Multivariable Fractional Polynomials

Axel Benner

August 10, 2004

Contents

1 Introduction

The `mfp` package is a collection of R [3] functions targeted at the use of fractional polynomials (FP) for modelling the relationship between a continuous covariate and the outcome in regression models, as introduced by Royston & Altman (1994) [4] and modified by Sauerbrei & Royston (1999) [6]. It combines backward elimination with a systematic search for a 'suitable' transformation to represent the influence of each continuous covariate on the outcome. An application of multivariable fractional polynomials (MFP) in modelling prognostic and diagnostic factors in breast cancer is given by [6]. The stability of the models selected is investigated in [5]. Briefly, fractional polynomials models are useful when one wishes to preserve the continuous nature of the covariates in a regression model, but suspects that some or all of the relationships may be non-linear. At each step of a 'backfitting' algorithm MFP constructs a fractional polynomial transformation for each continuous covariate while fixing the current functional forms of the other covariates. The algorithm terminates when the functional forms of the covariates do not change anymore.

2 Inventory of functions

`mfp.object` is an object representing a fitted `mfp` model. Class `mfp` inherits from either glm or coxph depending on the type of model fitted. In addition to the standard glm/coxph components the following components are included in an `mfp` object

x the final FP transformations that are contained in the design matrix x. The predictor "z" with 4 df would have corresponding columns "z.1" and "z.2" in x.

powers a matrix containing the best FP powers for each predictor. If a predictor has less than two powers a NA will fill the appropriate cell of the matrix.

pvalues a matrix containing the P-values from the closed tests. Briefly p.null is the P-value for the test of inclusion (see mfp), p.lin corresponds to the test of nonlinearity and p.FP the test of simplification. The best m=1 power (power2) and best m=2 powers (power4.1 and power4.2) are also given.

scale all predictors are shifted and rescaled before being power transformed if nonpositive values are encountered or the range of the predictor is reasonably large. If x' would be used instead of x where x' = (x+a)/b the parameters a (shift) and b (scale) are contained in the matrix scale.

df.initial a vector containing the degrees of freedom allocated to each predictor.

df.final a vector containing the degrees of freedom of each predictor at convergence of the backfitting algorithm.

dev the deviance of the final model.

dev.lin the deviance of the model that has every predictor included with 1 df (i.e. linear).

dev.null the deviance of the null model.

fptable the table of the final fp transformations.

3 Usage in R

Start with

```
>library(mfp)
```

An `mfp.object` will be created by application of function `mfp`.

```
>str(mfp)
```

```
function (formula = formula(data), data = parent.frame(), family = gaussian,
    subset, na.action, init, alpha = 0.05, select = 1, verbose = FALSE,
    x = TRUE, y = TRUE)
```

A typical predictor has the form `response` $\sim$ `terms` where `response` is the (numeric) response vector and `terms` is a series of terms, separated by + operators, which specifies a linear predictor for `response` and provided by the `formula` argument of the function call. Fractional polynomial terms are indicated by `fp`.
For `binomial` models the response can also be specified as a `factor`. If a Cox proportional hazards model is required then the outcome need to be specified using the `Surv()` notation.
The argument `family` describes the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function.
`alpha` sets the FP selection level for all predictors. Values for individual predictors are changed using the `fp` function. `select` sets the variable selection level for all predictors. Values for individual predictors are set using the `fp` function in the formula.
The function `fp` defines a fractional polynomial object for a single input variable.

```
>str(fp)
```

```
function (x, df = 4, select = NA, alpha = NA, scale = TRUE)
```

In addition to `alpha` and `select` the `scale` argument of the `fp` function denotes the use of pre-transformation scaling to avoid possible numerical problems.

3.1 Model selection

The original implementation of `mfp` uses two different selection procedures for a single continuous covariate x, a sequential selection procedure and a closed testing selection procedure (`ra2`). In the R implementation only the `ra2` algorithm is used. For the sequential selection procedure the actual Type I error rate may exceed the nominal value when the true relationship is a straight line. Therefore the procedure tends to favour more complex models over simple ones.
The `ra2` algorithm is described in [1] and in [7]. It uses a closed test procedure [2] which maintains approximately the correct Type I error rate for each component test. The procedure allows the complexity of candidate models to increase progressively from a prespecified minimum (a null model) to a prespecified maximum (an FP) according to an ordered sequence of test results.
The algorithm works as follows:

1. Perform a 4 df test at the α level of the best-fitting second-degree FP against the null model. If the test is not significant, drop x and stop, otherwise continue.

2. Perform a 3 df test at the α level of the best-fitting second-degree FP against a straight line. If the test is not significant, stop (the final model is a straight line), otherwise continue.

3. Perform a 2 df test at the α level of the best-fitting second-degree FP against the best-fitting first-degree FP. If the test is significant, the final model is the FP with $m = 2$, otherwise the FP with $m = 1$.

The tests in step 1, 2 and 3 are of overall association, non-linearity and between a simpler or more complex FP model, respectively.

4 Example

4.1 Cox proportional hazards model

We use the dataset `GBSG` which contains data from a study of the German Breast Cancer Study Group for patients with node-positive breast cancer.

```
>data(GBSG)
>str(GBSG)

`data.frame':        686 obs. of  11 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ htreat  : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 2 1 1 1 ...
 $ age     : int  70 56 58 59 73 32 59 65 80 66 ...
 $ menostat: Factor w/ 2 levels "1","2": 2 2 2 2 2 1 2 2 2 2 ...
 $ tumsize : int  21 12 35 17 35 57 8 16 39 18 ...
 $ tumgrad : Factor w/ 3 levels "1","2","3": 2 2 2 2 2 3 2 2 2 2 ...
 $ posnodal: int  3 7 9 4 1 24 2 1 30 7 ...
 $ prm     : int  49 62 53 61 27 1 182 193 1 1 ...
 $ esm     : int  67 78 272 30 66 14 1 26 60 4 ...
 $ rfst    : int  1814 2018 712 1807 772 448 2172 2161 471 2014 ...
 $ cens    : int  1 1 1 1 1 1 0 0 1 0 ...
```

The response variable is recurrence free survival time (`Surv(rfst, cens)`). Complete data on 7 prognostic factors is available for 686 patients. The median follow-up was about 5 years, 299 events were observed for recurrence free survival time. We use a Cox regression to model the hazard of

recurrence by the 7 prognostic factors of which 5 are continuous, age of the patients in years (`age`), tumor size in mm (`tumsize`), number of positive nodes (`posnodal`), progesterone receptor in fmol (`prm`), estrogen receptor in fmol (`esm`); two are binary, hormonal therapy (`htreat`), menopausal status (`menostat`); and one is ordered categorical with three levels, tumor grade (`tumgrad`).
We use `mfp` to build a model from the initial set of 7 covariates using the backfitting model selection algorithm. We set the global selection level to 0.05.
If one uses `fp()` in the formula a fractional polynomial transformation with pre-transformation scaling is estimated. This is done here for `tumsize`, `posnodal`, `prm`, and `esm`. Otherwise a linear form of the unscaled variable is used, as for `age`. Categorical factors are included without transformation. Alternatively a categorical variable can be applied to define different strata. In the example hormonal therapy (`htreat`) was used as stratification variable.
To be compatible with other implementations of multiple fractional polynomials in SAS and Stata we use the Breslow method for tie handling.

```
>f <- mfp(Surv(rfst, cens) ~ strata(htreat) +
+     age + fp(tumsize) + fp(posnodal) +
+     fp(prm) + fp(esm) + menostat + tumgrad,
+     family = cox, data = GBSG, verbose = T)

Loading required package: survival

        Variable        Deviance        Power(s)
-----------------------------------------------
Cycle 1
        esm                3103.474
                           3103.245        1
                           3101.075        -0.5
                           3099.18                 -0.5 3

        menostat2          3105.271
                           3103.245        1



        tumsize                 3107.028
                           3103.245        1
                           3099.637        -1
                           3099.372        -1 3

        tumgrad2           3110.756
                           3103.245        1



        age                3104.28
                           3103.245        1



        tumgrad3           3112.682
```

```
                    3103.245        1


        prm             3122.678
                    3103.245        1
                    3093.632        0
                    3091.834        0 3

        posnodal    3127.436
                    3093.632        1
                    3074.339        0
                    3069.143        0.5 3

Cycle 2
        esm             3075.836
                    3074.339        1
                    3073.715        2
                    3073.371        -1 2

        menostat2   3075.583
                    3074.339        1


        tumsize             3075.484
                    3074.339        1
                    3072.31                 -1
                    3071.963        -0.5 -0.5

        tumgrad2    3079.064
                    3074.339        1


        age             3075.398
                    3074.339        1


        tumgrad3    3078.587
                    3074.339        1


        prm             3103.264
                    3081.123        1
                    3073.535        0.5
                    3071.891        0 3
```

```
        posnodal        3127.183
                         3094.864        1
                         3073.535        0
                         3067.746        0.5 3

Cycle 3
        esm                 3075.356
                         3073.535        1
                         3072.404        2
                         3071.932        -0.5 2

        menostat2        3074.637
                         3073.535        1



        tumsize                3075.088
                         3073.535        1
                         3071.587        -1
                         3071.279        -1 3

        tumgrad2        3078.723
                         3073.535        1



        age                 3074.326
                         3073.535        1



        tumgrad3        3078.58
                         3073.535        1



Tansformation
          shift scale
esm           0  1000
menostat2     0     1
tumsize       0   100
tumgrad2      0     1
age           0     1
tumgrad3      0     1
prm           0  1000
posnodal      0    10
```

```
Fractional polynomials
          df.initial select alpha df.final power1
esm                4      1  0.05        1      1
menostat2          1      1  0.05        1      1
tumsize            4      1  0.05        1      1
tumgrad2           1      1  0.05        1      1
age                1      1  0.05        1      1
tumgrad3           1      1  0.05        1      1
prm                4      1  0.05        2    0.5
posnodal           4      1  0.05        2      0
          power2
esm            .
menostat2      .
tumsize        .
tumgrad2       .
age            .
tumgrad3       .
prm            .
posnodal       .


Null model: 3198.026
Linear model: 3103.245
Final model: 3073.535
```

After three cycles the final model is selected. None of the possible input variables were excluded from the model. Only for variables `prm` and `posnodal` nonlinear transformations were chosen. Prescaling was used for esm, prm, tumsize and posnodal.

```
>summary(f)

Call:
mfp(formula = Surv(rfst, cens) ~ strata(htreat) + age + fp(tumsize) +
    fp(posnodal) + fp(prm) + fp(esm) + menostat + tumgrad, data = GBSG,
    family = cox, verbose = T)

  n= 686

                 coef exp(coef) se(coef)     z
esm.1        0.000645     1.001 0.000459  1.41
menostat2.1  0.192629     1.212 0.183328  1.05
tumsize.1    0.005010     1.005 0.003959  1.27
tumgrad2.1   0.533831     1.705 0.250854  2.13
age.1       -0.008193     0.992 0.009204 -0.89
tumgrad3.1   0.585888     1.797 0.273398  2.14
prm.1       -0.061745     0.940 0.012081 -5.11
posnodal.1   0.487775     1.629 0.066366  7.35
                  p
esm.1       1.6e-01
menostat2.1 2.9e-01
```

```
tumsize.1   2.1e-01
tumgrad2.1  3.3e-02
age.1       3.7e-01
tumgrad3.1  3.2e-02
prm.1       3.2e-07
posnodal.1  2.0e-13

            exp(coef) exp(-coef) lower .95
esm.1           1.001      0.999     1.000
menostat2.1     1.212      0.825     0.846
tumsize.1       1.005      0.995     0.997
tumgrad2.1      1.705      0.586     1.043
age.1           0.992      1.008     0.974
tumgrad3.1      1.797      0.557     1.051
prm.1           0.940      1.064     0.918
posnodal.1      1.629      0.614     1.430
            upper .95
esm.1           1.002
menostat2.1     1.737
tumsize.1       1.013
tumgrad2.1      2.788
age.1           1.010
tumgrad3.1      3.070
prm.1           0.963
posnodal.1      1.855

Rsquare= 0.166   (max possible= 0.991 )
Likelihood ratio test= 124  on 8 df,   p=0
Wald test            = 120  on 8 df,   p=0
Score (logrank) test = 125  on 8 df,   p=0
```

The final model states a sqrt-transformation for prm `prm/1000`.
The function `plot.mfp` draws three plots: the linear predictor function, a plot of the partial residuals together with a lowess smooth, and smoothed martingale based residuals of the null model (1).

References

[1] Ambler, G., and Royston, P. Fractional polynomial model selection procedures: investigation of Type I error rate. *Journal of Statistical Simulation and Computation 69* (2001), 89–108.

[2] Marcus, R., Peritz, E., and Gabriel, K. On closed test procedures with special reference to ordered analysis of variance. *Biometrika 76* (1976), 655–660.

[3] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-00-3.

[4] Royston, P., and Altman, D. G. Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion). *Applied Statistics 43*, 3 (1994), 429–467.

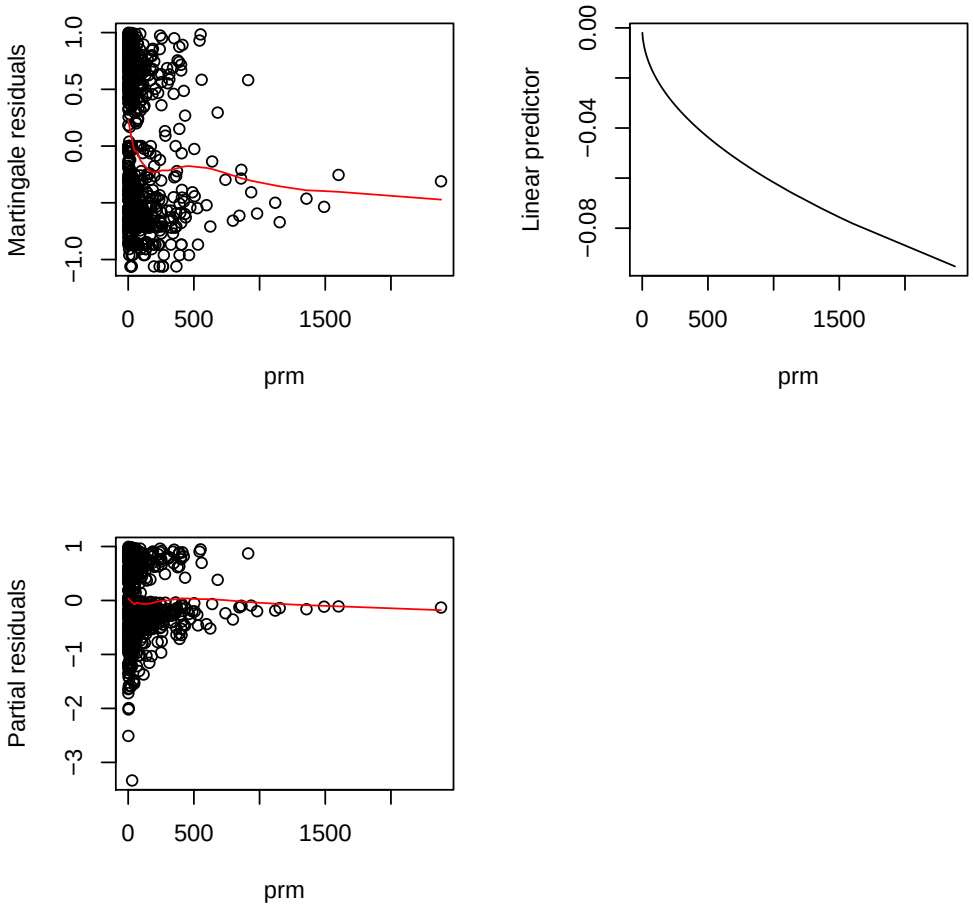

Figure 1: Smoothed null model martingale residuals, the plot of the estimated functional form of the influence of `prm` on the log relative risk of tumor recurrence, and the partial residuals plot for `prm`.

[5] Royston, P., and Sauerbrei, W. Stability of multivariable fractional polynomial models with selection of variables and transformations: a bootstrap investigation. *Statistics in Medicine 22* (2003), 639–659.

[6] Sauerbrei, W., and Royston, P. Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society (Series A) 162* (1999), 71–94.

[7] Sauerbrei, W., and Royston, P. Corrigendum: Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society (Series A) 165* (2002), 399–400.