segmented: An R Package to Fit Regression Models with Broken-Line Relationships

by Vito M. R. Muggeo

Introduction

Segmented or *broken-line* models are regression models where the relationships between the response and one or more explanatory variables are piecewise linear, namely represented by two or more straight lines connected at unknown values: these values are usually referred as breakpoints, changepoints or even joinpoints. Hereafter we use such words indistinctly.

Broken-line relationships are common in many fields, including epidemiology, occupational medicine, toxicology, and ecology, where sometimes it is of interest to assess threshold value where the effect of the covariate changes (Ulm, 1991; Betts et al., 2007).

Formulating the model, estimation and testing

A segmented relationship between the mean response $\mu = E[Y]$ and the variable Z, for observation $i = 1, 2, \dots, n$ is modelled by adding in the linear predictor the following terms

$$\beta_1 z_i + \beta_2 (z_i - \psi)_+ \qquad (1)$$

where $(z_i - \psi)_+ = (z_i - \psi) \times I(z_i > \psi)$ and $I(\cdot)$ is the indicator function equal to one when the statement is true. According to such parameterization, β_1 is the left slope, β_2 is the difference-in-slopes and ψ is the breakpoint. In this paper we tacitly assume a GLM with a known link function and possible additional covariates, x_i, with linear parameters δ, namely $link(\mu_i) = x_i'\delta + \beta_1 z_i + \beta_2(z_i - \psi)_+$; however, since the discussed methods only depend on (1), we leave out from our presentation the response, the link function, and the possible linear covariates.

Breakpoints and slopes of such segmented relationship are usually of main interest, although parameters relevant to the additional covariates may be of some concern. Difficulties in estimating and testing problems are well-known in such models, see for instance Hinkley (1971). A simple and common approach to estimate the model is via grid-search type algorithms: basically, given a grid of possible candidate values of $\{\psi_k\}_{k=1,\dots,K}$, one fits K linear models and seeks for the value corresponding to the model with the best fit. There are at least two drawbacks in using this procedure: (i) estimation might be quite cumbersome with more than one breakpoint and/or with large datasets and (ii) depending on sample size and configuration of data, estimating the model with fixed changepoint may lead the standard error of the other parameters to be too narrow, since uncertainty in the breakpoint is not taken into account.

The package **segmented** offers facilities to estimate and summarize generalized linear models with segmented relationships; virtually, no limit on the number of segmented variables and on the number of changepoint exists. **segmented** uses a method that allows the modeler to estimate simultaneously all the model parameters yielding also, at the possible convergence, the approximate full covariance matrix.

Estimation

Muggeo (2003) shows that the nonlinear term (1) has an approximate intrinsic linear representation which, to some extent, allows us to translate the problem into the standard linear framework: given an initial guess for the breakpoint, $\tilde{\psi}$ say, **segmented** attempts to estimate model (1) by fitting iteratively the linear model with linear predictor

$$\beta_1 z_i + \beta_2 (z_i - \tilde{\psi})_+ + \gamma I(z_i > \tilde{\psi})^- \qquad (2)$$

where $I(\cdot)^- = -I(\cdot)$ and γ is the parameter which may be understood as a re-parameterization of ψ and therefore accounts for the breakpoint estimation. At each iteration, a standard linear model is fitted, and the breakpoint value is updated via $\hat{\psi} = \tilde{\psi} + \hat{\gamma}/\hat{\beta}_2$; note that $\hat{\gamma}$ measures the gap, at the current estimate of ψ, between the two fitted straight lines coming from model (2). When the algorithm converges, the 'gap' should be small, i.e. $\hat{\gamma} \approx 0$, and the standard error of $\hat{\psi}$ can be obtained via the Delta method for the ratio $\frac{\hat{\gamma}}{\hat{\beta}_2}$ which reduces to $\mathrm{SE}(\hat{\gamma})/|\hat{\beta}_2|$ if $\hat{\gamma} = 0$.

The idea may be used to fit multiple segmented relationships, only by including in the linear predictor the appropriate constructed variables for the additional breakpoints to be estimated: at each step, every breakpoint estimate is updated through the relevant 'gap' and 'difference-in-slope' coefficients. Due to its computational facility, the algorithm is able to perform multiple breakpoint estimation in a very efficient way.

Testing for a breakpoint

If the breakpoint does not exist the difference-in-slopes parameter has to be zero, then a natural test for the existence of ψ is

$$H_0 : \beta_2(\psi) = 0. \qquad (3)$$

Note that here we write $\beta_2(\psi)$ to stress that the parameter of interest, β_2, depends on a nuisance parameter, ψ, which vanishes under H_0. Conditions for validity of standard statistical tests (Wald, for instance) are not satisfied. More specifically, the p-value returned by classical tests is heavily underestimated, with an empirical levels about three to five times larger than the nominal levels. **segmented** employs the Davies (1987) test for performing hypothesis (3). It works as follows: given K fixed ordered values of breakpoints $\psi_1 < \psi_2 < \ldots < \psi_K$ in the range of $\mathcal{Z}$, and relevant K values of the test statistic $\{S(\psi_k)\}_{k=1,\ldots,K}$ having a standard Normal distribution for fixed ψ_k, Davies provides an upper bound given by

$$p\text{-value} \approx \Phi(-M) + V \exp\{-M^2/2\}(8\pi)^{-1/2} \quad (4)$$

where $M = max\{S(\psi_k)\}_k$ is the maximum of the K test statistics, $\Phi(\cdot)$ is the standard Normal distribution function, and $V = \sum_k(|S(\psi_k) - S(\psi_{k-1})|)$ is the total variation of $\{S(\psi_k)\}_k$. Formula (4) is an upper bound, hence the reported p-value is somewhat overestimated and the test is slightly conservative. Davies does not provide guidelines for selecting number and location of the fixed values $\{\psi_k\}_k$, however a reasonable strategy is to use the quantiles of the distribution of $\mathcal{Z}$; some simulation experiments have shown that $5 \leq K \leq 10$ usually suffices. Formula (4) refers to one-sided hypothesis test, the alternative being $H_1 : \beta_2(\psi) > 0$. The p-value for the 'lesser' alternative is obtained by using $M = min\{S(\psi_k)\}_k$, while for the two-sided case let $M = max\{|S(\psi_k)|\}_k$ and double the (4) (Davies, 1987).

The Davies test is appropriate for testing for a breakpoint, but it does not appear useful for selecting the number of the joinpoints. Following results by Tiwari et al. (2005), we suggest using the BIC for this aim.

Examples

Black dots in Figure 1 plotted on the logit scale, show the percentages of babies with Down Syndrome (DS) on births for mothers with different age groups (Davison and Hinkley, 1997, p.371). It is well-known that the risk of DS increases with the mother's age, but it is important to assess where and how such a risk changes with respect to the mother age. Presumably, at least three questions have to answered: (i) does the mother's age increase the risk of DS?; (ii) is the risk constant over the whole range of age? and (iii) if the risk is age-dependent, does a threshold value exist?

In a wider context, the problem is to estimate the broken-line model and to provide point estimates and relevant uncertainty measures of all the model parameters. The steps to be followed are straightforward with **segmented**. First, a standard GLM is estimated and a broken-line relationship is added afterwards by re-fitting the overall model. The code below uses the dataframe `down` shipped with the package.

```
> library("segmented")
> data("down")
> fit.glm<-glm(cases/births~age, weight=
+   births, family=binomial, data=down)
> fit.seg<-segmented(fit.glm, seg.Z=~age,
+   psi=25)
```

`segmented` takes the original (G)LM object (`fit.glm`) and fits a new model taking into account the piecewise linear relationship. The argument `seg.Z` is a formula (without response) which specifies the variable, possibly more than one, supposed to have a piecewise relationship, while in the `psi` argument the initial guess for the breakpoint must be supplied.

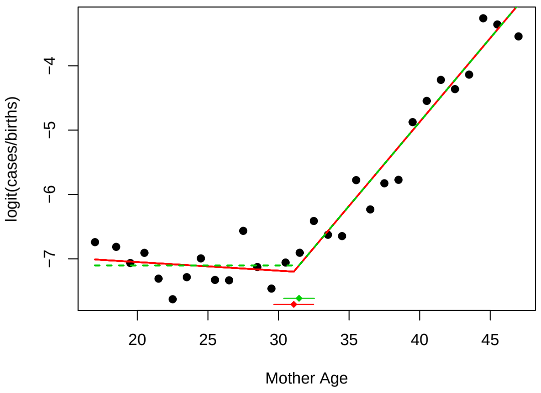

Figure 1: Scatter plot (on the logit scale) of proportion of babies with DS against mother's age and fits from models `fit.seg` and `fit.seg1`.

The estimated model can be visualized by the relevant methods `print()`, `summary()` and `print.summary()` of class `segmented`. The summary shown in Figure 2 is very similar to one of `summary.glm()`. Additional printed information include the estimated breakpoint and relevant (approximate) standard error (computed via $\text{SE}(\hat{\psi}) = \text{SE}(\hat{\gamma})/|\hat{\beta}_2|$), the t value for the 'gap' variable which should be 'small' ($|t| < 2$, say) when the algorithm converges, and the number of iterations employed to fit the model. The variable labeled with `U1.age` stands for the 'difference-in-slope parameter

```
> summary(fit.seg)

        ***Regression Model with Segmented Relationship(s)***

Call: segmented.glm(obj = fit.glm, seg.Z = ~age, psi = 25)

Estimated Break-Point(s):
    Est.  St.Err
31.0800  0.7242

t value for the gap-variable(s) V:  7.367959e-13

Meaningful coefficients of the linear terms:
               Estimate Std. Error     z value     Pr(>|z|)
(Intercept) -6.78243778 0.43140674 -15.7216777 1.074406e-55
age         -0.01341037 0.01794710  -0.7472162 4.549330e-01
U1.age       0.27422124 0.02323945  11.7998172           NA

(Dispersion parameter for binomial family taken to be 1)

Null     deviance: 625.210  on 29  degrees of freedom
Residual deviance:  43.939  on 26  degrees of freedom
AIC: 190.82

Convergence attained in 5 iterations with relative change 1.455411e-14
```

Figure 2: Output of `summary.segmented()`

of the variable age′ (β_2 in equation (1)) and the estimate of the gap parameter γ is omitted since it is just a trick to estimate ψ. Note, however, that the model degrees of freedom are correctly computed and displayed.

Also notice that the p-value relevant to `U1.age` is not reported, and `NA` is printed. The reason is that, as discussed previously, standard asymptotics do not apply. In order to test for a significant difference-in-slope, the Davies' test can be used. The use of `davies.test()` is straightforward and requires to specify the regression model (`lm` or `glm`), the 'segmented' variable whose a broken-line relationship is being tested, and the number of the evaluation points,

```
> davies.test(fit.glm,"age",k=5)

        Davies' test for a change in the slope

data:  Model =  binomial , link = logit
formula = cases/births ~ age
segmented variable = age
'Best' at = 32, n.points = 5, p-value < 2.2e-16
alternative hypothesis: two.sided
```

Currently `davies.test()` only uses the Wald statistic, i.e. $S(\psi_k) = \hat{\beta}_2/\mathrm{SE}(\hat{\beta}_2)$ for each fixed ψ_k, although alternative statistics could be used.

If the breakpoint exists, the limiting distribution of $\hat{\beta}_2$ is gaussian, therefore estimates (and standard errors) of the slopes can be easily computed via the function `slope()` whose argument `conf.level` specifies the confidence level (defaults to `conf.level=0.95`),

```
> slope(fit.seg)
$age
          Est.  St.Err. t value CI(95%).l CI(95%).u
slope1 -0.01341 0.01795 -0.7472  -0.04859   0.02177
slope2  0.26080 0.01476 17.6700   0.23190   0.28970
```

Davison and Hinkley (1997) discuss that it might be of some interest to test for a null left slope, and at this aim they use isotonic regression. On the other hand, the piecewise parameterization allows to face this question in a straightforward way since only a test for $H_0: \beta_1 = 0$ has be performed; for instance, a Wald test is available directly from the summary (see Figure 2, $t = -0.747$). Under a null-left-slope constraint, a segmented model may be fitted by omitting from the 'initial' model the segmented variable, namely

```
> fit.glm<-update(fit.glm,.~.-age)
> fit.seg1<-update(fit.seg)
```

While the fit is substantially unchanged, the (approximate) standard error of the breakpoint is noticeably reduced (compare the output in Figure 2)

```
> fit.seg1$psi
         Initial     Est.    St.Err
psi1.age      25 31.45333 0.5536572
```

Instead, as firstly observed in Hinkley (1971) and shown by some simulations, the breakpoint estimator coming from a null left slope model is more efficient as compared to the one coming from a nonnull left slope fit. Fitted values for both segmented models are displayed in Figure 1 where broken-lines and bars for the breakpoint estimates have been added via the relevant methods `plot()` and `lines()` detailed at the end of this section.

We continue our illustration of the **segmented** package by running a further example using the `plant` dataset in the package. This example may be instructive to describe how to fit multiple segmented relationships with also a zero constraint on the right slope. Data refer to variables, `y`, `time` and `group` which represent measurements of a plant organ over time for three attributes (levels of the factor `group`). The data have been kindly provided by Dr Zongjian Yang at School of Land, Crop and Food Sciences, The University of Queensland, Brisbane, Australia. Biological reasoning and empirical evidence as emphasized in Figure 3, indicate that non-parallel segmented relationships with multiple breakpoints may allow a well-grounded and reasonable fit. Multiple breakpoints are easily accounted in equation (1) by including additional terms $\beta_3(z_i - \psi_2)_+ + \ldots$ `segmented` allows a such extension in a straightforward manner by supplying multiple starting points in the `psi` argument.

To fit such a broken-line model within **segmented**, we first need to build the three different explanatory variables, products of the covariate `time` by the dummies of `group` [1],

```
> data("plant")
> attach(plant)
> X<-model.matrix(~0+group)*time
> time.KV<-X[,1]
> time.KW<-X[,2]
> time.WC<-X[,3]
```

Then we call `segmented` on a `lm` fit, by specifying multiple segmented variables in `seg.Z` and using a list to supply the starting values for the breakpoints in `psi`. We assume two breakpoints in each series,

```
> olm<-lm(y~0+group+ time.KV + time.KW + time.WC)
> os<-segmented(olm, seg.Z= ~ time.KV + time.KW
+   + time.WC, psi=list(time.KV=c(300,450),
+   time.KW=c(450,600), time.WC=c(300,450)))
Warning message:
max number of iterations attained
```

Some points are probably worth mentioning here. First, the starting linear model `olm` could be fitted via the more intuitive call `lm(y~group*time)`: even if `segmented()` would have worked providing the same results, a possible use of `slope()` would have not been allowed. Second, since there are multiple segmented variables, the starting values - obtained by visual inspection of the scatter-plots - have to supplied via a named list whose names have to match with the variables in `seg.Z`. Last but not least, the printed message suggests to re-fit the model because convergence is suspected. Therefore it could be helpful to trace out the algorithm and/or to increase the maximum number of the iterations,

```
> os<-update(os, control=seg.control(it.max=30,
+    display=TRUE))
0   1.433  (No breakpoint(s))
1   0.108
2   0.109
3   0.108
4   0.109
5   0.108
. . . .
29  0.108
30  0.109
Warning message:
max number of iterations attained
```

The optimized objective function (residual sum of squares in this case) alternates among two values and 'does not converge', in that differences never reach the (default) tolerance value of 0.0001; the function `draw.history()` may be used to visualize the values of breakpoints throughout the iterations. Moreover, increasing the number of maximum iterations, typically does not modify the result. This is not necessarily a problem. One could change the tolerance by setting `toll=0.001`, say, or better, stop the algorithm at the iteration with the best value. Also, one could stabilize the algorithm by shrinking the increments in breakpoint updates through a factor $h < 1$, say; this is attained via the argument `h` in the auxiliary function `seg.control()`,

```
> os<-update(os, control=seg.control(h=.3))
```

However, when convergence is not straightforward, the fitted model has to be inspected with particular care: if a breakpoint is understood to exist, the corresponding difference-in-slope estimate (and its t value) has to be large and furthermore the 'gap' coefficient (and its t value) has to be small (see the `summary(..)$gap`). If at the estimated breakpoint the coefficient of the gap variable is large (greater than two, say) a broken-line parameterization is somewhat questionable. Finally, a test for the existence of the breakpoint and/or comparing the BIC values would be very helpful in these circumstances.

Green diamonds in Figure 3 and output from `slope()` (not shown) show that the last slope for group "KW" may be set to zero. While a left slope is allowed by fitting only $(z - \psi)_+$ (i.e. by omitting the main variable z in the initial linear model as in the previous example), similarly a null right slope might be allowed by including only $(z - \psi)_-$. **segmented** does not handle such terms explicitly, however by noting that $(z - \psi)_- = -(-z + \psi)_+$, we can proceed as follows

[1] Of course, a corner-point parameterization (i.e. 'treatment' contrasts) is required to define the dummies relevant to the grouping variable; this is the default in R.

```
> neg.time.KW<- -time.KW
> olm1<-lm(y~0+group+time.KV+time.WC)
> os1<-segmented(olm1, seg.Z=~ time.KV + time.WC+
+   neg.time.KW, psi=list(time.KV=c(300,450),
+   neg.time.KW=c(-600,-450), time.WC=c(300,450)))
```

The 'minus' of the explanatory variable in group "KW" requires that the corresponding starting guess has to be supplied with reversed sign and, as consequence, the signs of estimates for the corresponding group will be reversed. The method segmented for confint() may be used to display (large sample) interval estimates for the breakpoints; confidence intervals are computed using $\hat{\psi} \mp z_{\alpha/2}\mathrm{SE}(\hat{\psi})$ where $\mathrm{SE}(\hat{\psi})$ comes from the Delta method for the ratio $\frac{\hat{\gamma}}{\hat{\beta}_2}$ and $z_{\alpha/2}$ is the quantile of the standard Normal. Optional arguments are parm to specify the segmented variable of interest (default to all variables) and rev.sgn to change the sign of output before printing (this is useful when the sign of the segmented variable has been changed to constrain the last slope as in example at hand).

```
> confint(os1,rev.sgn=c(FALSE,FALSE,TRUE))
$time.KV
              Est. CI(95%).l CI(95%).u
psi1.time.KV 299.9     256.9     342.8
psi2.time.KV 441.9     402.0     481.8

$time.WC
              Est. CI(95%).l CI(95%).u
psi1.time.WC 306.0     284.2     327.8
psi2.time.WC 460.1     385.5     534.7

$neg.time.KW
                  Est. CI(95%).l CI(95%).u
psi1.neg.time.KW 445.4     398.5     492.3
psi2.neg.time.KW 609.9     549.7     670.0
```

The slope estimates may be obtained using slope(); again, parm and rev.sgn may be specified when requested,

```
> slope(os1, parm="neg.time.KW", rev.sgn=TRUE)
$neg.time.KW
             Est.   St.Err.  t value CI(95%).l CI(95%).u
slope1 0.0022640 8.515e-05  26.580 0.0020970  0.002431
slope2 0.0008398 2.871e-04   2.925 0.0002771  0.001403
slope3 0.0000000        NA      NA        NA        NA
```

Notice that in light of the constrained right slope, standard errors, t-values, and confidence limits are not computed.

Figure 3 emphasizes the constrained fit which has been added to the current device via the relevant plot() method. More specifically, plot() allows to draw on the current or new device (depending on the logical value TRUE/FALSE of add) the fitted piecewise relationship for the variable term. To get sensible plots with fitted values to be superimposed to the observed points, the arguments const and rev.sgn have to be set carefully. The role of rev.sgn is intuitive and has been discussed above while const indicates a constant to be added to the fitted values before plotting,

```
> plot(os1, term="neg.time.KW", add=TRUE, col=3,
+    const=coef(os1)["groupRKW"], rev.sgn=TRUE)
```

const defaults to the model intercept, and for relationships by group the group-specific intercept is appropriate, as in the "KW" group example above. However when a 'minus' variable has been considered, simple algebra on the regression equation show that the correct constant for the other groups is given by the current estimate minus a linear combination of difference-in-slope parameters and relevant breakpoints. For the "KV" group we add the fitted lines after computing the 'adjusted' constant,

```
> const.KV<-coef(os1)["groupRKV"]-
+    coef(os1)["U1.neg.time.KW"]*
+    os1$psi["psi1.neg.time.KW","Est."]-
+    coef(os1)["U2.neg.time.KW"]*
+    os1$psi["psi2.neg.time.KW","Est."]
> plot(os1, "time.KV", add=TRUE, col=2, const=const.KV)
```

and similarly for group "WC".

Finally the estimated join points with relevant confidence intervals are added to the current device via the lines.segmented() method,

```
> lines(os1,term="neg.time.KW",col=3,rev.sgn=TRUE)
> lines(os1,term="time.KV",col=2,k=20)
> lines(os1,term="time.WC",col=4,k=10)
```

where term selects the segmented variable, rev.sgn says if the sign of the breakpoint values (point estimate and confidence limits) have to be reversed, k regulates the vertical position of the bars, and the remaining arguments refer to options of the drawn segments.

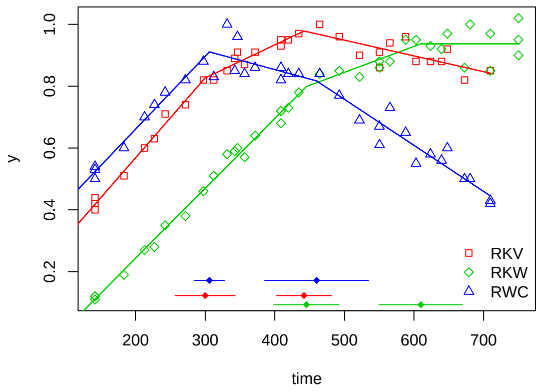

Figure 3: The plant dataset: data and constrained fit (model os1).

Conclusions

We illustrated the key-ideas of broken-line regression and how such a class of models may be fitted in R through the package **segmented**. Although alternative approaches could be undertaken to model nonlinear relationships, for instance via splines, the main appealing of segmented models lies on interpretability of the parameters. Sometimes a piecewise parameterization may provide a reasonable approximation to the shape of the underlying relationship, and threshold and slopes may be very informative and meaningful.

However it is well known that the likelihood in segmented models may not be concave, hence there is no guarantee the algorithm finds the global maximum; moreover it should be recognized that the method works by approximating the 'true' model (1) by (2), which could make the estimation problematic. A possible and useful strategy - quite common in the nonlinear optimization field - is to run the algorithm starting with different initial guesses for the breakpoint in order to assess possible differences. This is quite practicable due to computational efficiency of the algorithm. However, the more the clear-cut the relationship, the less important the starting values become.

The package is not concerned with estimation of the number of the breakpoints. Although the BIC has been suggested, in general nonstatistical issues related to the understanding of the mechanism of the phenomenon in study could help to discriminate among several competing models with a different number of joinpoints.

Currently, only methods for LM and GLM objects are implemented; however, due to the ease of the algorithm which only depends on the linear predictor, methods for other models (Cox regression, say) could be written straightforwardly following the skeleton of `segmented.lm` or `segmented.glm`.

Finally, for the sake of novices in breakpoint estimation, it is probably worth mentioning the difference existing with the other R package dealing with breakpoints. The **strucchange** package by Zeileis et al. (2002) substantially is concerned with regression models having a different set of parameters for each 'interval' of the segmented variable, typically the time; **strucchange** performs breakpoint estimation via a dynamic grid search algorithm and allows for testing for parameter instability. Such 'structural breaks models', mainly employed in economics and econometrics, are somewhat different from the broken-line models discussed in this paper, since they do not require the fitted lines to join at the estimated breakpoints.

Acknowledgements

This work was partially supported by grant 'Fondi di Ateneo (ex 60%) 2004 prot. ORPA044431: 'Verifica di ipotesi in modelli complessi e non-standard' ('Hypothesis testing in complex and nonstandard models'). The author would like to thank the anonymous referee for useful suggestions which improved the paper and the interface of the package itself.

Bibliography

M. Betts, G. Forbes, and A. Diamond. Thresholds in songbird occurrence in relation to landscape structure. *Conservation Biology*, 21:1046–1058, 2007.

R. B. Davies. Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika*, 74:33–43, 1987.

A. Davison and D. Hinkley. *Bootstrap methods and their application*. Cambridge University Press, 1997.

D. Hinkley. Inference in two-phase regression. *Journal of American Statistical Association*, pages 736–743, 1971.

V. Muggeo. Estimating regression models with unknown break-points. *Statistics in Medicine*, 22: 3055–3071, 2003.

R. Tiwari, K. A. Cronin, W. Davis, E. Feuer, B. Yu, and S. Chib. Bayesian model selection for join point regression with application to age-adjusted cancer rates. *Applied Statistics*, 54:919–939, 2005.

K. Ulm. A statistical methods for assessing a threshold in epidemiological studies. *Statistics in Medicine*, 10:341–349, 1991.

A. Zeileis, F. Leisch, K. Hornik, and C. Kleiber. `strucchange`: An R package for testing for structural change in linear regression models. *Journal of Statistical Software*, 7(2):1–38, 2002.

Vito M. R. Muggeo
Dipartimento Scienze Statistiche e Matematiche 'Vianelli'
Università di Palermo, Italy
`vmuggeo@dssm.unipa.it`