Validation

Olav Nikolai Risdal Breivik

Thanks to Anders Nielsen for borrowing course material

Norsk Regnesentral
ANVENDT DATAFORSKNING

Fish stock assessment models have evolved

Deterministic → Full parametric (observations stochastic, process deterministic) → State−space

Advantages with state-space formulation:

- The obvious tool for time series data
- Quantification of observation errors
- Quantification of process errors
- Process formulation of time-varying quantities
- Reasonable (low) number of model parameters
- Prediction as part of model formulation

Current validation practice include:

- Residuals
- Retrospective patterns of key outputs
- Leave-fleet-out runs (to check consistency between data sources)
- Jittered starting point analysis

One-step-ahead residuals

SAM provides functionality for efficiently calculating one-step-ahead residuals

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
     residual.fleet=cn,
     prop.mature=mo,
     stock.mean.weight=sw,
     catch.mean.weight=cw,
     dis.mean.weight=dw,
     land.mean.weight=lw,
     prop.f=pf,
     prop.m=pm,
     natural.mortality=nm,
     land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Calculate OSA-residuals
res = residuals(fit)
```

Given the correct model, the residuals should be independent and $\mathcal{N}(0,1)$ distributed

- Illustrate live

In RTMB with your own model

- Inform what are the observations

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

- Calculated one-step-ahead residuals:

```
res =oneStepPredict(obj)
```

- method `oneStepGeneric`
    - Most accurate approximation in general
    - Works in the discrete case

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.

Simulation study

SAM has built-in functionality for simulation experiment

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
      residual.fleet=cn,
      prop.mature=mo,
      stock.mean.weight=sw,
      catch.mean.weight=cw,
      dis.mean.weight=dw,
      land.mean.weight=lw,
      prop.f=pf,
      prop.m=pm,
      natural.mortality=nm,
      land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Calculate OSA-residuals
sim = simstudy(fit)
plot(sim)
```

- Illustrate live

Jitter analysis

SAM has built-in functionality for jitter experiment

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
     residual.fleet=cn,
     prop.mature=mo,
     stock.mean.weight=sw,
     catch.mean.weight=cw,
     dis.mean.weight=dw,
     land.mean.weight=lw,
     prop.f=pf,
     prop.m=pm,
     natural.mortality=nm,
     land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Do jitter:
jj = jit(fit)
jj
```

- Illustrate live

Leave-out analysis

SAM has built-in functionality for leave-out analysis

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
      residual.fleet=cn,
      prop.mature=mo,
      stock.mean.weight=sw,
      catch.mean.weight=cw,
      dis.mean.weight=dw,
      land.mean.weight=lw,
      prop.f=pf,
      prop.m=pm,
      natural.mortality=nm,
      land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Do leaveout:
ll = leaveout(fit)
plot(ll)
```

- Illustrate live

AIC

SAM has built-in functionality for using the AIC function

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
     residual.fleet=cn,
     prop.mature=mo,
     stock.mean.weight=sw,
     catch.mean.weight=cw,
     dis.mean.weight=dw,
     land.mean.weight=lw,
     prop.f=pf,
     prop.m=pm,
     natural.mortality=nm,
     land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Calculate AIC
AIC(fit)
```

- Illustrate live

Likelihood-ratio test

Assume we want to test whether a parameter is significant

- $H_0 : \theta \in \Omega_0$
- $H_a : \theta \in \Omega$

A likelihood-ratio test reject H_0 if

$$\text{LR} = \frac{\mathcal{L}(\hat{\theta}_0)}{\mathcal{L}(\hat{\theta})}$$

is small enough.

What is small enough?

- Many standard tests can be derived from the LR principle (e.g., t-test)
- For large sample size we have an approximation for the LR distribution

The log of the likelihood ratio is

$$\log(\mathrm{LR}) = \log \frac{\mathcal{L}(\hat{\theta}_0)}{\mathcal{L}(\hat{\theta})} = \ell(\hat{\theta}_0) - \ell(\hat{\theta})$$

Asymptotically we have that

$$-2\log(\mathrm{LR}) = 2[\ell(\hat{\theta}) - \ell(\hat{\theta}_0)] \sim \chi^2_p$$

where $p = dim(\Omega) - dim(\Omega_0)$

Procedure:

- Calculate $-2\log(\mathrm{LR})$
- Compare with χ^2_p

```
> qchisq(0.95,df = 1)
[1] 6.634897
```

Liklihood-ratio test and AIC

AIC is defined by $AIC = 2k - 2\ell(\hat{\theta})$

- k is the number of model parameters
- $\ell(\hat{\theta})$ is the maximum log-likelihood

We have that for including one parameter:

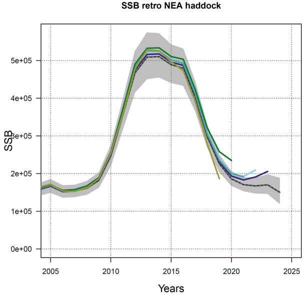
$$
\begin{aligned}
\Delta AIC &= AIC_0 - AIC \\
&= \Big(2(k-1) - 2\ell(\hat{\theta}_0)\Big) - \Big(2k - 2\ell(\hat{\theta})\Big) \\
&= 2[\ell(\hat{\theta}) - \ell(\hat{\theta}_0)] - 2
\end{aligned}
$$

Notice the similarity with a likelihood ratio-test?
The AIC needs to be reduced with approximately 5 to be significant on 0.01-level

- Do not hunt parameters to include based on a small improvement in AIC

Retro analysis

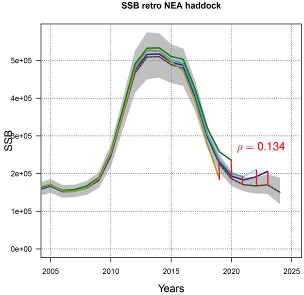
- Average relative difference from terminal estimate



$$\rho = \frac{1}{n}\sum_{y=y_0}^{Y-1}\frac{\hat{X}_{y|y}-\hat{X}_{y|Y}}{\hat{X}_{y|Y}}$$

Retro analysis

- Average relative difference from terminal estimate



$$\rho = \frac{1}{n}\sum_{y=y_0}^{Y-1}\frac{\hat{X}_{y|y} - \hat{X}_{y|Y}}{\hat{X}_{y|Y}}$$

Retro analysis

SAM has built-in functionality for retro analysis

```
library(stockassessment)
dat<-setup.sam.data(surveys=surveys,
     residual.fleet=cn,
     prop.mature=mo,
     stock.mean.weight=sw,
     catch.mean.weight=cw,
     dis.mean.weight=dw,
     land.mean.weight=lw,
     prop.f=pf,
     prop.m=pm,
     natural.mortality=nm,
     land.frac=lf)

confDef<-defcon(dat)
par = defpar(dat,confDef)
fit = sam.fit(dat,confDef,par)
#Do retro analysis:
ret = retro(fit,year = 5)
#Calculate mohn's rho
mohn(ret)
```

- Illustrate live

Retro analysis

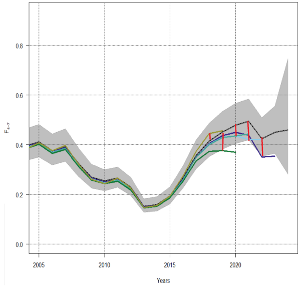
- SAM will by default remove last year of observation from all time series
  - This means that if a survey ended in 2014, SAM will remove year 2014 in first peal
  - This is not what you want
- You can manually set up which year that is removed from each fleet in each peal:

```
yearMat =  matrix(c(2023, 2024, 2024,2024,
     2022, 2023, 2023,2023,
     2021, 2022, 2022,2022,
     2020, 2021, 2021,2021,
     2019, 2020, 2020,2020),
nrow = 5, ncol = 4, byrow = TRUE)
ret = retro(fit,year = yearMat)
```

Retro analysis

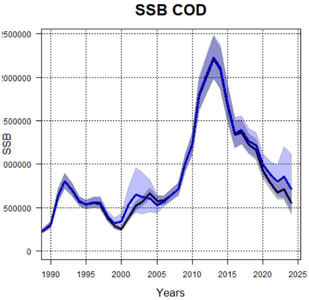
- Fishing mortality is very uncertain without data on catch in year
- If catch is missing in terminal year, default in ICES is to set `lag` = 1 when calculating Mohn's ρ for fishing mortality

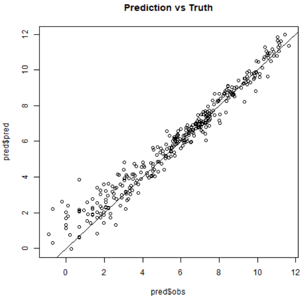
```
ret = retro(fit,year = 5)
#Calculate mohn's rho
mohn(ret,lag = 1)
```



Prediction and cross-validations

- Validate if the model is realistic w.r.t. coverage of confidence intervals
- Of special interest is the 2-3 year ahead predictions





- The only thing that is real is the observations
- When evaluating (and comparing) models we should look at their ability to predict observations.
- With state-space models we can (difficult to compare to other model types).

Implementing cross validations

- Set observations to unobserved. Simple implementation:

```
# function for cross-validation
xval <- function(fit, year=NULL, fleet=NULL, age=NULL, ...){
     data <- fit$data
     nam <- c("year", "fleet", "age")[c(length(year)>0,length(fleet)>0,length(age)>0)]
     if((length(year)==0) & (length(fleet)==0) & (length(age)==0)){
          idx <- rep(TRUE,nrow(data$aux))
     }else{
          idx <- !do.call(paste, as.data.frame(data$aux[,nam,drop=FALSE])) %in% do.call(paste, as.data.frame(cbind(year=year, fleet=fleet, age=age)))
     }
     idx <- !idx
     data$logobs[idx] <- NA
     idx2 <- which(is.na(data$logobs))
     conf <- fit$conf
     par <- defpar(data,conf)
     thisfit <- sam.fit(data, conf, par, rm.unidentified = TRUE, silent=TRUE,...)
     ret <- as.data.frame(cbind(data$aux[idx2,], obs=fit$data$logobs[idx2], pred=thisfit$pl$missing, predSd=thisfit$plsd$missing))
     ret <- ret[complete.cases(ret),]
     attr(ret, "fit") <- thisfit
     return(ret)
}
pred <- xval(fit, year=c(1988:1990,2013:2015))
```

- Exercise: Try it! See `crossvalidation.R` to get you started