intcox: Compendium to apply the iterative convex minorant algorithm to interval censored event data

Volkmar Henschel, Christiane Heiß, Ulrich Mansmann
University of Heidelberg
Department of Medical Biometry and Informatics
INF 305, 69120 Heidelberg, Germany

November 2, 2004

Abstract

Software that fits a multivariate proportional hazards model to interval censored event data is urgently needed in medical research. Pan describes the use of the iterative convex minorant algorithm (ICM) to achieve this purpose. The implementation of the ICM is presented as well as a bootstrap procedure to derive information for statistical inference on the regression coefficients. An example is studied and the `intcox` results are compared to results of alternative procedures available in commercial software products.

1 Introduction

The occurrence of an event represents important information on the prognosis or treatment efficacy of a disease. It often plays the role of the primary endpoint in clinical studies. But, like the recurrence of a tumor, the time of its occurrence cannot be exactly observed. Events which are known to occur only within intervals represent interval censored event data.
Especially of interest is the influence of a covariate vector x on the probability of the occurrence of events which is formalized by the survival function $S(t|x) = 1 - F(t|x)$ with F the cumulative distribution function.
In case of right censored event data, the quantification is achieved by applying the extended proportional hazard model of Cox, cf. Therneau and Grambsch [6] which makes the following assumption on the survival function

$$\begin{aligned} S(t|\mathbf{x}) &= \exp\{-\Lambda(t|\mathbf{x})\} \\ &= \exp\left\{-\int_0^t \lambda_0(s)\exp\{\boldsymbol{\beta}'\mathbf{x}\}\mathrm{d}s\right\}. \end{aligned}$$

The expression $\Lambda(t|\mathbf{x})$ is called the cumulative hazard which is the integral of the hazard function $\lambda(s|\mathbf{x})$ up to time t

$$\lambda(s,\mathbf{x}) = \lambda_0(s)\exp\{\boldsymbol{\beta}'\mathbf{x}\}.$$

The model is called proportional hazards model because the regression coefficients act via a factor on the hazard function. The exponential value of a component of the coefficient vector is called relative hazard of the corresponding factor.
In Pan [4], a proportional hazard model is fit to interval censored data by means of the iterative convex minorant algorithm (ICM). This model allows to infer relative hazards from interval censored event data.
The omnipresence of multivariate interval censored data in medical research creates a strong need for appropriate software. In spite of the high practical relevance, the algorithm of Pan is not implemented in a statistical software environment and available for public use. In this note, we describe an implementation of the ICM-based algorithm for interval censored event data in the R software.
After a short description of the mathematical background, a clinical example of interval censored event data is presented and analyzed. The results are compared to alternative procedures available in commercial software products.

2 Mathematical Background

The proportional hazards assumption combines the covariate vector $\mathbf{x}$ and the vector of regression coefficients $\boldsymbol{\beta}$ via a linear predictor with the baseline hazard $\lambda_0(t)$:

$$\lambda(t, \mathbf{x}) = \lambda_0(t) \exp\{\boldsymbol{\beta}'\mathbf{x}\}.$$

A straightforward calculation gives the likelihood contribution of an observation which takes place in the interval $[s, t]$ and allows to find the log-likelihood of the data

$$L(F_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} \ln\Big\{ (1 - F_0(S_i-))^{\exp(\boldsymbol{\beta}'\mathbf{x}_i)} - (1 - F_0(T_i))^{\exp(\boldsymbol{\beta}'\mathbf{x}_i)} \Big\}.$$

This calculations uses the following relationship between survival and distribution function in the proportional hazards model

$$\begin{aligned}(1 - F(t|\mathbf{x})) &= S(t|\mathbf{x}) \\ &= S_0(t|\mathbf{x})^{\exp\{\boldsymbol{\beta}'\mathbf{x}\}} \\ &= (1 - F_0(t|\mathbf{x}))^{\exp\{\boldsymbol{\beta}'\mathbf{x}\}}.\end{aligned}$$

The objective of the ICM-algorithm is to maximize the log-likelihood by a modified Newton-Raphson algorithm. The gradients needed for the maximization are $\nabla_1 L(F_0, \boldsymbol{\beta}) = \frac{\partial L(F_0, \boldsymbol{\beta})}{\partial F_0}$ and $\nabla_2 L(F_0, \boldsymbol{\beta}) = \frac{\partial L(F_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$. The baseline distribution function F_0 is considered to be piecewise constant and thus can be represented by a finite dimensional vector which is parameterized by the finite steps of the cumulative baseline hazard function. The derivative with respect to F_0 is the gradient of the log-likelihood with respect to the vector of the baseline cumulative distribution function values. The derivative with respect to $\boldsymbol{\beta}$ is the usual derivative of the log-likelihood with respect to the components of beta. The full Hessian in the in the original Newton-Raphson algorithm is replaced by the diagonal matrices of the negative second derivatives $G_1(F_0, \boldsymbol{\beta})$ and $G_2(F_0, \boldsymbol{\beta})$.

The update from $F^{(m)}$ to $F^{(m+1)}$ is done iteratively with control of the stepsize. Starting point is always a stepsize of $\alpha = 1$. The new candidates for F_0 and $\boldsymbol{\beta}$ result from

$$\begin{aligned} F_0^{(m+1)} =& \text{Proj}\Big[F_0^{(m)} + \alpha G_1(m)^{-1}\nabla_1 L(m), G_1(m), \mathcal{R}\Big] \\ \boldsymbol{\beta}^{(m+1)} =& \boldsymbol{\beta}^{(m)} + \alpha G_2(m)^{-1}\nabla_2 L(m). \end{aligned}$$

A projection into the restricted range $\mathcal{R}$ weighted by G is used to assure that $F_0^{(m+1)}$ is again a distribution function:

$$\text{Proj}[y, G, \mathcal{R}] = \arg\min_x \Big\{\sum_{i=1}^{k}(y_i - x_i)^2 G_{ii} : 0 \le x_1 \le \cdots \le x_k \le 1\Big\}.$$

In case of $L(F^{(m+1)}) < L(F^{(m)})$, α is halved and the step is reiterated. A numerical procedure for the restricted projection is the pool adjacent violators algorithm (PAVA), which is described in Robertson et al. [5].
Starting values are calculated by treating the data as right censored and using the classical proportional hazards model. An event within a bounded interval $[s, t]$ will be interpreted as event observed at time t. An event within an interval unbounded to the right $[s, \infty]$ will be interpreted as a right censored event at time s. The Breslow-estimator is used to get a starting value for the baseline hazard $\Lambda_0(t)$.

3 Example

Meisel et al. [3] present data on the shrinkage of aneurisms associated with cerebral arteriovenous malformations (cAVM) after embolization treatment. The time to a shrinkage of the aneurism to below 50% of the baseline volume was of interest. Several patients had multiple aneurisms. Each patient was inspected at a random inspection time *obs.t*. The censoring variable z was set to one, if at the inspection time sufficient shrinkage was observed, else the censoring indicator was set to zero.
Two covariates were considered: the degree of cAMV occlusion by embolization (dichotomized at 50%, variable *mo*) and the location of the aneurism, whether at the midline arteries or at other afferent cerebral arteries, variable *lok*.
The single aneurisms are not independent because aneurisms within a patient may shrink in the same way (because the share the same "environment"). Multiple aneurisms were observed per patient. This clustering of aneurisms is indicated by the grouping variable *gr*.
The data is loaded and inspected for the first five patients.

```
> library(survival)
> library(intcox)
> data(AA.data)
> AA.data[1:11, ]
```

```
       obs.t z mo gr lok
1  1.7698630 0  0  1   1
2  0.9972603 0  1  2   1
```

```
3  0.9972603 0  1  2   1
4  0.9972603 0  1  2   1
5  1.0712329 0  0  3   0
6  1.0712329 0  0  3   1
7  5.6547945 0  0  4   1
8  1.5780822 0  0  5   1
9  1.5780822 1  0  5   0
10 1.5780822 1  0  5   0
11 1.5780822 1  0  5   1
```

The data is analyzed by applying the `intcox` algorithm. The algorithm requires the interval censored observation given in interval format with left and right boundaries. The `Surv` function does not allow the use of current status censored data as given in the aneurism example. In case of right censored data the right boundaries are set arbitrary to `NA`.

```
> AA.data$t.left <- ifelse(AA.data$z == 1, 0, AA.data$obs.t)
> AA.data$t.right <- ifelse(AA.data$z == 1, AA.data$obs.t, NA)
```

The fit with `intcox` gives an object of class "coxph" without the standard errors of the regression coefficients. The summary function for class "coxph" allows to summarize the output of the estimation procedure.

```
> AA.fit <- intcox(Surv(t.left, t.right, type = "interval2") ~
+     mo + lok, data = AA.data)
> summary(AA.fit)

Call:
intcox(formula = Surv(t.left, t.right, type = "interval2") ~
    mo + lok, data = AA.data)

  n= 149

      coef exp(coef) se(coef)  z  p
mo  -1.007     0.365       NA NA NA
lok -0.831     0.435       NA NA NA

    exp(coef) exp(-coef) lower .95 upper .95
mo      0.365       2.74        NA        NA
lok     0.435       2.30        NA        NA

Rsquare= NA   (max possible= 0.678 )
Likelihood ratio test= NA  on 2 df,   p=NA
Wald test            = NA  on 2 df,   p=NA
Score (logrank) test = NA  on 2 df,   p=NA
```

Additionally to the components of a class "coxph" object there are given:

lambda0: the estimated cumulative baseline hazard;

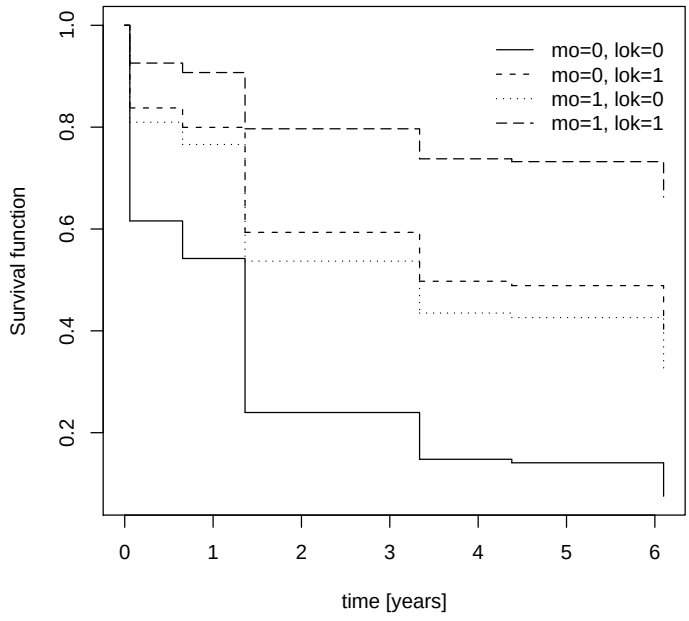
time.point: the corresponding time points at which the cumulative baseline hazard was estimated;

likeli.vec: a vector of the estimated loglik of each ICM step;

termination: an indicator for the reason of termination (see HTML Help for details), 1 indicates that the algorithm converged.

The estimated coefficients and cumulative baseline hazard can be used to estimated and plot group specific survival curves.

```
> surv.base <- exp(-AA.fit$lambda0)
> plot(AA.fit$time.point, surv.base, type = "s", xlab = "time [years]",
+     ylab = "Survival function", lty = 1)
> lines(AA.fit$time.point, surv.base^exp(AA.fit$coefficients["mo"]),
+     type = "s", lty = 2)
> lines(AA.fit$time.point, surv.base^exp(AA.fit$coefficients["lok"]),
+     type = "s", lty = 3)
> lines(AA.fit$time.point, surv.base^exp(sum(AA.fit$coefficients[c("mo",
+     "lok")])), type = "s", lty = 5)
> leg.names <- c("mo=0, lok=0", "mo=0, lok=1", "mo=1, lok=0", "mo=1, lok=1")
> legend(4, 1, leg.names, lty = c(1, 2, 3, 5), bty = "n")
```



It is of interest to calculate basic bootstrap confidence intervals [2] of the regression coefficients. A wild bootstrap procedure is used, c.f. Burr [1]. Because several patients present with multiple aneurisms, the bootstrap respected this clustering by sampling patients and not individual aneurisms. A patient enters with all of her/his aneurisms in the analysis. This procedure is in accordance with the analysis of marginal models as presented in chapter 8 of Therneau and Grambsch [6]. The bias between the ICM estimates and the median/mean of the bootstrap samples will also be assessed.

Remark. The number of replicates should be set to at least 999. The low number of nine is only chosen for a fast check by CRAN.

```
> set.seed(123)
> pat <- unique(AA.data$gr)
> intcox.boot.AA <- function(i, form) {
+     boot.sample <- sample(pat, length(pat), replace = T)
+     data.ind <- unlist(lapply(boot.sample, function(x, yy) which(yy ==
+         x), yy = AA.data$gr))
+     data.sample <- AA.data[data.ind, ]
+     boot.fit <- intcox(form, data = data.sample, no.warnings = TRUE)
+     return(list(coef = coef(boot.fit), term = boot.fit$termination))
+ }
> n.rep <- 9
> AA.boot <- lapply(1:n.rep, intcox.boot.AA, form = Surv(t.left,
+     t.right, type = "interval2") ~ mo + lok)
> AA.boot <- matrix(unlist(AA.boot), byrow = T, nrow = n.rep)
> colnames(AA.boot) <- c(names(coef(AA.fit)), "termination")
> inf.level <- 0.05
> mo.ord <- order(AA.boot[, "mo"])
> lok.ord <- order(AA.boot[, "lok"])
> pos.lower <- ceiling((n.rep + 1) * (inf.level/2))
> pos.upper <- ceiling((n.rep + 1) * (1 - inf.level/2))
> ci.mo <- AA.boot[mo.ord, "mo"][c(pos.lower, pos.upper)]
> ci.lok <- AA.boot[mo.ord, "lok"][c(pos.lower, pos.upper)]
> ci.mo

[1] -1.784836        NA

> ci.lok

[1] -1.063285        NA

> summary(AA.boot[, "mo"])

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.7850 -1.5960 -1.1650 -1.2330 -1.0590 -0.5932

> summary(AA.boot[, "lok"])

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.2000 -1.0630 -0.8941 -0.9276 -0.8401 -0.5790

> bias.mo <- c(mean.bias = coef(AA.fit)["mo"] - mean(AA.boot[,
+     "mo"]), median.bias = coef(AA.fit)["mo"] - median(AA.boot[,
+     "mo"]))
> bias.lok <- c(mean.bias = coef(AA.fit)["lok"] - mean(AA.boot[,
+     "lok"]), median.bias = coef(AA.fit)["lok"] - median(AA.boot[,
+     "lok"]))
> bias.mo

  mean.bias.mo median.bias.mo
     0.2260122      0.1577364
```

```
> bias.lok

  mean.bias.lok median.bias.lok
     0.09615781      0.06263106

> table(AA.boot[, "termination"])

1 2
8 1
```

The analysis shows a light bias between the the ICM estimates on the original data and the median/mean of the bootstrap samples which will not invalidate the statistical inference on the regression coefficient based on the basic bootstrap confidence interval which show the significant (level $\alpha = 0.05$) influence of both covariates on the shrinkage.
There are some caveats:

Single group setting: A single group model can not be calculated by the ICM algorithm. The calculation needs explicitly one or more covariates.

Stop after first step: There are situations where the ICM algorithm does not iterate. Then the starting values calculated from Cox model by interpreting the data as right censored give a likelihood value which can not be improved by the ICM algorithm. The occurrence of this situation will be announced by a warning.

Model selection: Look at the following model selection problem: Does the model `Surv(t.left,t.right,type="interval2")~ mo*lok` improve the fit to the data. One way to answer this question is to check the bootstrap confidence interval for the interaction regression coefficient. The calculation below renders a 95% bootstrap confidence interval of $[-1.22; 1.77]$ and a small bias which does not influences the statistical decision. One can conclude that there is no significant interaction between the covariates with respect to shrinkage.

Remark. The number of replicates should be set to at least 999. The low number of nine is only chosen for a fast check by CRAN.

```
> AA.int.fit <- intcox(Surv(t.left, t.right, type = "interval2") ~
+     mo * lok, data = AA.data)
> summary(AA.int.fit)

Call:
intcox(formula = Surv(t.left, t.right, type = "interval2") ~
    mo * lok, data = AA.data)

  n= 149

         coef exp(coef) se(coef)  z  p
mo     -1.144     0.318       NA NA NA
lok    -0.880     0.415       NA NA NA
mo:lok  0.246     1.279       NA NA NA
```

```
       exp(coef) exp(-coef) lower .95 upper .95
mo         0.318      3.140        NA        NA
lok        0.415      2.412        NA        NA
mo:lok     1.279      0.782        NA        NA

Rsquare= NA   (max possible= 0.679 )
Likelihood ratio test= NA  on 3 df,   p=NA
Wald test            = NA  on 3 df,   p=NA
Score (logrank) test = NA  on 3 df,   p=NA
```

```
> set.seed(234)
> n.rep <- 9
> AA.int.boot <- lapply(1:n.rep, intcox.boot.AA, form = Surv(t.left,
+     t.right, type = "interval2") ~ mo * lok)
> AA.int.boot <- matrix(unlist(AA.int.boot), byrow = T, nrow = n.rep)
> colnames(AA.int.boot) <- c(names(coef(AA.int.fit)), "termination")
> inf.level <- 0.05
> int.ord <- order(AA.int.boot[, "mo:lok"])
> pos.lower <- ceiling((n.rep + 1) * (inf.level/2))
> pos.upper <- ceiling((n.rep + 1) * (1 - inf.level/2))
> ci.int <- AA.int.boot[int.ord, "mo:lok"][c(pos.lower, pos.upper)]
> ci.int

[1] -0.8864028         NA

> summary(AA.int.boot[, "mo:lok"])

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.8864 -0.3491 -0.1379 -0.1192  0.3173  0.7166

> bias.int <- c(mean.bias = coef(AA.int.fit)["mo:lok"] - mean(AA.int.boot[,
+     "mo:lok"]), median.bias = coef(AA.int.fit)["mo:lok"] - median(AA.int.boot[,
+     "mo:lok"]))
> bias.int

  mean.bias.mo:lok median.bias.mo:lok
         0.3648784          0.3836323

> table(AA.int.boot[, "termination"])

1 2
8 1
```

Likelihood ratio test: There are caveats when using the calculated likelihood to perform a likelihood ratio test for model selection. First, we are not aware of an asymptotic theory for this test, second, the likelihood of the larger model may be smaller as the likelihood of the sparser model as can be seen in our example.

```
> AA.int.fit$loglik

[1] -84.54277
```

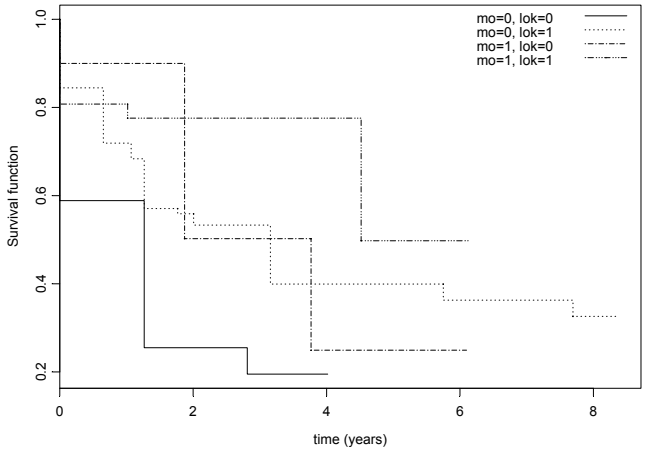


Figure 1: Turnbull's generalization of the Kaplan-Meier estimator

```
> AA.fit$loglik

[1] -84.51275
```

A non-parametric approach to the data uses Turnbull's [7] generalization of the Kaplan-Meier estimator which is implemented in S-Plus (Insightful Corporation). The following S-Plus code will not work in R.

```
> surv.formula<-censor(left,right,cens*3,type="interval")~mo+lok
> plot(kaplanMeier(surv.formula,data=aneur),lty=c(1,2,3,5),
+ xlab="time (years)",ylab="Survival function")
```

Figure 1 shows the results when applied to the four subgroups of patients. The fitting of an accelerated failure time model for interval censored data could be carried out by means of the SAS-procedure LIFEREG (SAS-Institute Inc. Cary, North Carolina, USA). Because of technical reasons, the value 0 for the lower end of an interval had to be replaced by the equivalent of 1 day: 1/365. The Weibull distribution was chosen for the error term. Then the estimates of the coefficients can be interpreted as log-transformed relative hazards: mo -1.64 [-2.88, -0.57], lok -1.22 [-2.21, -0.36]. The baseline hazard is determined by intercept 0.47 [-0.36, 1.37] and scale 1.67 [1.25, 2.01].
To reproduce the calculation, the dataset AA.data (columns separated by a semicolon) is written as comma seperated value file "AA.csv" to the working directory (shown by the getwd call) and is available for the given SAS procedure.

```
> getwd()

[1] "C:/Rdevel/intcox.Rcheck/intcox/doc"

> write.table(AA.data, file = "AA.csv", sep = ",", na = ".", col.names = NA)
```

```
PROC IMPORT OUT= WORK.aneur
            DATAFILE= "Pfad  \AA.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
data aneur;
    set aneur;
    if t_left=0 then t_left=1/365.25;
    gruppe=2*mo+lok;
run;
proc lifereg data=aneur;
    model (t_left,t_right)=mo lok / d=weibull;
    output out=outcdf cdf=cdf;
run;
data outcdf;
    set outcdf;
    svf=1-cdf;
run;
proc sort data=outcdf;
    by gruppe svf;
run;
symbol1 color=black i=spline line=1;
symbol2 color=black i=spline line=33;
symbol3 color=black i=spline line=41;
symbol4 color=black i=spline line=43;
AXIS1 label=none MINOR=(number=1)
LABEL=(justify=center angle=90 Rotate=0 'Survival function');
AXIS2 Label=none MINOR=(number=1) order=(0 to 9 by 1)
LABEL=(JUSTIFY=CENTER 'time (years)');
LEGEND across=1 label=none position=(top right inside)
value=('mo=0, lok=0' 'mo=0, lok=1' 'mo=1, lok=0' 'mo=1, lok=1');
proc gplot data=outcdf;
    plot svf*t_left=gruppe / vaxis=axis1 haxis=axis2 legend=legend;
run; quit;
goptions reset=all;
```

Figure 2 represents the result of the Weibull approach.

4 Discussion

The ICM-based algorithm of Pan follows the maximum likelihood rationale of estimation. The computational effort is low, because only the diagonal elements of the Hessian matrix are used, as well as the PAVA for the restricted regression. The ICM-algorithm is implemented in a R-package which includes a dynamically loaded C-routine for the PAVA. It has to be used with care.
Simulation studies showed a light positive bias in the estimated regression coefficients. From a practical point of view, this bias seems to be acceptable but has to be taken into consideration when the results of a study will be interpreted. The ICM gives a rather rough estimate of the baseline hazard, because

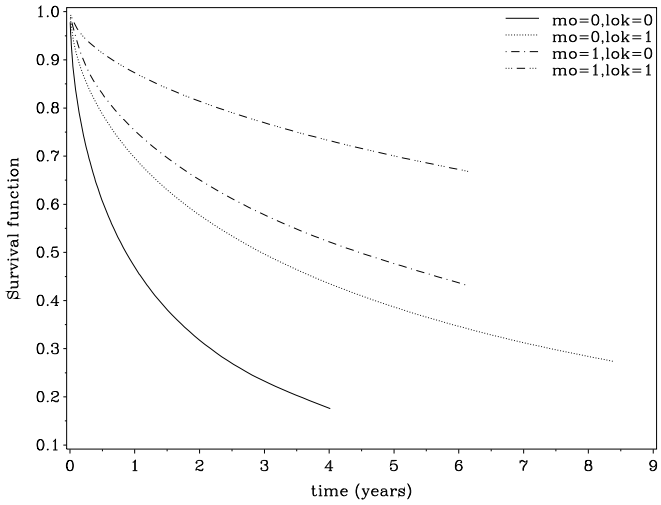


Figure 2: Accelerated failure time model

the PAVA smoothes this function to large intervals of constant hazard. As an example the obtained bias of the coefficients of a bootstrap of 999 samples from the `intcox.example`, see HTML Help, is shown in figure 3.

The algorithm is able to handle the combination of real right censored data and real interval censored data. It is not wise to transfer right censored data artificially to interval censored data by substituting a large number as right endpoint of a right censored event. This introduces bias into the relative hazards estimates.

We observed problems in the algorithm with respect to maximizing the likelihood in case of a high percentage of right censored data ($> 30\%$). In this case, the algorithm does not move away from the starting values calculated from the Cox model as described in the first paragraph of section 2.

This work was supported by DFG grant MA 1723/2-1.

References

[1] Deborah Burr. A comparison of certain bootstrap confidence intervals in the cox modell. *Journal of the American Statistical Association*, 89:1290–1302, 1994.

[2] A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application.* Cambridge University Press, Cambridge, 1997.

[3] H. J. Meisel, U. Mansmann, H. Alvarez, G. Rodesch, M. Brock, and P. Lasjaunias. Cerebral arteriovenous malformations and associated aneurysms: Analysis of 305 cases from a series of 662 patients. *Neurosurgery*, 46:793–802, 2000.

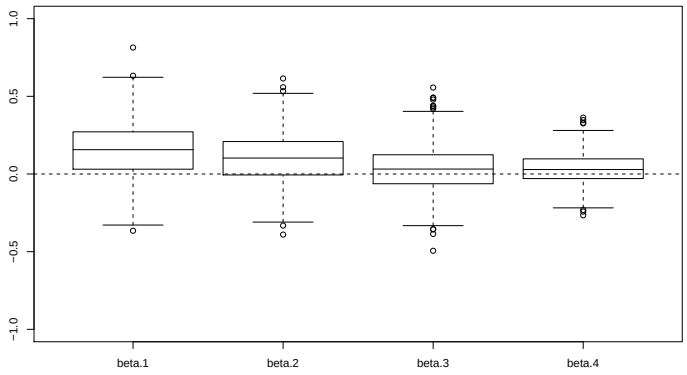


Figure 3: Bias in the coefficients of intcox.example

[4] Wei Pan. Extending the iterative convex minorant algorithm to the cox model for interval-censored data. *Journal of Computational and Graphical Statistics*, 78:109–120, 1999.

[5] Tim Robertson, F.T. Wright, and Richard L. Dykstra. *Order Restricted Statistical Inference*. Wiley, New York, 1988.

[6] Terry M. Therneau and Patricia M. Grambsch. *Modeling Survival Data: extending the Cox model*. Springer, New York, 2000.

[7] B. Turnbull. Nonparametric estimation of a survivorship function with doubly censored data. *Journal of the American Statistical Association*, 69:169–173, 1974.