Examples of focused model comparison: skew-normal models

Christopher Jackson
chris.jackson@mrc-bsu.cam.ac.uk

Abstract

The following example (from Claeskens and Hjort (2008)) illustrates how focused model comparison can be performed using the **fic** package in a situation where:

- a novel class of models is defined and fitted by custom R functions
- a simple model is extended in two different directions to define the models being compared

Keywords: models.

1. Skew-normal models

An outcome y_i and a covariate x_i are observed for individuals $i = 1, \ldots, n$. Four different models are compared: two normal models with a constant variance, one without (1) and one with (2) a linear regression term, and two "skew-normal" models without (3) and with (4) the linear regression term. The skew-normal model is defined by an error term $\sigma\epsilon_i$, where the ϵ_i are independently distributed with a density $f(u|\lambda) = \lambda\Phi(u)^{\lambda-1}\phi(u)$. All four models are nested in the "wide" model (4).

(1) $y_i \sim N(\beta_0, \sigma^2)$

(2) $y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$

(3) $y_i = \beta_0 + \sigma\epsilon_i$

(4) $y_i = \beta_0 + \beta_1 x_i + \sigma\epsilon_i$

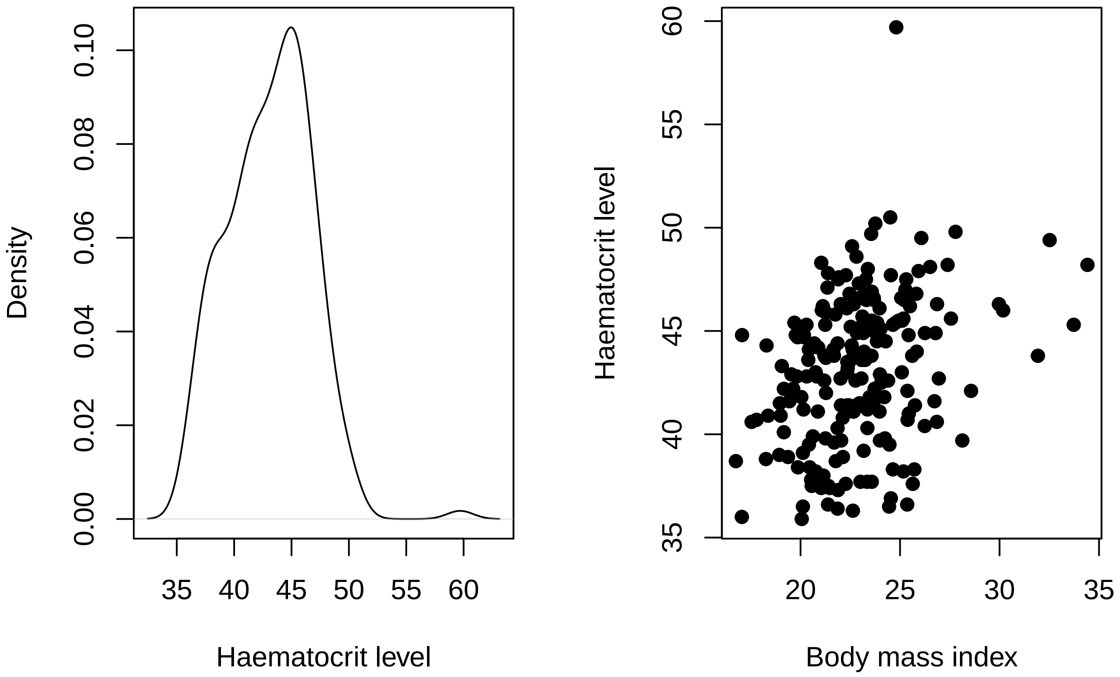
2. Fitting the models in R

To implement this class of models in R, firstly we define the log density function of the general skew-normal model with mean, scale and skewness parameters μ, σ, λ indicated by arguments `mean`, `sigma` and `lambda`. This defines the distribution of y_i in model (3) with mean $\mu_i = \beta_0$, and in (4) with $\mu_i = \beta_0 + \beta_1 x_i$. Models (1) and (2) are defined by setting $\lambda = 1$ in models (3) and (4) respectively.

```
ldsnorm <- function(x, mean, sd, lambda){
    log(lambda) + (lambda-1)*pnorm(x, mean, sd, log.p=TRUE) +
        dnorm(x, mean, sd, log=TRUE)
}
```

The models are fitted to data from the Australian Institute of Sports (Cook and Weisberg 1994), available as `ais` from the **sn** package (Azzalini 2018). The outcome y_i is haematocrit level `Hc`, and the covariate is body mass index `BMI`. Observe the skewed distribution of the outcome and a mild association between the variables.

```
if (!require("sn"))
    stop("The `sn` package should be installed to run code in this vignette")

## Loading required package:  sn
## Loading required package:  stats4
##
## Attaching package:  'sn'
## The following object is masked from 'package:stats':
##
##     sd

data(ais)
par(mfrow=c(1,2))
plot(density(ais$Hc), xlab="Haematocrit level", main="")
plot(ais$BMI, ais$Hc, pch=19,
     xlab="Body mass index", ylab="Haematocrit level")
```



The following defines the minus log likelihood for these data as a function of four parameters β_0, β_1, σ and λ.

```
mloglik <- function(b0, b1, sd, lambda){
    -sum(ldsnorm(ais$Hc, b0 + b1*ais$BMI, sd, lambda))
}
```

Then to obtain the maximum likelihood estimates under models 1 to 4, `mloglik` is rewritten as a function of a single vector `par`, containing either 2, 3 or 4 parameters to be minimised over, depending on the model. Note here that models 1 and 3 have β_2 fixed at 0, and models 1 and 2 have λ fixed at 1. The positive parameters σ and λ will be estimated by unconstrained maximisation on the log scale. These functions can be passed to the `nlm` function for optimisation.

```
fn1 <- function(par) mloglik(par[1], 0, exp(par[2]), 1)
fn2 <- function(par) mloglik(par[1], par[2], exp(par[3]), 1)
fn3 <- function(par) mloglik(par[1], 0, exp(par[2]), exp(par[3]))
fn4 <- function(par) mloglik(par[1], par[2], exp(par[3]), exp(par[4]))
```

`nlm` also requires plausible initial values for the parameters. A vector of these (`ini`) is obtained as follows by fitting model (2) with `lm` and extracting the coefficients with `coef` (for β_0 and

β_1) and the residual standard deviation for $\log(\sigma)$. An initial value of 0 is used for $\log(\lambda)$. [1]

```r
lm2 <- lm(Hc ~ BMI, data=ais)
cf <- unname(coef(lm2))
ini <- c(beta0=cf[1], beta1=cf[2],
         logsigma=log(summary(lm2)$sigma), loglambda=0)
```

The appropriate objective function (`fn1`–`fn4`) is then optimised for each model, starting from the given initial values[2].

```r
opt1 <- nlm(fn1, ini[c("beta0","logsigma")], hessian=TRUE)
opt2 <- nlm(fn2, ini[c("beta0","beta1","logsigma")], hessian=TRUE)
opt3 <- nlm(fn3, ini[c("beta0","logsigma","loglambda")], hessian=TRUE)
opt4 <- nlm(fn4, ini, hessian=TRUE)
```

Finally, the information required by the `fic` function (the estimates and covariance matrices) is extracted from the `nlm` results and arranged into a list, for each of the four models.

```r
mod1 <- list(est=opt1$estimate, vcov=solve(opt1$hessian) )
mod2 <- list(est=opt2$estimate, vcov=solve(opt2$hessian) )
mod3 <- list(est=opt3$estimate, vcov=solve(opt3$hessian) )
mod4 <- list(est=opt4$estimate, vcov=solve(opt4$hessian) )
```

3. Focused model comparison

We now perform a focused comparison of the four models. Two alternative focuses are investigated: the mean and median outcome at a covariate value of interest. Expressions for the mean and median of the skew normal, in the parameterisation used here, are given by Claeskens and Hjort (2008) and implemented in the following R functions:

```r
mean_snorm <- function(mu, sigma, lambda){
    f <- function(u){u*exp(ldsnorm(u, 0, 1, lambda))}
    mu + sigma * integrate(f, -Inf, Inf)$value
}
median_snorm <- function(mu, sigma, lambda){
    mu + sigma * qnorm(0.5^(1/lambda))
}
```

[1] Note these are the exact maximum likelihood estimates for model 2. The added value of `nlm` in fitting models 1 and 2, compared to simply using `lm`, is to conveniently provide the covariance matrix at the maximum likelihood estimates in the same form as for the skew normal models, making focused model comparison more convenient here.

[2] A warning message of `NA/Inf replaced by maximum positive value` can be ignored and is the result of `nlm` trying out extreme and implausible values on the way to finding the maximum likelihood.

As described in the main `fic` package vignette, the focus function supplied to `fic` should have arguments defined by a vector `par` of parameters of the biggest model (in this case the four-parameter model 4), and optionally also a matrix of covariate values `X`, and should return the corresponding focus quantity. In this example, the two focus functions are

```
focus1 <- function(par, X){
    mean_snorm(mu = par[1] + X %*% par[2],
               sigma = exp(par[3]), lambda = exp(par[4]))
}
focus2 <- function(par, X){
    median_snorm(mu = par[1] + X %*% par[2],
                 sigma = exp(par[3]), lambda = exp(par[4]))
}
```

The `inds` matrix, required by `fic`, is now constructed. Recall this indicates which parameters (columns) are included in each of the models (rows) being compared. The narrow model is in the first row, and the wide model in the last. The rows are given names to describe the models, so the output is easier to read.

The functions `fns` required to extract the estimates and covariance matrix from the fitted model objects are then defined.

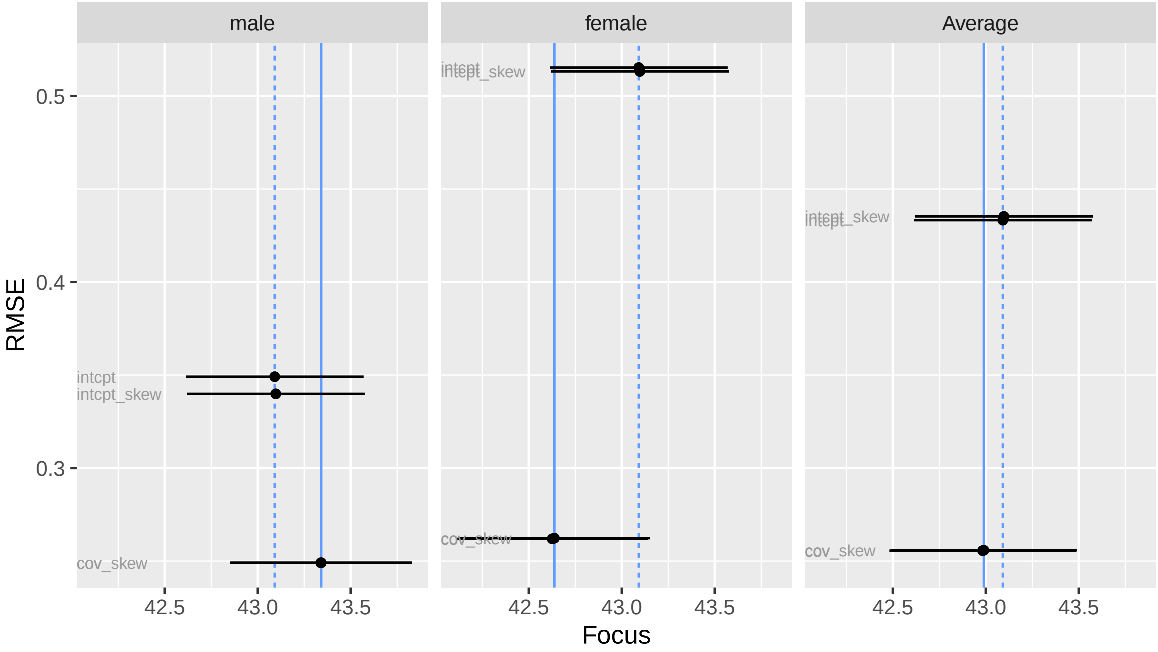
Finally `fic` is called to compare the four models for focuses defined by the mean and median for average men and women, with covariate values defined by `med.bmi`. The `sub` argument is supplied to ensure the focus estimates are returned too – note that `fic` can only automatically fit the submodels for standard R model classes such as `glm`.

```
inds <- rbind("intcpt"     =c(1,0,1,0),
              "cov"        =c(1,1,1,0),
              "intcpt_skew"=c(1,0,1,1),
              "cov_skew"   =c(1,1,1,1))

fns <- list(coef=function(x)x$est,
            vcov=function(x)x$vcov,
            nobs=function(x)nrow(ais))

med.bmi <- rbind(male=23.56,  female=21.82)

library(fic)
fmean <- fic(mod4, inds=inds, fns=fns, focus=focus1, X=med.bmi, FIC=TRUE,
             sub=list(mod1, mod2, mod3, mod4))
fmean

##       vals        mods  rmse rmse.adj   bias    se   FIC
## 1     male      intcpt 0.349    0.349 -0.250 0.244 13.14
## 4     male         cov 0.249    0.249  0.000 0.249  1.05
## 7     male intcpt_skew 0.340    0.340 -0.237 0.244 11.85
## 10    male    cov_skew 0.249    0.249  0.000 0.249  1.05
## 2   female      intcpt 0.515    0.515  0.454 0.244 43.51
## 5   female         cov 0.262    0.262  0.000 0.262  3.73
## 8   female intcpt_skew 0.513    0.513  0.452 0.244 43.09
## 11  female    cov_skew 0.262    0.262  0.000 0.262  3.78
## 3  Average      intcpt 0.433    0.433  0.358 0.244 26.02
## 6  Average         cov 0.256    0.256  0.000 0.256  1.30
## 9  Average intcpt_skew 0.435    0.435  0.360 0.244 26.38
## 12 Average    cov_skew 0.256    0.256  0.000 0.256  1.32
##    focus
## 1   43.1
## 4   43.3
## 7   43.1
## 10  43.3
## 2   43.1
## 5   42.6
## 8   43.1
## 11  42.6
## 3   43.1
## 6   43.0
## 9   43.1
## 12  43.0
```

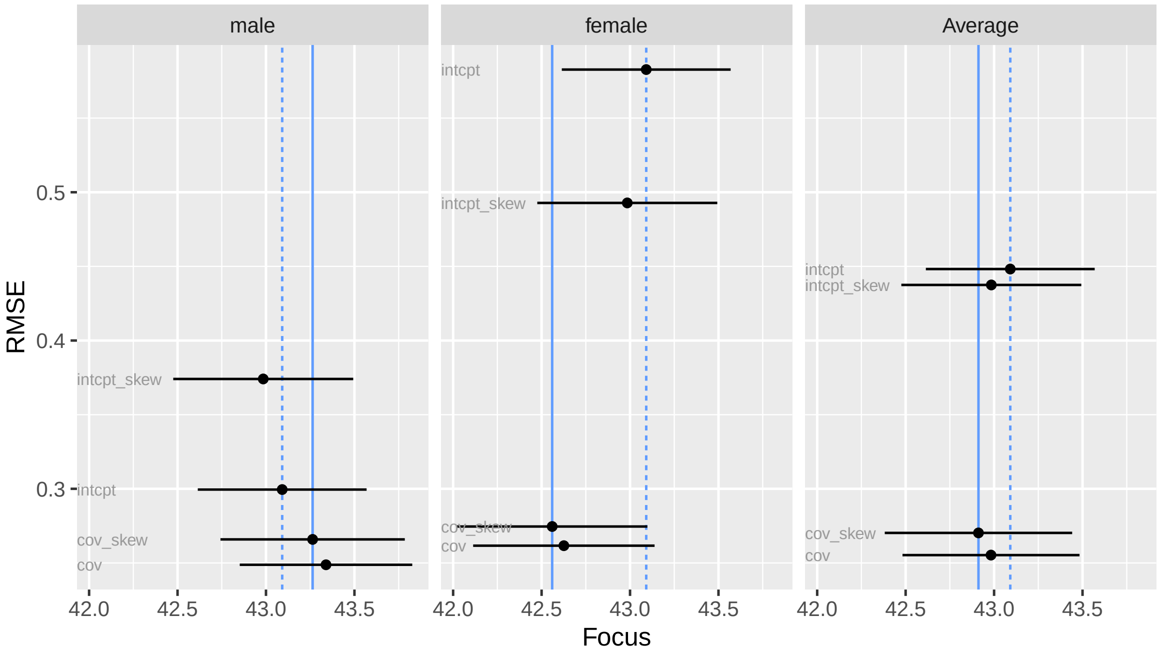
```
fmed <- fic(mod4, inds=inds, fns=fns, focus=focus2, X=med.bmi, FIC=TRUE,
            sub=list(mod1, mod2, mod3, mod4))
fmed
```

```
##       vals        mods  rmse rmse.adj   bias    se   FIC
## 1     male      intcpt 0.299    0.299 -0.174 0.244  8.44
## 4     male         cov 0.242    0.249  0.000 0.249  2.12
## 7     male intcpt_skew 0.374    0.374 -0.269 0.260 18.59
## 10    male    cov_skew 0.266    0.266  0.000 0.266  4.60
## 2   female      intcpt 0.583    0.583  0.529 0.244 59.87
## 5   female         cov 0.256    0.262  0.000 0.262  4.55
## 8   female intcpt_skew 0.493    0.493  0.419 0.260 40.33
## 11  female    cov_skew 0.275    0.275  0.000 0.275  6.50
## 3  Average      intcpt 0.448    0.448  0.376 0.244 30.29
## 6  Average         cov 0.249    0.255  0.000 0.255  2.24
## 9  Average intcpt_skew 0.437    0.437  0.352 0.260 28.37
## 12 Average    cov_skew 0.270    0.270  0.000 0.270  4.46
##    focus
## 1   43.1
## 4   43.3
## 7   43.0
## 10  43.3
## 2   43.1
## 5   42.6
## 8   43.0
## 11  42.6
## 3   43.1
## 6   43.0
## 9   43.0
## 12  42.9
```

```
ggplot_fic(fmean)
```



```
ggplot_fic(fmed)
```



For both the mean and the median, including the covariate is judged to be essential. Including the skewness is pointless for estimating the mean, since, as discussed by Claeskens and Hjort (2008), the skewness term provides no extra information. The utility of including the skewness is also doubtful for estimating the median too — given that the covariate is included, the focus estimates and RMSE are very similar whether or not the skewness is also included.

References

Azzalini A (2018). *The R Package* ***sn****: The Skew-Normal and Related Distributions Such As the Skew-t (version 1.5-3).* Università di Padova, Italia. URL http://azzalini.stat.unipd.it/SN.

Claeskens G, Hjort N (2008). *Model Selection and Model Averaging.* Cambridge University Press.

Cook RD, Weisberg S (1994). "An Introduction to Regression Graphics."