Validation with RTMB

Olav Nikolai Risdal Breivik

Validation

RTMB has built-in functionality for:

- One-observation-ahead residuals
- Simulation studies
- Laplace checks

And it is easy to do:

- Jitter studies

One-observation-ahead residuals

- RTMB provides functionality for efficiently calculating

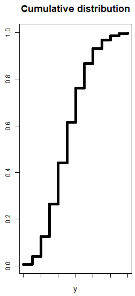
$$r_i = \Phi^{-1}\Big(P(Y_i \leq y_i | y_{i-1}, ..., y_1)\Big)$$

- Uniform randomization for discrete distribution:

$$r_i = \Phi^{-1}\Big(P(Y_i \leq y_i | y_{i-1}, ..., y_1) - U_i P(Y_i = y_i | y_{i-1}, ..., y_1)\Big)$$

  where $U_i \sim \text{unif}(0, 1)$

- If the model is correct, $r_i \sim N(0, 1)$ and independent.



- Inform what are the observations

```
nll = function(par){
...
y = OBS(y)
...
}
```

- Calculated one-step-ahead residuals:

```
res =oneStepPredict(obj, method = "oneStepGeneric" )
```

  - oneStepGeneric: Most accurate approximation in general
  - fullGaussian: Very fast, but assumes everything is Gaussian

For details, see :

Thygesen, U. H., Albertsen, C. M., Berg, C. W., Kristensen, K., and Nielsen, A. (2017). Validation of ecological state space models using the Laplace approximation. Environmental and Ecological Statistics, 24(2):317–339.

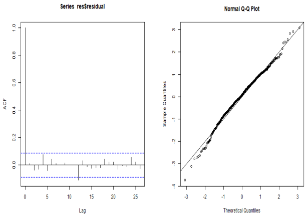
Exercise

Assume we observe $y_1, ..., y_n$ where $\log y_i \sim N(\mu_i, \sigma^2_{\text{logobs}})$ and

$$\mu \sim N(0, \Sigma).$$

Here γ is a stationary mean zero AR(1) process.

- Exercise 5 a: Investigate one-observation-ahead residuals
- Exercise 5 b: Remove auto-correlation and investigate residuals
- Code to get you started is in **residualsMVN.R**



.

Exercise comment

We observe a prediction dependent residual structure.

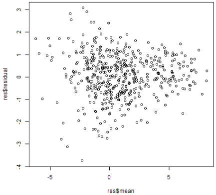
- Log-normality implies variance on natural scale:

$$v = \alpha\mu^{\beta}$$

  with $\beta = 2$, were μ is mean on natural scale
- The assumption $\beta = 2$ is not correct in our exercise
- Note that

$$v = \alpha\mu^{\beta} \;\Rightarrow\; \sigma^2_{\text{logobs}} = \log(\alpha\mu^{\beta-2} + 1)$$

- With RTMB we are in control and can estimate the β-parameter



Exercise comment

We observe a prediction dependent residual structure.

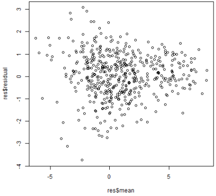
- Log-normality implies variance on natural scale:

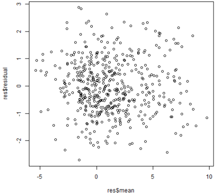
$$v = \alpha\mu^{\beta}$$

  with $\beta = 2$, were μ is mean on natural scale
- The assumption $\beta = 2$ is not correct in our exercise
- Note that

$$v = \alpha\mu^{\beta} \;\Rightarrow\; \sigma^2_{\text{logobs}} = \log(\alpha\mu^{\beta-2} + 1)$$

- With RTMB we are in control and can estimate the β-parameter
  - See **`residualsMVNsolution.R`**





Exercise comment

We observe a prediction dependent residual structure.

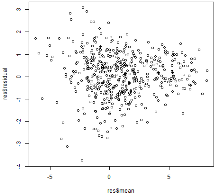
- Log-normality implies variance on natural scale:

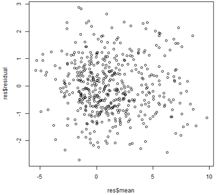
$$v = \alpha\mu^{\beta}$$

  with $\beta = 2$, were μ is mean on natural scale
- The assumption $\beta = 2$ is not correct in our exercise
- Note that

$$v = \alpha\mu^{\beta} \Rightarrow \sigma^2_{\text{logobs}} = \log(\alpha\mu^{\beta-2} + 1)$$

- With RTMB we are in control and can estimate the β-parameter
  - See **`residualsMVNsolution.R`**
- See **`residualsDiscrete.R`** for discrete example
- Also implemented procedure with compositional observations (Trijoulet et al., 2023)





Simulation study

RTMB has built-in functionality for simulating from the model

- Inform what are the observations

```
nll = function(par){
      ...
      y = OBS(y)
      ...
}
```

- Simulate from the model given hyperparameters

```
obj$simulate()
```

- We can then re-estimate the model

```
data = list(y = obj$simulate()$y)
objSim = MakeADFun(nll,par,random = "gamma")
optSim = nlminb(objSim$par,objSim$fn,objSim$gr)
```

- Repeat and verify that you can re-estimate model
- Code for the ar1-example available here: **`simulationAR1.R`**

Jitter analysis

Easy to verify if model is sensitive to starting values

- Sample new starting values
- Re-fit model
- Verify that results are stable
- Code example:

```
parJit <- lapply(1:nojit,
               function(i)relist(unlist(par)+rnorm(length(unlist(par)),sd=sd), par))
jj = lapply(parJit, function(p){
     obj = MakeADFun(nll,p,random = "gamma")
     opt = nlminb(obj$par,obj$fn,obj$gr)
     sdrep = sdreport(obj)
     pl = as.list(sdrep,what = "Est")
     rl = as.list(sdrep,what = "Est",report = TRUE)
     ret = list(opt = opt,pl = pl,rl = rl)
     ret
})
```

- Code for AR1 example: `jitter.R`

Laplace checker

- The expectation of the gradient of the negative loglikelihood is zero

$$E_\theta\left(\nabla \ell(\theta; X)\right) = 0$$

- If we simulate from the model, the average should be zero
- But this only holds for the real likelihood
- If the approximation is wrong, the average gradient is not zero
- Notice, even the smallest bias will be detected if we simulate enough, but models with modest bias can still be usefull.
- RTMB estimates parameter bias caused by the Laplace approximation
- ~~Verify that simulation are implemented correctly on C-side.~~
- Code for AR1 example: **`laplaceCheck.R`**

```
cc = checkConsistency(fit$obj,n = 100)
summary(cc)
>...
>$marginal$p.value
>[1] 0.2317515
>$marginal$bias
>logsd     logitRho
>-0.01318075 -0.01125501
```