ML-based inference for complex dependence models

Lídia André [1] Jennifer Wadsworth [2] Raphaël Huser [3]

[1]STOR-i CDT, Lancaster University, UK
[2]Department of Mathematics and Statistics, Lancaster University, UK
[3]King Abdullah University of Science and Technology (KAUST), Saudi Arabia

STOR-i excellence with impact

1. Motivation

- Various models in the multivariate extremes literature allow for interpolation between two dependence classes, when the variables are extreme together **(asymptotic dependence)**, or not **(asymptotic independence)**
- Such flexible models often rely on numerical integration and inversion of functions, which makes the evaluation of their likelihood **computationally costly**. This might limit the use of these models in practice as performing inference is not computationally efficient
- Likelihood-free approaches, such as **neural Bayes estimators [2]**, are appealing to perform inference on the vector of parameters of expensive models

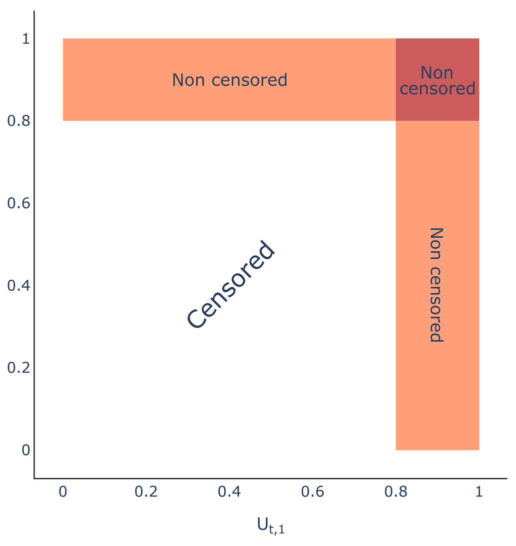
Goal: toolbox for simple fitting and comparison of complex dependence models

2. Modelling setup

- $\boldsymbol{U}_t = (U_{t,1}, U_{t,2}) \sim C_{\boldsymbol{\theta}}$ where C_θ is a copula model with parameters $\boldsymbol{\theta}$, and $U_{t,j} \sim \text{Unif}(0,1)$ for $j = 1, 2$
- Since the interest is in the extremal dependence, non-extreme values should be **censored**

Example of a censoring scheme [1]:

- Given a censoring level $\tau \in (0,1)$, set $\boldsymbol{U}_t = \boldsymbol{0}$ if $\max\{U_{t,1}, U_{t,2}\} \leq \tau$ ($t \in \{1, \dots, m\}$, $j = \{1, 2\}$)
- Indicator variables: $\mathcal{I}_{t,j} = \mathbb{I}\{\max\{U_{t,1}, U_{t,2}\} > \tau\}$ showing which variables are censored ($j = \{1, 2\}$)



Neural Bayes estimators (NBEs):

- Point estimators which minimise the Bayes risk, $r_\Omega(\hat{\theta}(\cdot))$, during the training step of the Neural Network (NN)
- The Bayes risk is approximated by

$$r_\Omega(\hat{\theta}(\cdot)) \approx \frac{1}{KJ} \sum_{m \in \mathcal{M}} \sum_{\theta \in \mathcal{V}} \sum_{U^{(m)} \in \mathcal{U}_\theta} \Pr(M = m) L(\theta, \hat{\theta}(U^{(m)}))$$

- $\mathcal{V}$ is a set of K samples from the prior $\boldsymbol{\theta} \sim \Omega(\cdot)$
- $\mathcal{U}_\theta$ is a set of J samples $u^{(m)} \mid \boldsymbol{\theta}, m \sim c(u^{(m)} \mid \boldsymbol{\theta})$, where m is a random sample size sampled from $M \sim \text{Unif}(m_1, m_2)$ ($M \in \mathbb{Z}$)
- $L(\cdot, \cdot)$ is the loss function (here the mean absolute error)

3. Architecture of the Neural Network

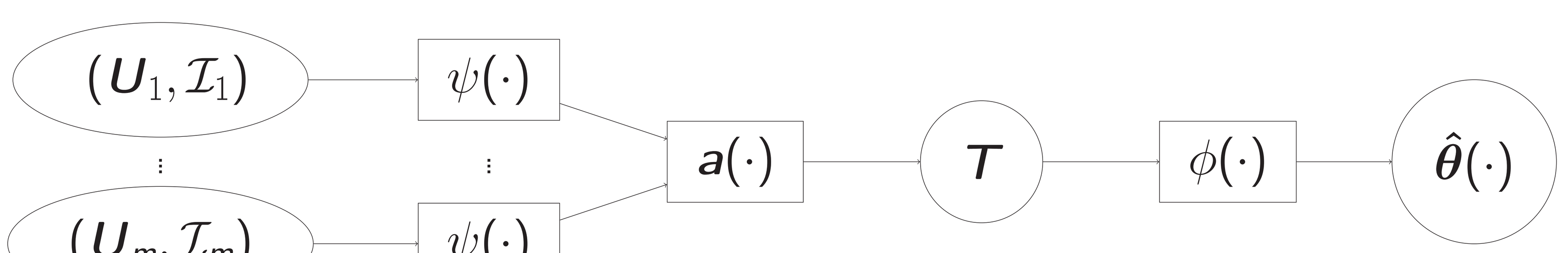


Figure 1: DeepSets framework [4]

- $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^q$ and $\boldsymbol{\phi} : \mathbb{R}^q \to \mathbb{R}^p$ are **deep neural networks** parameterised by $\boldsymbol{\gamma}_\psi$ and $\boldsymbol{\gamma}_\phi$,
- $\boldsymbol{a} : (\mathbb{R}^q)^m \to \mathbb{R}^q$ is a **permutation-invariant set** function (here the elementwise average)
- This framework ensures the NBEs are **invariant** to permutations of replicates of the data

4. Example

- The model of Wadsworth et al. (2017) [3] is able to capture both asymptotic dependence (AD) and asymptotic independence (AI) through parameter λ. AD if $\lambda > 0$ and AI if $\lambda \leq 0$
- It exploits the copula of the model constructed as

$$\begin{aligned} S &\sim \text{GP}(1, \lambda) \\ V &\sim \text{Beta}(\alpha, \alpha) \\ (V_1, V_2) &= \frac{(V, 1 - V)}{\|V, 1 - V\|_\infty} \\ (X_1, X_2) &= S(V_1, V_2) \end{aligned}$$

- (U_1, U_2) is obtained through rank transformation of (X_1, X_2)

5. Assessment of the NBE

To train the NN:

- Priors: $\alpha \sim \text{Unif}(0.2, 15)$, $\lambda \sim \text{Unif}(-2, 1)$
- Hyperparameters: $J = 5$, $K = 100\,000$, $M \sim \text{Unif}(100, 1500)$
- Censoring level: 0.8

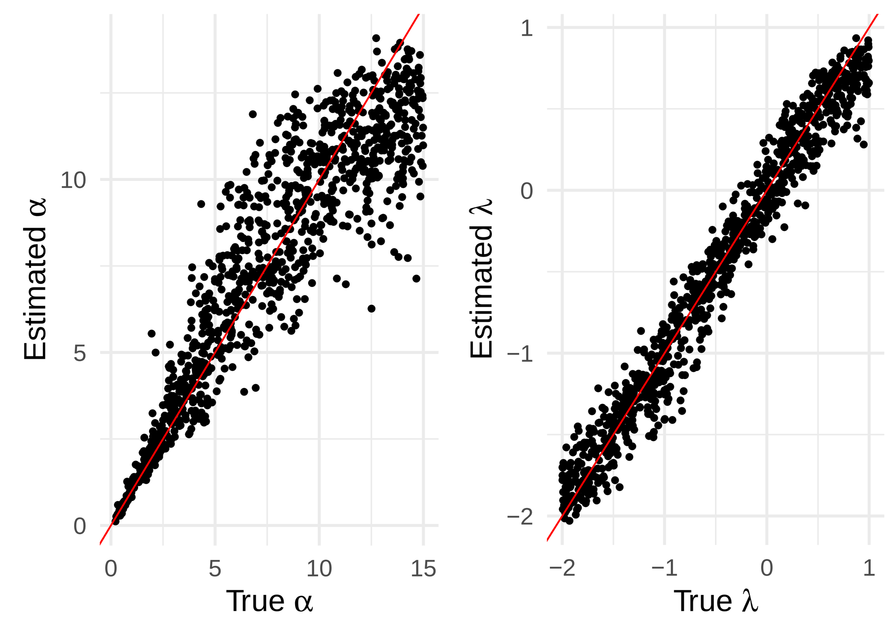


Figure 2: Assessment of the NBE with 1000 samples from the prior and sample size $m = 1000$

Comparison with censored maximum likelihood estimation:

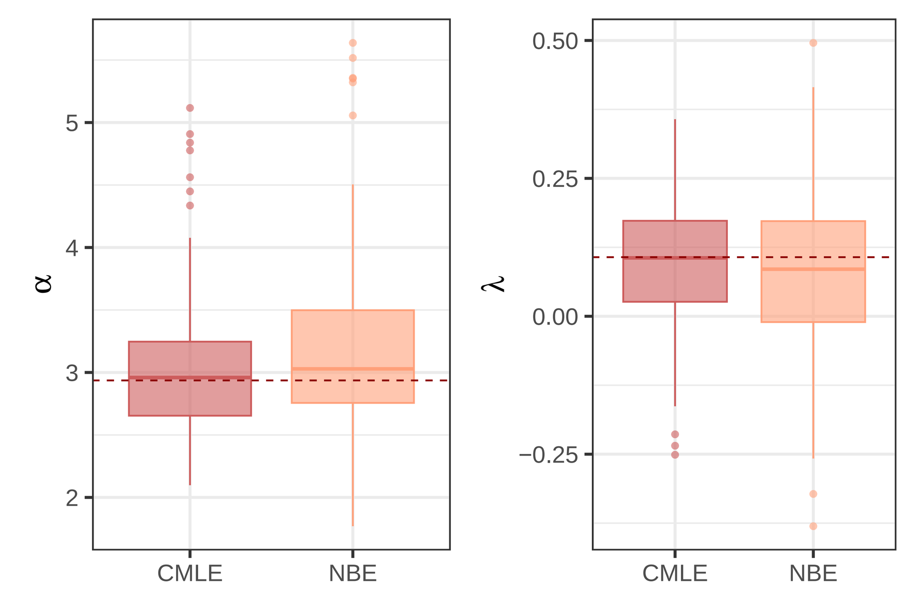


Figure 3: NBE compared with the censored maximum likelihood estimator (CMLE), for 100 samples of sample size 1000.

Average time to get an estimate:

- NBE: 0.5845 seconds and CMLE: 44.9979 seconds **(77 times faster)**
- In some experiments, it was $\sim$ **182 000 times faster** to get the NBE than the MLE, while in others, the NBE was $\sim$ **45 times faster** than the CMLE

6 - Further work: Model selection

- Use the NN as a **classifier**: have a prior on the model δ
  - 2 models: **binary classification** problem with $\delta \sim \text{Bernoulli}(0.5)$
  - n models: **multiclass classification** problem with $\delta \sim \text{Unif}(1, n)$, $\delta \in \mathbb{Z}$

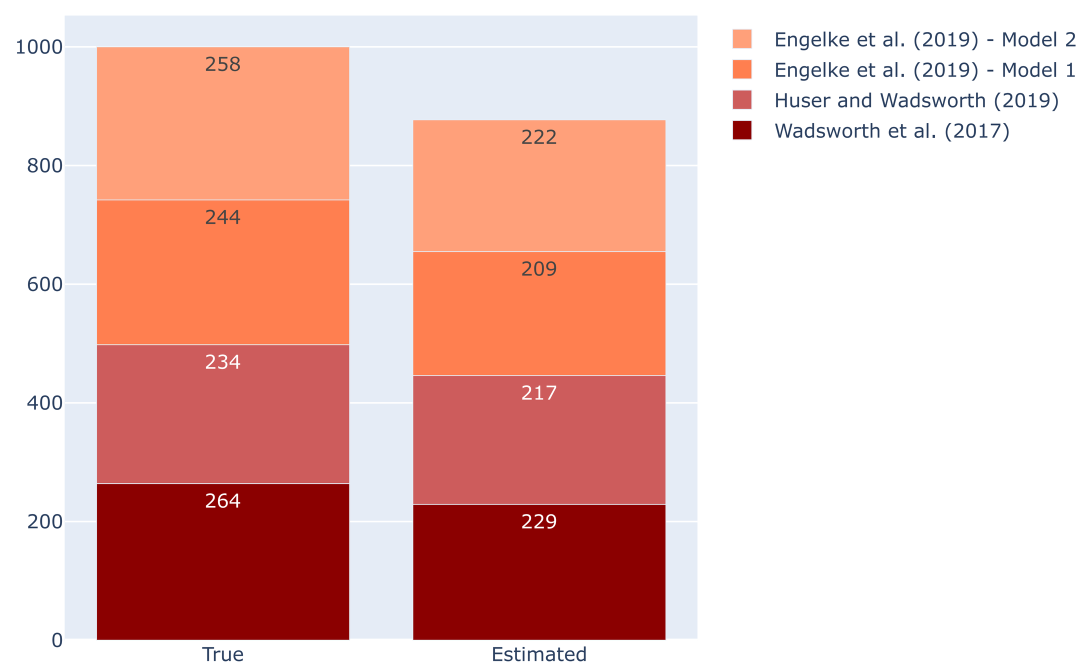


Figure 4: Correctly identified models from a trained NN.

References

[1] Richards, J., Sainsbury-Dale, M., Zammit-Mangion, A., and Hüser, R. (2023). Neural Bayes estimators for censored inference with peaks-over-threshold models. *(prepint)*

[2] Sainsbury-Dale, M., Zammit-Mangion, A., and Hüser, R. (2023). Likelihood-free parameter estimation with neural Bayes estimators. *The American Statistician*: 1-23

[3] Wadsworth, J. L., Tawn, J. A., Davison, A. C., and Elton, D. M. (2017). Modelling across extremal dependence classes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79 (1):149-175

[4] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B, Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. *Advances in Neural Information Processing Systems*, 30

l.andre@lancaster.ac.uk