Non-Life Insurance: Mathematics and Statistics

Solution sheet 11

Solution 11.1 Claim Frequency Modeling with GLM

(a) In this exercise we work with three tariff criteria. The first criterion (vehicle class) has 2 risk characteristics:

$$\beta_{1,1} \text{ (weight over 60 kg and more than two gears)} \quad \text{and} \quad \beta_{1,2} \text{ (other).}$$

The second criterion (vehicle age) also has 2 risk characteristics:

$$\beta_{2,1} \text{ (at most one year)} \quad \text{and} \quad \beta_{2,2} \text{ (more than one year).}$$

The third criterion (geographic zone) has 3 risk characteristics:

$$\beta_{3,1} \text{ (large cities)}, \quad \beta_{3,2} \text{ (middle-sized towns)} \quad \text{and} \quad \beta_{3,3} \text{ (smaller towns and countryside).}$$

We write N_{l_1,l_2,l_3} for the numbers of claims, v_{l_1,l_2,l_3} for the volumes and λ_{l_1,l_2,l_3} for the claim frequencies of the risk classes $(l_1, l_2, l_3), 1 \le l_1 \le 2, 1 \le l_2 \le 2, 1 \le l_3 \le 3$. We assume that all N_{l_1,l_2,l_3} are independent with

$$N_{l_1,l_2,l_3} \sim \text{Poi}(\lambda_{l_1,l_2,l_3} v_{l_1,l_2,l_3}),$$

and define

$$X_{l_1,l_2,l_3} = \frac{N_{l_1,l_2,l_3}}{v_{l_1,l_2,l_3}}.$$

In particular, we have

$$\lambda_{l_1,l_2,l_3} = \mathbb{E}\left[\frac{N_{l_1,l_2,l_3}}{v_{l_1,l_2,l_3}}\right] = \mathbb{E}\left[X_{l_1,l_2,l_3}\right].$$

We model

$$g(\lambda_{l_1,l_2,l_3}) = g\left(\mathbb{E}\left[X_{l_1,l_2,l_3}\right]\right) = \beta_0 + \beta_{1,l_1} + \beta_{2,l_2} + \beta_{3,l_3},$$

where $\beta_0 \in \mathbb{R}$ and where we use the log-link function, i.e. $g(\cdot) = \log(\cdot)$. In order to get a unique solution, we set $\beta_{1,1} = \beta_{2,1} = \beta_{3,1} = 0$. Moreover, we define

$$\boldsymbol{\beta} = (\beta_0, \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{3,3})' \in \mathbb{R}^{r+1},$$

where $r = 4$. Similarly as in Exercise 10.1, we relabel the risk classes with the index $m \in \{1, \ldots, M\}$, where $M = 2 \cdot 2 \cdot 3 = 12$, define $\boldsymbol{X} = (X_1, \ldots, X_M)'$ and the design matrix $Z \in \mathbb{R}^{M \times (r+1)}$ that satisfies

$$\log \mathbb{E}[\boldsymbol{X}] = Z\boldsymbol{\beta},$$

where the logarithm is applied componentwise to $\mathbb{E}[\boldsymbol{X}]$. Let $m \in \{1, \ldots, 12\}$. According to Example 7.8 of the lecture notes (version of February 7, 2022), $X_m = N_m/v_m$ belongs to the exponential dispersion family with cumulant function $b(\cdot) = \exp\{\cdot\}$, $\theta_m = \log \lambda_m$, $w_m = v_m$ and dispersion parameter $\phi = 1$, i.e. we have

$$[Z\boldsymbol{\beta}]_m = \log \mathbb{E}[X_m] = \log \mathbb{E}\left[\frac{N_m}{v_m}\right] = \log \lambda_m = \theta_m,$$

where $[Z\boldsymbol{\beta}]_m$ denotes the m-th element of the vector $Z\boldsymbol{\beta}$. Summarizing, we assume that $X_1, \dots, X_M$ are independent with

$$X_m \sim \text{EDF}(\theta_m = [Z\boldsymbol{\beta}]_m, \phi = 1, w_m = v_m, b(\cdot) = \exp\{\cdot\}),$$

for all $m \in \{1, \dots, M\}$. As $b(\cdot) = \exp\{\cdot\}$, we also have $b'(\cdot) = \exp\{\cdot\}$, where b' denotes the first derivative of b. In particular, the log-link function $g(\cdot) = \log(\cdot)$ is equal to the canonical link function $h(\cdot) = (b')^{-1}(\cdot) = \log(\cdot)$ in the Poisson model. Therefore, we can use equation (7.35) of the lecture notes (version of February 7, 2022): the MLE $\widehat{\boldsymbol{\beta}}^{\text{MLE}}$ of $\boldsymbol{\beta}$ is the solution of

$$Z'Vb'(Z\boldsymbol{\beta}) \,=\, Z'V\exp\{Z\boldsymbol{\beta}\} \overset{!}{=} Z'V\boldsymbol{X}, \tag{1}$$

where the weight matrix V is given by $V = \text{diag}(v_1, \dots, v_M)$, see also Proposition 7.10 of the lecture notes (version of February 7, 2022). Equation (1) has to be solved numerically. We refer to Listing 1 for the application of this GLM model in R. The resulting MLEs of the parameters $\beta_0, \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{3,3}$ are given in the first row of Table 1. We observe that insureds with a vehicle with weight over 60 kg and more than two gears tend to cause more claims than insureds with other vehicles. Analogously, if the vehicle is at most one year old, we expect more claims than if it is older. Regarding the geographic zone, we see that driving in middle-sized towns leads to fewer claims than driving in large cities. Moreover, driving in smaller towns and countryside leads to even fewer claims than driving in middle-sized towns. Similarly as the log-linear Gaussian regression model discussed in Exercise 10.1, the GLM framework allows for calculating parameter uncertainties and hypothesis testing. According to the R output, for the individual parameters we get the p-values listed in the second row of Table 1. These p-values are all substantially smaller than 0.05 and, thus, all the parameters are significantly different from zero.

	$\widehat{\beta}_0$	$\widehat{\beta}_{1,2}$	$\widehat{\beta}_{2,2}$	$\widehat{\beta}_{3,2}$	$\widehat{\beta}_{3,3}$
MLE	-1.4351	-0.2371	-0.5019	-0.4036	-1.6571
p-value	≈ 0	0.0009	≈ 0	≈ 0	≈ 0

Table 1: MLEs of the parameters $\beta_0, \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{3,3}$ and corresponding p-values.

Listing 1: R code for Exercise 11.1 (a).

```r
### Determine the design matrix Z
class <- factor(c(rep(1,6),rep(2,6)))
age <- factor(c(rep(1,3),rep(2,3),rep(1,3),rep(2,3)))
zone <- factor(c(rep(1:3,4)))
volumes <- c(1,2,5,4,9,70,2,3,6,8,15,50)*100
counts <- c(25,15,15,60,90,210,45,45,30,80,120,90)
Z <- model.matrix(counts ~ class + age + zone)

### Store design matrix Z (without intercept term), counts and volumes in one dataset
data <- as.data.frame(cbind(Z[,-1],counts,volumes))

### Apply GLM
d.glm <- glm(counts ~ class2 + age2 + zone2 + zone3, data=data, offset=log(volumes),
             family=poisson())
summary(d.glm)
```

(b) The plots of the observed and the fitted claim frequencies against the vehicle class, the vehicle age and the geographic zone are given in Figure 1, the corresponding R code in Listing 2. Note that the observed and the fitted marginal claim frequencies are always the same. This is a direct consequence of equation (1) above, which ensures that the observed and the fitted total marginal sums are the same (if we use the same volumes again), see also the remarks after Proposition 7.10 in the lecture notes (version of February 7, 2022). Moreover, in the marginal plot for the vehicle class we do not see that insureds with a vehicle with weight over 60 kg and more than two gears tend to cause more claims than insureds with other vehicles, as we would have expected after the discussion at the end of part (a). The reason for this peculiarity is that the MLE $\widehat{\beta}_{1,2}$ is driven by the risk cells with the biggest volumes ($v_6 = 7'000$ and $v_{12} = 5'000$). However, in these risk cells with the biggest volumes we observe very low claim frequencies. This implies that these risk cells have a small impact on the mean claim frequency. As a consequence, the resulting mean claim frequency is of similar size for both vehicles with weight over 60 kg and more than two gears and for other vehicles. For the other variables vehicle age and geographic zone we again see the same results as in part (a).

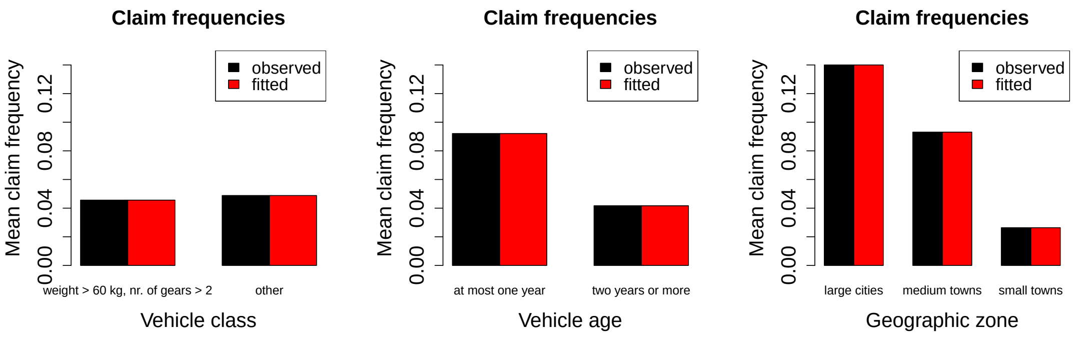

Figure 1: Observed and fitted claim frequencies against the vehicle class, the vehicle age and the geographical zone.

Listing 2: R code for Exercise 11.1 (b).

```
### Store features, observed numbers of claims and fitted numbers of claims in one dataset
data2 <- as.data.frame(cbind(class, age, zone, volumes, counts, fitted(d.glm)))
colnames(data2)[5:6] <- c("observed","fitted")

### Marginal claim frequencies for the two class categories
library(plyr)
class.comp <- ddply(data2, .(class), summarise, volumes=sum(volumes), observed=sum(observed),
                    fitted=sum(fitted))
par(mar=c(5.1, 4.6, 4.1, 2.1))
barplot(t(as.matrix(class.comp[,3:4]/class.comp[,2])), beside=TRUE,
        names.arg=c("weight > 60 kg, nr. of gears > 2", "other"), main="Claim frequencies",
        ylim=c(0,0.15), xlab="Vehicle class", ylab="Mean claim frequency", legend.text=FALSE,
        col=1:2, cex.names=0.95, cex.lab=1.5, cex.main=1.5, cex.axis=1.5)
legend("topright", legend=c("observed  ", "fitted  "), fill=1:2, cex=1.25)

### Marginal claim frequencies for the two age categories
age.comp <- ddply(data2, .(age), summarise, volumes=sum(volumes), observed=sum(observed),
                  fitted=sum(fitted))
barplot(t(as.matrix(age.comp[,3:4]/age.comp[,2])), beside=TRUE,
        names.arg=c("at most one year", "two years or more"), main="Claim frequencies",
        ylim=c(0,0.15), xlab="Vehicle age", ylab="Mean claim frequency", legend.text=FALSE,
        col=1:2, cex.names=0.95, cex.lab=1.5, cex.main=1.5, cex.axis=1.5)
legend("topright", legend=c("observed  ", "fitted  "), fill=1:2, cex=1.25)

### Marginal claim frequencies for the three zone categories
zone.comp <- ddply(data2, .(zone), summarise, volumes=sum(volumes), observed=sum(observed),
             fitted=sum(fitted))
barplot(t(as.matrix(zone.comp[,3:4]/zone.comp[,2])), beside=TRUE,
        names.arg=c("large cities", "medium towns", "small towns"), main="Claim frequencies",
        ylim=c(0,0.15), xlab="Geographic zone", ylab="Mean claim frequency", legend.text=FALSE,
        col=1:2, cex.names=0.95, cex.lab=1.5, cex.main=1.5, cex.axis=1.5)
legend("topright", legend=c("observed  ", "fitted  "), fill=1:2, cex=1.25)
```

(c) The Tukey-Anscombe plot given in Figure 2 can be generated by the R code of Listing 3. The plot looks rather fine in the sense that we do not observe any structure. However, we remark that we only have 12 observations in this example and, thus, it is difficult to detect possible patterns and to make a clear statement.

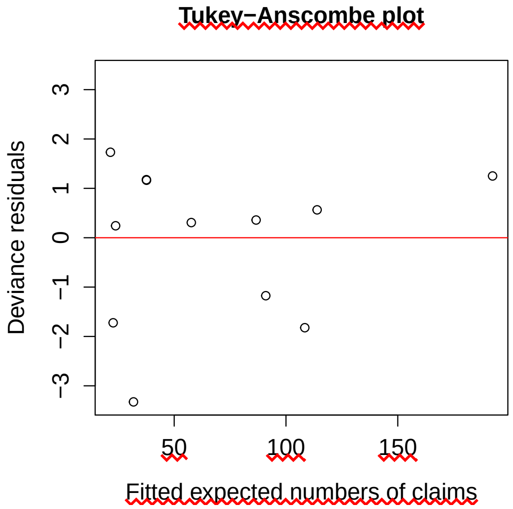

Figure 2: Tukey-Anscombe plot.

Listing 3: R code for Exercise 11.1 (c).

```
### Deviance residuals
dev.red <- residuals.glm(d.glm)

### Tukey-Anscombe plot
par(mar=c(5.1, 4.4, 4.1, 2.1))
plot(data2$fitted, dev.red, main="Tukey-Anscombe plot",
     xlab="Fitted expected numbers of claims", ylab="Deviance residuals",
     ylim=c(-max(abs(dev.red)),max(abs(dev.red))), cex.lab=1.25, cex.main=1.25, cex.axis=1.25)
abline(h=0, col="red")
```

(d) We perform two tests in order to check if there is statistical evidence that the classification into the geographic zones could be omitted. Note that in part (a) we have seen that we tend to have considerably fewer claims for drivers in smaller towns and countryside than for drivers in middle-sized towns. The same holds true for middle-sized towns and large cities. Thus, we would expect that the classification into the three different geographic zones is reasonable. Now we investigate this. The estimates of the expected values of X_m are given by

$$\widehat{\mu}_m = b'(\widehat{\theta}_m) = \exp\left\{\widehat{\theta}_m\right\} = \exp\left\{\left[Z\widehat{\boldsymbol{\beta}}^{\text{MLE}}\right]_m\right\},$$

for all $m = 1, \ldots, M$, and we write $\widehat{\boldsymbol{\mu}} = (\widehat{\mu}_1, \ldots, \widehat{\mu}_M)'$. According to page 207 of the lecture notes (version of February 7, 2022), the scaled deviance statistics is given by

$$\begin{aligned} D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}) &= \frac{2}{\phi}\sum_{m=1}^{M} w_m(X_m h(X_m) - b[h(X_m)] - X_m h(\widehat{\mu}_m) + b[h(\widehat{\mu}_m)]) \\ &= 2\sum_{m=1}^{M} v_m\left(X_m \log X_m - X_m - X_m \log \widehat{\mu}_m + \widehat{\mu}_m\right). \qquad (2) \end{aligned}$$

Moreover, since for the Poisson case we have $\phi = 1$, the scaled deviance statistics $D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}})$ and the deviance statistics $D(\boldsymbol{X}, \widehat{\boldsymbol{\mu}})$ are the same. In order to check whether there is statistical evidence that the classification into the geographic zones could be omitted, we define the null hypothesis

$$H_0 : \beta_{3,2} = \beta_{3,3} = 0.$$

Thus, in the reduced model we set the above $p = 2$ variables equal to 0. Then, we can recalculate $\widehat{\boldsymbol{\beta}}_{H_0}^{\text{MLE}}$ for this reduced model and define

$$\widehat{\boldsymbol{\mu}}_{H_0} = \exp\left\{ Z_{H_0} \widehat{\boldsymbol{\beta}}_{H_0}^{\text{MLE}} \right\},$$

where Z_{H_0} is the design matrix in the reduced model. According to formula (7.40) of the lecture notes (version of February 7, 2022), the test statistic

$$X^2 = D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}_{H_0}) - D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}).$$

has approximately a χ^2-distribution with $\text{df} = p = 2$ degrees of freedom. We get

$$X^2 \approx 389.882,$$

which corresponds to a p-value of approximately $2.179 \cdot 10^{-85}$, which is basically 0. Thus, we can reject H_0 at significance level of 5%. Since we can reject H_0 using this test, we can conclude that there seems to be no statistical evidence that the classification into the geographic zones could be omitted. For the R code used in part (d) we refer to Listing 4.

Listing 4: R code for Exercise 11.1 (d).

```r
### Deviance statistics of the full model
D.full <- d.glm$deviance

### Fit the reduced model
d.glm.2 <- glm(counts ~ class2 + age2, data=data, offset=log(volumes), family=poisson())
summary(d.glm.2)

### Deviance statistics of the reduced model
D.reduced <- d.glm.2$deviance

### Calculate the test statistic X^2
X.2 <- D.reduced-D.full

### Calculation of the corresponding p-value
pchisq(X.2, 2, lower.tail=FALSE)
```

Solution 11.2 Claim Frequency Modeling with Neural Networks

(a) The Poisson deviance statistics calculated on the datasets `trainset` and `testset` with the R code of Listing 5 are given in the first column of Table 2.

	GLM	NN (100 epochs)	NN (1'000 epochs)
deviance statistics `trainset`	1'314.7	709.7	111.6
deviance statistics `testset`	1'454.3	1'070.2	1'523.5

Table 2: Deviance statistics.

Listing 5: R code for Exercise 11.2 (a).

```r
### Apply GLM (on the training set)
d.glm <- glm(ClaimNb ~ VehPower+VehAge+DrivAge, data=trainset, offset=log(Exposure),
             family=poisson())
summary(d.glm)

### Deviance statistics on training set
predtrain <- predict(d.glm, trainset, type="response")
obstrain <- trainset$ClaimNb
(Deviancetrain <- 2*sum(log((obstrain/predtrain)^obstrain)-obstrain+predtrain))
d.glm$deviance     ### check deviance statistics on training set

### Deviance statistics on test set
predtestGLM <- predict(d.glm, testset, type="response")
obstest <- testset$ClaimNb
(Deviancetest <- 2*sum(log((obstest/predtestGLM)^obstest)-obstest+predtestGLM))
```

(b) We fit the neural network for 100 gradient descent steps, see the R code given in Listing 6 and use the resulting model to calculate the Poisson deviance statistics on the datasets `trainset` and `testset`, see the second column of Table 2. We observe that the neural network leads to smaller values of the deviance statistics on both the datasets `trainset` and `testset`. This is an indication that the neural network model has better predictive power than the GLM model. We remark that a simple GLM model like the one used in this exercise usually is not able to cope with interactions between the tariff criteria, in contrast to neural network models. This might explain the lower deviance statistics observed for the neural network model on the data `testset`. However, we do not further investigate this here.

(c) We perform the exact same fitting procedure as in part (b), with the only difference that we use 1'000 gradient descent steps instead of only 100. The resulting Poisson deviance statistics on the datasets `trainset` and `testset` are given in the third column of Table 2. On the one hand, we see that the deviance statistics on the dataset `trainset` used during training is smaller than for the GLM model of part (a) and the neural network model with 100 gradient descent steps of part (b). However, this "better" fit is deceiving. In fact, the deviance statistics on the dataset `testset` is bigger than for the GLM model of part (a) and the neural network model with 100 gradient descent steps of part (b). We emphasize that the dataset `testset` has not been seen during training and, thus, is the correct dataset to analyze the predictive power of a fitted model. We conclude that with 1'000 gradient descent steps we are in the situation of overfitting to the training data `trainset`. Therefore, the number of gradient descent steps has to be chosen carefully. Usually, one splits the available dataset into a learning set and a validation set. The learning set is then used to perform the gradient descent steps and to fit the model. The validation set can be used to track over-fitting to the learning set. As long as the deviance statistics on the validation set decreases, we are learning additional model structure. Once the deviance statistics on the validation set starts to increase again, we reach the phase of over-fitting where we are not learning (true) model structure anymore but rather peculiarities of the learning set, which is undesirable.

Listing 6: R code for Exercise 11.2 (b) and (c) (Neural network model).

```
### Features, volumes, responses and initial estimate
Ztrain <- model.matrix(data=trainset, ClaimNb ~ VehPower+VehAge+DrivAge)
trainset[,6:30] <- as.data.frame(Ztrain[,-1])
Ztest <- model.matrix(data=testset, ClaimNb ~ VehPower+VehAge+DrivAge)
testset[,6:30] <- as.data.frame(Ztest[,-1])
featlearn <- data.matrix(trainset[,6:30])
feattest <- data.matrix(testset[,6:30])
vollearn <- as.vector(log(trainset$Exposure))
voltest <- as.vector(log(testset$Exposure))
resplearn <- as.vector(trainset$ClaimNb)
resptest <- as.vector(testset$ClaimNb)
lambda0 <- sum(trainset$ClaimNb)/sum(trainset$Exposure)

### Keras model
seed1 <- 100
use_session_with_seed(seed1)
Design <- layer_input(shape=c(25), dtype="float32", name="Design")
LogVol <- layer_input(shape=c(1), dtype="float32", name="LogVol")
Network <- Design %>%
  layer_dense(units=20, activation="tanh") %>%
  layer_dense(units=10, activation="tanh") %>%
  layer_dense(units=1, activation="linear", name="Network",
              weights=list(array(0,dim=c(10,1)), array(log(lambda0),dim=c(1))))
Response <- list(Network, LogVol) %>%
  layer_add(name="Add") %>%
  layer_dense(units=1, activation=k_exp, name="Response", trainable=FALSE,
              weights=list(array(1,dim=c(1,1)), array(0,dim=c(1))))
model <- keras_model(inputs=c(Design, LogVol), outputs=c(Response))
model %>% compile(optimizer=optimizer_nadam(), loss="poisson")

### Prepare features and responses for keras and fit the neural network model
xlearn = list(Design=featlearn, LogVol=vollearn)
ylearn = list(Response=resplearn)
xtest = list(Design=feattest, LogVol=voltest)
ytest = list(Response=resptest)
epochs <- 100   ### c) 1000
model %>% fit(x=xlearn, y=ylearn, epochs=epochs, verbose=1)

### Deviance statistics on training set
predtrain <- as.vector(model %>% predict(xlearn))
obstrain <- trainset$ClaimNb
(Deviancetrain <- 2*sum(log((obstrain/predtrain)^obstrain)-obstrain+predtrain))

### Deviance statistics on test set
predtestNN <- as.vector(model %>% predict(xtest))
obstest <- testset$ClaimNb
(Deviancetest <- 2*sum(log((obstest/predtestNN)^obstest)-obstest+predtestNN))
```

Solution 11.3 Claim Severity Modeling with GLM

(a) In this exercise we work with three tariff criteria. The first criterion (area code) has 6 risk characteristics:

$$\beta_{1,1}\ (\text{A}),\quad \beta_{1,2}\ (\text{B}),\quad \beta_{1,3}\ (\text{C}),\quad \beta_{1,4}\ (\text{D}),\quad \beta_{1,5}\ (\text{E})\quad \text{and}\quad \beta_{1,6}\ (\text{F}).$$

The second criterion (brand of the vehicle) has 11 risk characteristics:

$$\beta_{2,1}\ (\text{B1}),\ \beta_{2,2}\ (\text{B10}),\ \ldots,\ \beta_{2,6}\ (\text{B14}),\ \beta_{2,7}\ (\text{B2}),\ \ldots,\ \beta_{2,11}\ (\text{B6}).$$

The third criterion (diesel/fuel) has 2 risk characteristics:

$$\beta_{3,1}\ (\text{diesel})\quad \text{and}\quad \beta_{3,2}\ (\text{regular fuel}).$$

Therefore, we consider risk classes $(l_1, l_2, l_3), 1 \le l_1 \le 6, 1 \le l_2 \le 11, 1 \le l_3 \le 2$. We write n_{l_1,l_2,l_3} for the numbers of claims in risk class (l_1, l_2, l_3) and we only consider risk classes with $n_{l_1,l_2,l_3} > 0$. The n_{l_1,l_2,l_3} individual claim sizes in risk class (l_1, l_2, l_3) are denoted by $Y^{(i)}_{l_1,l_2,l_3}$,

$i = 1, \ldots, n_{l_1,l_2,l_3}$. We assume that all $Y^{(i)}_{l_1,l_2,l_3}$ are independent with

$$Y^{(i)}_{l_1,l_2,l_3} \sim \Gamma(\gamma, c_{l_1,l_2,l_3}),$$

where $\gamma > 0$ is a global shape parameter and $c_{l_1,l_2,l_3} > 0$ a risk-class dependent scale parameter. The total claim amount Y_{l_1,l_2,l_3} in risk class (l_1, l_2, l_3) is then given by

$$Y_{l_1,l_2,l_3} = \sum_{i=1}^{n_{l_1,l_2,l_3}} Y^{(i)}_{l_1,l_2,l_3} \sim \Gamma(\gamma n_{l_1,l_2,l_3}, c_{l_1,l_2,l_3}).$$

For the average claim amount X_{l_1,l_2,l_3} in risk class (l_1, l_2, l_3) we have

$$X_{l_1,l_2,l_3} = \frac{Y_{l_1,l_2,l_3}}{n_{l_1,l_2,l_3}} \sim \Gamma(\gamma n_{l_1,l_2,l_3}, c_{l_1,l_2,l_3} n_{l_1,l_2,l_3}).$$

We model

$$g\left(\mathbb{E}\left[X_{l_1,l_2,l_3}\right]\right) = \beta_0 + \beta_{1,l_1} + \beta_{2,l_2} + \beta_{3,l_3},$$

where $\beta_0 \in \mathbb{R}$ and where we use the log-link function, i.e. $g(\cdot) = \log(\cdot)$, which leads to a multiplicative structure. In order to get a unique solution, we set $\beta_{1,1} = \beta_{2,1} = \beta_{3,1} = 0$. Moreover, we define

$$\boldsymbol{\beta} = (\beta_0, \beta_{1,2}, \ldots, \beta_{1,6}, \beta_{2,2}, \ldots, \beta_{2,11}, \beta_{3,2})' \in \mathbb{R}^{r+1},$$

where $r = 16$. Similarly as in Exercises 10.1 and 11.1, we relabel the risk classes with the index $m \in \{1, \ldots, M\}$, where $M = 6 \cdot 11 \cdot 2 = 132$, define $\boldsymbol{X} = (X_1, \ldots, X_M)'$ and the design matrix $Z \in \mathbb{R}^{M \times (r+1)}$ that satisfies

$$\log \mathbb{E}[\boldsymbol{X}] = Z\boldsymbol{\beta},$$

where the logarithm is applied componentwise to $\mathbb{E}[\boldsymbol{X}]$. Let $m \in \{1, \ldots, M\}$. According to Section 7.4.4 of the lecture notes (version of February 7, 2022), X_m belongs to the exponential dispersion family with cumulant function $b(\theta) = -\log(-\theta)$ for $\theta < 0$, $\theta_m = -c_m/\gamma$, $w_m = n_m$ and dispersion parameter $\phi = 1/\gamma$, i.e. we have

$$[Z\boldsymbol{\beta}]_m = \log \mathbb{E}[X_m] = \log \frac{\gamma n_m}{c_m n_m} = \log \frac{\gamma}{c_m} = \log\left(-\frac{1}{\theta_m}\right),$$

where $[Z\boldsymbol{\beta}]_m$ denotes the m-th element of the vector $Z\boldsymbol{\beta}$. Summarizing, we assume that $X_1, \ldots, X_M$ are independent with

$$X_m \sim \mathrm{EDF}\left(\theta_m = -\exp\{-[Z\boldsymbol{\beta}]_m\}, \phi = 1/\gamma, w_m = n_m, b(\theta) = -\log(-\theta)\right),$$

for all $m \in \{1, \ldots, M\}$. As $b(\theta) = -\log(-\theta)$, we have $b'(\cdot) = -1/\theta$, where b' denotes the first derivative of b. In particular, the log-link function $g(\cdot) = \log(\cdot)$ is not equal to the canonical link function $h(\mu) = (b')^{-1}(\mu) = -1/\mu$ in the gamma model. Therefore, we cannot use equation (7.35) of the lecture notes (version of February 7, 2022) in order to determine the MLE $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ of $\boldsymbol{\beta}$. However, according to Proposition 7.12 of the lecture notes (version of February 7, 2022), the MLE $\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ of $\boldsymbol{\beta}$ is the solution of

$$Z' V_{\boldsymbol{\theta}} \exp\{Z\boldsymbol{\beta}\} \overset{!}{=} Z' V_{\boldsymbol{\theta}} \boldsymbol{X}, \tag{3}$$

where the weight matrix $V_{\boldsymbol{\theta}}$ is given by $V_{\boldsymbol{\theta}} = \mathrm{diag}(-\theta_1 n_1, \ldots, -\theta_M n_M)$. Note that assuming a constant scale parameter γ for all risk cells $m = 1, \ldots M$, the dispersion parameter $\phi = 1/\gamma$ cancels from the weight matrix defined on page 205 of the lecture notes (version of February 7, 2022). Equation (3) has to be solved numerically. We refer to Listing 7 for the R code used in this exercise. The resulting MLEs of the parameters $\beta_0, \beta_{2,3}, \beta_{2,7}, \beta_{3,2}$ that are (statistically) significantly different from 0 (on a 10% level) are given in the first row of Table 3. We observe that we expect higher claim sizes in regions B11 and B2, compared to the reference region B1. Moreover, claim sizes tend to be higher if a car with regular fuel is involved compared to a diesel car. We remark that the parameters corresponding to the individual categorical levels of the covariate area code are not (statistically) significantly different from 0 (on a 10% level). However, this does not mean that the covariate area code itself is not statistically significant, see part (b).

	$\widehat{\beta}_0$	$\widehat{\beta}_{2,3}$	$\widehat{\beta}_{2,7}$	$\widehat{\beta}_{3,2}$
MLE	7.6116	0.5288	0.1991	0.1846
p-value	≈ 0	0.0585	0.0898	0.0321

Table 3: MLEs of the statistically significant parameters and corresponding p-values.

(b) The estimates of the expected values of X_m are given by

$$\widehat{\mu}_m = b'(\widehat{\theta}_m) = -\widehat{\theta}_m^{-1} = \exp\left\{\left[Z\widehat{\boldsymbol{\beta}}^{\mathrm{MLE}}\right]_m\right\},$$

for all $m = 1, \ldots, M$, and we write $\widehat{\boldsymbol{\mu}} = (\widehat{\mu}_1, \ldots, \widehat{\mu}_M)'$. According to page 207 of the lecture notes (version of February 7, 2022), the deviance statistics is given by

$$\begin{aligned} D(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}) &= 2\sum_{m=1}^{M} w_m(X_m h(X_m) - b[h(X_m)] - X_m h(\widehat{\mu}_m) + b[h(\widehat{\mu}_m)]) \\ &= 2\sum_{m=1}^{M} n_m(-1 - \log X_m + X_m/\widehat{\mu}_m + \log \widehat{\mu}_m). \end{aligned}$$

Estimating ϕ by

$$\widehat{\phi}_D = \frac{D(\boldsymbol{X}, \widehat{\boldsymbol{\mu}})}{M - r - 1},$$

see page 208 of the lecture notes (version of February 7, 2022), we have for the scaled deviance statistics

$$D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}) = 2(\ell_{\boldsymbol{X}}(\boldsymbol{X}) - \ell_{\boldsymbol{X}}(\widehat{\boldsymbol{\mu}})) = \frac{2}{\widehat{\phi}_D}\sum_{m=1}^{M} n_m(-1 - \log X_m + X_m/\widehat{\mu}_m + \log \widehat{\mu}_m).$$

We define the null hypothesis

$$H_0: \beta_{1,2} = \cdots = \beta_{1,6} = 0.$$

Thus, in the reduced model we set the above $p = 5$ variables equal to 0. Then, we can recalculate $\widehat{\boldsymbol{\beta}}_{H_0}^{\mathrm{MLE}}$ for this reduced model and define

$$\widehat{\boldsymbol{\mu}}_{H_0} = \exp\left\{Z_{H_0}\widehat{\boldsymbol{\beta}}_{H_0}^{\mathrm{MLE}}\right\},$$

where Z_{H_0} is the design matrix in the reduced model. According to formula (7.40) of the lecture notes (version of February 7, 2022), the test statistic

$$X^2 = D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}_{H_0}) - D^*(\boldsymbol{X}, \widehat{\boldsymbol{\mu}}).$$

has approximately a χ^2-distribution with df $= p = 5$ degrees of freedom. We get

$$X^2 \approx 14.917,$$

which corresponds to a p-value of approximately 1.07%. Thus, using the χ^2-test we can reject H_0 at significance level of 5%. We can conclude that there seems to be no statistical evidence that the area code could be omitted as tariff criterion, even though the individual categorical levels of the covariate area code are not (statistically) significantly different from 0 (on a 10% level), see part (a).

Listing 7: R code for Exercise 11.3.

```
### Apply GLM
d.glm <- glm(ClaimAmount ~ Area+VehBrand+VehGas, data=data, weights=ClaimNb,
             family=Gamma(link="log"))
summary(d.glm)

### Calculate the deviance statistics of the full model
D.full <- d.glm$deviance

### Fit the reduced model and calculate the deviance statistics
d.glm.2 <- glm(ClaimAmount ~ VehGas+VehBrand, data=data, weights=ClaimNb,
               family=Gamma(link="log"))
D.reduced <- d.glm.2$deviance

### Calculate the test statistic X^2 and the corresponding p-value
phi.est <- d.glm$deviance/d.glm$df.residual
round((X.2 <- D.reduced/phi.est-D.full/phi.est),3)
pchisq(X.2, 5, lower.tail=FALSE)
```

Solution 11.4 Neural Networks and Gradient Descent

(a) We model the regression function $\alpha : \mathcal{Z} \to \mathbb{R}_+$ with a a single hidden layer neural network with $r_1 \in \mathbb{N}$ hidden neurons. Our feature space is $\mathcal{Z} \subset \mathbb{R}^{r_0+1}$ with $r_0 = 1$, i.e. we have input dimension $r_0 = 1$. We assume that the first component of the covariates $\boldsymbol{z} = (1, z) \in \mathcal{Z}$ is equal to 1 for modeling an intercept. We define the parameter vectors

$$\boldsymbol{\beta}_j^{(1)} = \left(\beta_{j,0}^{(1)}, \beta_{j,1}^{(1)}\right) \in \mathbb{R}^{r_0+1},$$

for all $j = 1, \ldots, r_1$, and

$$\boldsymbol{\beta}^{(2)} = \left(\beta_0^{(2)}, \beta_1^{(2)}, \ldots, \beta_{r_1}^{(2)}\right) \in \mathbb{R}^{r_1+1}.$$

The hyperbolic tangent activation function is given by

$$\psi(x) = \tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}, \qquad \text{for } x \in \mathbb{R}.$$

For covariates $\boldsymbol{z} = (1, z) \in \mathcal{Z}$, the activations in the hidden layer are then given by

$$\boldsymbol{q}^{(1)}(\boldsymbol{z}) = \left(1, q_1^{(1)}(\boldsymbol{z}), \ldots, q_{r_1}^{(1)}(\boldsymbol{z})\right),$$

where

$$q_j^{(1)}(\boldsymbol{z}) = \psi\left(\langle \boldsymbol{\beta}_j^{(1)}, \boldsymbol{z}\rangle\right) = \psi\left(\beta_{j,0}^{(1)} + \beta_{j,1}^{(1)} z\right),$$

for all $j = 1, \ldots, r_1$. Since the codomain of $\alpha(\cdot)$ has to be $\mathbb{R}_+$, we define a log-linear regression approach as follows

$$\alpha(\boldsymbol{z}) = \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}) = \exp\left\{\langle \boldsymbol{\beta}^{(2)}, \boldsymbol{q}^{(1)}(\boldsymbol{z})\rangle\right\} = \exp\left\{\beta_0^{(2)} + \sum_{j=1}^{r_1} \beta_j^{(2)} q_j^{(1)}(\boldsymbol{z})\right\},$$

with resulting network parameter

$$\boldsymbol{\beta} \;=\; \left(\boldsymbol{\beta}_1^{(1)}, \ldots, \boldsymbol{\beta}_{r_1}^{(1)}, \boldsymbol{\beta}^{(2)}\right) \;\in\; \mathbb{R}^{\varrho}$$

having dimension $\varrho = (r_0 + 1)r_1 + r_1 + 1 = (1 + 1)r_1 + r_1 + 1 = 3r_1 + 1$.

(b) As we assume independent Pareto distributions with threshold $\theta > 0$ and covariate-dependent tail index $\alpha(\boldsymbol{z}_m) > 0$ for the data $\boldsymbol{Y} = (Y_1, \ldots, Y_M)$ with corresponding covariates $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$, the joint log-likelihood function $\ell_{\boldsymbol{Y}}(\boldsymbol{\beta})$ is given by

$$\begin{aligned}\ell_{\boldsymbol{Y}}(\boldsymbol{\beta}) \;&=\; \log \prod_{m=1}^{M} \frac{\alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\theta} \left(\frac{Y_m}{\theta}\right)^{-\alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)-1} \\ &=\; \sum_{m=1}^{M} \log \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m) - \log\theta - [\alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m) + 1] \log \frac{Y_m}{\theta}.\end{aligned}$$

In the saturated model we assume one parameter α_m per observation m. This parameter α_m is determined by maximizing the individual MLE for observation m, i.e. we have to maximize

$$g(\alpha_m) \;\overset{\text{def}}{=}\; \log(\alpha_m) - \log\theta - (\alpha_m + 1)\log\frac{Y_m}{\theta}$$

with respect to α_m, for all $m = 1, \ldots, M$. If we take the derivative with respect to α_m, we get

$$\frac{\partial g(\alpha_m)}{\partial \alpha_m} \;=\; \frac{1}{\alpha_m} - \log\frac{Y_m}{\theta},$$

for all $m = 1, \ldots, M$. This is equal to 0 if and only if

$$\alpha_m \;=\; \frac{1}{\log\frac{Y_m}{\theta}}, \tag{4}$$

for all $m = 1, \ldots, M$. For the second derivative of $g(\alpha_m)$ with respect to α_m we get

$$\frac{\partial^2 g(\alpha_m)}{\partial \alpha_m^2} \;=\; -\frac{1}{\alpha_m^2} \;<\; 0,$$

for all $m = 1, \ldots, M$. That is, in the saturated model we have parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ with α_m given as in (4), for all $m = 1, \ldots, M$. For the log-likelihood of the saturated model we then have

$$\begin{aligned}\ell_{\boldsymbol{Y}}(\boldsymbol{Y}) \;&=\; \sum_{m=1}^{M} \log\frac{1}{\log\frac{Y_m}{\theta}} - \log\theta - \left(\frac{1}{\log\frac{Y_m}{\theta}} + 1\right)\log\frac{Y_m}{\theta} \\ &=\; \sum_{m=1}^{M} -\log\log\frac{Y_m}{\theta} - \log\theta - 1 - \log\frac{Y_m}{\theta}.\end{aligned}$$

Finally, the (scaled) deviance statistics is given by

$$\mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta}) \;=\; 2(\ell_{\boldsymbol{Y}}(\boldsymbol{Y}) - \ell_{\boldsymbol{Y}}(\boldsymbol{\beta})) \;=\; 2\sum_{m=1}^{M} -\log\log\frac{Y_m}{\theta} - 1 - \log\alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m) + \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)\log\frac{Y_m}{\theta}.$$

(c) A neural network model with a large number of hidden neurons is heavily over-parametrized. Therefore, a maximum likelihood estimator would lead to overfitting of the model to the data (in-sample). Thus, we are only interested in finding a sufficiently good approximation which has also a good out-of-sample performance. We believe that such a 'good' parametrization can be reached for example by the gradient descent method.

(d) For the derivative of the hyperbolic tangent activation function ψ we have

$$\begin{aligned}\frac{\partial \psi(x)}{\partial x} &= \frac{\partial}{\partial x}\frac{e^{2x}-1}{e^{2x}+1} = \frac{2e^{2x}(e^{2x}+1)-2e^{2x}(e^{2x}-1)}{(e^{2x}+1)^2} = \frac{4e^{2x}}{(e^{2x}+1)^2} \\ &= \frac{(e^{2x}+1)^2-(e^{2x}-1)^2}{(e^{2x}+1)^2} = 1-\psi^2(x).\end{aligned}$$

In the gradient descent optimization algorithm the goal is to decrease a given loss function by iteratively updating the model parameters. In our case we would like to decrease the (scaled) deviance statistics $\mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta})$ derived in part (b) above. To this end, for a given $\boldsymbol{\beta}$, we move in the direction of the maximal local decrease of the deviance statistics, i.e. in the direction of the negative gradient $\nabla_{\boldsymbol{\beta}}\mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta})$ of the deviance statistics. We calculate

$$\nabla_{\boldsymbol{\beta}}\mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta}) = \frac{\partial \mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 2\sum_{m=1}^{M}\left[-\frac{1}{\alpha_{\boldsymbol{\theta}}(\boldsymbol{z}_m)}+\log\frac{Y_m}{\theta}\right]\frac{\partial \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\partial \boldsymbol{\beta}},$$

where we have

$$\begin{aligned}\frac{\partial \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\partial \beta_{j,0}^{(1)}} &= \alpha_{\boldsymbol{\theta}}(\boldsymbol{z}_m)\beta_j^{(2)}\left(1-\left[q_j^{(1)}(\boldsymbol{z})\right]^2\right), \\ \frac{\partial \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\partial \beta_{j,1}^{(1)}} &= \alpha_{\boldsymbol{\theta}}(\boldsymbol{z}_m)\beta_j^{(2)}\left(1-\left[q_j^{(1)}(\boldsymbol{z})\right]^2\right)z_m, \\ \frac{\partial \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\partial \beta_0^{(2)}} &= \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}), \\ \frac{\partial \alpha_{\boldsymbol{\beta}}(\boldsymbol{z}_m)}{\partial \beta_j^{(2)}} &= \alpha_{\boldsymbol{\beta}}(\boldsymbol{z})q_j^{(1)}(\boldsymbol{z}),\end{aligned}$$

for all $m = 1,\dots,M$ and $j = 1,\dots,r_1$. In one single step of the gradient descent optimization algorithm we have the update

$$\boldsymbol{\beta} \longrightarrow \boldsymbol{\beta} - \rho\nabla_{\boldsymbol{\beta}}\mathcal{L}_{\boldsymbol{Y}}(\boldsymbol{\beta}),$$

where $\rho > 0$ is the so-called learning rate. Note that one should carefully choose an appropriate stopping time of the algorithm in order to prevent from overfitting; and one should also carefully choose $\rho > 0$ because the gradient descent steps lead to a decrease locally.