A Note on the Computation of Empirical Likelihood Confidence Intervals for Hazard Integrals

Mai Zhou
Dr Bing Zhang Department of Statistics, University of Kentucky,
Lexington, KY 40536 USA   Email: maizhou@gmail.com

Abstract

Confidence intervals/regions based on the likelihood ratio (Wilks intervals) have many desirable properties. However, the computation of such confidence regions is non-trivial, especially in higher dimensions. We point out a computational shortcut for the confidence interval/region for hazard type parameters. Using the shortcut, the confidence intervals for hazard type parameters can be 100 times, even 1000 times faster than those for the mean parameters. Examples with right censored data are given with some R code.

Keywords and Phrases: Average hazard; Mean; Wilks confidence intervals, Right censored data.

1 Introduction

Wilks confidence intervals have many advantages over the Wald type confidence intervals. See for example (Meeker and Escobar 1995) [2]. Empirical likelihood-based confidence intervals share the same nice properties with those based on parametric likelihoods. The disadvantage of the Wilks type confidence intervals often cited is the difficulty of computation. Empirical likelihood adds an extra layer of complication. Faster computers certainly made this less painful, but the task is still non-trivial. We point out a shortcut/trick here for the computation of Wilks confidence intervals based on the empirical likelihood for *hazard type parameter*. Compared to the computation of such confidence intervals for the mean type parameter, the speeding up is at least 10 times (non-censored data), but can be 100 or even 1000 times (right censored data). Three examples with R codes are given.

2 Mean type parameter

Owen's (1988) [3] paper not only provided theoretical results for the empirical likelihood confidence intervals, but also provided a computational method for the empirical likelihood ratio in the i.i.d. data setting when dealing with a mean type parameters. These are widely adopted in subsequent papers on empirical likelihood and implemented in software like R package `emplik` [5].

Given the i.i.d. observations $X_1, X_2, \cdots, X_n$ with a distribution function $F(t)$, let us consider a constraint or null hypothesis:

$$\int g(t) dF(t) = \mu. \tag{1}$$

We call this a mean type parameter. The maximum empirical likelihood under this constraint can be computed as follows:

(a). Solving for λ

$$\sum_{i=1}^{n} \frac{g(x_i) - \mu}{n - \lambda(g(x_i) - \mu)} = 0 \ ,$$

typically by a Newton-Raphson iteration. Let us denote the solution by $\lambda^* = \lambda^*(\mu, g)$.

(b). Once we obtained λ^*, we can get the MLE probabilities under the mean constraint by

$$p_i = \frac{1}{n - \lambda^*(g(x_i) - \mu)} \ , \qquad i = 1, 2, \cdots, n. \tag{2}$$

(c). With the probabilities p_i, we can compute the log empirical likelihood easily by $\sum \log p_i$. The log empirical likelihood ratio is just $\sum \log np_i$.

We notice that for a fixed μ there are only two values of λ ($\lambda = 0$ and $\lambda = \lambda^*$) that will make (2) a probability. The solution we are interested is of course, λ^*. We notice that this probability depends on both μ and λ^*. We first fix μ and then look for λ^*.

3 Hazard Type Parameter: Poisson Empirical Likelihood)

For the i.i.d. data setting or right censored i.i.d. lifetime data setting, Pan and Zhou (2002) [4] discussed testing an integrated hazard type parameter,

$$H_0 : \int_0^\infty g(t) d\Lambda(t) = \theta \tag{3}$$

via the empirical likelihood ratio.

The empirical likelihood, in this case, is formulated by way of the cumulative hazard function. The maximization of the likelihood forces the cumulative hazard functions involved to be discrete, similar to the mean/distributions case.

The definition for the Poisson version of empirical likelihood is (see [4])

$$\log AL(\Lambda) = \sum_i \left\{ \Delta N(t_i) \log \Delta\Lambda(t_i) - \sum_j \Delta\Lambda(t_j) I[t_j \le t_i] \right\} \ . \tag{4}$$

It is easy to show that if $\Delta\Lambda$ takes the jump of the Nelson-Aalen estimator, $\log AL$ will be maximized.

The cumulative hazard function Λ that maximizes $\log AL$ under the additional constrain (3) above is specified via the jumps of the cumulative hazard function, as in (5) below.

Since the i.i.d. right-censored case does not add much complication and includes the non-censored case as a special case, we shall assume the given data are right-censored.

Suppose we observe (T_i, δ_i) with $T_i = \min(X_i, C_i)$, $\delta_i = I[X_i \leq C_i]$, $i = 1, 2, \cdots, n$. Non-censored data is just a special case with all $C_i = \infty$. We further assume the i.i.d. lifetimes X_i have a cumulative hazard function $\Lambda(t)$. Without loss of generality, we assume the data is ordered according to T.

The jumps of $\Lambda(t)$ that maximize the empirical likelihood (4) under the constraint (3) is given by the formula below (Pan and Zhou 2002 [4]):

$$\Delta\Lambda(t_i) = \frac{\Delta N(t_i)}{R(t_i) - \lambda g(t_i)} \,, \qquad i = 1, 2, \cdots, n-1 \tag{5}$$

where $N(t)$ and $R(t)$ are the standard notation in survival analysis:

$$N(t) = \sum_{i=1}^{n} I[T_i \leq t, \delta_i = 1], \qquad R(t) = \sum_{i=1}^{n} I[T_i \geq t] \,. \tag{6}$$

On the other hand, it is easy to show the last jump of a discrete cumulative hazard, $\Delta\Lambda(t_n)$, is always 1, so the above formula only covers the cases from $i = 1$ to $n - 1$.

The λ in (5) satisfy (the constraint)

$$\sum_{i=1}^{n-1} g(t_i) \frac{\Delta N(t_i)}{R(t_i) - \lambda g(t_i)} + g(t_n) = \theta \,, \tag{7}$$

which is typically solved by a Newton-Raphson: given θ solve for λ.

Remark. When the last observation is censored, the Nelson-Aalen estimator is not uniquely defined beyond the last observation, similar to the Kaplan-Meier. We took the convention that defines the last observation always as uncensored in the Kaplan-Meier, this amounts to take the smallest probability value in all non-unique values of the Kaplan-Meier. Equivalently we take the convention that the jump of a Nelson-Aalen estimator (and cumulative hazard under the null hypothesis) at last observation to be always one. For reference of the Nelson-Aalen and Kaplan-Meier estimators, see any standard survival analysis book, for example [1].

3.1 Computational Advantage

We point out a **subtle difference** between the mean and hazard case: that the formula (5) does not involve θ and there are infinitely many λ values (in fact an interval containing zero, called feasible values by Pan and Zhou (2002) [4]) that will make (5) a legitimate hazard function jumps. Clearly when $\lambda = 0$ these are the jumps of the Nelson-Aalen estimator. In contrast, the formula (2) involves μ, and once μ is given, there is only one λ value (besides zero) that make (2) a legitimate probability.

This enables us to solve the hazard problem by first **skipping** θ: start by picking a λ, find the hazard jumps by (5), then we can find both the log empirical likelihoods with and without constraint (3) (thus the likelihood ratio) by (4).

The log empirical likelihood ratio is just

$$\log AL(\text{jumps from (5)}) - \log AL(\text{jumps of the Nelson-Aalen})\,.$$

This will enable us to find a confidence interval for λ first. After we find the confidence interval/region for λ, we can easily turn it into an interval/region for θ:

Easy to verify here the λ and θ are strictly one-to-one and monotone, see (7). We may finally obtain the confidence interval/region for θ by only computing the θ corresponding to those λ on the boundary of the confidence region.

We summarize as follows: Finding the confidence interval for parameter specified by $\int g(t)d\Lambda(t)$, is particularly easy: try a bunch of λ values near zero (both positive and negative); compute the corresponding log empirical likelihood ratio for each λ until we find one positive λ with log empirical likelihood ratio = 3.84 and one negative λ with log empirical likelihood ratio also = 3.84. Denote those values by λ^- and λ^+. We shall call them the λ confidence interval. Notice so far we have not worried about θ values.

The 95% confidence interval for $\theta = \int g(t)d\Lambda(t)$ is just

$$\theta^- = \sum_{i=1}^{n-1} g(t_i)\frac{\Delta N(t_i)}{R(t_i) - \lambda^- g(t_i)} + g(t_n), \qquad \theta^+ = \sum_{i=1}^{n-1} g(t_i)\frac{\Delta N(t_i)}{R(t_i) - \lambda^+ g(t_i)} + g(t_n)\ .$$

In other words, the search of confidence interval is in terms of λ, and the target function is log empirical likelihood ratio. We only find the corresponding θ after we have found the confidence interval for λ.

Compare to the mean case, there we must search the confidence interval in terms of the mean μ. For each tentative value of μ, we need a Newton-Raphson iteration to find the corresponding λ before computing the related log empirical likelihood ratio.

Another way to describe this advantage is: given a $g(t)$ function, we can find the 95% confidence interval in terms of the λ when dealing with the hazard type parameters. The computation path that needs to be performed many times over is $\lambda \Longrightarrow$ empirical likelihood. After the λ confidence interval is found, we need to do the following only once: find the corresponding interval of θ related to the λ interval.

For the mean type parameter, the confidence interval must be calculated/searched starting from μ: the computation path that needs to be performed many times over is $\mu \Longrightarrow \lambda \Longrightarrow$ empirical likelihood. We remind reader the first step ($\mu \Longrightarrow \lambda$) is done by a Newton-Raphson.

3.2 Hazard Type Parameter: Binomial Empirical Likelihood

In Pan and Zhou (2002) [4] they discussed two slightly different versions of the hazard empirical likelihoods. One is the Poisson version of the hazard empirical likelihood discussed in the previous section, the other is the binomial version of the hazard empirical likelihood we shall discuss now.

Since the results are parallel and similar, we shall be brief and only discuss the differences.

(1) The empirical likelihood function is defined differently. The Poisson log empirical likelihood is defined in (4), while the binomial version is

$$\log EL = \sum_i \{\Delta N(t_i) \log \Delta\Lambda(t_i) + [R(t_i) - \Delta N(t_i)] \log[1 - \Delta\Lambda(t_i)]\} \ . \tag{8}$$

(2) The parameter defined by integration is different. For the binomial empirical likelihood the parameter is:

$$\theta_B = -\sum_{i=1}^{n-1} g(t_i) \log(1 - \Delta\Lambda(t_i)) \ . \tag{9}$$

Notice the last jump is excluded, since the jump is always 1. We may write the parameter θ_B as $-\int g(t) \log(1 - d\Lambda(t))$.

On the other hand, the MLE of the hazard jumps $\Delta\Lambda(t_i)$, with or without the constraint $-\int g(t) \log(1 - d\Lambda(t))$ equal to a specific value, has the same formula: either (5) or the Nelson-Aalen jump.

The computational ease of the confidence intervals for parameter θ_B in (9), is similar to the Poisson version as for this binomial version.

4 Two Sample Case

The same computational advantage of the hazard confidence intervals discussed above is also valid for the two-sample case, with parameter being the difference of the two integrated hazards.

Suppose we observe (T_i, δ_i), $i = 1, 2, \cdots, n$. The (T_i, δ_i) are defined same as in section 3. In addition, we also observe (S_j, d_j), $j = 1, 2, \cdots, m$; where $S_j = \min(Y_j, V_j)$, $d_j = I[Y_j \leq V_j]$. We assume the i.i.d. lifetimes Y_j have a cumulative hazard function $\Lambda_2(t)$.

The parameter of interest is

$$\int g_1(t) d\Lambda_1(t) - \int g_2(t) d\Lambda_2(t) = \theta_{II} \, . \tag{10}$$

We point out that if we take

$$g_1(t) = g_2(t) = \frac{R_1(t)R_2(t)}{R_1(t) + R_2(t)}$$

then the statistic θ_{II} in (10) is precisely the familiar log-rank statistic, see for example (Kalbfleisch and Prentice 2002) [1]. Here R_2 (and N_2 below) are similarly defined as in (6), but for sample two.

Remark. The g function for the log-rank statistic is random. However, as long as the function is 'predictable', as is the case here, then everything are still fine. See (Pan and Zhou 2002) [4] for details. We shall still call θ_{II} a parameter.

When maximizing the log empirical likelihood for the two sample under the constraint (10), we end up with the cumulative hazards with jumps given below, see for example (Zhou 2016 section 2.6) [6] . For given g_1 and g_2, the jump of the cumulative hazards for two samples at λ are

$$\Delta\Lambda_1(t_i) = \frac{\Delta N_1(t_i)}{R_1(t_i) - \lambda g_1(t_i)} \ , \qquad \Delta\Lambda_2(s_j) = \frac{\Delta N_2(s_j)}{R_2(s_j) + \lambda g_2(s_j)} \ . \tag{11}$$

The two-sample log empirical likelihood ratio (Poisson version) is just the sum of the two one-sample Poisson log empirical likelihood ratios:

$$\log(\text{two-sample ELR}) = \log(\text{ELR of sample 1}) + \log(\text{ELR of sample 2})$$

where ELR is short for Empirical Likelihood Ratio.

Finally, the computation of the confidence interval for θ_{II} can be summarized as following steps:

(a) Given g_1 and g_2 and data, pick a λ and compute the jumps of two cumulative hazards by (11).

(b) With the jumps of cumulative hazards, compute each sample's log empirical likelihood ratio. Sum the two one-sample log empirical likelihood ratios to get the two-sample log empirical likelihood ratio.

(c) Search in terms of λ, until we find two λs both with $-2\log$ (two-sample ELR) $= 3.84$. This is the confidence interval for λ. Denote the interval as $[\lambda^-, \lambda^+]$.

(d) We can convert the interval in terms of λ into the interval for θ_{II} by

$$\theta_{II}^- = \sum_{i=1}^{n-1} g_1(t_i) \frac{\Delta N_1(t_i)}{R_1(t_i) - \lambda^- g_1(t_i)} + g_1(t_n) - \sum_{j=1}^{m-1} g_2(s_j) \frac{\Delta N_2(s_j)}{R_2(s_j) + \lambda^- g_2(s_j)} - g_2(s_m)$$

$$\theta_{II}^+ = \sum_{i=1}^{n-1} g_1(t_i) \frac{\Delta N_1(t_i)}{R_1(t_i) - \lambda^+ g_1(t_i)} + g_1(t_n) - \sum_{j=1}^{m-1} g_2(s_j) \frac{\Delta N_2(s_j)}{R_2(s_j) + \lambda^+ g_2(s_j)} - g_2(s_m) \ .$$

In the above discussion we used the Poisson empirical likelihood. Parallel results are also valid for the binomial empirical likelihood. We shall not write out the details to save space.

5 Examples

Next we take the `pbc` data from the R `survival` package for our examples. We ignore the different treatments here and use it as a one-sample data.

5.1 Poisson Likelihood Case

We need to define a new function `emplikH1P` in R to compute the Poisson empirical log likelihood ratio for the given λ, before computing the confidence interval. Similar function for the binomial empirical likelihood will be called `emplikH1B` in next subsection.

Once this function is defined, the confidence interval for λ can easily be obtained by a call to the function `findUL` in package `emplik`. Here we take the function $g(t) = I[t \leq 5 * 365.25]$ which defines the hazard parameter via (3).

```
library(emplik)
library(survival)
data(pbc)
findUL(step=0.1, fun=emplikH1P, MLE=0, x=pbc$time, d=(pbc$status==2),
                                   fung=function(t){as.numeric(t <= 5*365.25)} )

## $Low
## [1] -0.1228794
##
## $Up
## [1] 0.159799
##
## $FstepL
## [1] 6.103516e-05
##
## $FstepU
## [1] 6.103516e-05
##
## $Lvalue
## [1] 3.84158
##
## $Uvalue
## [1] 3.840033
```

We see the 95% confidence interval for the λ is $[-0.1228794, 0.159799]$. We can turn this into a confidence interval for the parameter $\int I[t \leq 5 * 365.25] d\Lambda(t) = \theta$ by two easy calculations (we only include the relevant outputs to save space).

```
emplikH1P(lambda=0.159799, x=pbc$time, d=(pbc$status==2),
          fung=function(t){as.numeric(t <= 5*365.25)}, CIforTheta=TRUE)

## $MeanHaz
## [1] 0.2903727

emplikH1P(lambda=-0.1228794, x=pbc$time, d=(pbc$status==2),
          fung=function(t){as.numeric(t <= 5*365.25)}, CIforTheta=TRUE)

## $MeanHaz
## [1] 0.422227
```

We see the 95% confidence interval for the θ is $[0.2903727, 0.422227]$.

The `MLE` input for the R function `findUL` should be always 0, since $\lambda = 0$ in (5) gives the MLE (Nelson-Aalen). The `step` input for the `findUL` should be a positive number. It is the initial search step size for λ.

5.2 Binomial Likelihood Case

We first define a function `emplikH1B` which computes the binomial empirical likelihood ratio for the given λ (see appendix).

Now we can find the confidence interval.

```
findUL(step=0.2, fun=emplikH1B, MLE=0, x=pbc$time, d=(pbc$status==2),
               fung=function(t){as.numeric(t <= 5*365.25)} )
$Low
[1] -0.1226231

$Up
[1] 0.1595283

$FstepL
[1] 6.103516e-05

$FstepU
[1] 6.103516e-05

$Lvalue
[1] 3.84002

$Uvalue
[1] 3.840848
```

We see the 95% confidence interval in terms of λ is $[-0.1226231, 0.1595283]$. Next we find the corresponding interval for θ_B. We only include the relevant output.

```
emplikH1B(lambda=0.1595283, x=pbc$time, d=(pbc$status==2),
           fung=function(t){as.numeric(t <= 5*365.25)}, CIforTheta=TRUE)

## $IntHaz
## [1] 0.2908711

emplikH1B(lambda=-0.1226231, x=pbc$time, d=(pbc$status==2),
           fung=function(t){as.numeric(t <= 5*365.25)}, CIforTheta=TRUE)

## $IntHaz
## [1] 0.422962
```

We see the 95% confidence interval for θ_B is $[0.2908711, 0.422962]$.

5.3 Speed Comparison

Let us compare our hazard confidence interval calculation to the Owen (1988) mean confidence interval (even though it ignores censoring). The result is as follows. The mean procedure taking the censoring into account is available in R package `emplik`. They are slower than those of Owen (1988).

The one-sample mean confidence interval, no censor:

```
myx <- as.numeric( pbc$time <= 5*365.25 )
myfun99 <- function(theta,x){ el.test(x=x, mu=theta) }
findUL(step=0.1, fun=myfun99, MLE=0.5, x=myx)
## $Low
## [1] 0.4808036
##
## $Up
## [1] 0.5762452
##
## $FstepL
## [1] 6.103516e-05
##
## $FstepU
## [1] 6.103516e-05
##
## $Lvalue
## [1] 3.8391
##
## $Uvalue
## [1] 3.836355
```

The 95% confidence interval is $[0.4808036, 0.5762452]$. The actual values of the interval are not comparable to those of hazards since one ignores censoring the other does not. But a comparison of computing timing is telling.

```
system.time(for(i in 1:100) findUL(step=0.1, fun=emplikH1B, MLE=0, x=pbc$time,
                d=(pbc$status==2), fung=function(t){as.numeric(t <= 5*365.25)} ))
## user   system  elapsed
## 0.51     0.00     0.51

system.time( for(i in 1:100) findUL(step=0.1, fun=myfun99, MLE=0.52, x=myx) )
## user  system elapsed
## 5.24    0.00    5.23
```

We see the computation of hazard empirical confidence intervals are about 10 times faster, even when the mean procedure ignores the right censoring! (5.24 seconds vs. 0.51 second).

In the one sample case, if we always consider right censored data, we obtain the following: the mean vs. hazard Wilks confidence intervals calculation timing (of 100 such intervals): 180 seconds vs. 0.51 second.

As for the two sample case, the confidence interval for the difference of two means are even more complicated than just one mean. Two-sample mean difference confidence intervals can be obtained by profiling, as shown in Zhou (2020) [7]. Thus the speed advantage of the confidence interval for the difference of two integrated hazards will be even more profound.

After some computations we obtain the following timings:

For 100 Poisson type two-sample hazard difference Wilks confidence intervals: 0.84 second.

For 100 binomial type two-sample hazard difference Wilks confidence intervals: 0.77 second.

For 100 two-sample mean difference Wilks confidence intervals (not consider censor): 98.58 seconds.

For 100 two-sample mean difference Wilks confidence intervals (consider censor): 2593 seconds.

For the detailed R code of the above computations, please see the Appendix.

6 Discussion: Approximate Mean by Hazard

Even though the parameters in terms of hazard integral has their own merit (like weighted hazard difference from two samples) we sometimes want to work with means.

Can we take advantage of this simpler and faster computing algorithm meant for integrated hazards to help those for the mean? Unfortunately not, at least not directly. What we can do is to approximate the empirical likelihood solution for the mean parameter, and use the hazard solution as a starting point for the mean iteration.

Let us suppose the parameter of interest is $\mu = \int g(t) dF(t)$. Notice we have, for any given F^*

$$\int_0^\infty g(t)[1 - F^*(t-)] d\Lambda(t) = \int_0^\infty g(t) \frac{1 - F^*(t-)}{1 - F(t-)} dF(t) \, .$$

If we can chose a known or computable F^* to make the ratio

$$g(t) \frac{1 - F^*(t-)}{1 - F(t-)} \approx g(t)$$

then we are in business. However, the F in above is unknown and precisely will be the constrained MLE we are seeking.

In some cases, we may argue that the Kaplan-Meier $\hat{F}_{km}$ is a good guess, if it approximately satisfy the constraint.

Any way, we can obtain the 95% confidence interval for the parameter $\int g(t)[1-\hat{F}_{km}(t-)] d\Lambda(t)$. This provide us a good starting point for the confidence interval for $\int g(t) dF(t)$.

Another possibility is to iterate, that is to use the (constrained MLE) F solution of the "Use KM as F^*' problem (as the F^* in second round of computation). And so on.

References

[1] Kalbfleisch, J. D.; Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data.* New York: John Wiley & Sons. ISBN 047136357X.

[2] Meeker, WQ, Escobar, L. (1995). Teaching about Approximate Confidence Regions Based on Maximum Likelihood Estimation *The American Statistician*, Vol. 49, No. 1, pp. 48–53.

[3] Owen, A. (1988). Empirical likelihood ratio confidence intervals for a single functional. *Biometrika*, 75, 237–249.

[4] Pan, X. and Zhou, M. (2002). Empirical likelihood ratio in terms of cumulative hazard function for censored data. *Journal of Multivariate Analysis*, 80, 166–188.

[5] Zhou, M. (2020). emplik: Empirical Likelihood Ratio for Censored/Truncated Data. R package version 1.1-1 https://CRAN.R-project.org/package=emplik

[6] Zhou, M. (2016). *Empirical Likelihood Methods in Survival Analysis.* CRC Press, Taylor & Francis Group, Boca Raton.

[7] Zhou, M. (2020). Restricted mean survival time and confidence intervals by empirical likelihood ratio. *Journal of Biopharmaceutical Statistics*, Available online Dec. 2020.

7 Appendix

```
> system.time(for (i in 1:100) findUL(step=15, fun=emplikH2P, MLE=0, x1=pbcX1, d1=pbcd1, x2=pbcX2, d2=pbcd2,
                fun1=function(t){as.numeric(t <= 5*365.25)},fun2=function(t){as.numeric(t <= 5*365.25)} ))
user  system elapsed
0.84    0.00    0.84

> system.time(for (i in 1:100) findUL(step=15, fun=emplikH2B, MLE=0, x1=pbcX1, d1=pbcd1, x2=pbcX2, d2=pbcd2,
                fun1=function(t){as.numeric(t <= 5*365.25)},fun2=function(t){as.numeric(t <= 5*365.25)} ))
user  system elapsed
0.77    0.00    0.76
```

```
#########################################################################
### Data with No censor. Difference of 2 means of I[x <= 5*365.25].   ###
#########################################################################

pbcX1F <- as.numeric(pbcX1 <= 5*365.25)
pbcX2F <- as.numeric(pbcX2 <= 5*365.25)

MeanD <- function(r, x1, x2, theta) {
temp1 <- el.test(x=x1, mu=r)
temp2 <- el.test(x=x2, mu=r-theta)
return(temp1$"-2LLR" + temp2$"-2LLR")
}

ThetaD <- function(theta, x1, x2) {
temp <- optimize(f=MeanD, lower=-1, upper=1, x1=x1, x2=x2, theta=theta)  ##since this is a difference of 2 probabilities
cstar <- temp$minimum
val <- temp$objective
list("-2LLR"=val, cstar=cstar, Pval= 1-pchisq(val, df=1)) }

findUL(step=100, fun=ThetaD, MLE=0, x1=pbcX1F, x2=pbcX2F)

> system.time(for (i in 1:100) findUL(step=0.1, fun=ThetaD, MLE=0, x1=pbcX1F, x2=pbcX2F) )
user  system elapsed
98.58    0.11   98.68
```

Time: 98.5 seconds vs. 0.84 second (on last page)!

```
######################################################################
### Difference of two survival probabilities. Right censored data  ###
######################################################################

survDiff <- function(r, x1, d1, x2, d2, theta){
 temp1 <- el.cen.EM2(x=x1,d=d1,fun=function(x){as.numeric(x > 5*365.25)},mu=r)
 temp2 <- el.cen.EM2(x=x2,d=d2,fun=function(x){as.numeric(x > 5*365.25)},mu=r-theta)
 return(temp1$"-2LLR" + temp2$"-2LLR")
 }

DiffTheta <- function(theta, x1, d1, x2, d2){
 temp <- optimize(f=survDiff,
 lower=-1,
 upper=1,
 x1=x1, d1=d1, x2=x2, d2=d2,
 theta=theta)
 cstar <- temp$minimum
 val <- temp$objective
 list("-2LLR"=val, cstar=cstar)
 }

findUL(step=0.1, fun=DiffTheta, MLE=0, x1=pbcX1, d1=pbcd1, x2=pbcX2, d2=pbcd2)
$Low
[1] -0.1114954

$Up
[1] 0.09770364

$FstepL
[1] 0.0001113891

$FstepU
[1] 6.103516e-05

$Lvalue
[1] 3.83986

$Uvalue
[1] 3.83845

> system.time( for (i in 1:10) findUL(step=0.1, fun=DiffTheta, MLE=0, x1=pbcX1, d1=pbcd1, x2=pbcX2, d2=pbcd2) )
user  system elapsed
259.36    0.54  259.97
```

We point out this is only the time of 10 such calculations. So, the mean difference vs. hazard difference confidence interval calculation timing of 100 intervals: 2593 seconds and 0.84 second.

Finally the confidence interval of one sample mean, with right censored data

```
myfun9<-function(theta,x,d){
  el.cen.EM2(x, d,fun=function(x) {as.numeric(x<= 5*365.25)}, mu=theta)
  }

findUL(step=0.2, fun=myfun9, MLE=0.3, x=pbc$time, d=(pbc$status==2))
$Low
[1] 0.252829

$Up
[1] 0.3444198

$FstepL
[1] 6.103516e-05

$FstepU
[1] 6.103516e-05

$Lvalue
[1] 3.840855

$Uvalue
[1] 3.838925

system.time( findUL(step=0.2, fun=myfun9, MLE=0.3, x=pbc$time, d=(pbc$status==2)) )
user  system elapsed
1.76    0.02    1.78
system.time( findUL(step=0.2, fun=myfun9, MLE=0.3, x=pbc$time, d=(pbc$status==2)) )
user  system elapsed
1.78    0.00    1.78
system.time( for (i in 1:100) findUL(step=0.2, fun=myfun9, MLE=0.3, x=pbc$time, d=(pbc$status==2)) )
user  system elapsed
181.26    0.29  181.58
```

So, even in the one sample case, if we always take right censored data, the mean vs. hazard confidence interval calculation timing (of 100 such intervals): 180 seconds vs. 0.51 second.