



CPHshape: estimating a shape constrained baseline hazard in the Cox proportional hazards model

Rihong Hui
York University

Hanna Jankowski
York University

Abstract

We introduce the R package **CPHshape**, which computes the effect parameters and the nonparametric maximum likelihood estimator of a shape constrained baseline hazard in the Cox proportional hazards model. The functionality of the package is illustrated using reproducible examples which are based on simulated data.

Keywords: proportional hazards, shape constraints, maximum likelihood, PAVA, Cox model, nonparametric statistics.

1. Introduction

The proportional hazards model was introduced in [Cox \(1972\)](#) and has since become a staple in the analysis of survival data. Under the Cox model, the hazard function takes the form

$$h(x|z) = \exp(\beta^T z)h_0(x), \quad (1)$$

where z denotes the covariates, β are the associated effect parameters, and $h_0(x)$ is the baseline hazard function. The approach developed in [Cox \(1972\)](#) was to estimate the parameters β using partial likelihood, thus allowing h_0 to be arbitrary. However, as noted in [Cox \(1972, page 190\)](#): “Alternatively we may restrict $h_0(x)$ qualitatively, for example by assuming it to be monotonic or to be a step function (a suggestion of Professor J.W. Tukey).”

The nonparametric estimator of an increasing hazard was first developed in [Grenander \(1956\)](#). The increasing, decreasing, and u-shaped cases with censoring were explored further in [Mykityn and Santner \(1981\)](#), where the proportional hazards model is also briefly mentioned. There, they suggest a method of computation based on the pool adjacent violators algorithm (PAVA). An alternative isotonic estimator was also considered in [Chung and Chang \(1994\)](#). Most recently, the asymptotics of the increasing/decreasing hazard in the Cox model were

studied in [Lopuhaä and Nane \(2011\)](#), where it is shown that the maximum likelihood estimator (MLE) is consistent and converges pointwise at rate $n^{1/3}$. Precise limits are also derived there.

An increasing hazard is a popular assumption to make in lifetime modelling, and the u-shaped hazard also allows for a “burn-in” period in the system. As noted in [Lopuhaä and Nane \(2011, page 2\)](#), “the survival time after a successful medical treatment is usually modelled using a decreasing hazard function”. One example of this is given in [Cook, Walter, Cook, Griffith, Guyatt, Leasa, Jaeschke, and Brun-Buisson \(1998\)](#), where the incidence rate and risk factors of ventilator associated pneumonia are studied. There, the authors point out a decreasing baseline hazard for acquiring pneumonia, suggesting that “long-term survivors are patients at lower intrinsic risk” ([Cook et al. 1998, page 437](#)). However, a closer examination of the hazard observed in [Cook et al. \(1998, Figure 2\)](#) reveals it to be unimodal, with a peak incidence rate at about five days. Thus, the unimodal shape constraint would also be of interest in practice, as it allows for the modelling of an initial incubation period.

The R package **CPHshape** implements maximum likelihood estimation of the effect parameters β and $h_0(\cdot)$ with the following options on the shape constraint on h_0 : increasing, decreasing, unimodal, and u-shaped. We assume that the observed times come from a continuous model, and allow for right-censoring. Our consideration of this problem appears to be independent of the work of [Lopuhaä and Nane \(2011\)](#). The algorithm, now available through the package **CPHshape**, was first presented in June 2011 ([Hui and Jankowski 2011](#)).

1.1. About this document

This document is an introduction to the R package **CPHshape** which is available from the Comprehensive R Archive Network at <http://CRAN.R-project.org/package=CPHshape>. In [Section 2](#) we define the model, and [Section 3](#) gives details on the algorithm used in the estimation. [Section 4](#) is dedicated to examples which explain the full functionality of the package.

This document was created using Sweave ([Leisch 2002](#)) and LATEX ([Lamport 1986](#)) using R ([R Development Core Team 2011](#)). This means that all of the code has been checked by R, and can be reproduced exactly.

2. The likelihood

We assume that the data consist of IID samples of the triple (T_i, Δ_i, Z_i) for $i = 1, \dots, n$. If X_i denotes the actual lifetime variable and C_i the censoring time, then the observed data is $T_i = \min(X_i, C_i)$ and $\Delta_i = 1_{X_i \leq C_i}$. Thus, if $\Delta_i = 1$ then we observe the actual lifetime and if $\Delta_i = 0$ we observe only the censored time. Also, Z_i denotes the covariate vector $Z_i \in \mathbb{R}^p$. We are interested only in the distribution of X , and we assume that the hazard function of X can be modelled as in [\(1\)](#). We also assume that the survival random variable X , conditionally on $Z = z$, is continuous with density $f(x|z)$. The survival time X and the censoring time C are assumed to be conditionally independent given $Z = z$.

The assumption that the distribution of X is continuous is an important one for the package **CPHshape**, since ties for $\Delta_i = 1$ are not handled by the algorithm. The package does allow for ties in the censored observations though, i.e. when $\Delta_i = 0$. We note that the consistency and asymptotic results developed in [Lopuhaä and Nane \(2011\)](#) require additional assumptions.

Let (t_i, δ_i, z_i) , $i = 1, \dots, n$ denote the observations from this model. The full likelihood for the effect parameters β and the baseline hazard h_0 is

$$\begin{aligned} \mathcal{L}(\beta, h_0) &= \prod_{i=1}^n h(t_i|z_i)^{\delta_i} (1 - F(t_i|z_i)) \\ &= \left\{ \prod_{i=1}^n e^{\delta_i \beta^T z_i} \right\} \times \exp \left\{ - \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) \right\} \times \left\{ \prod_{i=1}^n h_0(t_i)^{\delta_i} \right\}, \end{aligned}$$

where $H_0(t) = \int_0^t h_0(s) ds$. Maximising the likelihood is the same as minimising the following criterion function

$$\varphi(\beta, h_0) = \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) - \sum_{i=1}^n \delta_i \log h_0(t_i) - \sum_{i=1}^n \delta_i \beta^T z_i, \quad (2)$$

which is convex in all parameters. Without loss of generality, in what follows we assume that the observations t_1, \dots, t_n are ordered. That is, $t_1 < t_2 < \dots < t_n$.

It is important to note that the likelihood provides information on the hazard function only on the interval $[0, t_n]$ when $\delta_n = 1$ and $[0, t_n)$ when $\delta_n = 0$. Values of $h_0(t)$ for $t > t_n$ (or $t \geq t_n$ if $\delta_n = 0$) are unknown, unless additional information on the shape of the hazard function is available.

Suppose next that $\delta_n = 1$ (that is, that the largest observation was uncensored), and also suppose that we are interested in the case where h_0 is an increasing function of t . Then, we can see that the term $h_0(t_n)$ in the likelihood function $\mathcal{L}(\beta, h_0)$ can become arbitrarily large. A similar situation arises if we wish to estimate the unimodal baseline hazard, where the term $h_0(t_m)$ is now unbounded (here t_m denotes the location of the mode, and is always an observation). To handle such situations, we simply remove the problematic term from the likelihood, and instead consider minimising, for example,

$$\varphi(\beta, h_0) = \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) - \sum_{i=1}^{n-1} \delta_i \log h_0(t_i) - \sum_{i=1}^{n-1} \delta_i \beta^T z_i,$$

for the increasing hazard setting. This is equivalent to the approach taken in [Grenander \(1956\)](#), where first an upper bound of M is assumed on h_0 , and then M is allowed to grow arbitrarily large. Thus, minimising the (modified) function $\varphi(\beta, h_0)$ given above yields an estimate of h_0 , and we set $h_0(t) = \infty$ for $t = t_n$, and therefore also for $t > t_n$.

3. Algorithm

Our overall approach is similar to that proposed in [Mykytyn and Santner \(1981\)](#). However, instead of PAVA, we use a graphical representation of the estimators, similar to that of [Grenander \(1956\)](#). Define

$$\begin{aligned} \varphi_1(\beta, h_0) &= \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) - \sum_{i=1}^n \delta_i \log h_0(t_i), \\ \varphi_2(\beta, h_0) &= \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) - \sum_{i=1}^n \delta_i \beta^T z_i. \end{aligned}$$

To minimise (2) we perform the following algorithm, where \mathcal{C} denotes the class of functions to which h_0 belongs. That is, \mathcal{C} denotes the class of positive functions on \mathbb{R}_+ which are either increasing, decreasing, u-shaped, or unimodal.

```

SET initial value for  $\hat{\beta} = \beta_{\text{ini}}$ 

find  $\hat{h}_0 = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi(\hat{\beta}, h_0) = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi_1(\hat{\beta}, h_0)$ 
set  $\varphi_{\text{new}} = \varphi(\hat{\beta}, \hat{h}_0)$ 

WHILE  $\delta < \varepsilon$  REPEAT
    set  $\varphi_{\text{old}} = \varphi_{\text{new}}$ 
    find  $\hat{\beta} = \operatorname{argmin}_{\beta \in \mathbb{R}^k} \varphi(\beta, \hat{h}_0) = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi_2(\hat{\beta}, h_0)$ 
    find  $\hat{h}_0 = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi(\hat{\beta}, h_0) = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi_1(\hat{\beta}, h_0)$ 
    set  $\varphi_{\text{new}} = \varphi(\hat{\beta}, \hat{h}_0)$ 
    set  $\delta = |\varphi_{\text{new}} - \varphi_{\text{old}}|$ 
END WHILE LOOP

```

The two inner minimisations are described below in detail. To minimize φ_2 we use the Newton-Raphson algorithm and the minimum of φ_1 is found exactly via a graphical representation.

Stopping criteria

Both the outer loop and the inner (Newton-Raphson) loop require a stopping criterion. Let ε denote the value used in the outer loop and let ε_{NR} denote the value used for stopping in the Newton-Raphson step. These are coded in the function `find.shapeCPH` as `eps` and `eps.beta`, respectively. Their default values are set to `eps=eps.beta=1e-5`. If, within the function `find.shapeCPH`, the setting `print=TRUE` is used, then successive iterations with values of δ (as defined above), as well as additional information, is printed to the screen. An example of this is given in Section 4.

3.1. Newton-Raphson algorithm

To minimise $\varphi_2(\beta, h_0) = \sum_{i=1}^n e^{\beta^T z_i} H_0(t_i) - \sum_{i=1}^n \delta_i \beta^T z_i$, we need to find the vector $\hat{\beta}$ such that

$$\partial_j \left\{ \sum_{i=1}^n e^{\hat{\beta}^T z_i} H_0(t_i) - \sum_{i=1}^n \delta_i \hat{\beta}^T z_i \right\} = \sum_{i=1}^n z_{ij} \left(e^{\hat{\beta}^T z_i} H_0(t_i) - \delta_i \right) = 0.$$

To do this, we employ the multi-dimensional Newton-Raphson algorithm. Let ∇_k denote the vector where $\nabla_{ki} = \partial_j \varphi_2(\beta_k, h_0)$, and let J_k denote the Jacobian $k \times k$ matrix where

$$(J_k)_{j,l} = \partial_l \partial_j \varphi_2(\beta_k, h_0)$$

$$= \sum_{i=1}^n z_{ij} z_{il} e^{\hat{\beta}^T z_i}.$$

Then, each iteration of the Newton-Raphson algorithm has $\beta_{k+1} = \beta_k - J_k^{-1} \nabla_k$. The algorithm is terminated when $\|\partial\varphi_2(\beta, h_0)\|^2 < \varepsilon_{NR}$. Here, $\partial\varphi_2$ denotes the gradient of φ_2 in the direction β , while h_0 is held fixed.

3.2. Weighted MLE

We next need to find $\hat{h}_0 = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi_1(\beta, h_0)$ for fixed β . Defining $w_i = e^{\beta^T z_i} > 0$, this is equivalent to minimising the criterion function

$$\varphi_1(\beta, h_0) = \sum_{i=1}^n w_i H_0(t_i) - \sum_{i=1}^n \delta_i \log h_0(t_i).$$

In what follows, we let i_1, \dots, i_m enumerate the indices $i \leq n$ such that $\delta_i = 1$.

Decreasing baseline hazard

To visualise the estimator, suppose first that $y_{i_k} = h_0(t_{i_k})$ is known for all i_k . Then to minimize φ_1 , we need to minimise the sum $\sum_{i=1}^n w_i H_0(t_i)$ by making the function $h_0(t)$ as small as possible for $t \neq y_{i_k}$, while satisfying the shape constraints. It is not difficult to see that the function h_0 must therefore have the form

$$h_0(t) = \begin{cases} y_1 & t \in [0, t_{i_1}] \\ y_2 & t \in (t_{i_1}, t_{i_2}] \\ y_3 & t \in (t_{i_2}, t_{i_3}] \\ \vdots & \\ y_m & t \in (t_{i_{m-1}}, t_{i_m}]. \end{cases} \quad (3)$$

If $\delta_n = 0$, then the function $h(t) = 0$ for $t > t_{i_m}$. Otherwise, $h(t)$ is unknown for $t > t_{i_m} = t_n$. We have thus argued that the MLE of a decreasing hazard is left-continuous and piecewise constant.

Next, note that any positive decreasing hazard can be written as a (positive) mixture of the basis functions $e_\tau(t) = \mathbb{I}_{[0, \tau]}(t)$. That is, $h_0(t) = \int e_\tau(t) d\mu_0(\tau)$, for some positive measure μ_0 . It is therefore sufficient to examine the directional derivative of φ_1 in the directions given by the basis functions, as $\tau > 0$ varies. Furthermore, (3) tells us that $\hat{h}_0 = \operatorname{argmin}_{h_0 \in \mathcal{C}} \varphi_1(\beta, h_0)$ will have the form $\hat{h}_0(t) = \int e_\tau(t) d\hat{\mu}_0(\tau)$, where $\hat{\mu}_0$ gives positive mass only to a finite number of τ , and each of these must be one of the uncensored observation points. For reasons which will become obvious shortly, we call τ such that $\hat{\mu}_0(\{\tau\}) > 0$ touch points.

The directional derivative is then

$$\begin{aligned} \nabla\varphi_1(\hat{h}_0)[e_\tau] &= \lim_{\varepsilon \rightarrow 0} \frac{\varphi_1(\hat{h}_0 + \varepsilon e_\tau) - \varphi_1(\hat{h}_0)}{\varepsilon} \\ &= \sum_{i=1}^n w_i \int_0^{t_i} \mathbb{I}_{[0, \tau]}(t) dt - \sum_{i=1}^n \delta_i \frac{\mathbb{I}_{[0, \tau]}(t_i)}{\hat{h}_0(t_i)} \\ &= \sum_{i=1}^n w_i \min(t_i, \tau) - \sum_{i: t_i \leq \tau} \frac{\delta_i}{\hat{h}_0(t_i)}. \end{aligned}$$

If \widehat{h}_0 minimises φ_1 , then this must always be greater than zero, with equality if τ is a touch point. Let τ_{k-1} and τ_k be two successive touch points. Recall that only uncensored observations may be touch points and the hazard is constant between two touch points. For $\tau_{k-1} < \tau < \tau_k$, we therefore have

$$\begin{aligned} \frac{|\{i_j : t_{i_j} \in (\tau_{k-1}, \tau]\}|}{\widehat{y}_k} &< \sum_{i=1}^n w_i \min(t_i, \tau) - \sum_{i=1}^n w_i \min(t_i, \tau_{k-1}) \\ &= \sum_{i:t_i > \tau_{k-1}} w_i (\min(\tau, t_i) - \tau_{k-1}) \\ &= \sum_{i:t_i > \tau_{k-1}} \left(\sum_{j=i+1}^n w_j \right) (\min(\tau, t_{i+1}) - t_i) \end{aligned}$$

Rearranging, this yields,

$$\frac{1}{\widehat{y}_k} < \frac{\sum_{i:t_i > \tau_{k-1}} \left(\sum_{j=i+1}^n w_j \right) (\min(\tau, t_{i+1}) - t_i)}{|\{i_j : t_{i_j} \in [\tau_k, \tau]\}|},$$

for $\tau \in (\tau_k, \tau_{k+1})$ with equality for $\tau = \tau_{k+1}$. It follows that the values of $\widehat{s}_k = 1/\widehat{y}_k$ can be found as the slopes of the greatest convex minorant of the sequence of points

$$\{(0, 0), (1, s_{i_1}), \dots, (m, s_{i_m})\},$$

where

$$s_{i_j} = \sum_{j=0}^{i_j-1} \left(\sum_{i=j+1}^n w_j \right) (t_{j+1} - t_j),$$

with the convention that $t_0 = 0$. The function `chull` available from R computes the convex hull of a specified set of points, and is easily modified to find the greatest convex minorant of a function. The algorithm used in `chull` is based on that developed in [Eddy \(1977\)](#). Note that, unlike for the Grenander estimator of a decreasing density, the MLE is *not* the derivative of the greatest convex minorant (or least concave majorant in the increasing case), but rather the values of $1/\widehat{y}_i$ are found via the greatest convex minorant. The MLE is then given via the formula (3), with the locations t_{i_k} corresponding to the indices i_k which are touch points in the greatest convex minorant above. An alternative derivation of this fact is given in [Lopuhaä and Nane \(2011\)](#) and is based on the work of [Grenander \(1956\)](#).

Other shape constraints

The remaining shape constraints (increasing, unimodal, u-shaped) are handled in a similar fashion, and we omit the details. We note that the unimodal and u-shaped cases do involve an additional level of difficulty, in that the mode or antimode (location of the minimum for the u-shaped constraint) must be specified in advance to use the approach discussed above. To find, for example, the unimodal estimator we then search over all possible locations of the mode and select the one with the smallest value of φ_1 . We handle the u-shaped setting in a similar fashion. This is the approach taken in [Mykytyn and Santner \(1981\)](#), but without the

weights w_i . Therefore, the algorithm in the u-shaped or unimodal case is considerably slower, particularly so for larger sample sizes. A related issue comes up in the estimation of a convex hazard, and we refer to [Jankowski and Wellner \(2009\)](#) for more details.

Functional form of the MLE

Each shape constraint results in a particular form of the estimator. Here, we summarise the different forms obtained for the four different shape constraints available in **CPHshape**. We note that these forms are valid under both the proportional hazards model and also the nonparametric MLE of a shape constrained hazard without covariates.

Decreasing:

$$\widehat{h}_0(t) = \begin{cases} y_1 & t \in [0, t_{i_1}] \\ y_2 & t \in (t_{i_1}, t_{i_2}] \\ y_3 & t \in (t_{i_2}, t_{i_3}] \\ \vdots & \\ y_m & t \in (t_{i_{m-1}}, t_{i_m}]. \end{cases}$$

If $\delta_n = 0$, then the function $\widehat{h}_0(t) = 0$ for $t > t_{i_m}$. Otherwise, $\widehat{h}_0(t)$ is unknown in this region.

Unimodal:

The mode, m_0 , must lie at one of the uncensored observation points. Let $t_{i_{k_0}} = m_0$ denote this observation point.

$$\widehat{h}_0(t) = \begin{cases} 0 & t \in [0, t_{i_1}) \\ y_1 & t \in [t_{i_1}, t_{i_2}] \\ y_2 & t \in [t_{i_2}, t_{i_3}) \\ \vdots & \\ y_{k_0-1} & t \in [t_{i_{k_0-1}}, m_0) \\ \infty & t = m_0 \\ y_{k_0+1} & t \in (m_0, t_{i_{k_0+1}}] \\ y_{k_0+2} & t \in (t_{i_{k_0+1}}, t_{i_{k_0+2}}] \\ \vdots & \\ y_{m-1} & t \in (t_{i_{m-2}}, t_{i_{m-1}}] \\ y_m & t \in (t_{i_{m-1}}, t_{i_m}]. \end{cases}$$

If $\delta_n = 0$, then the function $\widehat{h}_0(t) = 0$ for $t > t_{i_m}$. Otherwise, $\widehat{h}_0(t)$ is unknown in this region.

Increasing:

$$\widehat{h}_0(t) = \begin{cases} 0 & t \in [0, t_{i_1}) \\ y_1 & t \in [t_{i_1}, t_{i_2}] \\ y_2 & t \in [t_{i_2}, t_{i_3}) \\ \vdots & \\ y_{m-1} & t \in [t_{i_{m-1}}, t_{i_m}]. \end{cases}$$

If $\delta_n = 1$ then the function $\widehat{h}_0(t) = \infty$ for $t \geq t_n$. Otherwise, $\widehat{h}_0(t)$ is unknown for $t \geq t_{i_m}$.

U-shaped:

The antimode, a_0 , must lie in between two observation points. Let $t_{i_{k_0}} < a_0 < t_{i_{k_0+1}}$ denote the two adjacent observation points with $\delta_i = 1$.

$$\widehat{h}_0(t) = \begin{cases} y_1 & t \in [0, t_{i_1}] \\ y_2 & t \in (t_{i_1}, t_{i_2}] \\ \vdots & \\ y_{k_0} & t \in (t_{i_{k_0-1}}, t_{i_{k_0}}] \\ 0 & t \in (t_{i_{a_0}}, t_{i_{k_0+1}}) \\ y_{k_0+1} & t \in [t_{i_{k_0+1}}, t_{i_{k_0+2}}) \\ y_{k_0+2} & t \in [t_{i_{k_0+2}}, t_{i_{k_0+3}}) \\ \vdots & \\ y_{m-1} & t \in [t_{i_{m-1}}, t_{i_m}]. \end{cases}$$

If $\delta_n = 1$ then the function $\widehat{h}_0(t) = \infty$ for $t \geq t_n$. Otherwise, $\widehat{h}_0(t)$ is unknown for $t \geq t_{i_m}$.

Functions for plotting the MLE are provided as part of the package **CPHshape**. The fitted hazard functions are always piecewise continuous, and points of (either left or right) continuity are marked as filled dots, while points of discontinuity are not marked. The location of the mode for the unimodal hazard is also indicated with a dashed vertical line. Some examples are given in the following section.

4. Examples

The main function in **CPHshape** is `find.shapeCPH`. This function finds the maximum likelihood estimator of β and h_0 in the model described in Section 2. We begin, however, with the function `find.shapeMLE`, which finds the maximum likelihood estimator of a shape constrained hazard function without the proportional assumption on the covariate input (equivalently, assume that $\beta = 0$). This function illustrates nicely the graphical representation algorithm described in Section 3.

4.1. MLE of a shape constrained hazard (no covariates)

Suppose first that we observe X_1, \dots, X_n exponential random variables with mean 1. The true hazard function in this case is $h(x) = 1$, which falls into the decreasing category (in fact, it is also increasing, unimodal, and u-shaped). However, suppose that we wish to find the nonparametric MLE of a decreasing hazard based on the observed data. To do this, we use the function `find.shapeMLE`.

```
> library(CPHshape)
> set.seed(12345)
> x <- rexp(10)
> mle <- find.shapeMLE(x, type="decreasing")
```

The result of `find.shapeMLE` is an object of type "CPHshape". The MLE itself is specified completely by

```
> mle$h.range
[1] 0.00000000 0.02393814 1.81800304 6.40218924

> mle$h.val
[1] 8.5509578 0.8863219 0.2181412
```

To read this properly we need to understand the functional form of the MLE given in (3). Thus, the MLE is equal to 8.55 on $[0, 0.024]$, 0.886 on $(0.024, 1.82]$, and 0.022 on $(1.82, 6.40]$. The type of model fit, and the location of the mode/antimode (if necessary) is reported in

```
> mle$mode
[1] NA
```



```
> mle$type
```

```
[1] "decreasing"
```

We note that for the u-shaped constraint, the antimode is not unique and the MLE is equal to zero over a range of functions. In this case the algorithm returns the midpoint of this range as the antimode. However, it is not necessary for the user to read the output of the MLE directly. The functions `find.hazard` and `find.cumulative` can evaluate the value of the fitted hazard and cumulative hazard (respectively) at a user specified point t , and plotting functions are provided for objects of type "CPHshape". For this example, we also compare the estimator with the true hazard function (shown in red).

```
> find.hazard(1, mle)
```

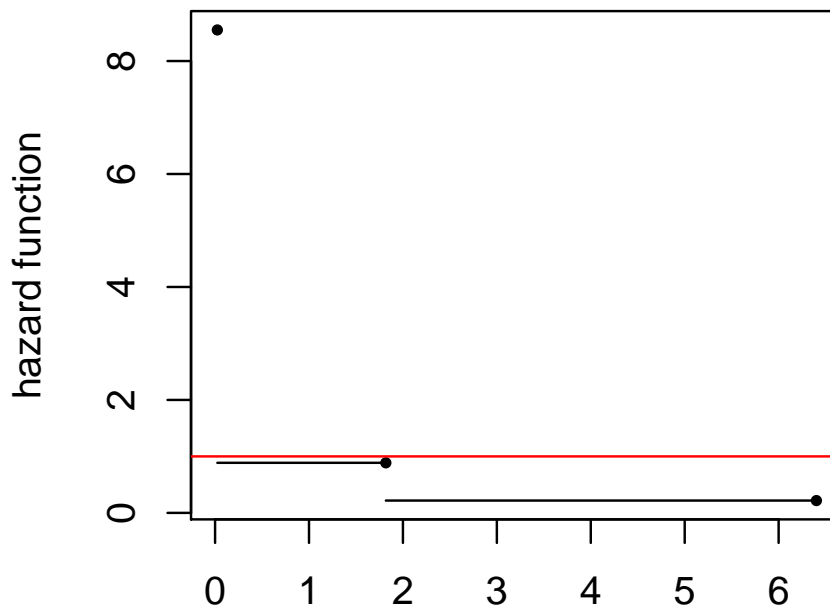
```
[1] 0.8863219
```

```
> find.cumulative(1, mle)
```

```
[1] 1.069799
```

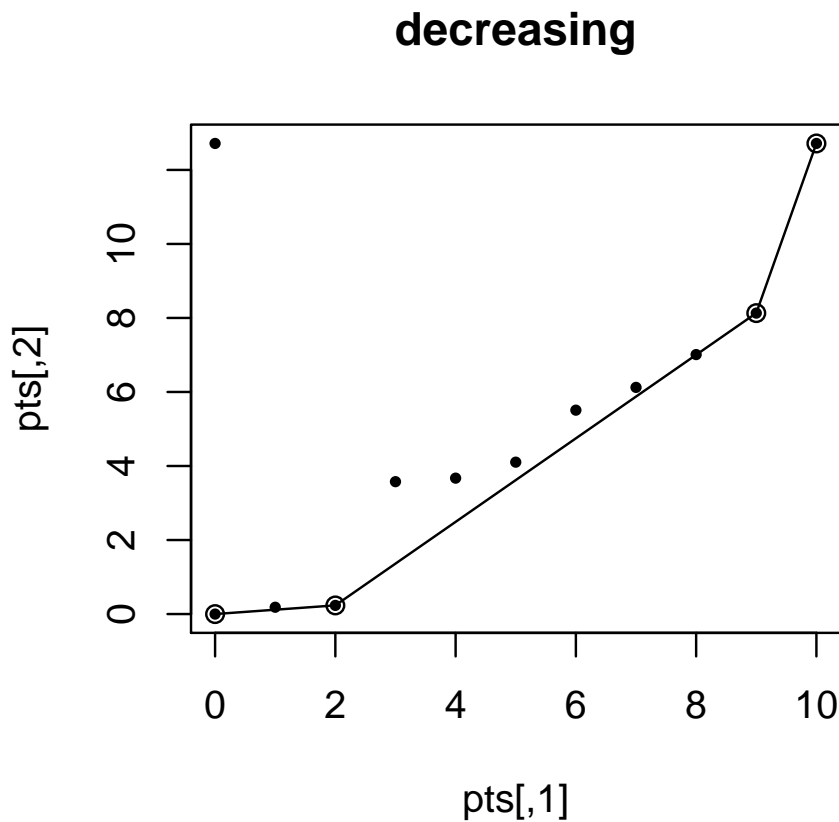
```
> plot(mle)
```

```
> abline(h=1, col="red")
```



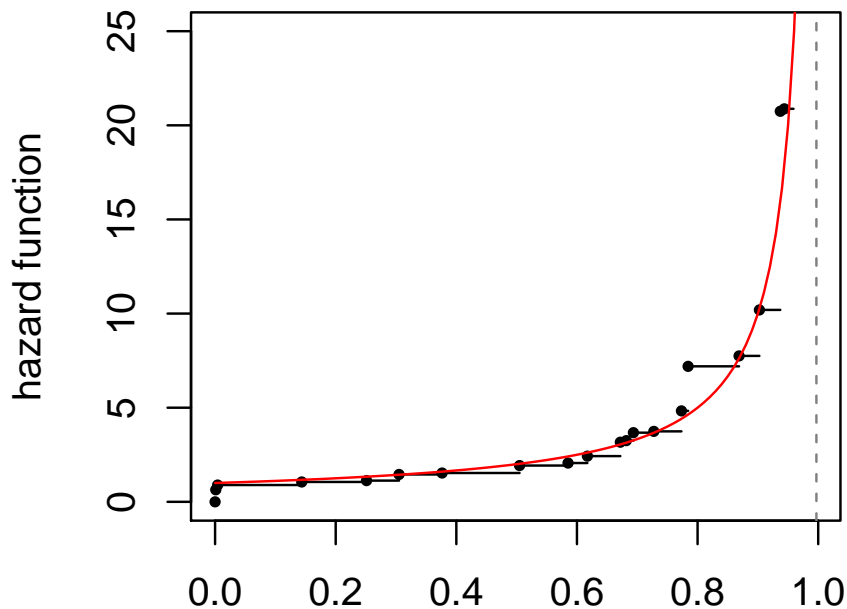
As described in Section 3, the MLE is found by using a graphical representation. Setting `plot=TRUE` in the function `find.shapeMLE` plots this graphical representation in R. The plot below confirms that the MLE we found for the above sample is piecewise constant with three “pieces”.

```
> mle <- find.shapeMLE(x, type="decreasing", plot=TRUE)
```



Suppose next that we observe X_1, \dots, X_n uniform random variables. Then the true hazard function is increasing and equal to $h(x) = (1-x)^{-1}$. The true hazard function is again shown in red.

```
> set.seed(12345)
> x <- runif(500)
> mle <- find.shapeMLE(x, type="increasing")
> plot(mle, ylim=c(0,25))
> htrue <- function(x) 1/(1-x)
> plot(htrue, col="red", add=TRUE)
```



4.2. MLE in the proportional hazards model

The full functionality of the **CPHshape** is best demonstrated with the function `find.shapeCPH`. We illustrate it here by first generating some data with two covariates from a uniform baseline model with random right censoring.

```
> set.seed(12345)
> n <- 200
> beta1 <- 1
> beta2 <- 2
> z1 <- rbinom(n,1,0.5)
> z2 <- runif(n, -1,1)
> w <- exp(beta1*z1+beta2*z2)
> temp <- runif(n)
> x <- 1-temp^(1/w)
> u <- runif(n, 0, 1)
> delta <- 1*(x <=u)
> x <- pmin(x,u)
```

We now fit an increasing baseline hazard. By setting the option `print=TRUE` we can see the

iterations of the main algorithm described above.

```
> z <- cbind(z1,z2)
> mle <- find.shapeCPH(x, z=z, delta=delta, type="increasing", print=TRUE)

iter=i phi[i] |phi[i]-phi[i-1]| beta(s)
0 -49.76003 NA 1 1
1 -67.4077 17.64767 0.9295706 1.899814
2 -68.48837 1.080672 0.9807406 2.053447
3 -68.92084 0.4324651 1.040361 2.106185
4 -69.14369 0.2228495 1.086093 2.137023
5 -69.26223 0.1185384 1.119647 2.158539
6 -69.32627 0.06403928 1.144271 2.174143
7 -69.36124 0.03497721 1.162431 2.185603
8 -69.3805 0.01925569 1.175881 2.194072
9 -69.39116 0.0106621 1.185877 2.200356
10 -69.39709 0.00592888 1.193324 2.205032
11 -69.4004 0.003307256 1.198881 2.20852
12 -69.40225 0.001849167 1.203034 2.211124
13 -69.40328 0.001035711 1.206142 2.213071
14 -69.40386 0.0005808498 1.208468 2.214529
15 -69.40419 0.0003260686 1.21021 2.21562
16 -69.40437 0.0001831759 1.211516 2.216438
17 -69.40448 0.0001029586 1.212495 2.217051
18 -69.40453 5.789395e-05 1.213229 2.217511
19 -69.40457 3.256384e-05 1.213779 2.217855
20 -69.40458 1.832048e-05 1.214192 2.218114
21 -69.40459 1.03089e-05 1.214502 2.218307
22 -69.4046 5.801542e-06 1.214734 2.218453
```

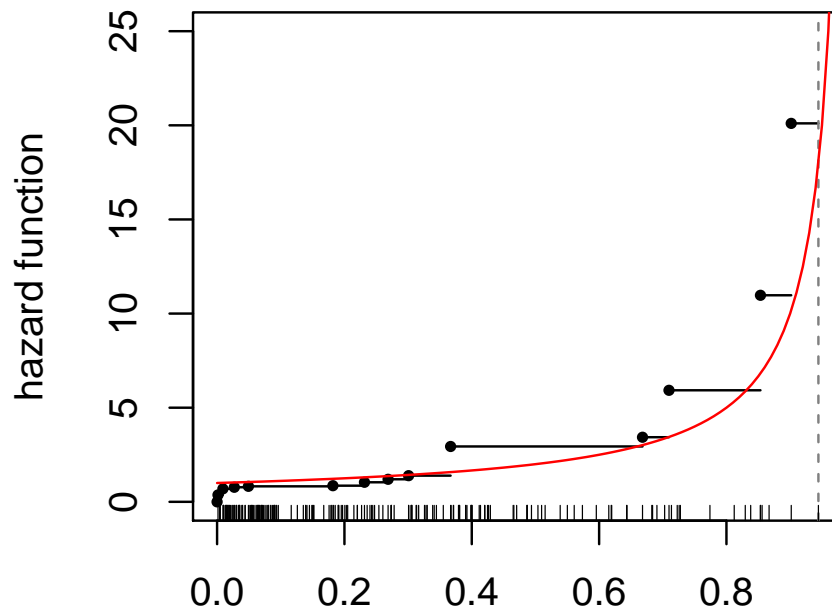
The log-likelihood is found to be 69.4046 (equal to $-\phi$ in the output) after 22 iterations. The resulting MLE for β is

```
> mle$beta

[1] 1.214734 2.218453
```

and a plot of the fitted baseline hazard is as follows.

```
> plot(mle, ylim=c(0,25))
> htrue <- function(x) 1/(1-x)
> plot(htrue, col="red", add=TRUE)
> rug(x)
```



We have added again the true hazard function in red to the plot.

Lastly, one may also be interested in viewing the fitted hazard function for a particular choice of covariates. This can be easily accomplished as follows.

```
> z1 <- 1
> z2 <- 0.5
> w <- exp(mle$beta[1]*z1+mle$beta[2]*z2)
> mleZ <- mle
> mleZ$h.val <- w*mle$h.val
```

The new object `mleZ` is of class "CPHshape" and can be plotted or evaluated using `find.hazard`, `find.cumulative`.

Acknowledgements

The second author thanks Georges Monette for many helpful discussions on R.

References

- Chung D, Chang MN (1994). “An isotonic estimator of the baseline hazard function in Cox’s regression model under order restriction.” *Statist. Probab. Lett.*, **21**(3), 223–228.
- Cook DJ, Walter SD, Cook RJ, Griffith LE, Guyatt GH, Leasa D, Jaeschke RZ, Brun-Buisson C (1998). “Incidence of and Risk Factors for Ventilator-Associated Pneumonia in Critically Ill Patients.” *Annals of Internal Medicine*, **129**(6), 433–440.
- Cox DR (1972). “Regression models and life-tables.” *J. Roy. Statist. Soc. Ser. B*, **34**, 187–220.
- Eddy WF (1977). “A new convex hull algorithm for planar sets.” *ACM Transactions on Mathematical Software*, **3**, 398–403.
- Grenander U (1956). “On the theory of mortality measurement. II.” *Skand. Aktuarietidskr.*, **39**, 125–153 (1957).
- Hui R, Jankowski H (2011). “Shape constrained maximum likelihood estimation for the proportional hazards model.” Poster presented at the annual Statistical Society of Canada conference in Halifax, Canada, URL <http://www.math.yorku.ca/~hkj/Students/CPHposter.pdf>.
- Jankowski HK, Wellner JA (2009). “Computation of nonparametric convex hazard estimators via profile methods.” *Journal of Nonparametric Statistics*, **21**, 505–518.
- Lamport L (1986). *L^AT_EX: A Document Preparation System*. Addison–Wesley Pub. Co., Reading, MA.
- Leisch F (2002). “Dynamic generation of statistical reports using literate data analysis.” In W Härdle, B Rönz (eds.), *COMPSTAT 2002 - Proceedings in Computational Statistics*.
- Lopuhaä HP, Nane GF (2011). “Shape constrained nonparametric estimators of the baseline distribution in Cox proportional hazards model.” *Preprint*.
- Mykityn SW, Santner TJ (1981). “Maximum likelihood estimation of the survival function based on censored data under hazard rate assumptions.” *Comm. Statist. A—Theory Methods*, **10**(14), 1369–1387.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.

Affiliation:

Rihong Hui
Hanna K. Jankowski
Department of Mathematics and Statistics
York University
Toronto, Canada
E-mail: hkj@mathstat.yorku.ca
URL: <http://www.math.yorku.ca/~hkj/>