Different models for different purposes: focused model comparison in R

Chris Jackson <chris.jackson@mrc-bsu.cam.ac.uk>,
Howard Thom <howard.thom@bristol.ac.uk>,
Gerda Claeskens <gerda.claeskens@kuleuven.be>

Abstract

Typical methods of model comparison are used to pick one "best" model, no matter what the estimates from the model are used for. "Focused" model comparison, by contrast, considers that different models may be better for different purposes. Different models may be preferred for estimating different "focus" quantities, functions of the basic parameters.

In the "focused information criterion" of Claeskens and Hjort (2006), data are assumed to be generated by a "wide" model, in which all models we would consider are nested. Fitting the wide model to the observed data, however, may give estimates that are not sufficiently precise. Therefore we might accept some bias in the estimate in return for greater precision. The optimal submodel for a particular focus is the one which minimises the mean squared error of the estimate of that focus from the submodel, assuming that the wide model is true.

The `fic` package calculates this error, and shows the bias-variance tradeoff directly, for comparisons within any class of models fitted by maximum likelihood. The tradeoff between bias and variance is shown directlyThere are shortcuts for commonly-used model classes such as GLMs and parametric survival models. Covariate selection problems in Cox regression models are also supported.

Keywords: FIC,model comparison,AIC,BIC.

1. Introduction: principles for model comparison

To compare a set of statistical models fitted to the same data by maximum likelihood, it is common to rank them according to some "criterion". For example, Akaike's information criterion (AIC, Akaike (1973)) takes the form

$$-2\log \ell(\hat{\theta};\mathbf{x}) + 2p$$

where $\ell(\hat{\theta};\mathbf{x})$ is the maximised likelihood for the model fitted to the dataset $\mathbf{x}$, the likelihood is maximised at parameters $\hat{\theta}$, and p is the number of parameters.

The Bayesian information criterion (BIC, Schwarz (1978)) is

$$-2\log \ell(\hat{\theta};\mathbf{x}) + p\log(n)$$

These two criteria are based on very different principles. Thus they often rank models differently. The AIC is designed to choose models with better predictive ability, thus it tends to favour bigger models as the sample size increases. BIC is an approximation to Bayesian model comparison by Bayes factors, and prefers models with higher posterior probability under an implicit weak prior (with an amount of information equivalent to one observation, see Kass and Wasserman (1995)). If there is a "true" model, the BIC will tend to select it as the sample size increases. In many situations there may not be a true model, and collecting more data will uncover more complexity in the process generating the data, in which case AIC may be more suitable. See e.g. Burnham and Anderson (2003), Claeskens and Hjort (2008) for more theory behind these, and other similar model comparison criteria.

Both of these methods give a single ranking of models according how well they fit a given dataset. However, different models may be better for different purposes, for example to estimate different quantities of interest. Such quantities are termed the "focus" of a model, and this is the idea behind "focused" model comparison.

This paper describes the **fic** R package. This compares a set of models according to how accurately they estimate a focus quantity. The models should all be nested in a single "wide" model that is assumed to generate the data. Section 2 gives an informal introduction to the principles, and sets out the formulae, as developed by Claeskens and Hjort (2003) and Claeskens and Hjort (2008). Section 3 explains how general-purpose software is constructed to enable these quantities to be evaluated for any class of models and focuses, with the minimum of user effort. A worked example of using the **fic** package for covariate selection in GLMs is given in Section 4. A set of additional package vignettes demonstrate the use of the package in a variety of less common situations. The extension of the theory to deal with Cox proportional hazards regression models is described in Section 5 with a worked example.

2. Focused model comparison: principles and formulae

Suppose the range of models we are willing to use is bounded by

- a *wide model*, in which all models we would use are nested, with parameters $(\boldsymbol{\theta}, \boldsymbol{\gamma})$
- a *narrow model*, the smallest model we are willing to use, defined by setting $\boldsymbol{\gamma} = \boldsymbol{\gamma}_0$ in the wide model.

A typical example is covariate selection in regression models, where $\boldsymbol{\theta}$ are the coefficients of "protected" covariates which are always included, and $\boldsymbol{\gamma}$ are the coefficients of optional covariates that may or may not be included. More generally, we wish to choose the appropriate level of flexibility for the distribution of some outcome. For example, choosing between a Poisson versus a Negative Binomial model for a count outcome, or an exponential versus a Weibull survival model, where the former is a constained version of the latter.

Suppose also that the purpose of the model is to estimate some *focus* quantity, which could be any function of the basic parameters

$$\mu = g(\boldsymbol{\theta}, \boldsymbol{\gamma})$$

In focused model comparison, we prefer models which give better estimates of μ. A typical way to define "better" is by the *mean square error*. The mean square error of the estimate $\hat{\mu}_S$ under a submodel S of the wide model, compared to the true value μ, is

$$E\left\{(\hat{\mu}_S - \mu)^2\right\}$$

This expectation is calculated under the assumption that the data are generated from the wide model. While we believe the wide model is the most realistic, we also accept that there may not be enough data to give sufficiently precise estimates of μ. Therefore we are willing to accept some bias in this estimate, in return for a smaller variance, by selecting a smaller model than the wide model. The submodel S with the lowest mean square error is the one which makes the optimal trade-off between bias and variance.

The mean squared error MSE_S under model S can be decomposed as a sum of the squared bias B_S^2 and the variance V_S.

$$\begin{aligned} MSE_S = E\left\{(\hat{\mu}_S - \mu)^2\right\} &= \left\{E(\hat{\mu}_S) - \mu\right\}^2 + E\left\{(\hat{\mu}_S - E(\hat{\mu}_S))^2\right\} \\ &= B_S^2 + V_S \end{aligned} \tag{1}$$

Estimators for these quantities are constructed by Claeskens and Hjort (2003) under an asymptotic framework in which the data are assumed to be n independent identically distributed observations from the wide model, but reparameterised so that $\boldsymbol{\gamma} = \gamma_0 + \boldsymbol{\delta}/\sqrt{n}$. Thus as the sample size increases, we aim to detect more subtle departures from the narrow model.

An obvious estimate for the bias B_S is $\hat{B_S} = \hat{\mu_S} - \hat{\mu_W}$, where $\hat{\mu}_W$ is the estimate of the focus quantity under the wide model, which is assumed to be unbiased. However, Claeskens and Hjort (2003) derive a more accurate, asymptotically unbiased, estimate for the *squared* bias as

$$\widehat{B_S^2} = \hat{\omega}^T (I - G_S)(\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}^T - Q)(I - G_S)\hat{\omega} \tag{2}$$

where:

- $\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\gamma}}\sqrt{n}$, where $\hat{\boldsymbol{\gamma}}$ is the estimate of $\boldsymbol{\gamma}$ under the wide model.

- $\omega^T\boldsymbol{\delta}$ is the bias of the estimate of $\sqrt{n}\mu$ under the narrow model N, that is, the asymptotic mean of $\sqrt{n}(\hat{\mu}_N - \mu)$. Thus ω acts as a linear transformation from the biases of the basic parameters $\boldsymbol{\gamma}$ to the biases of the focus parameter μ.

- ω is estimated as $\hat{\omega} = J_{10}J_{00}^{-1}\frac{d\mu}{d\theta} - \frac{d\mu}{d\gamma}$ using Taylor approximation arguments, where J is the information (inverse covariance) matrix under the wide model [1] and subscripts 0 and 1 select the rows and columns forming the submatrices of J that correspond to parameters $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ respectively. The partial derivatives of the focus μ are evaluated at the estimates from the wide model.

- $G_S = \pi^T Q_S \pi Q^{-1}$ is an estimate of the transformation that maps the wide model estimate of $\boldsymbol{\delta}$ to the submodel S estimate, where $Q_S = (\pi Q^{-1}\pi^T)^{-1}$, $Q^{-1} = J_{11}$ and π is the projection matrix consisting of 0s and 1s which maps a vector of the same length as $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ to a subvector containing the elements corresponding to submodel S.

[1] Note that Claeskens and Hjort (2008) calculate the MSE of the focus multiplied by $\sqrt{n}$ rather than the MSE of the focus, thus they define J instead as the information matrix divided by n. Thus their definition of ω is the same as ours since n cancels, but their definitions of Q_S and Q are different.

Occasionally the estimate (2) of squared bias is negative. Claeskens and Hjort (2003) also present an adjusted version of (2), which assumes the bias is zero in these cases.

$$\widehat{B_S^{2*}} = max\left\{0, \widehat{B_S^2}\right\} \tag{3}$$

The corresponding estimate of the bias is

$$\widehat{B_S*} = sign(\hat{\psi}_W - \hat{\psi}_S)\sqrt{\widehat{B_S^{2*}}} \tag{4}$$

where $\hat{\psi}_W = \hat{\omega}^T\hat{\boldsymbol{\delta}}$ and $\hat{\psi}_S = \hat{\omega}^T G_S\hat{\boldsymbol{\delta}}$ are estimates of $\omega^T\boldsymbol{\delta}$ under the wide model and submodel respectively.

The estimate for the variance of $\hat{\mu}_S$ under the wide model, derived by Claeskens and Hjort (2003), is

$$\hat{V_S} = (\hat{\tau_0}^2 + \hat{\omega}^T Q_S^0 \hat{\omega})$$

where $\hat{\tau_0}^2$ estimates the variance of the narrow model focus estimate (using "delta method" principles, $\hat{\tau}_0^2 = \frac{d\mu}{d\theta}^T J_{00}^{-1}\frac{d\mu}{d\theta}$), and the additional term $(\hat{\omega}^T Q_S^0 \hat{\omega})$ is the increase in variance we accept by using a wider but still misspecified model S, with $Q_S^0 = \pi^T Q_S \pi$.

Thus we compare models on the basis of the root mean square error, estimated by

$$\sqrt{\widehat{MSE}_S} = \sqrt{\widehat{B_S^2} + \hat{V_S}} \tag{5}$$

or the alternative version which, while not asymptotically unbiased, is based on an interpretable estimate (4) of the bias.

$$\sqrt{\widehat{MSE}_S^*} = \sqrt{\widehat{B_S^{2*}} + \hat{V_S}} \tag{6}$$

Claeskens and Hjort (2003) define the "focused information criterion" (FIC), which has a slightly simpler form due to excluding terms common to all submodels S, and is related to the MSE as

$$FIC_S = n\widehat{MSE}_S - \hat{\tau}_0^2 + \hat{\omega}^T Q\hat{\omega} \tag{7}$$

Models with lower FIC give better estimates of the focus quantity. However we prefer to use the (root) MSE as the model comparison statistic, due to its direct interpretation as the error of the focus estimate.

2.1. Average MSE over a range of focuses

Often we want a model that performs well in a range of situations. In covariate selection problems, for example, we might want to estimate a focus quantity accurately for a defined range of covariate values. Thus the quantities defined above may now depend on the covariate value u. We might simply define the "averaged MSE" for submodel S,

$$MSE_S^{(ave)} = \int MSE_S(u)dW(u)du$$

as a weighted average of the mean squared errors (1) for focuses defined by different covariate values u, weighted by their prevalence $W(u)$. However Claeskens and Hjort (2008) derived an alternative formula, so that if correction of the squared bias estimate, as in (3) is required, it only needs to be performed once:

$$\widehat{MSE_S^{(ave)}} = I_S + \widehat{V_S^{(ave)}} \tag{8}$$

where

$$I_S = Tr((I - G_S)(\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}^T - Q)(I - G_S)^T A) \tag{9}$$

is an estimate of the squared bias, and

$$\widehat{V_S^{(ave)}} = \tau_0^{2(ave)} + II_S \tag{10}$$

is an estimate of the variance, with $\tau_0^{2(ave)} = \int \tau_0(u)du$, $II_S = Tr(Q_S^0 A)$, and

$$\begin{aligned} A &= J_{10}J_{00}^{-1}B_{00}J_{00}^{-1}J_{01} - J_{10}J_{00}^{-1}B_{01} - B_{10}J_{00}^{-1}J_{01} + B_{11} \\ B &= \int \begin{pmatrix} d\mu(u)/d\boldsymbol{\theta} \\ d\mu(u)/d\boldsymbol{\gamma} \end{pmatrix} \begin{pmatrix} d\mu(u)/d\boldsymbol{\theta} \\ d\mu(u)/d\boldsymbol{\gamma} \end{pmatrix}^T dW(u) \quad = \quad \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix} \end{aligned}$$

An analogue of the alternative MSE estimate (6) can then be defined, based on a bias estimator which is corrected when (9) is negative, by substituting $max(IS, 0)$ for IS in (8).

If the focus is defined as the log likelihood, and if we average over the observed distribution of covariates in the data, then model comparison using this procedure is asymptotically equivalent to model comparison by AIC (Claeskens and Hjort 2008, 2003).

3. Software for focused model comparison

In order to calculate the MSE (1) for focused model comparison of a submodel S against a wide model, we simply need to know

- the estimates $\hat{\boldsymbol{\theta}}_W$ and $\hat{\boldsymbol{\gamma}}_W$ and their covariance matrix under the wide model,
- the focus function $\mu(\boldsymbol{\theta}, \boldsymbol{\gamma})$
- the definition of which parameters are included in submodel S and which are included in the narrow model N.

Additionally if we want to know the focused information criterion (7) we will need the sample size n.

Derivatives of the focus function, required by (2), can be calculated numerically in general, for which robust software exists — **numDeriv** (Gilbert and Varadhan 2016) is used here. Analytic derivatives are implemented in **fic** for two built-in focuses (the outcome probability in logistic regression, and the mean outcome in linear regression), but we have noticed no loss in accuracy from using numerical methods.

This knowledge allows the **fic** package to implement focused model comparison for any class of models and focuses. The estimates and covariance matrix are routinely computed by functions for fitting models by maximum likelihood. Therefore the user simply needs to supply

- an R object containing the wide model
- a definition of the focus function $\mu()$
- indicators for what submodels they want to compare

In addition the software needs to know where to look inside the wide model object for the estimates and covariance matrix, but this information can be built into the software for a range of commonly-used models.

4. Example: covariate selection in logistic regression

The use of the `fic` package is illustrated for covariate selection in logistic regression, using the example originally presented in Claeskens and Hjort (2008) (Example 6.1). The dataset was originally presented by Hosmer and Lemeshow (1989). Data are taken from $n = 189$ women with newborn babies, and the binary outcome is whether the baby is born with a weight of less than 2500g. We build a logistic regression model to predict the outcome, but are uncertain about what covariates should be included.

The data are provided as an object `birthwt` in the `fic` package. This is the same as `birthwt` in `MASS` (Venables and Ripley 2002) with the addition of a few extra columns defining interactions and transformations as in Claeskens and Hjort (2008).

The following covariates are always included (coefficient vector $\boldsymbol{\theta}$)

- x_1 Weight of mother in kg, `lwtkg`

The following covariates will be selected from (coefficient vector $\boldsymbol{\gamma}$)

- z_1 age, in years, `age`
- z_2 indicator for smoking, `smoke`
- z_3 history of hypertension, `ht`
- z_4 uterine irritability, `ui`
- interaction $z_5 = z_1 z_2$ between smoking and age, `smokeage`
- interaction $z_6 = z_2 z_4$ between smoking and uterine irritability, `smokeui`

Firstly the wide model, that includes all the above covariates, is defined and fitted.

```r
library(fic)
wide.glm <- glm(low ~ lwtkg + age + smoke + ht + ui + smokeage + smokeui,
                data=birthwt, family=binomial)
```

The *focus function* is then defined. This should be an R function, mapping the parameters `par` of the wide model to the quantity of interest. The focus can optionally have an second argument. If supplied, this must be called `X`, and can be used to supply covariate values at which the focus function should be evaluated. [2] Here we take the probability of low birth weight as the focus, for two covariate categories:

1. smokers with average or typical values of the other covariates. These values are given in the order supplied when specifying the model (for smokers: intercept, `lwtkg`=58.24, `age`=22.95, `smoke`=1, `ht`=0, `ui`=0, `smokeage`=22.95, `smokeui`=0).

2. non-smokers with average values of the other covariates

```
focus <- function(par, X)plogis(X %*% par)
vals.smoke <-    c(1, 58.24, 22.95, 1, 0, 0, 22.95, 0)
vals.nonsmoke <- c(1, 59.50, 23.43, 0, 0, 0, 0, 0)
X <- rbind("Smokers"=vals.smoke, "Non-smokers"=vals.nonsmoke)
```

We can illustrate this function by calculating the probability of low birth weight, given the parameters of the fitted wide model, for each group. This is about twice as high for smokers.

```
focus(coef(wide.glm), X=X)

##              [,1]
## Smokers     0.345
## Non-smokers 0.168
```

The `fic` function can then be used to calculate the mean square error of the focus for one or more given submodels. For illustration we will compare two models, both including maternal weight, one including age and smoking, but the other including age, smoking and hypertension.

```
mod1.glm <- glm(low ~ lwtkg + age + smoke, data=birthwt, family=binomial)
mod2.glm <- glm(low ~ lwtkg + age + smoke + ht, data=birthwt, family=binomial)
```

We supply the following arguments to the `fic` function.

- `wide`: the fitted wide model. All the model fit statistics are computed using the estimates and covariance matrix from this model. `fic` will automatically recognise that this is a GLM fitted by the `glm` function in R, and extract the relevant information.

- `inds`: indicators for which parameters are included in the submodels, that is, which elements of $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ are fixed to γ_0. This should have number of rows equal to the number of submodels to be assessed, and number of columns equal to $dim(\boldsymbol{\theta}) + dim(\boldsymbol{\gamma})$, the total number of parameters in the wide model, 8 in the case of `wide.glm`, which includes the intercept and the coefficients of seven covariates. It contains 1s in the positions where the parameter is included in the submodel, and 0s in positions where

[2] Note we could also have written the focus function as `function(par,X)plogis(par[1] + X %*% par[-1])` then we could have omitted the dummy covariate values of 1 for the intercept at the start of `vals.smoke` and `vals.nonsmoke`.

the parameter is excluded. This should always be 1 in the positions defining the narrow model, as specified in `inds0` below. If just one submodel is to be assessed, `inds` can also be supplied as a vector of length $dim(\boldsymbol{\theta}) + dim(\boldsymbol{\gamma})$.

Note that `inds` indexes *parameters* rather than *linear model terms*, that is, in covariate selection problems where a variable is a factor with more than two levels, `inds` should contain separate entries for the coefficient of each factor level relative to the baseline level, not just one entry indicating the presence of the factor as a whole. A utility to construct this in the presence of factors is illustrated in Section 5.1.

- `inds0` vector of indicators for which parameters are included in the narrow model, in the same format as `inds`. This can be omitted, in which case the narrow model is assumed to be given by the first row of `inds`. In this case, just the first two parameters are included, the intercept and the coefficient of `lwtkg`.

```
inds <- rbind(mod1 = c(1,1,1,1,0,0,0,0),
              mod2 = c(1,1,1,1,1,0,0,0))
inds0 <- c(1,1,0,0,0,0,0,0)
```

- `focus` the focus function. As well as an R function, this argument can alternatively be supplied as a character string naming a built-in focus function supplied by the **fic** package. Currently these just include `"prob_logistic"`, the outcome probability in a logistic regression, and `"mean_normal"`, the mean outcome in a normal linear regression.

The main `fic` function then returns the model fit statistics and the estimate of the focus quantity for each model.

```
fic1 <- fic(wide=wide.glm, inds=inds, inds0=inds0, focus=focus, X=X)
fic1

##          vals mods   rmse rmse.adj   bias     se   FIC
## 1     Smokers mod1 0.0723   0.0723 0.0459 0.0558 1.187
## 4     Smokers mod2 0.0556   0.0572 0.0000 0.0572 0.783
## 2 Non-smokers mod1 0.0804   0.0804 0.0731 0.0334 1.305
## 5 Non-smokers mod2 0.0596   0.0596 0.0484 0.0348 0.755
## 3         ave mod1 0.0764   0.0764 0.0610 0.0460 1.005
## 6         ave mod2 0.0576   0.0576 0.0329 0.0473 0.528
##   focus
## 1 0.398
## 4 0.366
## 2 0.243
## 5 0.215
## 3 0.320
## 6 0.291
```

The object returned by `fic` is a data frame containing one row for each combination of focus covariate values indicated in the column `vals` and submodels indicated in the column `mods`. The focus estimate is returned in the final column `focus`, while the remaining columns contain the following model comparison statistics:

- `rmse` The root mean square error of the submodel focus estimate, calculated assuming the wide model is true (equation 5),
- `rmse.adj` Alternative estimate of the root mean square error (equation 6) based on the adjusted bias estimator (3–4).
- `bias` The estimated bias $\widehat{B_S*}$ (equation 4), which will be zero if $\widehat{B_S^2}$ (2) is negative.
- `se` The standard error $\sqrt{\hat{V}}$ of the submodel focus estimate, calculated assuming the wide model is true.
- `FIC` The FIC as originally defined by Claeskens and Hjort (2003) (equation 7).

The submodels are fitted automatically within the `fic` function in order to produce the focus estimate, so it was not really necessary to fit `mod1.glm` and `mod2.glm` by hand, as above. As the wide model has class `glm`, it is recognised as a GLM, so `fic` assumes that our submodels correspond to models with different covariates included, as indicated by `inds`. The focus estimates from the submodels can then be returned alongside the model comparison statistics.

As well as the specific covariate categories, `fic` calculates model comparison statistics which are averaged over the categories, indicated by a value of `ave` in the column `vals`. An equally-weighted average is computed by default. Arbitrary weights can be supplied in the argument `Xwt` to `fic`.

Recall that `mod2` contains one more covariate than `mod1`. For each of the two focuses, and the average, the unadjusted and adjusted bias estimates are lower due to the inclusion of this covariate, while the standard error `se` is higher. Given the lower `rmse` and `rmse.adj` under `mod2`, the reduction in bias is deemed to be worth the increase in uncertainty.

4.1. Comparing a wide range of models

We may want to examine a broad range of models, particularly in regression contexts. The function `all_inds` (a wrapper around `expand.grid`) creates a matrix of indicators that defines all submodels with different covariates, spanned by a given wide model (here `wide.glm`) and a narrow model (here defined by `inds0`). This function works only for all classes of model objects `x` for which the `terms(x)` function is understood, which includes standard R regression models such as `lm` and `glm`. Factors are handled naturally.

```
combs <- all_inds(wide.glm, inds0)
```

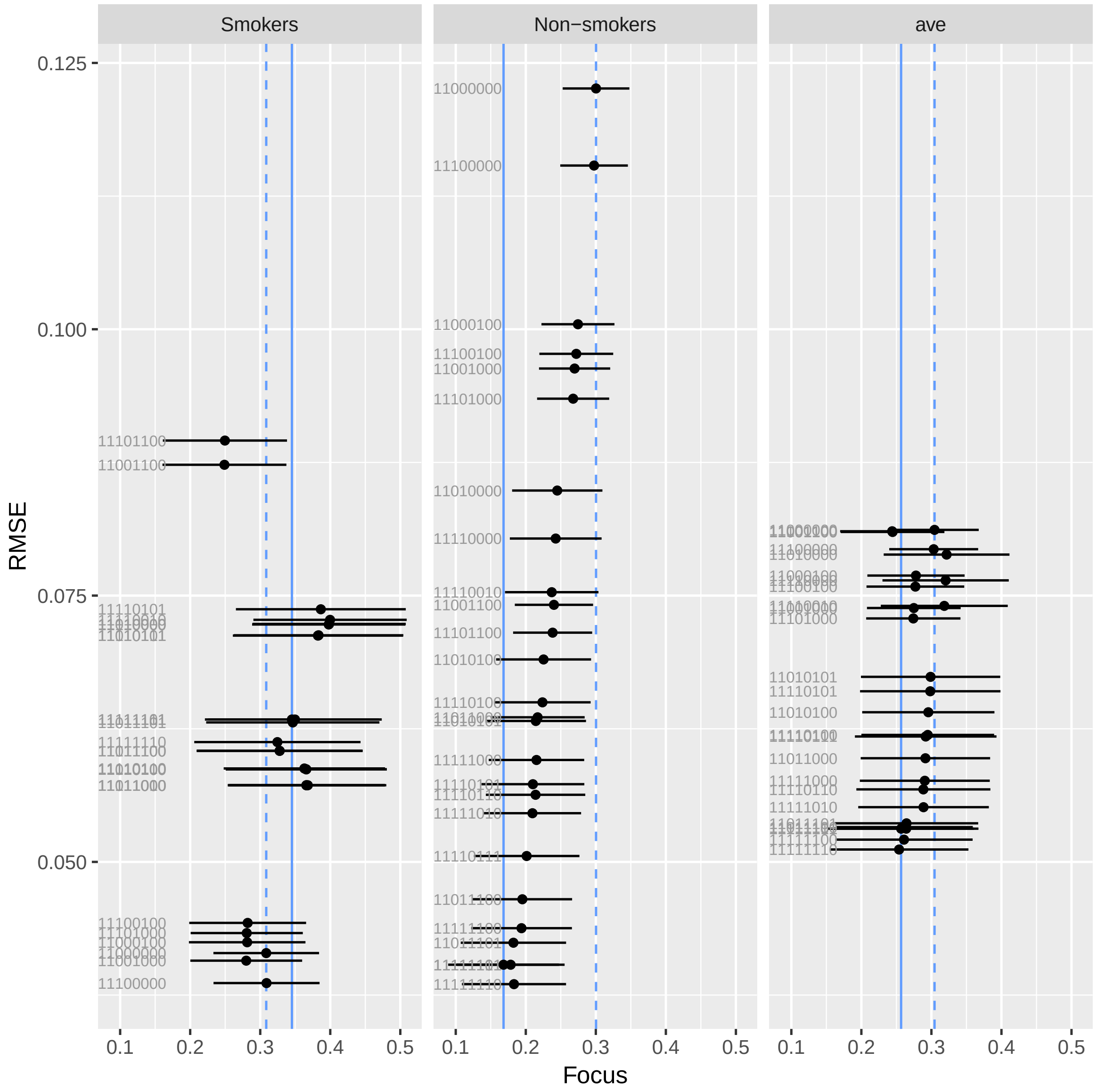
The resulting matrix can be used as the `inds` argument to `fic` to compare all submodels in this example, again for a focus defined by the probability of low birth weight at covariate values defined by `X`. Before calling `fic` again, we redefine `combs` to exclude models with interactions but not both corresponding main effects.

```
combs <- with(combs,
              combs[!((smoke==0 & smokeage==1) |
                      (smoke==0 & smokeui==1) |
                      (age==0 & smokeage==1) |
                      (ui==0 & smokeui==1)),])
ficres <- fic(wide=wide.glm, inds=combs, inds0=inds0, focus=focus, X=X)
```

Notice that some of the `rmse` elements of `ficres` are `NaN`, since the first squared bias estimate $\widehat{B^2}$ is negative. The alternative estimate (6), `rmse.adj`, might be preferred in these cases, since it is consistent with the bias and variance estimates.

A comparison of many models can be illustrated by a scatterplot of the focus estimate against the RMSE of each submodel. The default `plot` method for `fic` objects accomplishes this using base R graphics: try `plot(ficres)`. Alternatively a graph can be plotted using `ggplot2` if this package is installed. This is illustrated here.

```
ggplot_fic(ficres)
```



There is one panel for each of the two covariate categories (smokers and non-smokers) defining the focus (probability of low birth weight) and an average over the two categories. The solid blue line is the focus estimate under the wide model, and the dashed blue line is the focus estimate under the narrow model. An informal illustration of the uncertainty around the estimate of the focus quantity from each submodel is given by the estimate $\pm 1.96 \times \sqrt{\hat{V}}$. Note that this underestimates the uncertainty if inference is based on a selected model — "post-selection" inference (e.g. CURRENT REFS) aims to obtain estimates of error for a selected model which account for the range of models being searched over.

Each submodel is labelled faintly using the row names of the matrix supplied as the `inds` argument to `fic`. In this case, these names were automatically constructed by the function `all_inds` and contain a string of binary 0/1 indicators for the inclusion of eight parameters. For smokers, the narrow model (labelled `11000000`) and similar smaller models give estimates of the probability of low birth weight with the lowest MSE, while by contrast, for non-smokers, the wide model (labelled `11111111`) and similar larger models give the most accurate estimates of the focus quantity. Note that in this dataset, there are 115 non-smokers and 74 smokers, thus more data enables bigger models to be identified for non-smokers. Wider models are also preferred for describing the average population.

The model with the optimal MSE could be identified if necessary, with the `summary` function, though in general, if there are multiple models with similar estimation performance and different results, the reasons for their differences should be explored in greater depth with the aid of background knowledge.

```
summary(ficres)

## $min
##             index focus
## Smokers         5 0.280
## Non-smokers    20 0.183
## ave            20 0.254
##                                                                pars
## Smokers                                    (Intercept),lwtkg,htTRUE
## Non-smokers (Intercept),lwtkg,age,smokeTRUE,htTRUE,uiTRUE,smokeage
## ave         (Intercept),lwtkg,age,smokeTRUE,htTRUE,uiTRUE,smokeage
##
## $ranges
##             min(focus) max(focus) min(RMSE) max(RMSE)
## Smokers          0.249      0.400    0.0403    0.0896
## Non-smokers      0.168      0.300    0.0385    0.1226
## ave              0.244      0.322    0.0512    0.0812
##
## attr(,"class")
## [1] "summary.fic"
```

4.2. Calling “fic” for an unfamiliar class of models

Above, the `fic` function recognised the fitted model objects as GLMs, that is, objects of class `"glm"` returned by the `glm()` function in base R. But the package can be used to calculate focused model comparison statistics for any class of models, not just the special classes it recognises. To do this, it needs to know where two things are stored inside the fitted model objects:

1. `coef`: the vector of maximum likelihood estimates $(\hat{\boldsymbol{\theta}}, \hat{\gamma})$,
2. `vcov`: the covariance matrix of the maximum likelihood estimates, $(nJ)^{-1}$.

plus, if the classic FIC is required, also `nobs`: the number of observations n contributing to the model fit. Given a fitted model object called `mod`, the `fic()` function assumes by default that `coef(mod)`, and `vcov(mod)` respectively return these pieces of information, likewise `nobs(mod)` if FIC is required.

If one or more of these assumptions is not true, the defaults can be changed by supplying the argument `fns` to `fic()`, which should be a named list of three components. Each component should be a function with one argument (the fitted model) which extracts the required information from the fitted model and returns it. For example, the first component of the list below is a function which, when applied to a `glm` object, returns the maximum likelihood estimates of the regression coefficients.

```
fns <- list(coef = function(x)coef(x),
            nobs = function(x)nobs(x),
            vcov = function(x)vcov(x))
fic1 <- fic(wide=wide.glm, inds=inds, inds0=inds0, focus=focus, fns=fns,
            X=X, sub=sub)
```

A full worked example of a using `fic` for a novel class of models, defined and fitted by custom R functions, is given in the package vignette "Examples of focused model comparison: skew-normal models".

4.3. Other classes of models

The **fic** package also has built-in methods for the following classes of models.

- Linear models fitted with `lm` in base R. Examples of covariate selection and polynomial order selection in linear models are given in an additional **fic** vignette, which also illustrates the use of quantiles of the outcome distribution as focuses.

- Parametric survival models fitted with `flexsurvreg` in the **flexsurv** package (Jackson 2016), or `survreg` in the **survival** package (Therneau 2015).

  The additional **fic** vignette "Examples of focused model comparison: parametric survival models" illustrates the use of `fic` to compare parametric baseline hazard functions of different levels of complexity. `fic` needs to be set up carefully here since the same model can be presented in several different parameterisations. Focused comparison requires the submodels to be defined by fixing parameters of the wide model to special values.

- Multi-state models fitted with the **msm** package (Jackson 2011).

  A multi-state model might consist of several regression models, one for each transition between states in a multi-state structure, which poses a challenge to defining and interpreting a focused model comparison. The package vignette "Examples of focused model comparison: multi-state models" gives a worked example. In this example, the focus of interest is a complicated model summary function provided by the **msm** package, which needs to be converted carefully into the format expected by **fic**.

5. Focused covariate selection in Cox proportional hazards regression

In a Cox regression model, time-to-event outcomes t_i are observed on individuals i, potentially with right-censoring. At time t, individual i is assumed to have a hazard $h_i(t)$ which is proportional to their covariate values. We wish to compare models that have different sets of covariates. In the most general "wide" model, $h_i(t) = h_0(t)\exp(\boldsymbol{\theta}^T x_i + \boldsymbol{\gamma}^T z_i)$. The baseline hazard $h_0(t)$ is left unspecified, while $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ are estimated by maximum partial likelihood.

We compare submodels S of this wide model, which include different subsets of covariates, according to how accurately they estimate some focus quantity $\mu = \mu(\boldsymbol{\theta}, \boldsymbol{\gamma}, H_0()|\mathbf{x}, t)$, where $H_0()$ is the cumulative baseline hazard function, which can be estimated nonparametrically by various methods. Typical focus quantities might depend on time t as well as covariate values $\mathbf{x}$, e.g. the probability $S(t|\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\gamma}, H_0())$ that a person with covariates $\mathbf{x}$ will survive t years.

Again the mean square error $MSE_S = B_S^2 + V_S$ of μ of the focus quantity under submodel S has an asymptotically unbiased estimator $\widehat{B_S^2} + \widehat{V_S}$, like that in Section 2, which was derived by Hjort and Claeskens (2006) under the same theoretical principles:

$$\begin{aligned}
\widehat{B_S^2} &= (\hat{\omega} - \hat{\kappa})(I - G_S)(\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}^T - Q)(I - G_S)(\hat{\omega} - \hat{\kappa}) \\
\widehat{V_S} &= \left\{\hat{\tau}_0(t)^2 + (\hat{\omega} - \hat{\kappa})^T Q_S^0 (\hat{\omega} - \hat{\kappa})\right\}/n \\
\hat{\omega} &= J_{10}J_{00}^{-1}\frac{d\mu}{d\boldsymbol{\theta}} - \frac{d\mu}{d\boldsymbol{\gamma}} \\
\hat{\kappa}(t) &= (J_{10}J_{00}^{-1}F_0(t) - F_1(t))\frac{d\mu}{dH_0},
\end{aligned}$$

where $Q_S^0, G_S, \hat{\boldsymbol{\delta}}, J_{10}, J_{00}$ are defined as in Section 2, except in terms of the partial likelihood instead of the likelihood. Newly-defined quantities are

$$F(t) = \int_0^t \left\{G_n^{(1)}(u)/G_n^{(0)}(u)\right\} dH_0(u) = \begin{pmatrix} F_0(t) \\ F_1(t) \end{pmatrix}$$

where $F_0(t)$ and $F_1(t)$ have p and q components respectively,

- $G_n^{(0)}(u) = \frac{1}{n}\sum_{i=1}^n Y_i(u)\exp(x_i^T\boldsymbol{\theta} + z_i^T\boldsymbol{\gamma})$
- $G_n^{(1)}(u) = \frac{1}{n}\sum_{i=1}^n Y_i(u)\exp(x_i^T\boldsymbol{\theta} + z_i^T\boldsymbol{\gamma})\begin{pmatrix} x_i \\ z_i \end{pmatrix}$

both evaluated at the estimates of $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$ and $H_0()$ from the wide model,

$$\tau_0(t)^2 = \left(\frac{d\mu}{dH_0}\right)^2 \int_0^t \frac{dH_0(u)}{g^{(0)}(u, \beta, 0)} + \left\{\frac{d\mu}{d\beta} - \frac{d\mu}{dH_0}F_0(t)\right\}^T J_{00}^{-1}\left\{\frac{d\mu}{d\beta} - \frac{d\mu}{dH_0}F_0(t)\right\}$$

and $Y_i(t) = I(t_i \geq t)$ is the indicator for individual i being at risk at time t, and

As before, if $\widehat{B_S^2} < 0$ we could also use an alternative estimator for MSE_S which assumes that the bias is zero in these cases.

5.1. Example: malignant melanoma

To illustrate the method, Hjort and Claeskens (2006) study a dataset from 205 patients with malignant melanoma, earlier analysed in detail by Andersen *et al.* (1993). This dataset is also provided in the **fic** package.

We compare models ranging from a wide model `wide` that includes 7 terms in the regression model formula, to a narrow model that includes only sex.

```
library(survival)
wide <- coxph(Surv(years, death==1) ~ sex + thick_centred + infilt + epith +
                ulcer + depth + age, data=melanoma)

## Warning in model.matrix.default(Terms, mf, contrasts = contrast.arg):  partial
argument match of ’contrasts’ to ’contrasts.arg’
## Warning in seq.default(along = temp):  partial argument match of ’along’ to
’along.with’
## Warning in seq.default(1L, by = 2L, length = lastrow.length):  partial argument
match of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
## Warning in seq.default(1L, by = 2L, length = start):  partial argument match
of ’length’ to ’length.out’
```

In this example, we need to deal with *factor* terms in the model when setting up the `inds` and `inds0` indicators to supply to `fic`. Specifically, `infilt` and `depth` are factors with 4 and 3 levels respectively, represented in the model by 3 and 2 model parameters respectively, instead of one parameter for each. The remaining terms in the model are each associated with one parameter. Thus the wide model, with 7 terms, includes 10 parameters.

The function `expand_inds` can be used to construct `inds` or `inds0` terms in the presence of factors[3]. We supply a vector of 7 elements, indicating the presence or absence of each of the 7 terms in the model formula. In this case, only the first term, `sex`, is included in the narrow model. Then to create an `inds0` vector of 10 elements, indicating the presence or absence of each of the wide model's 10 parameters in the narrow model, we call

```
inds0 <- expand_inds(c(1,0,0,0,0,0,0), wide)

## Warning in model.matrix.default(Terms, mf, contrasts = contrast.arg):  partial
argument match of ’contrasts’ to ’contrasts.arg’

inds0
```

[3] This function only works for classes of models for which the `model.matrix` function is understood and returns objects with an `"assign"` attribute. This includes all the commonly-used models in base R.

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    0    0    0    0    0    0    0    0     0
```

Note that in Cox regression there is no intercept parameter, therefore the model parameters include only the regression coefficients. In fully parametric regression models, for example GLMs, the vector supplied to `expand_inds` should contain an additional element indicating the presence of the intercept.

The `fic` package includes three built-in alternative focuses, as specified through the `focus` argument to `fic`.

- `focus="hr"`: the hazard ratio between an individual with covariates X and an individual with covariates 0 (which by definition of the Cox model is independent of time t)
- `focus="survival"`: the survival probability at time t, for an individual with covariates X
- `focus="cumhaz"`: the cumulative hazard at time t, for an individual with covariates X

The required covariate values and/or time point(s) are supplied as the `X` and `t` arguments to `fic`.

It is possible to supply alternative focuses, though this is slightly trickier than for standard full likelihood models. It requires the user to supply a list of three functions as the `focus` argument to `fic`, one returning the focus, and two returning its derivatives with respect to $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ and $H_0(t)$ respectively. The functions take arguments `par`, `H0`, `X` and `t` representing the parameter vector, cumulative hazard, covariate matrix and time. For examples, examine the code for the following lists of functions, which are used for the three built-in focuses above.

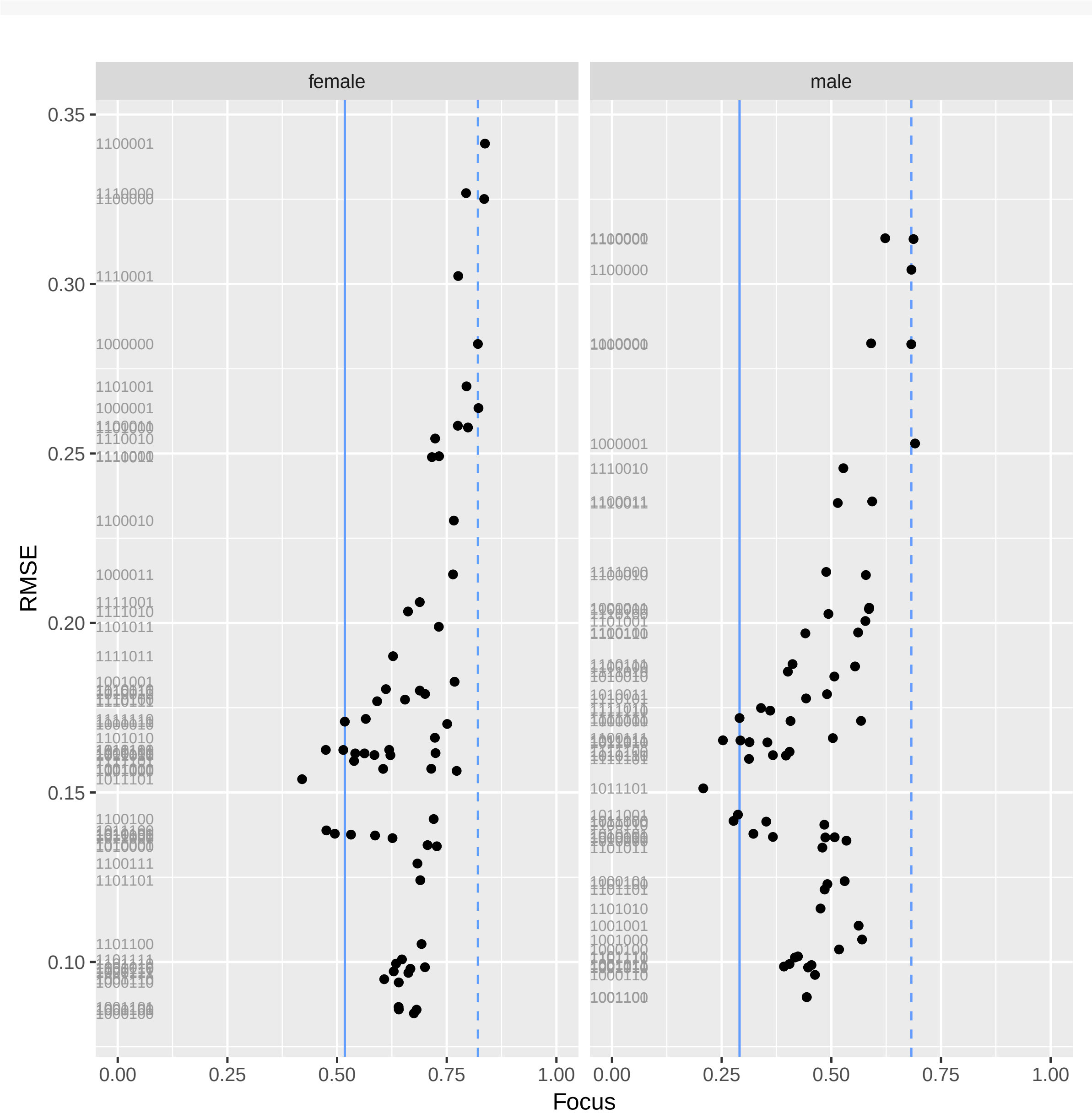
```
list(focus=fic:::cox_hr, deriv=fic:::cox_hr_deriv,
     dH=fic:::cox_hr_dH)
list(focus=fic:::cox_cumhaz, deriv=fic:::cox_cumhaz_deriv,
     dH=fic:::cox_cumhaz_dH)
list(focus=fic:::cox_survival, deriv=fic:::cox_survival_deriv,
     dH=fic:::cox_survival_dH)
```

In the melanoma example, all possible submodels spanning the wide and narrow model are compared. As before, a matrix of indicators describing these models is constructed, and the submodels are fitted automatically within the `fic` function.

```
combs <- all_inds(wide,inds0)

## Warning in model.matrix.default(Terms, mf, contrasts = contrast.arg):  partial
argument match of 'contrasts' to 'contrasts.arg'
```

The focus is defined as the 5 year survival probability (`focus="survival",t=5`) for the covariate values defined by `X`, here taken as men and women separately, with average age and mean observed tumour thickness among men and women, infiltration level 4, epithelioid cells and ulceration present, and invasion depth 2. The utility `newdata_to_X` is used to convert the user-defined data frame `newdata` that identifies these covariate values, with one variable per covariate or factor, to a design matrix `X`, with one column for each of the 10 parameter values.[4]



The models give a big range of estimates for the focus survival probability. Generally, the models returning higher survival estimates are those models closer to the narrow model, and the error of these estimates is high. The most accurate models, as judged by the MSE of this focus, return survival estimates of around 0.6–0.7 for men and 0.4–0.5 for women. If desired, the model with the lowest RMSE could be identified with the `summary` method.

[4] Note the intercept is excluded here, as Cox models don't have a regression intercept term. In GLMs, the design matrix usually includes an intercept.

```
summary(ficall)

## $min
##        index focus                                  pars
## female    61 0.629 sexmale,epith,ulcer,depth2,depth3,age
## male      21 0.447           sexmale,epith,depth2,depth3
##
## $ranges
##        min(focus) max(focus) min(RMSE) max(RMSE)
## female      0.420      0.837    0.0291     0.341
## male        0.208      0.691    0.0408     0.314
##
## attr(,"class")
## [1] "summary.fic"
```

However, the reasons for the variation in results between models should be investigated further, with the aid of clinical background knowledge.

6. Discussion

In focused model comparison, we explicitly connect statistical models with the scientific questions that the models were designed to address. Different models might give better estimates of different focus quantities of interest. Focused information criteria enable models to be compared according to how well they balance the bias and variance of an estimate of interest. The criteria can be decomposed into bias and variance terms that allow this trade-off to be examined directly.

The **fic** package performs focused model comparison easily for any class of models fitted by maximum likelihood. The package is designed to be easily extensible, by adding new model classes and focuses. The vignettes illustrate some examples, and further contributions of code and worked examples would be welcome. More advanced models, beyond standard maximum likelihood, might need novel focused criteria to be implemented. Some generalisations were discussed by Claeskens and Hjort (2008), for example, for mixed models, missing data, and where a submodel is on the boundary of the parameter space. Gueuning and Claeskens (2018) developed focused information criteria for situations where one or more of the models being compared is not identifiable from the data, for example, high-dimensional regression. *ANY MORE RELEVANT LITERATURE?*

Model uncertainty is an area of ongoing research. It is often not wise to highlight only the results of the single model that is selected by a criterion. If multiple well-performing models give different estimates, then the uncertainty about the model choice should be acknowledged. If a single "best" estimate is desired, then this could be produced by a model-averaged estimator (see, e.g. Hjort and Claeskens (2003), Claeskens and Hjort (2008), *[any more recent literature]*). Alternatively, if we wish to highlight a single best model, this could provide the point estimate, but coupled with a confidence interval that acknowledges the range of models that have been selected from. In either case, determining appropriate confidence intervals or standard errors is challenging *[BEST RECENT REFS?]* Bootstrapping, e.g. by resampling the data, or resampling parameters from their asymptotic sampling distributions under a wide model, is an attractive method of handling model uncertainty that is simple to implement in software, though its theoretical properties in this context are not well understood. See, e.g. Claeskens and Hjort (2008), Breiman (1996), Jackson *et al.* (2010), *[ANYTHING ELSE]* for examples and discussion.

Estimation is often followed by decision making. Formal decision theory might also lend itself to focused model comparison. For example we might consider the "focus" to be the decision among discrete actions with optimal expected loss. Again, bootstrapping might provide a route to focused model comparison, e.g. by calculating the average loss, over a set of bootstrap resamples, of decisions made under competing models.

Focused model comparison principles might also be used for Bayesian inference. In realistic situations, the true process is typically more complex than the biggest model that can be identified from the data. In a Bayesian perspective, we might use a theoretically-realistic model, but with a prior that encodes external information to enable the parameters to be identified from the data, e.g. through shrinkage, regularisation or elicited judgements. However, the appropriate model or prior might still be uncertain, and we may have multiple competing models. Claeskens and Hjort (2008) present some equivalents of FIC for Bayesian estimators where there is prior information about departures from the narrow model. Various criteria, with justifications related to cross-validation or predictive loss, have been developed to compare Bayesian models (Spiegelhalter *et al.* 2002; Plummer 2008; Watanabe 2013; Vehtari *et al.* 2017) or average their estimates (Yao *et al.* 2018). These might also be extended to enable models to be compared for different focuses. *[ANYTHING ELSE BAYESIAN?]*

References

Akaike H (1973). "Information theory and an extension of the maximum likelihood principle." In B Petrov, F Csaki (eds.), *2nd International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.

Andersen PK, Borgan O, Gill RD, Keiding N (1993). *Statistical models based on counting processes*. Springer, New York.

Breiman L (1996). "Bagging predictors." *Machine learning*, **24**(2), 123–140.

Burnham KP, Anderson DR (2003). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer.

Claeskens G, Hjort N (2003). "The focused information criterion (with discussion)." *Journal of the American Statistical Association*, **98**(464), 900–945.

Claeskens G, Hjort N (2008). *Model selection and model averaging*. Cambridge University Press.

Gilbert P, Varadhan R (2016). *numDeriv: Accurate Numerical Derivatives*. R package version 2016.8-1, URL `https://CRAN.R-project.org/package=numDeriv`.

Gueuning T, Claeskens G (2018). "A High-dimensional Focused Information Criterion." *Scandinavian Journal of Statistics*, **45**(1), 34–61.

Hjort N, Claeskens G (2003). "Frequentist model average estimators." *Journal of the American Statistical Association*, **98**(464), 879–899.

Hjort NL, Claeskens G (2006). "Focused information criteria and model averaging for the Cox hazard regression model." *Journal of the American Statistical Association*, **101**(476), 1449–1464.

Hosmer DW, Lemeshow S (1989). *Applied Logistic Regression*. John Wiley & Sons.

Jackson C (2016). "flexsurv: A Platform for Parametric Survival Modeling in R." *Journal of Statistical Software*, **70**(8), 1–33. doi:10.18637/jss.v070.i08.

Jackson CH (2011). "Multi-State Models for Panel Data: The msm Package for R." *Journal of Statistical Software*, **38**(8).

Jackson CH, Sharples LD, Thompson SG (2010). "Structural and parameter uncertainty in Bayesian cost-effectiveness models." **59**(2), 233–253.

Kass RE, Wasserman L (1995). "A reference Bayesian test for nested hypotheses with large samples." *Journal of the American Statistical Association*, **90**, 928–934.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523–539.

Schwarz G (1978). "Estimating the dimension of a model." *The Annals of Statistics*, **6**(2), 461–464.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit (with discussion)." **64**(4), 583–639.

Therneau TM (2015). *A Package for Survival Analysis in S*. Version 2.38, URL https://CRAN.R-project.org/package=survival.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC." *Statistics and Computing*, **27**(5), 1413–1432.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition edition. Springer.

Watanabe S (2013). "A widely applicable Bayesian information criterion." *Journal of Machine Learning Research*, **14**(Mar), 867–897.

Yao Y, Vehtari A, Simpson D, Gelman A (2018). "Using stacking to average Bayesian predictive distributions (with discussion)." *Bayesian Analysis*, **13**(3), 917–1003.