}$$

Δ_K defines the family of categorical distributions with K levels (all levels having a strictly positive probability). These distributions belong to the vector-valued parameter EF which we have met in Sects. 2.1.4 and 5.7.

The idea behind mixture distributions is to mix K different distributions with a mixture probability $\boldsymbol{p} \in \Delta_K$. For instance, we can mix K different EDF densities f_k by considering

$$Y \sim \sum_{k=1}^K p_k f_k(y; \theta_k, v/\varphi_k) = \sum_{k=1}^K p_k \exp\left\{ \frac{y\theta_k - \kappa_k(\theta_k)}{\varphi_k/v} + a_k(y; v/\varphi_k) \right\}, \tag{6.23}$$

with cumulant functions $\theta_k \in \boldsymbol{\Theta}_k \mapsto \kappa_k(\theta_k)$, exposure $v > 0$ and dispersion parameters $\varphi_k > 0$, for $1 \le k \le K$.

At the first sight, this does not look very spectacular and parameter estimation seems straightforward. If we consider the log-likelihood of n independent random variables $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$ following mixture density (6.23) we receive log-likelihood function

$$(\boldsymbol{\theta}, \boldsymbol{p}) \mapsto \ell_{\boldsymbol{Y}}(\boldsymbol{\theta}, \boldsymbol{p}) = \sum_{i=1}^n \ell_{Y_i}(\boldsymbol{\theta}, \boldsymbol{p}) = \sum_{i=1}^n \log\left(\sum_{k=1}^K p_k f_k(Y_i; \theta_k, v_i/\varphi_k) \right), \tag{6.24}$$

for canonical parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^\top \in \boldsymbol{\Theta} = \boldsymbol{\Theta}_1 \times \cdots \times \boldsymbol{\Theta}_K$ and mixture probability $\boldsymbol{p} \in \Delta_K$. Unfortunately, MLE of $(\boldsymbol{\theta}, \boldsymbol{p})$ in (6.24) is not that simple. Note, the summation over $1 \leq k \leq K$ is inside of the logarithmic function, and the use of the Newton–Raphson algorithm may be cumbersome. The Expectation-Maximization (EM) algorithm presented in Sect. 6.3.3, below, makes parameter estimation feasible. In a nutshell, the EM algorithm leads to a sequence of parameter estimates for $(\boldsymbol{\theta}, \boldsymbol{p})$ that monotonically increases the log-likelihood in each iteration of the algorithm. Thus, we can receive an approximation to the MLE of $(\boldsymbol{\theta}, \boldsymbol{p})$.

Nevertheless, model fitting may still be difficult for the following reasons. Firstly, the log-likelihood function of a mixture distribution does not need to be bounded, we highlight this in Example 6.13, below. In that case, MLE is not a well-defined problem. Secondly, even in very simple situations, the log-likelihood function (6.24) can have multiple local maximums. This usually happens if the data is clustered and the clusters are well separated. In that case of multiple local maximums, convergence of the EM algorithm does not guarantee that we have found the global maximum. Thirdly, convergence of the log-likelihood function through the EM algorithm does not guarantee that also the sequence of parameter estimates of $(\boldsymbol{\theta}, \boldsymbol{p})$ converges. The latter needs additional examination and regularity conditions.

Figure 6.10 (lhs) shows a density of a mixture distribution mixing $K = 3$ gamma densities with shape parameters $\alpha_k = 1, 20, 40$ (orange, green and blue) and mixture probability $\boldsymbol{p} = (0.7, 0.1, 0.2)^\top$; the mixture components are already multiplied with $\boldsymbol{p}$. The resulting mixture density in red color is continuous. Figure 6.10 (rhs) replaces the blue gamma component of the plot on the left-hand side by a Pareto component (in blue). As a result we observe that the resulting mixture density in red is no longer continuous. This example is often used in practice, however, the discontinuity may be a serious issue in applications and one may use a Lomax (Pareto Type II) component instead, we refer to Sect. 2.2.5.

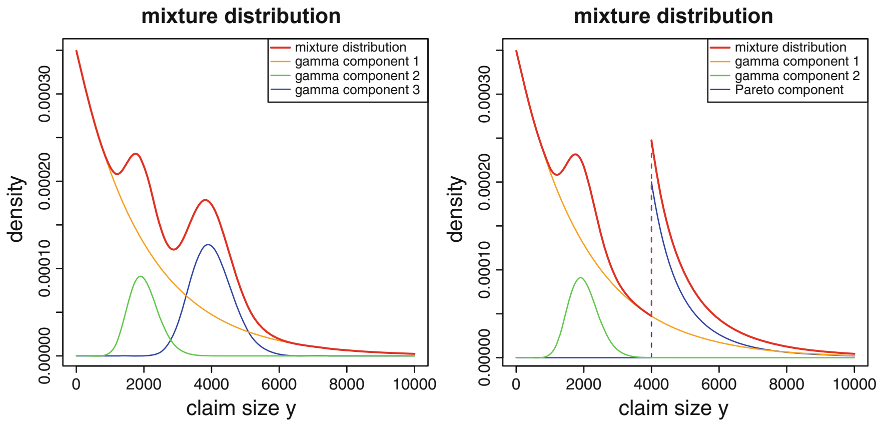


Fig. 6.10 (lhs) Mixture distribution mixing three gamma densities, and (rhs) mixture distributions mixing two gamma components and a Pareto component with mixture probabilities $\boldsymbol{p} = (0.7, 0.1, 0.2)^\top$ for orange, green and blue components (the density components are already multiplied with $\boldsymbol{p}$)

6.3.2 Incomplete and Complete Log-Likelihoods

A mixture distribution can be defined (brute force) by just defining a mixture density as in (6.23). Alternatively, we could define a mixture distribution in a more constructive way. In the following we discuss this constructive derivation which will allow us to efficiently fit mixture distributions to data $\boldsymbol{Y}$. For our outline we focus on (6.23), but all results presented below hold true in much more generality.

Choose a categorical random variable Z with $K \geq 2$ levels having probabilities $\mathbb{P}[Z = k] = p_k > 0$ for $1 \leq k \leq K$, that is, with $\boldsymbol{p} \in \Delta_K$. The main idea is to sample in a first step level $Z = k \in \{1, \ldots, K\}$, and in a second step $Y|_{\{Z=k\}} \sim f_k(y; \theta_k, v/\varphi_k)$, based on the selected level $Z = k$. The random tuple (Y, Z) has joint density

$$(Y, Z) \;\sim\; f_{\boldsymbol{\theta},\boldsymbol{p}}(y, k) = p_k f_k(y; \theta_k, v/\varphi_k),$$

and the marginal density of Y is exactly given by (6.23). In this interpretation we have a hierarchical model (Y, Z). If only Y is available for parameter estimation, then we are in the situation of *incomplete information* because information about the first hierarchy Z is missing. If both Y and Z are available we say that we have *complete information*.

For the subsequent derivations we use a different coding of the categorical random variable Z, namely, Z can be represented in the following one-hot encoding version

$$\boldsymbol{Z} = (Z_1, \ldots, Z_K)^\top = (\mathbb{1}_{\{Z=1\}}, \ldots, \mathbb{1}_{\{Z=K\}})^\top, \tag{6.25}$$

these are the K corners of the $(K-1)$-unit simplex Δ_K. One-hot encoding differs from dummy coding (5.21). One-hot encoding does not lead to a full rank design matrix because there is a redundancy, that is, we can drop one component of $\boldsymbol{Z}$ and still have the same information. One-hot encoding $\boldsymbol{Z}$ of Z allows us to extend the *incomplete (data) log-likelihood* $\ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$, see (6.23)–(6.24), under complete information $(Y, \boldsymbol{Z})$ as follows

$$\begin{aligned}
\ell_{(Y,\boldsymbol{Z})}(\boldsymbol{\theta}, \boldsymbol{p}) &= \log\left(\prod_{k=1}^{K} \left(p_k f_k(Y; \theta_k, v/\varphi_k)\right)^{Z_k}\right) \\
&= \log\left(\prod_{k=1}^{K} \left(p_k \exp\left\{\frac{Y\theta_k - \kappa_k(\theta_k)}{\varphi_k/v} + a_k(Y; v/\varphi_k)\right\}\right)^{Z_k}\right) \qquad (6.26) \\
&= \sum_{k=1}^{K} Z_k \left(\log(p_k) + \frac{Y\theta_k - \kappa_k(\theta_k)}{\varphi_k/v} + a_k(Y; v/\varphi_k)\right).
\end{aligned}$$

$\ell_{(Y,\boldsymbol{Z})}(\boldsymbol{\theta}, \boldsymbol{p})$ is called *complete (data) log-likelihood*. As a consequence of this last expression we observe that under complete information $(Y_i, \boldsymbol{Z}_i)_{1\le i\le n}$, the MLE of $\boldsymbol{\theta}$ and $\boldsymbol{p}$ can be determined completely analogously to above. Namely, θ_k is estimated from all observations Y_i for which $\boldsymbol{Z}_i$ belongs to level k, and the level indicators $(\boldsymbol{Z}_i)_{1\le i\le n}$ are used to estimate the mixture probability $\boldsymbol{p}$. Thus, the objective function nicely decouples under complete information into independent parts for θ_k and $\boldsymbol{p}$ estimation. There remains the question of how to fit this model under incomplete information $\boldsymbol{Y}$. The next section will discuss this problem.

6.3.3 Expectation-Maximization Algorithm for Mixtures

The EM algorithm is a general purpose tool for parameter estimation under incomplete information. The EM algorithm has been introduced within the EF by Sundberg [348, 349]. Sundberg's developments have been based on the vector-valued parameter EF with statistics $S(\boldsymbol{Y}) \in \mathbb{R}^k$, see (3.17), and he solved the estimation problem under the assumption that $S(\boldsymbol{Y})$ is not fully known. These results have been generalized to MLE under incomplete data in the celebrated work of Dempster et al. [96] and Wu [385]. The monograph of McLachlan–Krishnan [267] gives the theory behind the EM algorithm, and it also provides a historical review in Section 1.8. In actuarial science the EM algorithm is increasingly used to solve various kinds of problems of incomplete data. Mixture models of Erlang distributions are considered in Lee–Lin [240], Yin–Lin [396] and Fung et al. [146, 147]; general Erlang mixtures are universal approximators to positive distributions (in the weak convergence sense), and regularized Erlang mixtures and mixtures of experts models are determined using the EM algorithm to receive approximations to the true underlying model. Miljkovic–Grün [278], Parodi [295] and Fung et al. [148] consider the EM algorithm for mixtures of general distributions, in particular, mixtures of small and large claims distributions. Verbelen et al. [371], Blostein–Miljkovic [40], Grün–Miljkovic [173] and Fung et al. [147] use the EM algorithm for censored and/or truncated observations, and dispersion modeling is performed with the EM algorithm in Tzougas–Karlis [359]. (Inhomogeneous) phase-type and matrix Mittag–Leffler distributions are fitted with the EM algorithm in Asmussen et al. [14], Albrecher et al. [8] and Bladt [37], and the EM algorithm is used to fit mixture density networks (MDNs) in Delong et al. [95]. Parameter uncertainty is investigated in O'Hagan et al. [289] using the bootstrap method. The present section is mainly based on McLachlan–Krishnan [267].

As mentioned above, the EM algorithm is a general purpose tool for parameter estimation under incomplete data, and we describe the variant of the EM algorithm which is useful for our mixture distribution setup given in (6.26). We give a justification for its functioning below. The EM algorithm is an iterative algorithm that performs a Bayesian expectation step (E-step) to infer the latent variable $\boldsymbol{Z}$, given the model parameters and Y. Next, it performs a maximization step (M-step) for MLE of the parameters given the observation Y and the estimated latent variable $\widehat{\boldsymbol{Z}}$. More specifically, the E-step and the M-step look as follows.

- **E-step.** Calculate the posterior probability of the event that a given observation Y has been generated from the k-th component of the mixture distribution. Bayes' rule allows us to infer this posterior probability (for given $\boldsymbol{\theta}$ and $\boldsymbol{p}$) from (6.26)

$$\mathbb{P}_{\boldsymbol{\theta},\boldsymbol{p}}[Z_k = 1|Y] \;=\; \frac{p_k f_k(Y;\theta_k, v/\varphi_k)}{\sum_{l=1}^K p_l f_l(Y;\theta_l, v/\varphi_l)}.$$

The posterior (Bayesian) estimate for Z_k after having observed Y is given by

$$\widehat{Z}_k(\boldsymbol{\theta},\boldsymbol{p}|Y) \overset{\text{def.}}{=} \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{p}}[Z_k|Y] = \mathbb{P}_{\boldsymbol{\theta},\boldsymbol{p}}[Z_k = 1|Y] \qquad \text{for } 1 \le k \le K. \tag{6.27}$$

This posterior mean $\widehat{\boldsymbol{Z}} = \widehat{\boldsymbol{Z}}(\boldsymbol{\theta},\boldsymbol{p}|Y) = (\widehat{Z}_1(\boldsymbol{\theta},\boldsymbol{p}|Y),\ldots,\widehat{Z}_K(\boldsymbol{\theta},\boldsymbol{p}|Y))^\top \in \Delta_K$ is used as an estimate for the (unobserved) latent variable $\boldsymbol{Z}$; note that this posterior mean depends on the unknown parameters $(\boldsymbol{\theta},\boldsymbol{p})$.

- **M-step.** Based on Y and $\widehat{\boldsymbol{Z}}$ the parameters $\boldsymbol{\theta}$ and $\boldsymbol{p}$ are estimated with MLE.

Alternation of these two steps provide the following recursive algorithm. We assume to have independent responses $(Y_i, \boldsymbol{Z}_i)$, $1 \le i \le n$, following the mixture distribution (6.26), where, for simplicity, we assume that only the volumes $v_i > 0$ are dependent on i.

EM algorithm for mixture distributions

(0) Choose an initial parameter $(\widehat{\boldsymbol{\theta}}^{(0)}, \widehat{\boldsymbol{p}}^{(0)}) \in \boldsymbol{\Theta} \times \Delta_K$.

(1) Repeat for $t \ge 1$ until a stopping criterion is met:

- **E-step.** Given parameter $(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}) \in \boldsymbol{\Theta} \times \Delta_K$ estimate the latent variables $\boldsymbol{Z}_i$, $1 \le i \le n$, by their conditional expectations, see (6.27),

$$\widehat{\boldsymbol{Z}}_i^{(t)} \;=\; \widehat{\boldsymbol{Z}}\left(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)} \middle| Y_i\right) \;=\; \mathbb{E}_{\widehat{\boldsymbol{\theta}}^{(t-1)},\widehat{\boldsymbol{p}}^{(t-1)}}[\boldsymbol{Z}_i|Y_i] \;\in\; \Delta_K. \tag{6.28}$$

- **M-step.** Calculate the MLE $(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)}) \in \boldsymbol{\Theta} \times \Delta_K$ based on (complete) observations $((Y_1, \widehat{\boldsymbol{Z}}_1^{(t)}), \ldots, (Y_n, \widehat{\boldsymbol{Z}}_n^{(t)}))$, i.e., solve the score equations, see (6.26),

$$\nabla_{\boldsymbol{\theta}} \left(\sum_{i=1}^{n} \sum_{k=1}^{K} \widehat{Z}_{i,k}^{(t)} \frac{Y_i \theta_k - \kappa_k(\theta_k)}{\varphi_k / v_i} \right) = 0, \tag{6.29}$$

$$\nabla_{\boldsymbol{p}_-} \left(\sum_{i=1}^{n} \sum_{k=1}^{K} \widehat{Z}_{i,k}^{(t)} \log(p_k) \right) = 0, \tag{6.30}$$

where $\boldsymbol{p}_- = (p_1, \ldots, p_{K-1})^\top$ and setting $p_K = 1 - \sum_{k=1}^{K-1} p_k \in (0, 1)$.

Remarks 6.11

- The E-step uses Bayes' rule. This motivates to consider the EM algorithm in this Bayesian chapter; alternatively, it also fits to the MLE chapters.
- We have formulated the M-step in (6.29)–(6.30) in a general way because the canonical parameter $\boldsymbol{\theta}$ and the mixture probability $\boldsymbol{p}$ could be modeled by GLMs, and, henceforth, they may be feature $\boldsymbol{x}_i$ dependent. Moreover, (6.29) is formulated for a mixture of single-parameter EDF distributions, but, of course, this holds in much more generality.
- Equations (6.29)–(6.30) are the score equations received from (6.26). There is a subtle point here, namely, $Z_k \in \{0, 1\}$ in (6.26) are observations, whereas $\widehat{Z}_{i,k}^{(t)} \in (0, 1)$ in (6.29)–(6.30) are their estimates. Thus, in the EM algorithm the unknown latent variables are replaced by their estimates which, in our setup, results in two different types of variables with disjoint ranges. This may matter in software implementations, for instance, a categorical GLM may ask for a categorical random variable $Z \in \{1, \ldots, K\}$ (of `factor` type), whereas $\widehat{\boldsymbol{Z}}$ is in the interior of the unit simplex Δ_K.
- For mixture distributions one can replace the latent variables $\boldsymbol{Z}_i$ by their conditionally expected values $\widehat{\boldsymbol{Z}}_i$, see (6.29)–(6.30). In general, this does not hold true in EM algorithm applications: in our case we benefit from the fact that Z_k influences the complete log-likelihood *linearly*, see (6.26). In the general (non-linear) case of the EM algorithm application, different from mixture distribution problems, one needs to calculate the conditional expectation of the log-likelihood function.

- If we calculate the scores element-wise we receive

$$\frac{\partial}{\partial \theta_k} \sum_{i=1}^n \frac{Y_i \theta_k - \kappa_k(\theta_k)}{\varphi_k / (v_i \widehat{Z}_{i,k}^{(t)})} = 0,$$

$$\frac{\partial}{\partial p_k} \sum_{i=1}^n \left(\widehat{Z}_{i,k}^{(t)} \log(p_k) + \widehat{Z}_{i,K}^{(t)} \log(p_K) \right) = 0,$$

recall normalization $p_K = 1 - \sum_{k=1}^{K-1} p_k \in (0, 1)$.

From the first score equation we see that we receive the classical MLE/GLM framework, and all tools introduced above for parameter estimation can directly be used. The only part that changes are the weights $v_i \mapsto v_i \widehat{Z}_{i,k}^{(t)}$. In the homogeneous case, i.e., in the null model we have MLE after the t-th iteration of the EM algorithm

$$\widehat{\theta}_k^{(t)} = h_k \left(\frac{\sum_{i=1}^n v_i \widehat{Z}_{i,k}^{(t)} Y_i}{\sum_{i=1}^n v_i \widehat{Z}_{i,k}^{(t)}} \right),$$

where h_k is the canonical link that corresponds to cumulant function κ_k.

If we choose the null model for the mixture probabilities we receive MLEs

$$\widehat{p}_k^{(t)} = \frac{1}{n} \sum_{i=1}^n \widehat{Z}_{i,k}^{(t)} \qquad \text{for } 1 \le k \le K. \tag{6.31}$$

In Sect. 6.3.4, below, we will present an example that uses the null model for the mixture probabilities $\boldsymbol{p}$, and we present an other example that uses a logistic categorical GLM for these mixture probabilities.

Justification of the EM Algorithm So far, we have neither given any argument why the EM algorithm is reasonable for parameter estimation nor have we said anything about convergence. The purpose of this paragraph is to justify the above EM algorithm. We aim at solving the incomplete log-likelihood maximization problem, see (6.24),

$$(\widehat{\boldsymbol{\theta}}^{\text{MLE}}, \widehat{\boldsymbol{p}}^{\text{MLE}}) = \underset{(\boldsymbol{\theta}, \boldsymbol{p})}{\arg\max}\, \ell_{\boldsymbol{Y}}(\boldsymbol{\theta}, \boldsymbol{p}) = \underset{(\boldsymbol{\theta}, \boldsymbol{p})}{\arg\max} \sum_{i=1}^n \log \left(\sum_{k=1}^K p_k f_k(Y_i; \theta_k, v_i/\varphi_k) \right),$$

subject to existence and uniqueness. We introduce some notation. Let $f(y, z; \boldsymbol{\theta}, \boldsymbol{p}) = \exp\{\ell_{(y,z)}(\boldsymbol{\theta}, \boldsymbol{p})\}$ be the joint density of $(Y, \boldsymbol{Z})$ and let $f(y; \boldsymbol{\theta}, \boldsymbol{p}) =$

$\exp\{\ell_y(\boldsymbol{\theta}, \boldsymbol{p})\}$ be the marginal density of Y. This allows us to rewrite the incomplete log-likelihood as follows for any value of z

$$\ell_Y(\boldsymbol{\theta}, \boldsymbol{p}) = \log f(Y; \boldsymbol{\theta}, \boldsymbol{p}) = \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})}{f(z|Y; \boldsymbol{\theta}, \boldsymbol{p})}\right),$$

thus, we bring in the complete log-likelihood by using Bayes' rule. Choose an arbitrary categorical distribution $\pi \in \Delta_K$ with K levels. We have using the previous step

$$\begin{aligned}
\ell_Y(\boldsymbol{\theta}, \boldsymbol{p}) = \log f(Y; \boldsymbol{\theta}, \boldsymbol{p}) \; &= \; \sum_z \pi(z) \log f(Y; \boldsymbol{\theta}, \boldsymbol{p}) \\
&= \sum_z \pi(z) \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})/\pi(z)}{f(z|Y; \boldsymbol{\theta}, \boldsymbol{p})/\pi(z)}\right) \\
&= \sum_z \pi(z) \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})}{\pi(z)}\right) + \sum_z \pi(z) \log\left(\frac{\pi(z)}{f(z|Y; \boldsymbol{\theta}, \boldsymbol{p})}\right) \\
&= \sum_z \pi(z) \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})}{\pi(z)}\right) + D_{\mathrm{KL}}\left(\pi || f(\cdot|Y; \boldsymbol{\theta}, \boldsymbol{p})\right) \qquad (6.32) \\
&\geq \sum_z \pi(z) \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})}{\pi(z)}\right),
\end{aligned}$$

the inequality follows because the KL divergence is always non-negative, see Lemma 2.21. This provides us with a lower bound for the incomplete log-likelihood $\ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$ for any categorical distribution $\pi \in \Delta_K$ and any $(\boldsymbol{\theta}, \boldsymbol{p}) \in \boldsymbol{\Theta} \times \Delta_K$:

$$\begin{aligned}
\ell_Y(\boldsymbol{\theta}, \boldsymbol{p}) &\geq \sum_z \pi(z) \log\left(\frac{f(Y, z; \boldsymbol{\theta}, \boldsymbol{p})}{\pi(z)}\right) \qquad (6.33) \\
&= \mathbb{E}_{\boldsymbol{Z}\sim\pi}\left[\ell_{(Y,\boldsymbol{Z})}(\boldsymbol{\theta}, \boldsymbol{p}) \middle| Y\right] - \sum_z \pi(z) \log(\pi(z)) \overset{\text{def.}}{=} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{p}; \pi).
\end{aligned}$$

Thus, we have a lower bound $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{p}; \pi)$ on the incomplete log-likelihood $\ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$. This lower bound is based on the conditionally expected complete log-likelihood $\ell_{(Y,\boldsymbol{Z})}(\boldsymbol{\theta}, \boldsymbol{p})$, given Y, and under an arbitrary choice π for $\boldsymbol{Z}$. The difference between this arbitrary π and the true conditional posterior distribution is given by the KL divergence $D_{\mathrm{KL}}(\pi || f(\cdot|Y; \boldsymbol{\theta}, \boldsymbol{p}))$, see (6.32).

The general idea of the EM algorithm is to make this lower bound $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{p}; \pi)$ as large as possible in $\boldsymbol{\theta}$, $\boldsymbol{p}$ and π by iterating the following two alternating steps for $t \geq 1$:

$$\widehat{\pi}^{(t)} = \underset{\pi}{\arg\max}\ \mathcal{Q}\left(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}; \pi\right), \tag{6.34}$$

$$(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)}) = \underset{\boldsymbol{\theta}, \boldsymbol{p}}{\arg\max}\ \mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{p}; \widehat{\pi}^{(t)}\right). \tag{6.35}$$

The first step (6.34) can be solved explicitly and it results in the E-step. Namely, from (6.32) we see that maximizing $\mathcal{Q}(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}; \pi)$ in π is equivalent to minimizing the KL divergence $D_{\mathrm{KL}}(\pi || f(\cdot|Y; \widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}))$ in π because the left-hand side of (6.32) is independent of π. Thus, we have to solve

$$\widehat{\pi}^{(t)} = \underset{\pi}{\arg\max}\ \mathcal{Q}\left(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}; \pi\right) = \underset{\pi}{\arg\min}\ D_{\mathrm{KL}}\left(\pi \,\middle\|\, f(\cdot|Y; \widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)})\right).$$

This optimization is solved by choosing the density $\widehat{\pi}^{(t)} = f(\cdot|Y; \widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)})$, see Lemma 2.21, and this gives us exactly (6.28) if we calculate the corresponding conditional expectation of the latent variable $\boldsymbol{Z}$. Moreover, importantly, this step provides us with an identity in (6.33):

$$\ell_Y(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}) = \mathcal{Q}\left(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}; \widehat{\pi}^{(t)}\right). \tag{6.36}$$

The second step (6.35) then increases the right-hand side of (6.36). This second step is equivalent to

$$(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)}) = \underset{\boldsymbol{\theta}, \boldsymbol{p}}{\arg\max}\ \mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{p}; \widehat{\pi}^{(t)}\right) = \underset{\boldsymbol{\theta}, \boldsymbol{p}}{\arg\max}\ \mathbb{E}_{\boldsymbol{Z}\sim\widehat{\pi}^{(t)}}\left[\ell_{(Y,\boldsymbol{Z})}(\boldsymbol{\theta}, \boldsymbol{p}) \middle| Y\right], \tag{6.37}$$

and this maximization is solved by the solution of the score equations (6.29)–(6.30) of the M-step. In this step we explicitly use the linearity in $\boldsymbol{Z}$ of the log-likelihood $\ell_{(Y,\boldsymbol{Z})}$, which allows us to calculate the objective function in (6.37) explicitly resulting in replacing $\boldsymbol{Z}$ by $\widehat{\boldsymbol{Z}}^{(t)}$. For other incomplete data problems, where we do not have this linearity, this step will be more complicated.

Summarizing, alternating optimizations (6.34) and (6.35) gives us a sequence of parameters $(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)})_{t\geq 0}$ with monotonically increasing incomplete log-likelihoods

$$\ldots \leq \ell_Y(\widehat{\boldsymbol{\theta}}^{(t-1)}, \widehat{\boldsymbol{p}}^{(t-1)}) \leq \ell_Y(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)}) \leq \ell_Y(\widehat{\boldsymbol{\theta}}^{(t+1)}, \widehat{\boldsymbol{p}}^{(t+1)}) \leq \ldots. \tag{6.38}$$

Therefore, the EM algorithm converges supposed that the incomplete log-likelihood $\ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$ is a bounded function.

Remarks 6.12

- In general, the log-likelihood function $(\boldsymbol{\theta}, \boldsymbol{p}) \mapsto \ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$ does not need to be bounded. In that case the EM algorithm may not converge (unless it converges to a local maximum). An illustrative example is given in Example 6.13, below, which shows what can go wrong in MLE of mixture distributions.
- Even if the log-likelihood function $(\boldsymbol{\theta}, \boldsymbol{p}) \mapsto \ell_Y(\boldsymbol{\theta}, \boldsymbol{p})$ is bounded, one may not expect a unique solution to the parameter estimation problem with the EM algorithm. Firstly, a monotonically increasing sequence (6.38) only guarantees that we have convergence of that sequence. But the sequence may not converge to the global maximum and different starting points of the algorithm need to be explored. Secondly, convergence of sequence (6.38) does not necessarily imply that the parameters $(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)})$ converge for $t \to \infty$. On the one hand, we may have an identifiability issue because the components f_k of the mixture distribution may be exchangeable, and secondly one needs stronger conditions to ensure that not only the log-likelihoods converge but also their arguments (parameters) $(\widehat{\boldsymbol{\theta}}^{(t)}, \widehat{\boldsymbol{p}}^{(t)})$. This is the point studied in Wu [385].
- Even in very simple examples of mixture distributions we can have multiple local maximums. In this case the role of the starting point plays a crucial role. It is advantageous that in the starting configuration every component k shares roughly the same number of observations for the initial estimates $(\widehat{\boldsymbol{\theta}}^{(0)}, \widehat{\boldsymbol{p}}^{(0)})$ and $\widehat{\boldsymbol{Z}}^{(1)}$, otherwise one may start in a so-called spurious configuration where only a few observations almost fully determine a component k of the mixture distribution. This may result in similar singularities as in Example 6.13, below. Therefore, there are three common ways to determine a starting configuration of the EM algorithm, see Miljkovic–Grün [278]: (a) Euclidean distance-based initialization: cluster centers are selected at random, and all observations are allocated to these centers according to the shortest Euclidean distance; (b) K-means clustering allocation; or (c) completely random allocation to K bins. Using one of these three options, f_k and $\boldsymbol{p}$ are initialized.
- We have formulated the EM algorithm in the homogeneous situation. However, we can easily expand it to GLMs by, for instance, assuming that the canonical parameters θ_k are modeled by linear predictors $\langle \boldsymbol{\beta}_k, \boldsymbol{x} \rangle$ and/or likewise for the mixture probabilities $\boldsymbol{p}$. The E-step will not change in this setup. For the M-step, we will solve a different maximization problem, however, this maximization problem respects monotonicity (6.38), and therefore a modified version of the above EM algorithm applies. We emphasize that the crucial point is monotonicity (6.38) that makes the EM algorithm a valid procedure.

6.3.4 Lab: Mixture Distribution Applications

In this section we are going to present different mixture distribution examples that use the EM algorithm for parameter estimation. On the one hand this illustrates the functioning of the EM algorithm, and on the other hand it also highlights pitfalls that need to be avoided.

Example 6.13 (Gaussian Mixture) We directly fit a mixture model to the observation $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$. Assume that the log-likelihood of $\boldsymbol{Y}$ is given by a mixture of two Gaussian distributions

$$\ell_{\boldsymbol{Y}}(\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{p}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{2} p_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(Y_i - \theta_k)^2\right\}\right),$$

with $\boldsymbol{p} \in \Delta_2$, mean vector $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top \in \mathbb{R}^2$ and standard deviations $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)^\top \in \mathbb{R}_+^2$. Choose estimate $\widehat{\theta}_1 = Y_1$, then we have

$$\lim_{\sigma_1 \to 0} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{-\frac{1}{2\sigma_1^2}(Y_1 - \widehat{\theta}_1)^2\right\} = \lim_{\sigma_1 \to 0} \frac{1}{\sqrt{2\pi}\sigma_1} = \infty.$$

For any $i \neq 1$ we have $Y_i \neq \widehat{\theta}_1$ (note that the Gaussian distribution is absolutely continuous and observations are distinct, a.s.). Henceforth for $i \neq 1$

$$\lim_{\sigma_1 \to 0} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{-\frac{1}{2\sigma_1^2}(Y_i - \widehat{\theta}_1)^2\right\} = \lim_{\sigma_1 \to 0} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma_1^2}(Y_i - \widehat{\theta}_1)^2 - \log \sigma_1\right\} = 0.$$

If we choose any $\widehat{\theta}_2 \in \mathbb{R}$, $\boldsymbol{p} \in \Delta_2$ and $\sigma_2 > 0$, we receive for $\widehat{\theta}_1 = Y_1$

$$\begin{aligned}\lim_{\sigma_1 \to 0} \ell_{\boldsymbol{Y}}(\widehat{\boldsymbol{\theta}}, \boldsymbol{\sigma}, \boldsymbol{p}) = \lim_{\sigma_1 \to 0} \log\left(\sum_{k=1}^{2} p_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(Y_1 - \widehat{\theta}_k)^2\right\}\right) \\ + \sum_{i=2}^{n} \log\left(\frac{p_2}{\sqrt{2\pi}\sigma_2}\right) - \frac{1}{2\sigma_2^2}(Y_i - \widehat{\theta}_2)^2 = \infty.\end{aligned}$$

Thus, we can make the log-likelihood of this mixture Gaussian model arbitrarily large by fitting a degenerate Gaussian model to one observation in one mixture component, and letting the remaining observations be described by the other mixture component. This shows that the MLE problem may not be well-posed for mixture distributions because the log-likelihood can be unbounded.

If the data has well separated clusters, the log-likelihood of a mixture Gaussian distribution will have multiple local maximums. One can construct for any given number $B \in \mathbb{N}$ a data set $\boldsymbol{Y}$ such that the number of local maximums exceeds this number B, see Theorem 3 in Améndola et al. [11]. ■

Example 6.14 (Gamma Claim Size Modeling) In this example we consider claim size modeling of the French MTPL example given in Chap. 13.1. In view of Fig. 13.15 this seems quite difficult because we have three modes and heavy-tailedness. We choose a mixture of 5 distribution functions, we choose four gamma distributions and the Lomax distribution

$$Y \sim \sum_{k=1}^{4} \left(p_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} y^{\alpha_k - 1} \exp\{-\beta_k y\} \right) + p_5 \frac{\beta_5}{M} \left(\frac{y+M}{M} \right)^{-(\beta_5+1)}, \tag{6.39}$$

with shape parameters α_k and scale parameters β_k, $1 \le k \le 4$, for the gamma densities; scale parameter M and tail parameter β_5 for the Lomax density; and with mixture probability $\boldsymbol{p} \in \Delta_5$. The idea behind this choice is that three gamma distributions take care of the three modes of the empirical density, see Fig. 13.15, the fourth gamma distribution models the remaining claims in the body of the distribution, and the Lomax distribution takes care of the regularly varying tail of the data. For the gamma distribution, we refer to Sect. 2.1.3, and for the Lomax distribution, we refer to Sect. 2.2.5.

We choose the null model for both the mixture probabilities $\boldsymbol{p} \in \Delta_5$ and the densities f_k, $1 \le k \le 5$. This model can directly be fitted with the EM algorithm as presented above, in particular, we can estimate the mixture probabilities by (6.31). The remaining shape, scale and tail parameters are directly estimated by MLE. To initialize the EM algorithm we use the interpretation of the components as explained above. We partition the entire data into $K = 5$ bins according to their claim sizes Y_i being in $(0, 300]$, $(300, 1'000]$, $(1'000, 1'200]$, $(1'200, 5'000]$ or $(5'000, \infty)$. The first three intervals will initialize the three modes of the empirical density, see Fig. 13.15 (lhs). This will correspond to the categorical variable taking values $Z = 1, 2, 3$; the fourth interval will correspond to $Z = 4$ and it will model the main body of the claims; and the last interval will correspond to $Z = 5$, modeling the Lomax tail of the claims. These choices provide the initialization given in Table 6.1 with upper indices $^{(0)}$. We remark that we choose a fixed threshold of $M = 2'000$ for the Lomax distribution, this choice will be further discussed below.

Based on these choices we run the EM algorithm for mixture distributions. We observe convergence after roughly 80 iterations, and the resulting parameters after 100 iterations are presented in Table 6.1. We observe rather large shape parameters $\widehat{\alpha}_k^{(100)}$ for the first three components $k = 1, 2, 3$. This indicates that these three components model the three modes of the empirical density and these three modes collect almost $\widehat{p}_1^{(100)} + \widehat{p}_2^{(100)} + \widehat{p}_3^{(100)} \approx 50\%$ of all claims. The remaining claims are modeled by the gamma density $k = 4$ having mean 1'304 and by the Lomax distribution having tail parameter $\widehat{\beta}_5^{(100)} = 1.416$, thus, this tail has finite first moment $M/(\widehat{\beta}_5^{(100)} - 1) = 4'812$ and infinite second moment.

Table 6.1 Parameter choices in the mixture model (6.39)

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$\widehat{p}_k^{(0)}$	0.13	0.18	0.25	0.39	0.05
$\widehat{\alpha}_k^{(0)}$	2.43	11.24	1'299.44	5.63	–
$\widehat{\beta}_k^{(0)}$	0.019	0.018	1.141	0.003	0.517
$\widehat{\mu}_k^{(0)} = \widehat{\alpha}_k^{(0)}/\widehat{\beta}_k^{(0)}$	125	623	1'138	1'763	–
$\widehat{p}_k^{(100)}$	0.04	0.03	0.42	0.25	0.26
$\widehat{\alpha}_k^{(100)}$	93.05	650.94	1'040.37	1.34	–
$\widehat{\beta}_k^{(100)}$	1.207	1.108	0.888	0.001	1.416
$\widehat{\mu}_k^{(100)} = \widehat{\alpha}_k^{(100)}/\widehat{\beta}_k^{(100)}$	77	588	1'172	1'304	–

Figure 6.11 shows the resulting estimated mixture distribution. It gives the individual mixture components (top-lhs), the resulting mixture density (top-rhs), the QQ plot (bottom-lhs) and the log-log plot (bottom-rhs). Overall we find a rather good fit; maybe the first mode is a bit too spiky. However, this plot may also be misleading because the empirical density plot relies on kernel smoothing having a given bandwidth. Thus, the true observations may be more spiky than the plot indicates. The third mode suggests that there are two different values in the observations around 1'100, this is also visible in the QQ plot. Nevertheless, the overall result seems satisfactory. These results (based on 13 estimated parameters) are also summarized in Table 6.2.

We mention a couple of limitations of these results. Firstly, the log-likelihood of this mixture model is unbounded, similarly to Example 6.13 we can precisely fit one degenerate gamma mixture component to an individual observation Y_i which results in an infinite log-likelihood value. Thus, the found solution corresponds to a local maximum of the log-likelihood function and we should not state AIC values in Table 6.2, see also Remarks 4.28. Secondly, it is crucial to initialize three components to the three modes, if we randomly allocate all claims to 5 bins as initial configuration, the EM algorithm only finds mode $Z = 3$ but not necessarily the first two modes, at least, in our specifically chosen random initialization this was the case. In fact, the likelihood value of our latter solution was worse than in the first calibration which shows that we ended up in a worse local maximum.

We may be tempted to also estimate the Lomax threshold M with MLE. In Fig. 6.12 we plot the maximal log-likelihood as a function of M (if we start the EM algorithm always in the same configuration given in Table 6.1). From this figure a threshold of $M = 1'600$ seems optimal. Choosing this threshold of $M = 1'600$ leads to a slightly bigger log-likelihood of -199'304 and a slightly smaller tail parameter of $\widehat{\beta}_5^{(100)} = 1.318$. However, overall the model is very similar to the one with $M = 2'000$. In general, we do *not* recommend to estimate M with MLE, but this should be treated as a hyper-parameter selected by the modeler. The reason for this recommendation is that this threshold is crucial in deciding for large claims modeling and its estimation from data is, typically, not very robust; we also refer to Remarks 6.15, below.

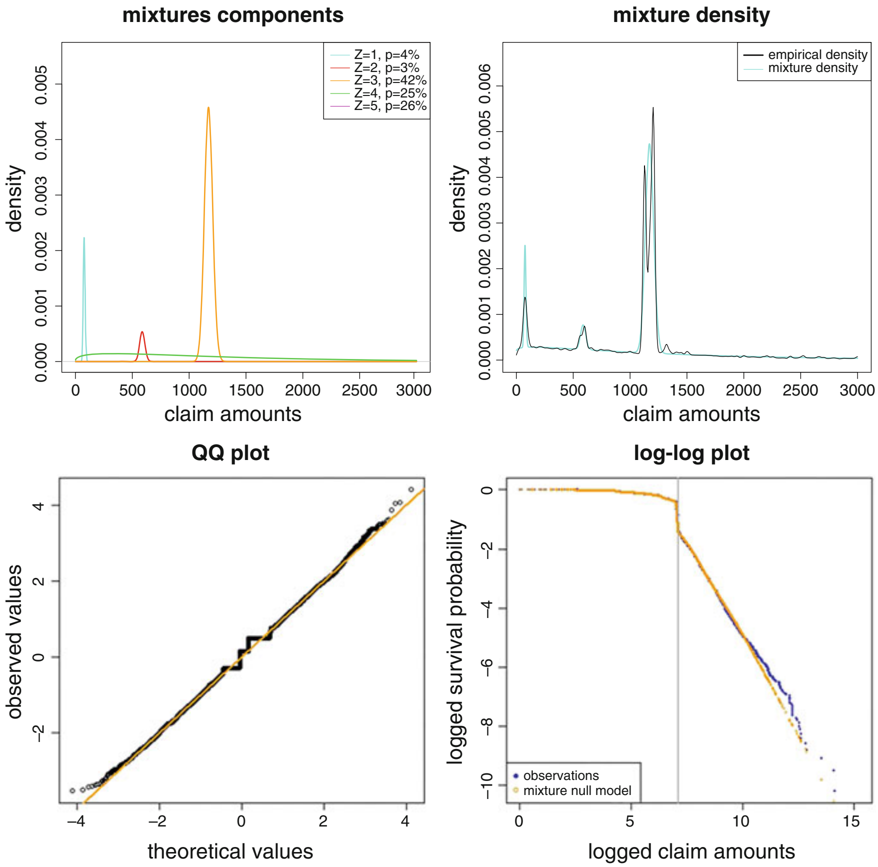


Fig. 6.11 Mixture null model: (top-lhs) individual estimated gamma components $f_k(\cdot; \widehat{\alpha}_k^{(100)}, \widehat{\beta}_k^{(100)})$, $1 \le k \le K$, and Lomax component $f_5(\cdot; \widehat{\beta}_5^{(100)})$, (top-rhs) estimated mixture density $\sum_{k=1}^{4} \widehat{p}_k^{(100)} f_k(\cdot; \widehat{\alpha}_k^{(100)}, \widehat{\beta}_k^{(100)}) + \widehat{p}_5^{(100)} f_5(\cdot; \widehat{\beta}_5^{(100)})$, (bottom-lhs) QQ plot of the estimated model, (bottom-rhs) log-log plot of the estimated model

Table 6.2 Mixture models for French MTPL claim size modeling

	# Param.	$\ell_{\boldsymbol{Y}}(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{p}})$	AIC	$\widehat{\mu} = \mathbb{E}_{\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{p}}}[Y]$
Empirical				2’266
Null model ($M = 2000$)	13	$-$199’306	398’637	2’381
Logistic GLM ($M = 2000$)	193	$-$198’404	397’193	2’176

In a next step we enhance the mixture modeling by including feature information $\boldsymbol{x}_i$ to explain the responses Y_i. In view of Fig. 13.17 we have decided to only model the mixture probabilities $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{x})$ feature dependent because feature information seems to mainly influence the heights of the peaks. We do not consider features `VehPower` and `VehGas` because these features do not seem to contribute, and

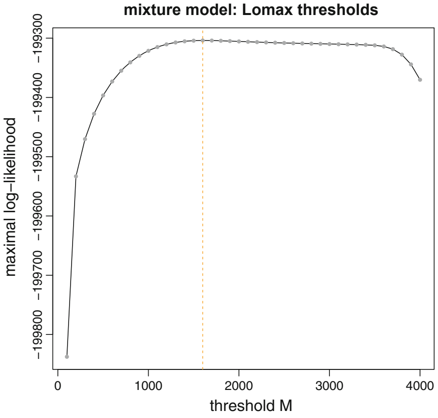


Fig. 6.12 Choice of Lomax threshold M

we do not consider `Density` because of the high co-linearity with `Area`, see Fig. 13.12 (rhs). Thus, we are left with the features `Area`, `VehAge`, `DrivAge`, `BonusMalus`, `VehBrand` and `Region`. Pre-processing of these features is done as in Listing 5.1, except that we keep `Area` categorical. Using these features $\boldsymbol{x} \in \mathcal{X} \subset \{1\} \times \mathbb{R}^q$ we choose a logistic categorical GLM for the mixture probabilities

$$\boldsymbol{x} \mapsto (p_1(\boldsymbol{x}), \ldots, p_{K-1}(\boldsymbol{x}))^\top = \frac{\exp\{\boldsymbol{X}\boldsymbol{\gamma}\}}{1 + \sum_{l=1}^{4} \exp\langle \boldsymbol{\gamma}_l, \boldsymbol{x}\rangle}, \tag{6.40}$$

that is, we choose $K = 5$ as reference level, feature matrix $\boldsymbol{X} \in \mathbb{R}^{(K-1)\times(K-1)(q+1)}$ is defined in (5.71), and with regression parameter $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1^\top, \ldots, \boldsymbol{\gamma}_{K-1}^\top)^\top \in \mathbb{R}^{(K-1)(q+1)}$; this regression parameter $\boldsymbol{\gamma}$ should not be confused with the shape parameters $\beta_1, \ldots, \beta_4$ of the gamma components and the tail parameter β_5 of the Lomax component, see (6.39). Note that the notation in this section slightly differs from Sect. 5.7 on the logistic categorical GLM. In this section we consider mixture probabilities $\boldsymbol{p}(\boldsymbol{x}) \in \Delta_{K=5}$ (which corresponds to one-hot encoding), whereas in Sect. 5.7 we model $(p_1(\boldsymbol{x}), \ldots, p_{K-1}(\boldsymbol{x}))^\top$ with a categorical GLM (which corresponds to dummy coding), and normalization provides us with $p_K(\boldsymbol{x}) = 1 - \sum_{l=1}^{K-1} p_l(\boldsymbol{x}) \in (0, 1)$.

This logistic categorical GLM requires that we replace in the M-step the probability estimation (6.31) by Fisher's scoring method for GLMs as outlined in Sect. 5.7.2, but there is a small difference to that section. In the working residuals (5.74) we use dummy coding $T(Z) \in \{0, 1\}^{K-1}$ of a categorical variable Z, this now needs to be replaced by the estimated vector $(\widehat{Z}_1(\boldsymbol{\theta}, \boldsymbol{p}|Y), \ldots, \widehat{Z}_{K-1}(\boldsymbol{\theta}, \boldsymbol{p}|Y))^\top \in (0, 1)^{K-1}$ which is used as an estimate for the latent variable $T(Z)$. Apart from that everything is done as described in Sect. 5.7.2; in R this can be done with the procedure `multinom` from the package `nnet` [368]. We start the EM algorithm exactly in the final configuration of the

Table 6.3 Parameter choices in the mixture models: upper part null model, lower part GLM for estimated mixture probabilities $\widehat{\boldsymbol{p}}(\boldsymbol{x}_i)$

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Null: $\widehat{p}_k^{(100)}$	0.04	0.03	0.42	0.25	0.26
Null: $\widehat{\alpha}_k^{(100)}$	93.05	650.94	1’040.37	1.34	–
Null: $\widehat{\beta}_k^{(100)}$	1.207	1.108	0.888	0.001	1.416
Null: $\widehat{\mu}_k^{(100)} = \widehat{\alpha}_k^{(100)}/\widehat{\beta}_k^{(100)}$	77	588	1’172	1’304	–
GLM: average mixture probabilities	0.04	0.03	0.42	0.25	0.26
GLM: $\widehat{\alpha}_k^{(100)}$	94.03	597.20	1’043.38	1.28	–
GLM: $\widehat{\beta}_k^{(100)}$	1.223	1.019	0.891	0.001	1.365
GLM: $\widehat{\mu}_k^{(100)} = \widehat{\alpha}_k^{(100)}/\widehat{\beta}_k^{(100)}$	77	586	1’172	1’268	–

estimated mixture null model, and we run this algorithm for 20 iterations (which provides convergences).

The resulting parameters are given in the lower part of Table 6.3. We observe that the resulting parameters remain essentially the same, the second mode $Z = 2$ is a bit less spiky, and the tail parameter is slightly smaller. The summary of this model is given on the last line of Table 6.2. Regression modeling adds another $4 \cdot 45 = 180$ parameters to the model because we have $q = 45$ feature components in $\boldsymbol{x}$ (different from the intercept component). In view of AIC we give preference to the logistic mixture probability case (though AIC has to be interpreted with care, here, because we do not consider the MLE but rather a local maximum).

Figure 6.13 plots the individual estimated mixture probabilities $\boldsymbol{x}_i \mapsto \widehat{\boldsymbol{p}}(\boldsymbol{x}_i) \in \Delta_5$ over the insurance policies $1 \le i \le n$; these plots are inspired by the thesis of Frei [138]. The upper plots consider these probabilities against the estimated claim sizes $\widehat{\mu}(\boldsymbol{x}_i) = \sum_{k=1}^{5} \widehat{p}_k(\boldsymbol{x}_i)\widehat{\mu}_k$ and the lower plots against the ranks of $\widehat{\mu}(\boldsymbol{x}_i)$, the latter gives a different scaling on the x-axis because of the heavy-tailedness of the claims. The plots on the left-hand side show all individual policies $1 \le i \le n$, and the plots on the right-hand side show a quadratic spline fit to these observations. Not surprisingly, we observe that the claim size estimate $\widehat{\mu}(\boldsymbol{x}_i)$ is mainly driven by the large claims probability $\widehat{p}_5(\boldsymbol{x}_i)$ describing the Lomax contribution.

In Fig. 6.14 we compare the QQ plots of the mixture null model and the one where we model the mixture probabilities with the logistic categorical GLM. We see that the latter (more complex) model clearly outperforms the more simple one, in fact, this QQ plot looks quite convincing for the French MTPL claim size data. Finally, we perform a Wald test (5.32). We simultaneously treat all parameters that belong to the same feature variable (similar to the ANOVA analysis); for instance, for the 22 `Regions` the corresponding part of the regression parameter $\boldsymbol{\gamma}$ contains $4 \cdot 21 = 84$ components. The resulting p-values of dropping such components are all close to 0 which says that we should not eliminate one of the feature variables. This closes the example. ■

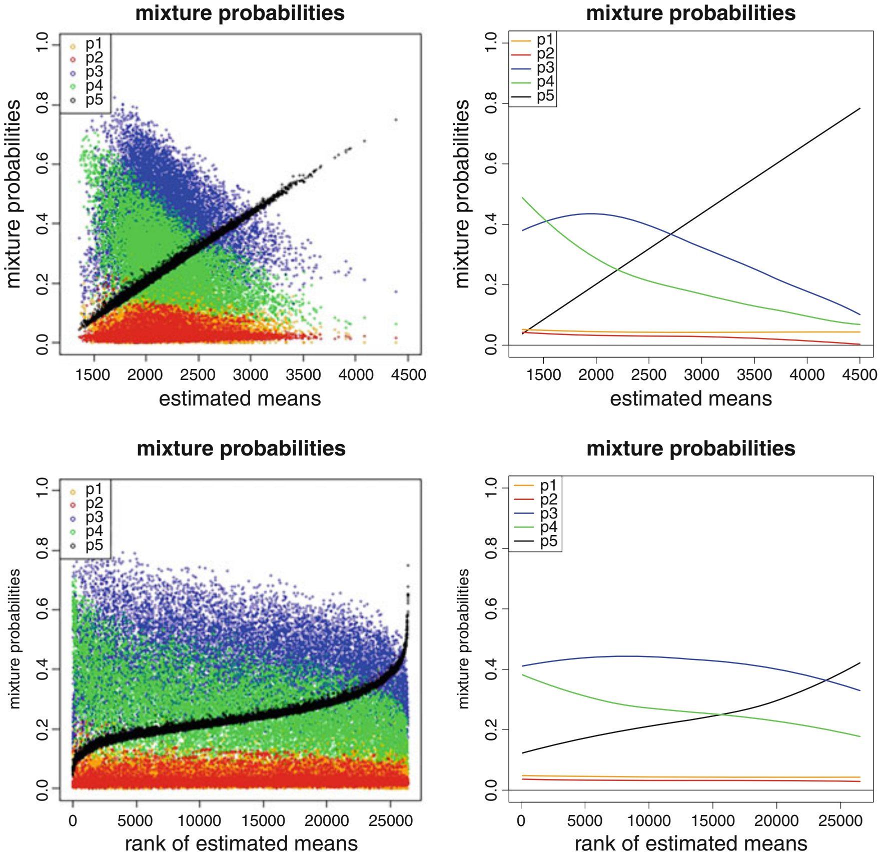


Fig. 6.13 Mixture probabilities $\boldsymbol{x}_i \mapsto \widehat{\boldsymbol{p}}(\boldsymbol{x}_i)$ on individual policies $1 \le i \le n$: (top) against the estimated means $\widehat{\mu}(\boldsymbol{x}_i)$ and (bottom) against the ranks of the estimated means $\widehat{\mu}(\boldsymbol{x}_i)$; (lhs) over policies $1 \le i \le n$ and (rhs) quadratic spline fit

Remarks 6.15

- In Example 6.14 we have chosen a mixture distribution with four gamma components and one Lomax component. The reason for choosing the Lomax component has been two-fold. Firstly, we need a regularly varying tail to model the heavy-tailed property of the data. Secondly, we have preferred the Lomax distribution over the Pareto distribution because this provides us with a continuous density in (6.39). The results in Example 6.14 have been satisfactory. In most practical approaches, however, this approach will not work, even when fixing the threshold M of the Lomax component. Often, the nature of the data is such that the chosen gamma mixture distribution is not able to fully explain the small data in the body of the distribution, and in that situation the Lomax tail will assist in fitting the small claims. The typical result is that the Lomax part

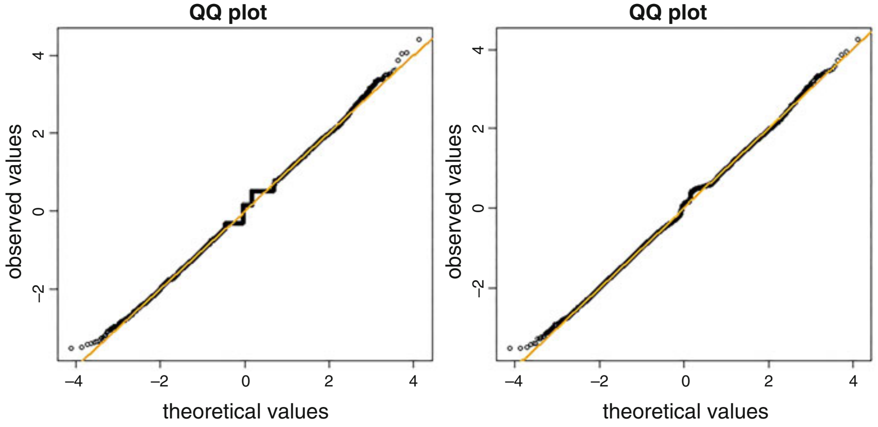


Fig. 6.14 QQ plots of the mixture models: (lhs) null model and (rhs) logistic categorical GLM for mixture probabilities

then pays more attention to small claims (through the log-likelihood function of numerous small claims) and the fitting of the tail turns out to be poor (because a few large claims do not sufficiently contribute to the log-likelihood). There are two ways to solve this dilemma. Either one works with composite distributions, see (6.56) below, and one drops the continuity property of the density; this is the approach taken in Fung et al. [148]. Or one fits the Lomax distribution solely to large observations in a first step, and then fixes the parameters of the Lomax distribution during the second step when fitting the full model to all data, this is the approach taken in Frei [138]. Both of these two approaches have been providing good results on real insurance data.

- There is an asymptotic theory for the optimal selection of the number of mixture components, we refer to Khalili–Chen [214] and Khalili [213]. Fung et al. [148] combine this asymptotic theory of mixture component selection with feature selection within these mixture components using LASSO and SCAD regularization.
- In Example 6.14 we have only modeled the mixture probabilities feature dependent, but not the parameters of the gamma mixture components. Introducing regressions for the gamma mixture components needs some care in fitting. For policy independent shape parameters $\alpha_1, \ldots, \alpha_4$, we can estimate the regression functions for the means of the mixture components without explicitly specifying α_k because these shape parameters cancel in the score equations. However, these shape parameters will be needed in the E-step, which requires also MLE of α_k. For more discussion on shape parameter estimation we refer to Sect. 5.3.7 (GLM with constant shape parameter) and Sect. 5.5.4 (double GLM).

6.4 Truncated and Censored Data

6.4.1 Lower-Truncation and Right-Censoring

A common problem in insurance is that we often have truncated or censored observations. Truncation naturally occurs if we sell insurance products that have a deductible $d > 0$ because in that case only the insurance claim $(Y - d)_+$ is compensated, and claims below the deductible d are usually not reported to the insurance company. This case is called *lower-truncation*, because claims below the deductible are not observed. If we lower-truncate an original claim $Y \sim f(\cdot; \theta)$ with lower-truncation point $\tau \in \mathbb{R}$ we obtain the density

$$f_{(\tau,\infty)}(y; \theta) = \frac{f(y; \theta)\mathbb{1}_{\{y>\tau\}}}{1 - F(\tau, \theta)}, \tag{6.41}$$

if $F(\cdot; \theta)$ is the distribution function corresponding to the density $f(\cdot; \theta)$. The lower-truncated density $f_{(\tau,\infty)}(y; \theta)$ only considers claims that fall into the interval (τ, ∞). Obviously, we can define upper-truncation completely analogously by considering an interval $(-\infty, \tau]$ instead. Figure 6.15 (lhs) gives an example of a lower-truncated density, and Fig. 6.15 (rhs) gives an example of a lower- and upper-truncated density.

Censoring occurs by selling insurance products with a maximal cover $M > 0$ because in that case only the insurance claim $Y \wedge M = \min\{Y, M\}$ is compensated, and the exact claim size above the maximal cover M may not be available. This case is called *right-censoring* because the exact claim amount above M is not known. Right-censoring of an original claim $Y \sim F(\cdot; \theta)$ with censoring point $M \in \mathbb{R}$

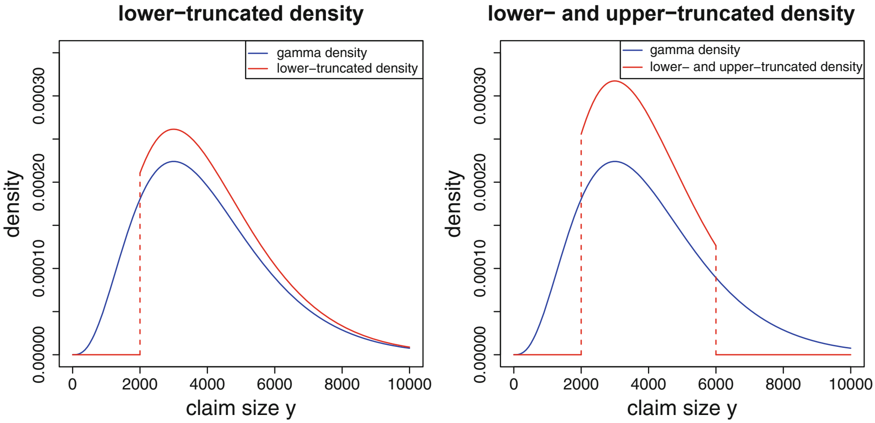


Fig. 6.15 (lhs) Lower-truncated gamma density with $\tau = 2'000$, and (rhs) lower- and upper-truncated gamma density with truncation points $2'000$ and $6'000$

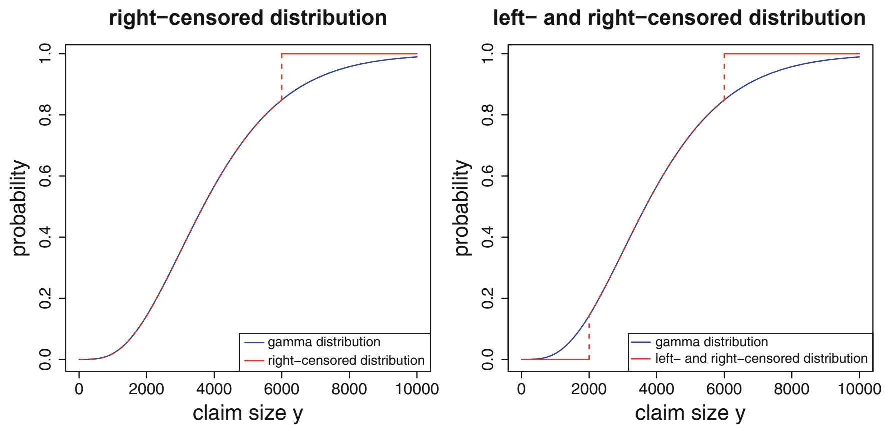


Fig. 6.16 (lhs) Right-censored gamma distribution with $M = 6'000$, and (rhs) left- and right-censored gamma distribution with censoring points $2'000$ and $6'000$

gives the distribution

$$F_{Y \wedge M}(y; \theta) = F(y; \theta)\mathbb{1}_{\{y<M\}} + \mathbb{1}_{\{y \geq M\}},$$

that is, we have a point mass in the censoring point M. We can define left-censoring analogously by considering the claim $Y \vee M = \max\{Y, M\}$. Figure 6.16 (lhs) shows a right-censored gamma distribution with censoring point $M = 6'000$, and Fig. 6.16 (rhs) shows a left- and right-censored example with censoring points $2'000$ and $6'000$.

Often in re-insurance, deductibles (also called retention levels) and maximal covers are combined, for instance, an excess-of-loss (XL) insurance cover of size $u > 0$ above the retention level $d > 0$ covers the claim

$$(Y - d)_+ \wedge u = (Y - d)\mathbb{1}_{\{d \leq Y < d+u\}} + u\mathbb{1}_{\{Y \geq d+u\}} = (Y - d)_+ - (Y - (d + u))_+.$$

Obviously, truncation and censoring pose some challenges in regression modeling because at the same time we need to consider the density $f(\cdot; \theta)$ and the distribution function $F(\cdot; \theta)$ to estimate a parameter θ. Both cases can be understood as missing data problems, with censoring providing the number of claims but not necessarily the exact claim size, and with truncation leaving also the number of claims unknown. These two cases are studied in Fung et al. [147] within the mixture of experts models using a variant of the EM algorithm. We use their techniques within the EDF framework for right-censored or lower-truncated data. This is done in the next sections.

6.4.2 Parameter Estimation Under Right-Censoring

Assume we have a fixed censoring point $M > 0$ that applies to independent observations Y_i following EDF densities $f(\cdot; \theta_i, v_i/\varphi)$; for simplicity we assume to work with an absolutely continuous EDF in this section. The (incomplete) log-likelihood function of canonical parameters $\boldsymbol{\theta} = (\theta_i)_{1\le i\le n}$ for observations $\boldsymbol{Y} \wedge M$ is given by

$$\ell_{\boldsymbol{Y}\wedge M}(\boldsymbol{\theta}) = \sum_{i:\, Y_i<M} \log f(Y_i; \theta_i, v_i/\varphi) + \sum_{i:\, Y_i\wedge M=M} \log\left(1 - F(M; \theta_i, v_i/\varphi)\right). \tag{6.42}$$

We interpret this as an incomplete data problem because the claim sizes Y_i above the censoring point M are not known. The complete log-likelihood is given by

$$\ell_{\boldsymbol{Y}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(Y_i; \theta_i, v_i/\varphi).$$

Similarly to (6.32) we calculate a lower bound to the incomplete log-likelihood. We focus on one component of $\boldsymbol{Y}$ and drop the lower index i in Y_i for this consideration. Firstly, if $Y \wedge M < M$ we are in the situation of full claim size information and, obviously, we have log-likelihood in that case $Y < M$

$$\ell_{Y\wedge M}(\theta) = \ell_Y(\theta) = \frac{Y\theta - \kappa(\theta)}{\varphi/v} + a(Y; v/\varphi). \tag{6.43}$$

In the second case $Y \wedge M = M$ we do not have precise claim size information. In that case we have conditional density of claim $Y|_{\{Y\wedge M=M\}} = Y|_{\{Y\ge M\}}$ above M

$$f(z|Y \ge M; \theta, v/\varphi) = \frac{f(z; \theta, v/\varphi)\mathbb{1}_{\{z\ge M\}}}{1 - F(M; \theta, v/\varphi)} = \frac{f(z; \theta, v/\varphi)\mathbb{1}_{\{z\ge M\}}}{\exp\{\ell_{Y\wedge M}(\theta)\}}, \tag{6.44}$$

the latter follows because $Y \wedge M = M$ has the corresponding point mass in censoring point M (we work with an absolutely continuous EDF here). Choose an arbitrary density π having the same support as $Y|_{\{Y\ge M\}}$, and consider a random variable $Z \sim \pi$. Using (6.44) and the EDF structure on the last line, we have for $Y \ge M$

$$\begin{aligned}
\ell_{Y\wedge M}(\theta) &= \int \pi(z)\, \ell_{Y\wedge M}(\theta)\, d\nu(z) \\
&= \int \pi(z) \log\left(\frac{f(z; \theta, v/\varphi)/\pi(z)}{f(z|Y \ge M; \theta, v/\varphi)/\pi(z)}\right) d\nu(z) \\
&= \int \pi(z) \log\left(\frac{f(z; \theta, v/\varphi)}{\pi(z)}\right) d\nu(z) + D_{\mathrm{KL}}\left(\pi || f(\cdot|Y \ge M; \theta, v/\varphi)\right)
\end{aligned}$$

$$
\begin{aligned}
&\geq \int \pi(z) \log\left(\frac{f(z;\theta,v/\varphi)}{\pi(z)}\right) d\nu(z) \\
&= \frac{\mathbb{E}_\pi\left[Z\right]\theta - \kappa(\theta)}{\varphi/v} + \mathbb{E}_\pi\left[a(Z;v/\varphi)\right] - \mathbb{E}_\pi\left[\log \pi(Z)\right] \overset{\text{def.}}{=} \mathcal{Q}(\theta;\pi).
\end{aligned}
$$

This allows us to explore the E-step and the M-step similarly to (6.34) and (6.35).

The **E-step** in the case $Y \geq M$ for given canonical parameter estimate $\widehat{\theta}^{(t-1)}$ reads as

$$
\begin{aligned}
\widehat{\pi}^{(t)} &= \underset{\pi}{\arg\max}\, \mathcal{Q}\left(\widehat{\theta}^{(t-1)};\pi\right) = \underset{\pi}{\arg\min}\, D_{\mathrm{KL}}\left(\pi \,\middle\|\, f(\cdot|Y \geq M;\widehat{\theta}^{(t-1)}, v/\varphi)\right) \\
&= f(\cdot|Y \geq M;\widehat{\theta}^{(t-1)}, v/\varphi).
\end{aligned}
$$

This allows us to calculate the estimation of the claim size above M, i.e., under $\widehat{\pi}^{(t)}$

$$
\widehat{Y}^{(t)} = \mathbb{E}_{\widehat{\pi}^{(t)}}\left[Z\right] = \int z\, f(z|Y \geq M;\widehat{\theta}^{(t-1)}, v/\varphi)\, d\nu(z). \tag{6.45}
$$

Note that this is an estimate of the censored claim $Y|_{\{Y \geq M\}}$. This completes the E-step.

The **M-step** considers in the EDF case for censored claim sizes $Y \geq M$

$$
\begin{aligned}
\widehat{\theta}^{(t)} &= \underset{\theta}{\arg\max}\, \mathcal{Q}\left(\theta;\widehat{\pi}^{(t)}\right) = \underset{\theta}{\arg\max}\, \frac{\mathbb{E}_{\widehat{\pi}^{(t)}}\left[Z\right]\theta - \kappa(\theta)}{\varphi/v} \\
&= \underset{\theta}{\arg\max}\, \ell_{\widehat{Y}^{(t)}}(\theta),
\end{aligned} \tag{6.46}
$$

the latter uses that the normalizing term $a(\cdot;v/\varphi)$ is not relevant for the MLE of θ. That is, (6.46) describes the regular MLE step under the observation $\widehat{Y}^{(t)}$ in the case of a censored observation $Y \geq M$; and if $Y < M$ we simply use the log-likelihood (6.43).

EM algorithm for right-censored data within the EDF

(0) Choose an initial parameter $\widehat{\boldsymbol{\theta}}^{(0)} = (\widehat{\theta}_i^{(0)})_{1 \leq i \leq n}$.

(1) Repeat for $t \geq 1$:

- **E-step.** Given parameter $\widehat{\boldsymbol{\theta}}^{(t-1)} = (\widehat{\theta}_i^{(t-1)})_{1 \leq i \leq n}$, estimate for the right-censored claims $Y_i \geq M$ their sizes by, see (6.45),

$$
\widehat{Y}_i^{(t)} = \int z\, f\left(z \,\middle|\, Y_i \geq M;\widehat{\theta}_i^{(t-1)}, v_i/\varphi\right) d\nu(z).
$$

This provides us with an estimated observation

$$\widehat{\boldsymbol{Y}}^{(t)} = \left(Y_i \mathbb{1}_{\{Y_i < M\}} + \widehat{Y}_i^{(t)} \mathbb{1}_{\{Y_i \geq M\}} \right)^{\top}_{1 \leq i \leq n}.$$

- **M-step.** Calculate the MLE $\widehat{\boldsymbol{\theta}}^{(t)} = (\widehat{\theta}_i^{(t)})_{1 \leq i \leq n}$ based on observation $\widehat{\boldsymbol{Y}}^{(t)}$, i.e., solve

$$\widehat{\boldsymbol{\theta}}^{(t)} = \underset{\boldsymbol{\theta}}{\arg\max}\; \ell_{\widehat{\boldsymbol{Y}}^{(t)}}(\boldsymbol{\theta}).$$

Note that the above EM algorithm uses that the log-likelihood $\ell_{\boldsymbol{Y}}(\boldsymbol{\theta})$ of the EDF is linear in the observations that interact with parameter $\boldsymbol{\theta}$. We revisit the gamma claim size example of Sect. 5.3.7.

Example 6.16 (Right-Censored Gamma Claim Sizes) We revisit the gamma claim size GLM introduced in Sect. 5.3.7. The claim sizes are illustrated in Fig. 13.22. In total we have $n = 656$ observations Y_i, and they range from 16 SEK to 211'254 SEK. We right-censor this data at $M = 50'000$, this results in 545 uncensored observations and 111 censored observations equal to M. Thus, for the 17% largest claims we assume to not have any knowledge about the exact claim sizes. We use the EM algorithm for right-censored data to fit a GLM to this problem.

In order to calculate the E-step we need to evaluate the conditional expectation (6.45) under the gamma model

$$\begin{aligned}\widehat{Y}^{(t)} &= \int z\, f(z|Y \geq M; \widehat{\theta}^{(t-1)}, v/\varphi)\, d\nu(z) \qquad (6.47)\\ &= \int_M^\infty z\, \frac{\frac{\beta^\alpha}{\Gamma(\alpha)} z^{\alpha-1} \exp\{-\beta z\}}{1 - \mathcal{G}(\alpha, \beta M)}\, dz \;=\; \frac{\alpha}{\beta} \frac{1 - \mathcal{G}(\alpha+1, \beta M)}{1 - \mathcal{G}(\alpha, \beta M)},\end{aligned}$$

with shape parameter $\alpha = v/\varphi$, scale parameter $\beta = -\widehat{\theta}^{(t-1)} v/\varphi$, see (5.45), and scaled incomplete gamma function

$$\mathcal{G}(\alpha, y) = \frac{1}{\Gamma(\alpha)} \int_0^y z^{\alpha-1} \exp\{-z\}\, dz \;\in\; (0,1) \qquad \text{for } y \in (0, \infty). \qquad (6.48)$$

Thus, we receive a simple formula that allows us to efficiently calculate the E-step, and the M-step is exactly the gamma GLM explained in Sect. 5.3.7 for the (estimated) data $\widehat{\boldsymbol{Y}}^{(t)}$.

For the modeling we choose exactly the features as used for model Gamma GLM2, this gives $q + 1 = 7$ regression parameter components and additionally we set for the dispersion parameter $\widehat{\varphi}^{\text{MLE}} = 1.427$, this is the MLE in model Gamma

Table 6.4 Comparison of the complete log-likelihood and the incomplete log-likelihood (right-censoring $M = 50'000$) results

	# Param.	Log-likelihood $\ell_{\boldsymbol{Y}}(\widehat{\theta}^{\mathrm{MLE}}, \widehat{\varphi}^{\mathrm{MLE}})$	Dispersion est. $\widehat{\varphi}^{\mathrm{MLE}}$	Average amount	Rel. change
Gamma GLM2 (complete data)	$7+1$	$-7'129$	1.427	25'130	
Crude GLM2 (right-censored)	$7+1$	$-7'158$		18'068	-28%
EM est. GLM2 (right-censored)	$7+1$	$-7'132$		26'687	$+6\%$

GLM2. This dispersion parameter we keep fixed in all our models studied in this example. In a first step we simply fit a gamma GLM to the right-censored data $Y_i \wedge M$. We call this model 'crude GLM2', and it underestimates the empirical claim sizes by 28% because it ignores the fact of having right-censored data.

To initialize the EM algorithm for right-censored data we use the model crude GLM2. We then iterate the algorithm for 15 steps which provides convergence. The results are presented in Table 6.4. We observe that the resulting log-likelihood of the model fitted on the censored data and evaluated on the complete data $\ell_{\boldsymbol{Y}}$ (which is available here) is almost the same as for model Gamma GLM2, which has been estimated on the complete data. Moreover, this right-censored EM algorithm fitted model slightly over-estimates the average claim sizes.

Figure 6.17 shows the estimated means $\widehat{\mu}_i$ on an individual claims level. The x-axis always gives the estimates from the complete log-likelihood model Gamma GLM2. The y-axis on the left-hand side shows the estimates from the crude GLM and the right-hand side the estimates from the EM algorithm fitted counterpart (fitted on the right-censored data). We observe that the crude model underestimates the claims (being below the diagonal), and the largest estimate lies below $M = 50'000$

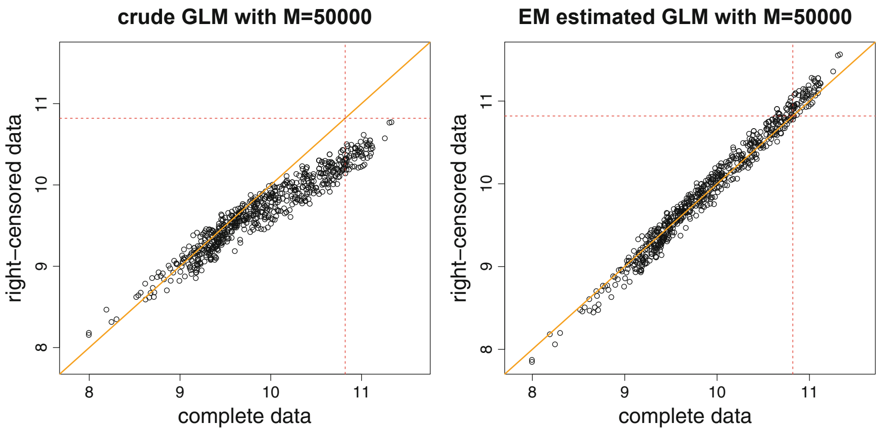


Fig. 6.17 Comparison of the estimated means $\widehat{\mu}_i$ in model Gamma GLM2 against (lhs) the crude GLM and (rhs) the EM fitted right-censored model; both axis are on the log-scale, the dotted lines shows the censoring point $\log(M)$

in our example (horizontal dotted line). The EM algorithm fitted model, considering the fact that we have right-censored data, corrects for the censoring, and the resulting estimates resemble the ones from the complete log-likelihood model quite well. In fact, we probably slightly over-estimate under right-censoring, here. Note that all these considerations have been done under an identical dispersion parameter estimate $\widehat{\varphi}^{\text{MLE}}$. For the complete log-likelihood case, this is not really needed for mean estimation because it cancels in the score equations for mean estimation. However, a reasonable dispersion parameter estimate is crucial for the incomplete case as it enters $\widehat{Y}^{(t)}$ in the E-step, see (6.47), thus, the caveat here is that we need a reasonable dispersion estimate from the right-censored data (which we did not discuss, here, and which requires further research). ■

6.4.3 Parameter Estimation Under Lower-Truncation

Compared to censoring we have less information under truncation because not only the claim sizes below the lower-truncation point are unknown, but we also do not know how many claims there are below that truncation point τ. Assume we work with responses belonging to the EDF. The incomplete log-likelihood is given by

$$\ell_{\boldsymbol{Y}>\tau}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(Y_i; \theta_i, v_i/\varphi) - \log\left(1 - F(\tau; \theta_i, v_i/\varphi)\right),$$

assuming that $\boldsymbol{Y} = (Y_i)_{1\le i\le n} > \tau$ collects all claims above the truncation point $Y_i > \tau$, see (6.41). We proceed as in Fung et al. [147] to construct a complete log-likelihood; there are different ways to do so, but this proposal is convenient for parameter estimation. Firstly, we equip each observed claim $Y_i > \tau$ with an independent count random variable $K_i \sim p(\cdot; \theta_i, v_i/\varphi)$ that determines the number of claims below the truncation point that correspond to claim i above the truncation point. Secondly, we assume that these claims are given by independent observations $Z_{i,1}, \ldots, Z_{i,K_i} \le \tau$, a.s., with a distribution obtained from an un-truncated version of Y_i, i.e., we consider the upper-truncated version of $f(\cdot; \theta_i, v_i/\varphi)$ for $Z_{i,j}$. This gives us the complete log-likelihood

$$\ell_{(\boldsymbol{Y},\boldsymbol{K},\boldsymbol{Z})}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \Bigg(\log\left(\frac{f(Y_i; \theta_i, v_i/\varphi)}{1 - F(\tau; \theta_i, v_i/\varphi)}\right) \tag{6.49}$$

$$+ \log p(K_i; \theta_i, v_i/\varphi) + \sum_{j=1}^{K_i} \log\left(\frac{f(Z_{i,j}; \theta_i, v_i/\varphi)}{F(\tau; \theta_i, v_i/\varphi)}\right)\Bigg),$$

with $\boldsymbol{K} = (K_i)_{1 \le i \le n}$, and $\boldsymbol{Z}$ collects all (latent) claims $Z_{i,j} \le \tau$, an empty sum is set equal to zero. Next, we assume that K_i is following the geometric distribution

$$\mathbb{P}_{\theta_i}[K_i = k] \; = \; p(k; \theta_i, v_i/\varphi) \; = \; F(\tau; \theta_i, v_i/\varphi)^k \left(1 - F(\tau; \theta_i, v_i/\varphi)\right). \tag{6.50}$$

As emphasized in Fung et al. [147], this complete log-likelihood is an artificial construct that supports parameter estimation of lower-truncated data. It does *not* claim that the true un-truncated data follows this model (6.49) but it provides a distributional extension below the truncation point $\tau > 0$ that is convenient for parameter estimation. Namely, inserting this geometric distribution assumption into (6.49) gives us complete log-likelihood

$$\ell_{(\boldsymbol{Y},\boldsymbol{K},\boldsymbol{Z})}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\log f(Y_i; \theta_i, v_i/\varphi) + \sum_{j=1}^{K_i} \log f(Z_{i,j}; \theta_i, v_i/\varphi) \right). \tag{6.51}$$

Within the EDF this allows us to do the same EM algorithm considerations as above; note that this expression no longer involves the distribution function. We consider one observation $Y_i > \tau$ and we drop the lower index i. This gives us complete observation $(Y, K, Z = (Z_j)_{1 \le j \le K})$ and conditional density

$$f(k, z | y; \theta, v/\varphi) = \frac{f(y, k, z; \theta, v/\varphi)}{f_{(\tau,\infty)}(y; \theta, v/\varphi)} = \frac{f(y, k, z; \theta, v/\varphi)}{\exp\{\ell_{Y=y>\tau}(\theta)\}},$$

where $\ell_{Y>\tau}(\theta)$ is the log-likelihood of the lower-truncated datum $Y > \tau$. Choose an arbitrary density π modeling the random vector (K, Z) below the truncation point τ. This gives us for the random vector $(K, Z) \sim \pi$

$$\begin{aligned}
\ell_{Y>\tau}(\theta) &= \int \pi(k, z)\, \ell_{Y>\tau}(\theta)\, d\nu(k, z) \\
&= \int \pi(k, z) \log \left(\frac{f(Y, k, z; \theta, v/\varphi)/\pi(k, z)}{f(k, z|Y; \theta, v/\varphi)/\pi(k, z)} \right) d\nu(k, z) \\
&= \int \pi(k, z) \log \left(\frac{f(Y, k, z; \theta, v/\varphi)}{\pi(k, z)} \right) d\nu(k, z) + D_{\mathrm{KL}}\left(\pi || f(\cdot|Y; \theta, v/\varphi)\right) \\
&\ge \int \pi(k, z) \log \left(\frac{f(Y, k, z; \theta, v/\varphi)}{\pi(k, z)} \right) d\nu(k, z) \\
&= \mathbb{E}_\pi \left[\ell_{(Y,K,Z)}(\theta) \middle| Y \right] - \mathbb{E}_\pi \left[\log \pi(K, Z) \right] \\
&= \log f(Y; \theta, v/\varphi) + \mathbb{E}_\pi \left[\sum_{j=1}^{K} \log f(Z_j; \theta, v/\varphi) \right] - \mathbb{E}_\pi \left[\log \pi(K, Z) \right] \\
&\overset{\text{def.}}{=} \mathcal{Q}(\theta; \pi),
\end{aligned}$$

where the second last identity uses that the log-likelihood (6.51) has a simple form under the geometric distribution chosen for K; this is exactly the step where we benefit from this specific choice of the probability extension below the truncation point. There is a subtle point here. Namely, $\ell_{Y>\tau}(\theta)$ is the log-likelihood of the lower-truncated datum $Y > \tau$, whereas $\log f(Y; \theta, v/\varphi)$ is the log-likelihood not using any lower-truncation.

The **E-step** for given canonical parameter estimate $\widehat{\theta}^{(t-1)}$ reads as

$$\begin{aligned}
\widehat{\pi}^{(t)} &= \underset{\pi}{\arg\max}\ \mathcal{Q}\left(\widehat{\theta}^{(t-1)}; \pi\right) \;=\; \underset{\pi}{\arg\min}\ D_{\mathrm{KL}}\left(\pi \,\middle\|\, f(\cdot|Y; \widehat{\theta}^{(t-1)}, v/\varphi)\right) \\
&= f\left(\cdot \,\middle|\, Y; \widehat{\theta}^{(t-1)}, v/\varphi\right) \\
&= p\left(\cdot; \widehat{\theta}^{(t-1)}, v/\varphi\right) \prod_{j=1}^{\cdot} \frac{f(\cdot_j; \widehat{\theta}^{(t-1)}, v/\varphi)}{F(\tau; \widehat{\theta}^{(t-1)}, v/\varphi)}.
\end{aligned}$$

The latter describes a compound distribution for $\sum_{j=1}^{K} Z_j$ with a geometric count random variable K and independent i.i.d. random variables $Z_1, Z_2, \ldots,$ having upper-truncated densities $f_{(-\infty,\tau]}(\cdot; \widehat{\theta}^{(t-1)}, v/\varphi)$. This allows us to calculate the expected compound claim below the truncation point

$$\begin{aligned}
\widehat{Y}^{(t)}_{\leq\tau} = \mathbb{E}_{\widehat{\pi}^{(t)}}\left[\sum_{j=1}^{K} Z_j\right] &\;=\; \mathbb{E}_{\widehat{\pi}^{(t)}}[K]\ \mathbb{E}_{\widehat{\pi}^{(t)}}[Z_1] \\
&= \frac{F(\tau; \widehat{\theta}^{(t-1)}, v/\varphi)}{1 - F(\tau; \widehat{\theta}^{(t-1)}, v/\varphi)} \int z\, f_{(-\infty,\tau]}(z; \widehat{\theta}^{(t-1)}, v/\varphi)\, d\nu(z).
\end{aligned}$$

This completes the E-step.

The **M-step** considers within the EDF

$$\begin{aligned}
\widehat{\theta}^{(t)} &= \underset{\theta}{\arg\max}\ \mathcal{Q}\left(\theta; \widehat{\pi}^{(t)}\right) \\
&= \underset{\theta}{\arg\max}\ \frac{\left(Y + \mathbb{E}_{\widehat{\pi}^{(t)}}\left[\sum_{j=1}^{K} Z_j\right]\right)\theta - (1 + \mathbb{E}_{\widehat{\pi}^{(t)}}[K])\kappa(\theta)}{\varphi/v} \\
&= \underset{\theta}{\arg\max}\ \frac{v(1 + \mathbb{E}_{\widehat{\pi}^{(t)}}[K])}{\varphi}\left[\left(\frac{Y + \widehat{Y}^{(t)}_{\leq\tau}}{1 + \mathbb{E}_{\widehat{\pi}^{(t)}}[K]}\right)\theta - \kappa(\theta)\right].
\end{aligned}$$

That is, the M-step applies the classical MLE step, we only need to change weights and observations

$$\begin{aligned} v \mapsto v^{(t)} &= v\left(1+\mathbb{E}_{\widehat{\pi}^{(t)}}\left[K\right]\right) = \frac{v}{1-F(\tau;\widehat{\theta}^{(t-1)}, v/\varphi)}, \\ Y \mapsto \widehat{Y}^{(t)} &= \frac{Y+\widehat{Y}^{(t)}_{\leq\tau}}{1+\mathbb{E}_{\widehat{\pi}^{(t)}}\left[K\right]} = \frac{Y+\mathbb{E}_{\widehat{\pi}^{(t)}}\left[K\right]\mathbb{E}_{\widehat{\pi}^{(t)}}\left[Z_1\right]}{1+\mathbb{E}_{\widehat{\pi}^{(t)}}\left[K\right]}. \end{aligned}$$

Note that this uses the specific structure of the EDF, in particular, we benefit from linearity here which allows for closed-form solutions.

EM algorithm for lower-truncated data within the EDF

---

(0) Choose an initial parameter $\widehat{\boldsymbol{\theta}}^{(0)} = (\widehat{\theta}_i^{(0)})_{1\leq i\leq n}$.
(1) Repeat for $t \geq 1$:

- **E-step.** Given parameter $\widehat{\boldsymbol{\theta}}^{(t-1)} = (\widehat{\theta}_i^{(t-1)})_{1\leq i\leq n}$, estimate the number of claims $\boldsymbol{K}$ and the corresponding claim sizes $Z_{i,j}$ by

$$\begin{aligned} \widehat{K}_i^{(t)} &= \frac{F(\tau;\widehat{\theta}_i^{(t-1)}, v_i/\varphi)}{1-F(\tau;\widehat{\theta}_i^{(t-1)}, v_i/\varphi)}, \\ \widehat{Z}_{i,1}^{(t)} &= \int z\, f_{(-\infty,\tau]}(z;\widehat{\theta}_i^{(t-1)}, v_i/\varphi)\, d\nu(z). \end{aligned} \tag{6.52}$$

  This provides us with estimated weights and observations for $1 \leq i \leq n$

$$v_i^{(t)} = v_i\left(1+\widehat{K}_i^{(t)}\right) \qquad \text{and} \qquad \widehat{Y}_i^{(t)} = \frac{Y_i + \widehat{K}_i^{(t)}\widehat{Z}_{i,1}^{(t)}}{1+\widehat{K}_i^{(t)}}.$$

- **M-step.** Calculate the MLE $\widehat{\boldsymbol{\theta}}^{(t)} = (\widehat{\theta}_i^{(t)})_{1\leq i\leq n}$ based on observations $\widehat{\boldsymbol{Y}}^{(t)} = (\widehat{Y}_i^{(t)})_{1\leq i\leq n}^{\top}$ and weights $\widehat{\boldsymbol{v}}^{(t)} = (\widehat{v}_i^{(t)})_{1\leq i\leq n}^{\top}$, i.e., solve

$$\widehat{\boldsymbol{\theta}}^{(t)} = \underset{\boldsymbol{\theta}}{\arg\max}\ \ell_{\widehat{\boldsymbol{Y}}^{(t)}}(\boldsymbol{\theta};\widehat{\boldsymbol{v}}^{(t)}/\varphi) = \underset{\boldsymbol{\theta}}{\arg\max}\ \sum_{i=1}^{n}\log f(\widehat{Y}_i^{(t)};\theta_i,\widehat{v}_i^{(t)}/\varphi).$$

Remarks 6.17 Essentially, the above algorithm uses that the MLE in the EDF is based on a sufficient statistics of the observations, and in our case this sufficient statistics is $\widehat{Y}_i^{(t)}$.

Example 6.18 (Lower-Truncated Claim Sizes) We revisit the gamma claim size GLM introduced in Sect. 5.3.7, see also Example 6.16 on right-censored claims. We choose as lower-truncation point $\tau = 1'000$, i.e., we get rid of the very small claims that mainly generate administrative expenses at a rather small claim compensation. We have 70 claims below this truncation point, and there remain $n = 586$ claims above the truncation point that can be used for model fitting in the lower-truncated case. We use the EM algorithm for lower-truncated data to fit a GLM to this problem.

In order to calculate the E-step we need to evaluate the conditional expectation (6.52) under the gamma model for truncation probability

$$F(\tau; \widehat{\theta}^{(t-1)}, v/\varphi) = \int_0^\tau \frac{\beta^\alpha}{\Gamma(\alpha)} z^{\alpha-1} \exp\{-\beta z\}\, dz = \mathcal{G}(\alpha, \beta\tau),$$

with shape parameter $\alpha = v/\varphi$ and scale parameter $\beta = -\widehat{\theta}^{(t-1)} v/\varphi$. In complete analogy to (6.47) we have

$$\widehat{Z}_1^{(t)} = \int z\, f_{(\infty,\tau]}(z; \widehat{\theta}^{(t-1)}, v/\varphi)\, d\nu(z) \;=\; \frac{\alpha}{\beta} \frac{\mathcal{G}(\alpha+1, \beta\tau)}{\mathcal{G}(\alpha, \beta\tau)}.$$

For the modeling we choose again the features as used for model Gamma GLM2, this gives $q+1 = 7$ regression parameter components and additionally we set for the dispersion parameter $\widehat{\varphi}^{\text{MLE}} = 1.427$. This dispersion parameter we keep fixed in all the models studied in this example. In a first step we simply fit a gamma GLM to the lower-truncated data $Y_i > \tau$. We call this model ‘crude GLM2’, and it overestimates the true claim sizes because it ignores the fact of having lower-truncated data.

To initialize the EM algorithm for lower-truncated data we use the model crude GLM2. We then iterate the algorithm for 10 steps which provides convergence. The results are presented in Table 6.5. We observe that the resulting log-likelihood fitted on the lower-truncated data and evaluated on the complete data $\ell_{\boldsymbol{Y}}$ (which is available here) is the same as for model Gamma GLM2 which has been estimated on the complete data. Moreover, this lower-truncated EM algorithm fitted model slightly under-estimates the average claim sizes.

Figure 6.18 shows the estimated means $\widehat{\mu}_i$ on an individual claims level. The x-axis always gives the estimates from the complete log-likelihood model Gamma GLM2. The y-axis on the left-hand side shows the estimates from the crude GLM and the right-hand side the estimates from the EM algorithm fitted counterpart (fitted on the lower-truncated data). We observe that the crude model overestimates

Table 6.5 Comparison of the complete log-likelihood and the incomplete log-likelihood (lower-truncation $\tau = 1'000$) results

	# Param.	Log-likelihood $\ell_{\boldsymbol{Y}}(\widehat{\theta}^{\text{MLE}}, \widehat{\varphi}^{\text{MLE}})$	Dispersion est. $\widehat{\varphi}^{\text{MLE}}$	Average amount	Rel. change
Gamma GLM2 (complete data)	$7+1$	$-7'129$	1.427	25’130	
Crude GLM2 (lower-truncated)	$7+1$	$-7'133$		26’879	+7%
EM est. GLM2 (lower-truncated)	$7+1$	$-7'129$		24’900	−1%

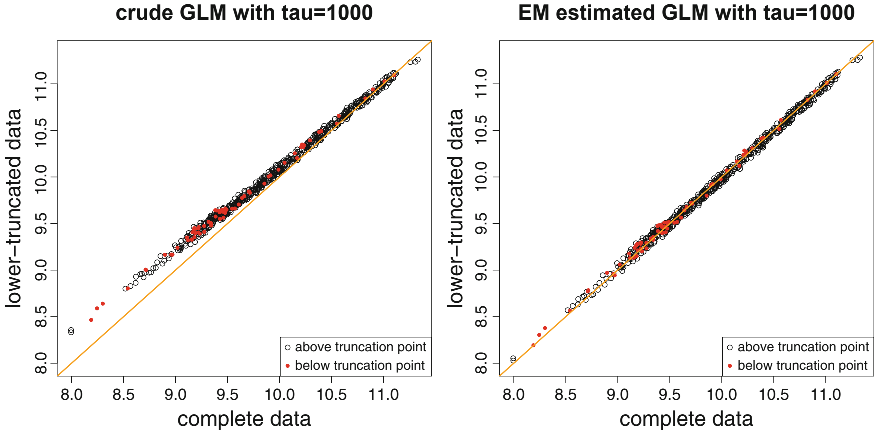


Fig. 6.18 Comparison of the estimated means $\widehat{\mu}_i$ in model Gamma GLM2 against (lhs) the crude GLM and (rhs) the EM fitted lower-truncated model; both axis are on the log-scale

the claims (being above the orange diagonal), in particular, this applies to claims with lower expected claim amounts. The EM algorithm fitted model, considering the fact that we have lower-truncated data, corrects for the truncation, and the resulting estimates almost completely coincide with the ones from the complete log-likelihood model. Again we remark that we use an identical dispersion parameter estimate $\widehat{\varphi}^{\mathrm{MLE}}$, and it is an open problem to select a reasonable value from lower-truncated data. ■

Example 6.19 (Zero-Truncated Claim Counts and the Hurdle Poisson Model) In Sect. 5.3.6, we have been studying the ZIP model that has assigned an additional probability weight to the event $\{N = 0\}$ of having zero claims. This model can be understood as a hierarchical model with a latent variable Z indicating whether we have an excess zero claim or not, see (5.41). In that situation we have a mixture distribution of a Poisson distribution and a degenerate distribution. Fitting in Example 5.25 has been done brute force by using a general purpose optimizer, but we could also use the EM algorithm for mixture distributions.

An alternative way of modeling excess zeros is the hurdle approach which combines a lower-truncated count distribution with a point mass in zero. For the Poisson case this reads as, see (5.42),

$$f_{\text{hurdle Poisson}}(k; \lambda, v, \pi_0) = \begin{cases} \pi_0 & \text{for } k = 0, \\ (1-\pi_0)\frac{e^{-v\lambda}\frac{(v\lambda)^k}{k!}}{1-e^{-v\lambda}} & \text{for } k \in \mathbb{N}, \end{cases} \tag{6.53}$$

for $\pi_0 \in (0, 1)$ and $\lambda, v > 0$. If we ignore any observation $\{N = 0\}$ we obtain a lower-truncated Poisson model, also called zero-truncated Poisson (ZTP) model. This ZTP model can be fitted with the EM algorithm for lower-truncated data. In the following we only consider insurance policies i with $N_i > 0$. The log-likelihood of the ZTP model $N > 0$ is given by (we consider one single component only and drop the lower index in the notation)

$$\theta \; \mapsto \; \ell_{N>0}(\theta) = N\theta - ve^{\theta} - \log(N!) + N\log(v) - \log(1 - e^{-ve^{\theta}}), \tag{6.54}$$

with exposure $v > 0$ and canonical parameter $\theta \in \boldsymbol{\Theta} = \mathbb{R}$ such that $\lambda = \exp\{\theta\}$. The ZTP model provides for the random variable K the following geometric distribution (for the number of claims below the truncation point), see (6.50),

$$\mathbb{P}_{\theta}[K = k] \;=\; \mathbb{P}_{\theta}[N = 0]^{k}\, \mathbb{P}_{\theta}[N > 0] \;=\; e^{-kve^{\theta}} \left(1 - e^{-ve^{\theta}}\right).$$

In view of (6.51), this gives us complete log-likelihood (note that $Z_j = 0$ for all j)

$$\begin{aligned}\ell_{(N,K,Z)}(\theta) &= N\theta - ve^{\theta} - \log(N!) + N\log(v) + \sum_{j=1}^{K} \left(Z_j\theta - ve^{\theta} - \log(Z_j!) + Z_j\log(v)\right)\\ &= N\theta - (1+K)\, ve^{\theta} - \log(N!) + N\log(v).\end{aligned}$$

We can now directly apply a simplified version of the EM algorithm for lower-truncated data. For the E-step we have, given parameter $\widehat{\theta}^{(t-1)}$,

$$\widehat{K}^{(t)} = \frac{\mathbb{P}_{\widehat{\theta}^{(t-1)}}[N=0]}{1 - \mathbb{P}_{\widehat{\theta}^{(t-1)}}[N=0]} = \frac{e^{-ve^{\widehat{\theta}^{(t-1)}}}}{1 - e^{-ve^{\widehat{\theta}^{(t-1)}}}} \qquad \text{and} \qquad \widehat{Z}_1^{(t)} = 0.$$

This provides us with the estimated weights and observations (set $Y = N/v$)

$$v^{(t)} = v\left(1 + \widehat{K}^{(t)}\right) = \frac{v}{1 - e^{-ve^{\widehat{\theta}^{(t-1)}}}} \qquad \text{and} \qquad \widehat{Y}^{(t)} = \frac{Y}{1 + \widehat{K}^{(t)}} = \frac{N}{v^{(t)}}. \tag{6.55}$$

Thus, the EM algorithm iterates Poisson MLEs, and the E-Step modifies the weights $v^{(t)}$ in each step of the loop correspondingly. We remark that the ZTP model has an EF representation which allows one to directly estimate the corresponding parameters without using the EM algorithm, see Remark 6.20, below.

We revisit the French MTPL claim frequency data, and, in particular, we use model Poisson GLM3 as a benchmark, we refer to Tables 5.5 and 5.10. The feature engineering is done exactly as in model Poisson GLM3. We then select only the insurance policies from the learning data $\mathcal{L}$ that have suffered at least one claim, i.e., $N_i > 0$. These are $m = 22'434$ out of $n = 610'206$ insurance policies. Thus, we only consider $m/n = 3.68\%$ of all insurance policies, and we fit the lower-truncated log-likelihood (ZTP model) to this data

$$\ell_{N>0}(\boldsymbol{\beta}) = \sum_{i=1}^{m} N_i\theta_i - v_i e^{\theta_i} - \log(N_i!) + N_i\log(v_i) - \log(1 - e^{-v_i e^{\theta_i}}),$$

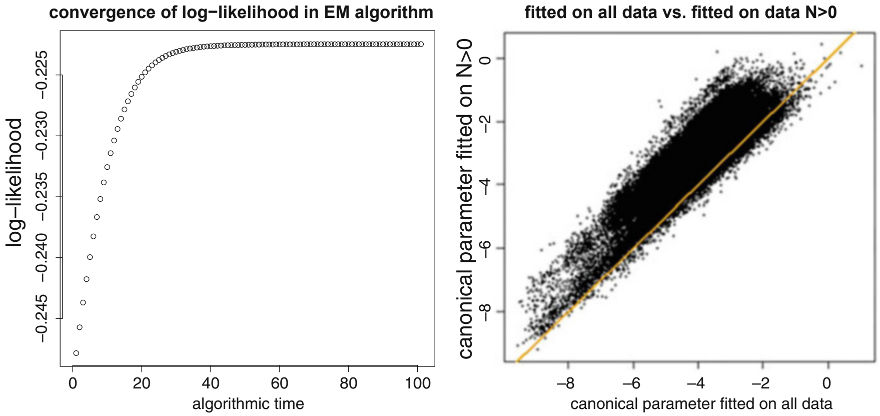


Fig. 6.19 (lhs) Convergence of the EM algorithm for the lower-truncated data in the Poisson hurdle case; (rhs) canonical parameters of the Poisson GLMs fitted on all data $\mathcal{L}$ vs. fitted only on policies with $N_i > 0$

Table 6.6 Run times, number of parameters, AICs, in-sample and out-of-sample deviance losses (units are in 10^{-2}) and in-sample average frequency of the Poisson null model and the Poisson, negative-binomial, ZIP and hurdle Poisson GLMs

	Run time	# Param.	AIC	In-sample loss on $\mathcal{L}$	Out-of-sample loss on $\mathcal{T}$	Aver. freq.
Poisson null	–	1	199’506	25.213	25.445	7.36%
Poisson GLM3	15 s	50	192’716	24.084	24.102	7.36%
NB GLM3 $\widehat{\alpha}^{\text{MLE}}_{\text{NB}} = 1.810$	85 s	51	192’113	20.722	20.674	7.38%
ZIP GLM3 (null π_0)	270 s	51	192’393	–	–	7.37%
Hurdle Poisson GLM3	300 s	100	**191’851**	–	–	7.39%

where $1 \le i \le m$ runs over all insurance policies with at least one claim and where the canonical parameter θ_i is given by the linear predictor $\theta_i = \langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle$. We fit this model using the EM algorithm for lower-truncated data. In each loop this requires that the offset $o_i^{(t)} = \log(v_i^{(t)})$ is adjusted according to (6.55); for the discussion of offsets we refer to Sect. 5.2.3. Convergence of the EM algorithm is achieved after roughly 75 iterations, see Fig. 6.19 (lhs).

In our first analysis we do not consider the Poisson hurdle model, but we simply consider model Poisson GLM3. However, this Poisson model with regression parameter $\boldsymbol{\beta}$ is fitted only on the data $N_i > 0$ (exactly using the results of the EM algorithm for lower-truncated data $N_i > 0$). The resulting predictive model is presented in Table 6.7. We observe that model Poisson GLM3 that is only fitted on the data $N_i > 0$ is clearly not competitive, i.e., we cannot simply extrapolate this estimated model to $\{N_i = 0\}$. This extrapolation results in a Poisson GLM that has a much too large average frequency of 15.11%, see last column of Table 6.7; this bias can clearly be seen in Fig. 6.19 (rhs) where we compare the two fits. From this we conclude that either the Poisson model assumption in general does not

Table 6.7 Number of parameters, in-sample and out-of-sample deviance losses on all data (units are in 10^{-2}), out-of-sample lower-truncated log-likelihood $\ell_{N>0}$ and in-sample average frequency of the Poisson null model and model Poisson GLM3 fitted on all data $\mathcal{L}$ and fitted on the data $N_i > 0$ only

	#	In-sample	Out-of-sample		Aver.
	Param.	Loss on $\mathcal{L}$	Loss on $\mathcal{T}$	$\ell_{N>0}$	freq.
Poisson null	1	25.213	25.445	–	7.36%
Poisson GLM3 fitted on all data	50	24.084	24.102	−0.2278	7.36%
Poisson GLM3 fitted on $N_i > 0$	50	28.064	28.211	−0.2195	15.11%

match the data, or that we have excess zeros (which do not influence the estimation procedure if we only consider the policies with at least one claim). Let us compare the lower-truncated log-likelihood $\ell_{N>0}$ out-of-sample only on the policies with at least one claim (ZTP model). We observe that the EM fitted model provides a better description of the data, as we have a bigger log-likelihood than the model fitted on all data $\mathcal{L}$ (i.e. -0.2195 vs. -0.2278 for the ZTP log-likelihood). Thus, the lower-truncated fitting procedure finds a better model on $\{N_i > 0\}$ when only fitted on these lower-truncated claim counts.

This analysis concludes that we need to fit the full hurdle Poisson model (6.53). That is, we cannot simply extrapolate the model fitted on the ZTP log-likelihood $\ell_{N>0}$ because, typically, $\pi_0(\boldsymbol{x}_i) \neq \exp\{-v_i e^{\langle \boldsymbol{\beta}, \boldsymbol{x}_i\rangle}\}$, the latter coming from the Poisson GLM with regression parameter $\boldsymbol{\beta}$. We model the zero claim probability $\pi_0(\boldsymbol{x}_i)$ by the logistic Bernoulli GLM indicating whether we have claims or not. We set up the logistic GLM for $p(\boldsymbol{x}_i) = 1 - \pi_0(\boldsymbol{x}_i)$ of describing the indicator $Y_i = \mathbb{1}_{\{N_i>0\}}$ of having claims. The difficulty compared to the Poisson model is that we cannot easily integrate the time exposure v_i as a pro rata temporis variable like in the Poisson case. We therefore make the following considerations. The canonical link in the logistic Bernoulli GLM is the logit function $p \mapsto \text{logit}(p) = \log(p/(1-p)) = \log(p) - \log(1-p)$ for $p \in (0,1)$. Typically, in our application, $p \ll 1$ is fairly small because claims are rare events. This implies $\log(p/(1-p)) \approx \log(p)$, i.e., the logit link behaves similarly to the log-link for small default probabilities p. This motivates to integrate the logged exposures $\log v_i$ as offsets into the logistic probabilities. That is, we make the following model assumption

$$(\boldsymbol{x}, v) \;\mapsto\; \text{logit}(p(\boldsymbol{x}_i, v_i)) = \log(v_i) + \langle \widetilde{\boldsymbol{\beta}}, \boldsymbol{x}_i\rangle,$$

with offset $o_i = \log(v_i)$ and regression parameter $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{q+1}$. We fit this model using the R command `glm` using `family=binomial()`. The results then allow us to define the estimated hurdle Poisson model by, recall $p(\boldsymbol{x}_i, v_i) = 1 - \pi_0(\boldsymbol{x}_i, v_i)$,

$$f_{\text{hurdle Poisson}}(k; \boldsymbol{x}_i, v_i) = \begin{cases} 1 - p(\boldsymbol{x}_i, v_i) = \left(1 + \exp\{\log(v_i) + \langle \widetilde{\boldsymbol{\beta}}, \boldsymbol{x}_i\rangle\}\right)^{-1} & \text{for } k = 0, \\ \frac{p(\boldsymbol{x}_i, v_i)}{1 - e^{-\mu(\boldsymbol{x}_i, v_i)}} \, e^{-\mu(\boldsymbol{x}_i, v_i)} \frac{\mu(\boldsymbol{x}_i, v_i)^k}{k!} & \text{for } k \in \mathbb{N}, \end{cases}$$

Table 6.8 Contingency table of the observed numbers of policies against predicted numbers of policies with given claim counts `ClaimNb` (in-sample)

	Numbers of claims `ClaimNb`					
	0	1	2	3	4	5
Observed number of policies	587’772	21’198	1’174	57	4	1
Poisson predicted number of policies	587’325	22’064	779	34	3	0.3
NB predicted number of policies	587’902	20’982	1’200	100	15	4
ZIP predicted number of policies	587’829	21’094	1’191	79	9	4
Hurdle Poisson predicted number of policies	587’772	21’119	1’233	76	6	1

where $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{q+1}$ is the regression parameter from the logistic Bernoulli GLM, and where $\mu(\boldsymbol{x}_i, v_i) = v_i \exp\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle$ is the Poisson GLM estimated with the EM algorithm on the lower-truncated data $N_i > 0$ (ZTP model). The results are presented in Table 6.6.

Table 6.6 compares the hurdle Poisson model to the approaches studied in Table 5.10. Firstly, fitting the hurdle Poisson model is more time intensive, the EM algorithm takes some time and we need to fit the Bernoulli logistic GLM which is of a similar complexity as fitting model Poisson GLM3. The results in terms of AIC look convincing. The hurdle Poisson model provides an excellent model for the indicator of having a claim (here it outperforms model ZIP GLM3). It also tries to optimally fit a ZTP model to all insurance policies having at least one claim. This can also be seen from Table 6.8 which determines the expected number of policies that suffer the different numbers of claims.

We close this example by concluding that the hurdle Poisson model provides the best description, at the price of using more parameters. The ZIP model could be lifted to a similar level, however, we consider fitting the hurdle approach to be more convenient, see also Remark 6.20, below. In particular, feature engineering seems simpler in the hurdle approach because the different effects are clearly separated, whereas in the ZIP approach it is more difficult to suitably model the excess zeros, see also Listing 5.10. This closes this example. ■

Remark 6.20 In (6.54) we have been considering the ZTP model for different exposures $v > 0$. If we set these exposures to $v = 1$, we obtain the ZTP log-likelihood

$$\ell_{N>0}(\theta) = N\theta - \left(e^{\theta} + \log(1 - e^{-e^{\theta}})\right) - \log(N!).$$

Note that this describes a single-parameter linear EF with cumulant function

$$\kappa(\theta) = e^{\theta} + \log(1 - e^{-e^{\theta}}),$$

for canonical parameter in the effective domain $\theta \in \boldsymbol{\Theta} = \mathbb{R}$. The mean of this EF model is given by

$$\mu = \mathbb{E}_\theta[N] = \kappa'(\theta) = \frac{e^\theta}{1 - e^{-e^\theta}} = \frac{\lambda}{1 - e^{-\lambda}},$$

where we set $\lambda = e^\theta$. The variance is given by

$$\mathrm{Var}_\theta(N) \;=\; \kappa''(\theta) \;=\; \mu \left(\frac{e^\lambda - (1+\lambda)}{e^\lambda - 1} \right) \;=\; \mu \left(1 - \mu e^{-\lambda}\right) \;>\; 0.$$

Note that the term in brackets is positive but less than one. The latter implies that the ZTP model has under-dispersion. Alternatively to the EM algorithm, we can also directly fit a GLM to this ZTP model. The only difficulty is that we need to appropriately integrate the time exposures. The original Poisson model suggests that if we choose the canonical parameter being equal to the linear predictor, we should integrate the logged exposures as offsets into the linear predictors. Along these lines, if we choose the canonical link $h = (\kappa')^{-1}$ of the ZTP model, we receive that the canonical parameter θ is equal to the linear predictor $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$, and we can directly integrate the logged exposures as offsets into the canonical parameters, see (5.25). This then allows us to directly fit this ZTP model with exposures using Fisher's scoring method. In this case of a concave log-likelihood function, the result will be identical to the solution of the EM algorithm found in Example 6.19, and, in fact, this direct approach is more straightforward and more time-efficient. Similar considerations can be done for other hurdle models.

6.4.4 *Composite Models*

In Sect. 6.3.1 we have promoted to mix distributions in cases where the data cannot be modeled by a single EDF distribution. Alternatively, one can also consider to compose densities which leads to so-called *composite models* (also called splicing models). This idea has been introduced to the actuarial literature by Cooray–Ananda [81] and Scollnik [332]. Assume we have two absolutely continuous densities $f^{(i)}(\cdot; \theta_i)$ with corresponding distribution functions $F^{(i)}(\cdot; \theta_i)$, $i = 1, 2$. These two densities can easily be composed at a splicing value τ and with weight $p \in (0, 1)$ by considering the following composite density

$$f(y; p, \theta_1, \theta_2) \;=\; p\, \frac{f^{(1)}(y; \theta_1) \mathbb{1}_{\{y \le \tau\}}}{F^{(1)}(\tau; \theta_1)} + (1-p)\, \frac{f^{(2)}(y; \theta_2) \mathbb{1}_{\{y > \tau\}}}{1 - F^{(2)}(\tau; \theta_2)}, \qquad (6.56)$$

supposed that both denominators are non-zero. In this notation we treat splicing value τ as a hyper-parameter that is chosen by the modeler, and is not estimated from data. In view of (6.41) we can rewrite this in terms for lower- and upper-truncated densities

$$f(y; p, \theta_1, \theta_2) \;=\; p\, f^{(1)}_{(-\infty,\tau]}(y; \theta_1) + (1-p)\, f^{(2)}_{(\tau,\infty)}(y; \theta_2).$$

In this notation, we see that a composite model can also be interpreted as a mixture model with mixture probability $p \in (0, 1)$ and mixing densities $f^{(1)}_{(-\infty,\tau]}$ and $f^{(2)}_{(\tau,\infty)}$ having disjoint supports $(\infty, \tau]$ and (τ, ∞), respectively.

These disjoint supports allow for simpler MLE, i.e., we do not need to rely on the 'EM algorithm for mixture distributions' to fit this model. The log-likelihood of $Y \sim f(y; p, \theta_1, \theta_2)$ is given by

$$\begin{aligned}\ell_Y(p, \theta_1, \theta_2) = \Big(\log(p) + \log f^{(1)}_{(-\infty,\tau]}(Y; \theta_1)\Big)\, \mathbb{1}_{\{Y \le \tau\}} \\ + \Big(\log(1-p) + \log f^{(2)}_{(\tau,\infty)}(Y; \theta_2)\Big)\, \mathbb{1}_{\{Y > \tau\}}.\end{aligned}$$

This shows that the log-likelihood nicely decouples in the composite case and all parameters can directly be estimated with MLE: parameter θ_1 uses all observations smaller or equal to τ, parameter θ_2 uses all observations bigger than τ, and p is estimated by the proportions of claims below and above the splicing point τ. This holds for a null model as well as for a GLM approach for θ_1, θ_2 and p.

Nevertheless, the EM algorithm may still be used for parameter estimation, namely, truncation may ask for the 'EM algorithm for truncated data'. Alternatively, we could also use the 'EM algorithm for censored data' to estimate the truncated densities, because we have knowledge of the number of claims above and below the splicing point τ, thus, we could right- or left-censor these claims. The latter may lead to more stability in the estimation procedure since we use more information in parameter estimation, i.e., the two truncated densities will not be independent because they simultaneously consider all claim counts (but not identical claim sizes due to censoring).

For composite models one sometimes requires more regularity in the densities, we may, e.g., require continuity in the density in the splicing point which provides mixture probability

$$p \;=\; \frac{f^{(2)}(\tau; \theta_2) F^{(1)}(\tau; \theta_1)}{f^{(1)}(\tau; \theta_1)(1 - F^{(2)}(\tau; \theta_2)) + f^{(2)}(\tau; \theta_2) F^{(1)}(\tau; \theta_1)}.$$

This reduces the number of parameters to be estimated but complicates the score equations. If we require a differential condition in τ we receive requirement

$$p \;=\; \frac{f^{(2)}_y(\tau; \theta_2) F^{(1)}(\tau; \theta_1)}{f^{(1)}_y(\tau; \theta_1)(1 - F^{(2)}(\tau; \theta_2)) + f^{(2)}_y(\tau; \theta_2) F^{(1)}(\tau; \theta_1)},$$

where $f_y^{(i)}(y;\theta_i)$ denotes the first derivative w.r.t. y. Together with the continuity this provides requirement for having differentiability in τ

$$\frac{f^{(2)}(\tau;\theta_2)}{f^{(1)}(\tau;\theta_1)} = \frac{f_y^{(2)}(\tau;\theta_2)}{f_y^{(1)}(\tau;\theta_1)}.$$

Again this reduces the degrees of freedom in parameter estimation but complicates the score equations. We refrain from giving an example and close this section; we will consider a deep composite regression model in Sect. 11.3.2, below, where we replace the fixed splicing point by a quantile for a fixed quantile level.