Implementation Details and Further Simulations
-
Prediction intervals for load sharing systems in accelerated life testing

Kevin Leckey* Christine H. Müller* Sebastian Szugat*
Reinhard Maurer†

April 4, 2019

1 Experimental data

The data of the fatigue tests performed at the TU Dortmund University can be found in the R-package `loadshare` using the command `loadData()`. An overview over the experiments is given in the table below.

Table 1: The experiments with the stress levels and the observed number of breaks and load cycles until failure in millions.

beam	stress range s in MPa	number of observed breaks	number of load cycles until last observed break (in millions)
TR01	200	15	3.39
TR02	455	9	0.21
TR03	200	12	3.47
TR04	150	6	5.21
TR05	98	3	7.38
SB01	200	17	5.66
SB02	100	18	16.19
SB03	60	18	85.16
SB04	80	19	21.63
SB05	80	19	66.47
SB06	50	1	28.62

*Faculty of Statistics, TU Dortmund University, D-44227 Dortmund, Germany
†Faculty of Architecture and Civil Engineering, TU Dortmund University, D-44227 Dortmund, Germany

2 R implementation details

Several R functions for methods provided in the main article are based on slighted transformed representations. These transformations are made for a variety of different reasons, mostly either to improve the numeric stability or to speed up the calculations. After a brief reminder on the notation and model (more thorough definitions can be found in the main article), we present the calculations made to obtain the new representations. The corresponding functions of our package `loadshare` are stated at the beginning of each paragraph. Note that the function `predfail` can be used to compute prediction intervals based on any of the following methods by specifying the variable `method` in the function accordingly: `naive, 3depth, wald, LR, delta`.

Recall that $\boldsymbol{w} = (w_{1,0}, \ldots, w_{I_0,0}, \ldots, w_{1,J}, \ldots, w_{I_J,J})$ denotes the observed realizations of the (exponentially distributed) waiting times between failures. Also recall that f_λ is the density function of the $\text{Exp}(\lambda)$ distribution. Let $N = \sum_{j=0}^{J} I_j$ denote the total number of observations. Throughout the section we assume that $\lambda_{\boldsymbol{\theta}}(i,s)$ is chosen according to the Basquin model, i.e.

$$\lambda_{\boldsymbol{\theta}}(i,s) = \mathrm{e}^{-\theta_1} \left(\frac{I \cdot s}{I - i} \right)^{\theta_2}.$$

Maximum likelihood estimation (MLE). The computation of the MLE is implemented in the `estML` function. Since multivariate optimization in R via `optim` is more time consuming and less accurate than working with univariate functions, our implementation of the MLE in the Basquin model computes the optimal parameter $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \widehat{\theta}_2)$ in two steps. In the first step, θ_2 is fixed and the (log-)likelihood function is optimized (analytically) with respect to θ_1. To this end, note that the log-likelihood function in the Basquin model is given by

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{w}}(\boldsymbol{\theta}) &= \sum_{j=0}^{J} \sum_{i=1}^{I_j} \ln\left(f_{\lambda_{\boldsymbol{\theta}}(i-1,s_j)}(w_{i,j}) \right) \\
&= -\theta_1 N + \theta_2 \sum_{j=0}^{J} \sum_{i=1}^{I_j} \ln\left(\frac{I s_j}{I - i + 1} \right) - \mathrm{e}^{-\theta_1} \sum_{j=0}^{J} \sum_{i=1}^{I_j} \left(\frac{I s_j}{I - i + 1} \right)^{\theta_2} w_{i,j}
\end{aligned}$$

For fixed θ_2, let $\mathcal{L}^1_{\boldsymbol{w},\theta_2}$ be the function given by $\mathcal{L}^1_{\boldsymbol{w},\theta_2}(\theta_1) = \mathcal{L}_{\boldsymbol{w}}(\theta_1, \theta_2)$. It is straightforward to compute the (unique) root of the first derivative of $\mathcal{L}^1_{\boldsymbol{w},\theta_2}$ which reveals that the function attains its global maximum at

$$\theta_1^*(\theta_2) = \ln\left(\frac{1}{N} \sum_{j=0}^{J} \sum_{i=1}^{I_j} \left(\frac{I s_j}{I - i + 1} \right)^{\theta_2} w_{i,j} \right). \tag{1}$$

Therefore, it is sufficient to find the position $\widehat{\theta}_2$ of the global maximum of

$$\mathcal{L}_2 : [0, \infty) \to \mathbb{R}, \quad \theta_2 \mapsto \mathcal{L}_{\boldsymbol{w}}(\theta_1^*(\theta_2), \theta_2).$$

The first derivative $\dot{\mathcal{L}}_2$ of $\mathcal{L}_2$ can be computed explicitly and is given by

$$\dot{\mathcal{L}}_2(\theta_2) = -\dot{\theta}_1^*(\theta_2)N + \sum_{j=0}^{J}\sum_{i=1}^{I_j} \ln\left(\frac{Is_j}{I-i+1}\right),$$
$$\dot{\theta}_1^*(\theta_2) = \frac{\sum_{j=0}^{J}\sum_{i=1}^{I_j} \ln\left(\frac{Is_j}{I-i+1}\right)\left(\frac{Is_j}{I-i+1}\right)^{\theta_2} w_{i,j}}{\sum_{j=0}^{J}\sum_{i=1}^{I_j}\left(\frac{Is_j}{I-i+1}\right)^{\theta_2} w_{i,j}}.$$

The solution $\widehat{\theta}_2$ to $\dot{\mathcal{L}}_2(\widehat{\theta}_2) = 0$ is computed numerically using the `uniroot` function in R. Finally, the MLE is retrieved by setting $\widehat{\boldsymbol{\theta}} = (\theta_1^*(\widehat{\theta}_2), \widehat{\theta}_2)$.

Information matrix estimation. The estimated information matrix can be computed via `predImat_basq`. Recall that the information matrix is estimated via (Equation (4) in the article):

$$\boldsymbol{I}(\boldsymbol{\theta}) := \sum_{j=0}^{J}\sum_{i=1}^{I_j} \frac{1}{\lambda_{\boldsymbol{\theta}}(i-1, s_j)^2}\, \dot{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}(i-1, s_j)\, \dot{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}(i-1, s_j)^\top.$$

Note that gradient of the rates in the Basquin model is given by

$$\dot{\lambda}_{\boldsymbol{\theta}}(i, s) = \begin{pmatrix} -1 \\ \ln\left(\frac{Is}{I-i}\right) \end{pmatrix} \lambda_{\boldsymbol{\theta}}(i, s). \tag{2}$$

Hence the estimator $\boldsymbol{I}(\boldsymbol{\theta})$ becomes

$$\boldsymbol{I}(\boldsymbol{\theta}) = \begin{pmatrix} N & -\sum_{j=0}^{J}\sum_{i=0}^{I_j-1} \ln\left(\frac{Is_j}{I-i}\right) \\ -\sum_{j=0}^{J}\sum_{i=0}^{I_j-1} \ln\left(\frac{Is_j}{I-i}\right) & \sum_{j=0}^{J}\sum_{i=0}^{I_j-1} \left(\ln\left(\frac{Is_j}{I-i}\right)\right)^2 \end{pmatrix},$$

which is the formula used for the R-implementation of $\boldsymbol{I}(\boldsymbol{\theta})$. Note that this estimation does not depend on $\boldsymbol{\theta}$. The inverse of a 2×2 matrix (which is required for the δ-method) can be computed efficiently using Cramer's rule.

Sign depth (3-depth). The (normalized) 3-depth is implemented as `normed3depth`. Kustosz and Szugat (2016) provide the asymptotic α-quantiles of the rescaled 3-depth for all α with $1000\alpha \in \mathbb{N}$. In the rare occasions where we need quantiles outside this grid, we approximate them by linear interpolation. Since these asymptotic quantiles can be a poor approximation for datasets with a small size N, we use the exact quantiles (obtained via simulation) rather than asymptotic ones if $N \leq 100$. The simulated quantiles are taken from the unpublished work Horn and Müller (2019).

Confidence sets. Confidence sets can be computed via the function `ConSet`. Rather than approximating the whole confidence $\widehat{C}_\alpha$, e.g., by listing all its points inside a chosen grid, we compute a $3 \times m$ matrix M_C, $m \geq 1$, that approximates the "border" of the confidence set in order to reduce the total number of points that need to be listed. More precisely, each column (a_i, b_i, c_i) is chosen in such a way that

1. $(a_i, c_i) \in \widehat{C}_\alpha$ and $(b_i, c_i) \in \widehat{C}_\alpha$,

2. $a_i = \min\{\theta_1; (\theta_1, c_i) \in \widehat{C}_\alpha\}$ and $b_i = \max\{\theta_1; (\theta_1, c_i) \in \widehat{C}_\alpha\}$,

where the minimum and maximum (in 2.) is only verified up to a chosen precision $\Delta > 0$ (the default choice is $\Delta = 0.05$). Moreover, the potential candidates for c_i are given by a grid with mesh width Δ (by default, only candidates inside the interval $[0, 5]$ are checked; if no valid c_i can be found or if one c_i turns out to be at the border of $[0, 5]$ then the interval is automatically expanded).

Prediction intervals based on confidence sets. The conversion of a confidence set to a prediction interval can be done via `predUnionConSet`. These prediction intervals are based on the simpler formula in Equation (13) of the main article, i.e.

$$\left[\min\{b_{\alpha(2)/2}(\theta);\ \theta \in \widehat{C}_{\alpha(1)}\}, \max\{b_{1-\alpha(2)/2}(\theta);\ \theta \in \widehat{C}_{\alpha(1)}\}\right].$$

In order to simplify this prediction interval, note that, for every $\theta_2 > 0$ and $\alpha \in (0, 1)$, the function $\theta_1 \mapsto b_\alpha(\theta_1, \theta_2)$ is increasing in θ_1. This is due to the fact that

$$\sum_{i=1}^{n} E_i/\lambda_i \sim HypoExp(\lambda_1, \ldots, \lambda_n) \quad \text{for i.i.d. } E_1, \ldots, E_n \sim Exp(1),$$

which, in combination with $\lambda_{(\theta_1,\theta_2)}(i, s) = \mathrm{e}^{-\theta_1}\, \lambda_{(0,\theta_2)}(i, s)$, implies $b_\alpha(\theta_1, \theta_2) = \mathrm{e}^{\theta_1}\, b_\alpha(0, \theta_2)$. Therefore we may restrict the minimum/maximum above to only be taken over all $\boldsymbol{\theta}$ on the border of $\widehat{C}_{\alpha(1)}$. This is done using the matrix representation M_C for $\widehat{C}_{\alpha(1)}$ given in the previous paragraph.

Prediction intervals based on the δ-method. A prediction based on the δ-method can be done via `pred_delta`. The implementation is based on the following calculations. Let $F_{W,\boldsymbol{\theta}}$ be the cdf of the Hypoexponential distribution. Recall that $H_\alpha(\boldsymbol{\theta}, b) = F_{\boldsymbol{W},\boldsymbol{\theta}}(b) - \alpha$ and that the implicit function theorem implies

$$\dot{\boldsymbol{b}}_\alpha(\boldsymbol{\theta}) = -\left(\frac{\partial}{\partial \tilde{b}} H_\alpha(\tilde{\boldsymbol{\theta}}, \tilde{b})\Big|_{(\tilde{\boldsymbol{\theta}},\tilde{b})=(\boldsymbol{\theta},b_\alpha(\boldsymbol{\theta}))}\right)^{-1} \frac{\partial}{\partial \tilde{\boldsymbol{\theta}}} H_\alpha(\tilde{\boldsymbol{\theta}}, \tilde{b})\Big|_{(\tilde{\boldsymbol{\theta}},\tilde{b})=(\boldsymbol{\theta},b_\alpha(\boldsymbol{\theta}))}$$

Whenever possible, calculations involving $F_{W,\boldsymbol{\theta}}$ are based on the matrix representation (Equation (10) in the main article) to avoid numeric instabilities:

$$F_{W,\boldsymbol{\theta}}(w) = 1 - \boldsymbol{e}_1^\top \exp(\boldsymbol{D}w)\mathbf{1}_{I_c - I_0} = 1 - \boldsymbol{e}_1^\top \sum_{k=0}^{\infty} \frac{\boldsymbol{D}^k w^k}{k!} \mathbf{1}_{I_c - I_0}, \tag{3}$$

where the matrix $\boldsymbol{D}$ can be found in the main article. In particular, a matrix representation for the the density function can be obtained by taking the derivative with respect to w. This yields

$$\frac{\partial}{\partial \tilde{b}} H_\alpha(\tilde{\boldsymbol{\theta}}, \tilde{b})\Big|_{(\tilde{\boldsymbol{\theta}},\tilde{b})=(\boldsymbol{\theta},b)} = \frac{\partial}{\partial \tilde{b}} F_{W,\boldsymbol{\theta}}(\tilde{b})\Big|_{\tilde{b}=b} = -\mathbf{e}_1^\top \mathbf{D} \exp(\mathbf{D}b)\mathbf{1}_{I_c - I_0}.$$

However, the derivative of H_α with respect to $\boldsymbol{\theta}$ cannot be deduced easily from (3). The R-implementation of this derivative is based on the following calculations. First recall that the cdf $F_{W,\boldsymbol{\theta}}$ of $HypoExp(\lambda_{\boldsymbol{\theta}}(I_0,s),\dots\lambda_{\boldsymbol{\theta}}(I_1,s))$ is given by

$$F_{W,\boldsymbol{\theta}}(b) = \sum_{i=I_0}^{I_1} a_i(\boldsymbol{\theta})\ \left(1-\mathrm{e}^{-b\,\lambda_{\boldsymbol{\theta}}(i,s)}\right) = 1-\sum_{i=I_0}^{I_1} a_i(\boldsymbol{\theta}) e^{-b\,\lambda_{\boldsymbol{\theta}}(i,s)},$$

$$a_i(\boldsymbol{\theta}) = \prod_{k=I_0,k\neq i}^{I_1} \frac{\lambda_{\boldsymbol{\theta}}(k,s)}{\lambda_{\boldsymbol{\theta}}(k,s)-\lambda_{\boldsymbol{\theta}}(i,s)} = \prod_{k=I_0,k\neq i}^{I_1} \frac{1}{1-\lambda_{\boldsymbol{\theta}}(i,s)/\lambda_{\boldsymbol{\theta}}(k,s)}.$$

Moreover, the choice for $\lambda_{\boldsymbol{\theta}}$ yields $\lambda_{\boldsymbol{\theta}}(i,s)/\lambda_{\boldsymbol{\theta}}(k,s) = \left(\frac{I-k}{I-i}\right)^{\theta_2}$. Hence

$$\dot{a}_i(\boldsymbol{\theta}) := \frac{\partial a_i}{\partial \boldsymbol{\theta}} = \left(0,\ a_i(\boldsymbol{\theta}) \sum_{j=I_0,j\neq i}^{I_1} \frac{\ln\left(\frac{I-j}{I-i}\right)\left(\frac{I-j}{I-i}\right)^{\theta_2}}{1-\left(\frac{I-j}{I-i}\right)^{\theta_2}}\right)^\top.$$

Combined with the derivative (2) for $\lambda_{\boldsymbol{\theta}}$ we thus obtain

$$\frac{\partial}{\partial\tilde{\boldsymbol{\theta}}} H_\alpha(\tilde{\boldsymbol{\theta}},\tilde{b})\bigg|_{(\tilde{\boldsymbol{\theta}},\tilde{b})=(\boldsymbol{\theta},b)} = b\sum_{i=I_0}^{I_1} \dot{\lambda}_{\boldsymbol{\theta}}(i,s) a_i(\boldsymbol{\theta})\,\mathrm{e}^{-b\lambda_{\boldsymbol{\theta}}}(i,s) - \sum_{i=I_0}^{I_1} \dot{a}_i(\boldsymbol{\theta}) e^{-b\lambda_{\boldsymbol{\theta}}(i,s)},$$

which is the formula used in the implementation of $\dot{b}_\alpha(\boldsymbol{\theta})$.

3 More results of the simulation study

This section contains some additional simulations to deduce the influence of one-sided outliers. Recall that the simulation study in the main article was based on the following parameters for $K \geq 1$:

$$I = 10, \quad \boldsymbol{\theta} = (28,3), \quad \lambda_{\boldsymbol{\theta}}(i,s) = \mathrm{e}^{-\theta_1}\left(\frac{sI}{I-i}\right)^{\theta_2},$$

$$s_j = \begin{cases} 200, & \text{for } j = 1,\dots,K, \\ 100, & \text{for } j = K+1,\dots,2K, \\ 80, & \text{for } j = 2K+1,\dots,3K, \\ 60, & \text{for } j = 3K+1, \end{cases}$$

$$W_{i,j} \sim Exp(\lambda_{\boldsymbol{\theta}}(i-1,s_j)) \quad \text{for } i = 1,\dots,6 \text{ and } j = 1,\dots 3K+1.$$

We assumed that the realization of the vector

$$\mathbf{W}_* = (W_{1,1},\dots,W_{6,1},\dots,W_{1,3K},\dots,W_{6,3K},W_{1,3K+1},\dots,W_{3,3K+1})$$

was observed and used to predict the the sixth wire break of the new experiment, i.e.

$$w_{1,3K+1} + w_{2,3K+1} + w_{3,3K+1} + W_{4,3K+1} + W_{5,3K+1} + W_{6,3K+1},$$

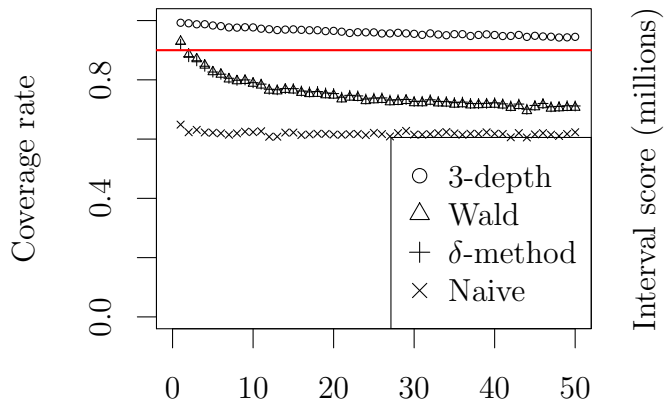
where $w_{1,3K+1}, w_{2,3K+1}, w_{3,3K+1}$ denote the realizations of $W_{1,3K+1}, W_{2,3K+1}, W_{3,3K+1}$. Such prediction intervals are computed by four different approaches: The naive/plug-in approach, confidence set based approaches using either Wald's test or the 3-depth test to compute the sets, and the δ-method.

In order to compare the outlier robustness of these methods, we considered the effect of *symmetric outliers* in the main article, i.e. each waiting time had a 20% chance to become an outlier and outliers were created by either increasing its value by a factor of 10 or by decreasing the waiting time to the absolute value 1. In order to compare the symmetric case with one-sided outliers, we took Figure 6 (now Figure 3) from the main article and added two new figures (Figure 1 and Figure 2) with different outlier scenarios. In summary, outliers in the the three different figures are produced as follows: Each waiting time $W_{i,j}$ in $\mathbf{W}_*$ has a 20% chance to be replaced by another time $\widetilde{W}_{i,j}$ chosen according to one of the following three different scenarios:

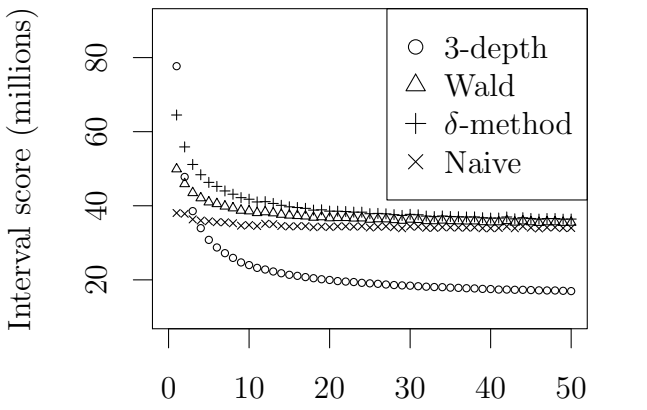
1. Upper outliers: $\widetilde{W}_{i,j} = 10W_{i,j}$,
2. Lower outliers: $\widetilde{W}_{i,j} = 1$,
3. Symmetric outliers: $P(\widetilde{W}_{i,j} = 10W_{i,j}) = P(\widetilde{W}_{i,j} = 1) = \frac{1}{2}$

Note that the waiting times $W_{4,3K+1}, W_{5,3K+1}, W_{6,3K+1}$ will never be changed to outliers since outliers in the prediction will barely ever be predicted correctly.

Lower outliers cause all methods, most noticeably the naive approach, to perform slightly worse than for uncontaminated data but the score remains within a range of 10-15 million for sufficiently large K (which is a value similar to the uncontaminated data in Figure 5 of the main article). Upper outliers, however, affect all methods but the robust 3-depth drastically as can be seen in Figure 1.

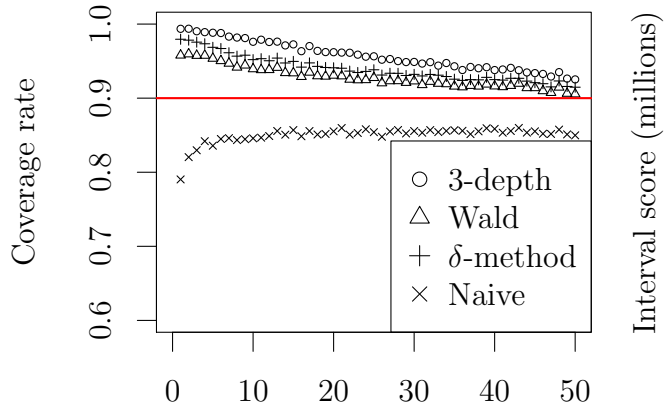


(a) Coverage rate

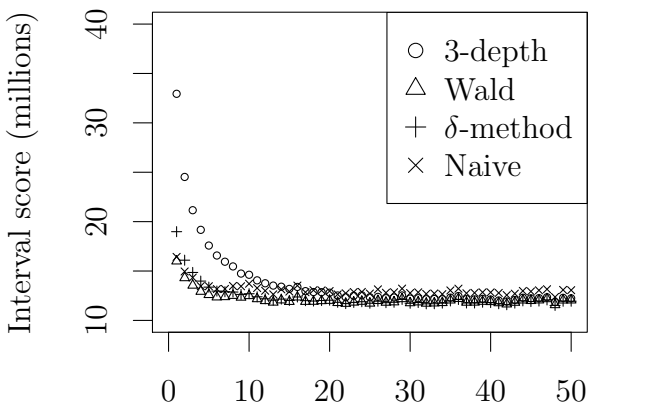


(b) Interval score

Figure 1: Coverage rates (a) and interval scores (b) of different methods applied to contaminated data in the contamination scenario 1 (upper outliers)

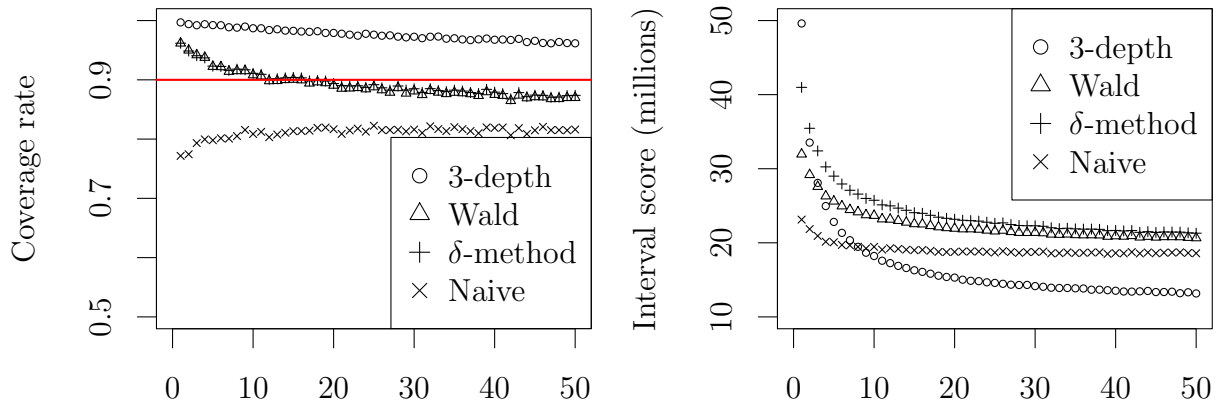


(a) Coverage rate



(b) Interval score

Figure 2: Coverage rates (a) and interval scores (b) of different methods applied to contaminated data in the contamination scenario 2 (lower outliers)



(a) Coverage rate

(b) Interval score

Figure 3: Coverage rates (a) and interval scores (b) of different methods applied to contaminated data in the contamination scenario 3 (symmetric outliers)

References

Horn, M. and Müller, C. H. (2019). Tests based on sign depth for multiple regression. *In preparation.*

Kustosz, C. and Szugat, S. (2016). *rexpar: Simplicial Depth for Explosive Autoregressive Processes.* R package version 1.1.