DTDA: An Updated and Expanded R Package for the Statistical Analysis of Doubly Truncated Data

Jacobo de Uña-Álvarez

(joint work with Carla Moreira and Rosa M. Crujeiras)

Universidade de Vigo

jacobo@uvigo.es

VIII Xornada de Usuarios de R en Galicia

Santiago de C., Oct 14, 2021
Double truncation: definition

- Target variable $X$ observed only when $U \leq X \leq V$
- In that case truncation couple $(U, V)$ observed too
- Prominent example: interval sampling of time-to-event data:
  - Subjects with event within $[d_0, d_1]$ recruited
  - $X$: time-to-event
  - $\tau = d_1 - d_0$: interval width
  - $V$: time from birth to $d_1$
  - $U = V - \tau$
- Sample: iid triplets $(X_i, U_i, V_i)$, $1 \leq i \leq n$
- $(X_1, U_1, V_1)$ follows the cond cdf of $(X, U, V)$ given $U \leq X \leq V$
Doubly truncated data: interval sampling

Red segments are not observed
Doubly truncated data: fields of application

- **Astronomy**: quasar luminosities
- **Epidemiology**: AIDS, cancer, Parkinson’s Disease, Acute Coronary Syndrome, *autopsyConfirmed* neurodegenerative diseases
- **Engineering**: time to failure after installation of a device
- **Social Sciences/Finance**: marriage lengths, age at insolvency for companies
- (...)
Doubly truncated data: sampling bias

- Sampling probability for $X$:

$$G(x) = P(U \leq X \leq V | X = x) = P(U \leq x \leq V)$$

(last equality requires $(U, V) \perp X$)

- $G(x)$ may be constant, or may be not
- NPMLE $G_n(x)$ can be computed from the $(X_i, U_i, V_i)$'s
- NPMLE $F_n(x)$ of the target cdf $F(x) = P(X \leq x)$ is an IPWE:

$$F_n(x) = \frac{\sum_{i=1}^{n} I(X_i \leq x) G_n(X_i)^{-1}}{\sum_{j=1}^{n} G_n(X_j)^{-1}}$$

Weight attached to $X_i$: $W_i = G_n(X_i)^{-1} / \sum_{j=1}^{n} G_n(X_j)^{-1}$

- Iterative methods to compute $G_n$ (and $F_n$) needed (DTDA)
Doubly truncated data: simulated example

![Graph showing doubly truncated data with lines for naive, corrected, target, and samp.prob.](image)
DTDA package v3.0

- Maintainer: Carla Moreira
- Launched on April 11, 2021
- Update and expansion of the original DTDA (September 21, 2009)
- Main improvements:
  - Computational savings through parallel computing (bootstrapping!)
  - Smoothing methods to estimate density and hazard functions
  - New real datasets
  - Simulation of doubly truncated data (interval sampling)

- 46K downloads, 593 last month:
  - https://cranlogs.r-pkg.org/badges/grand-total/DTDA
  - https://cranlogs.r-pkg.org/badges/DTDA
Three iterative algorithms to compute $F_n$:

- `efron.petrosian(X, U = NA, V = NA, wt = NA, error = NA, nmaxit = NA, boot = TRUE, B = NA, alpha = NA, display.F = FALSE, display.S = FALSE)`

- `lynden(X, U = NA, V = NA, error = NA, nmaxit = NA, boot = TRUE, B = NA, alpha = NA, display.F = FALSE, display.S = FALSE)`

- `shen(X, U = NA, V = NA, wt = NA, error = NA, nmaxit = NA, boot = TRUE, boot.type = "simple", B = NA, alpha = NA, display.FS = FALSE, display.UV = FALSE, plot.joint = FALSE, plot.type = NULL)`

Function `shen()` computes and returns $G_n$ too.
Smoothing methods for density and hazard functions:

- `densityDT(X, U, V, bw = "DPI2", from, to, n, wg = NA)`
- `hazardDT(X, U, V, bw = "LSCV", from, to, n, wg = NA)`

Alternatively, use

- `density(X, bw = "nrd0", weights = W)`

with

- `W <- shen(...)$biasf^-1`
- `W <- W / sum(W)`

but if so take care with bandwidth selection!
Random generation of doubly truncated data (interval sampling):

```r
rsim.DT(n, tau, model = NULL)
```

...and many real datasets:

- Quasars
- AIDS
- ChildCancer
- AIDS.DT
- EquipSRounded
- PDearly, PDlate
- ACS, ACSred
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)

> library(DTDA)
> head(PDearly)  # two cases with missing info for V

<table>
<thead>
<tr>
<th>X</th>
<th>U</th>
<th>V</th>
<th>SNP_A10398G</th>
<th>SNP_PGC1a</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>30</td>
<td>38</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>46</td>
<td>39</td>
<td>47</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>34</td>
<td>42</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>54</td>
<td>49</td>
<td>57</td>
<td>A</td>
</tr>
</tbody>
</table>

> PDearly <- na.omit(PDearly)
> attach(PDearly)
> shen(X, U, V, display.FS=TRUE, display.UV=TRUE) -> res

n.iterations 56
S0 9.716169e-07
events 97
B 500
alpha 0.05
Boot simple
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)

Shen estimator

Survival

Marginal U

Marginal V
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)

> #sampling probability:
> plot(res$time, res.biasf, type = "s", ylim = c(0,1))

> #save the weights
> #useful to construct consistent estimators later:
> W <- res.biasf^-1
> W <- W / sum(W)

> #estimation of the mean:
> mean(X) #naive
[1] 46.97938
> sum(W * res$time) #correct
[1] 43.20277
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)

```r
> densityDT(X, U, V, from = min(X), to = max(X), + n = 500) -> res.dpi2  #bw default is DPI2
> plot(res.dpi2, type = "l", ylim = c(.02, .075))

> densityDT(X, U, V, bw = "DPI1", from = min(X), to = max(X), + n = 500) -> res2.dpi1
> lines(res2.dpi1$x, res2.dpi1$y, col = 2)

> res.dpi2$bw; res2.dpi1$bw
[1] 2.855729
[1] 2.470739

> density(X) -> resn.d  #naive approach
> lines(resn.d$x, resn.d$y, lwd = 2)
> resn.d$bw  #ordinary rule-of-thumb undersmooths
[1] 2.046491
```
DTDA in practice: Parkinson’s Disease (Clark et al. 2011)
DTDA in practice: AIDS (Kalbfleisch and Lawless 1989)

```r
> head(AIDS.DT)
X   U   V  AGE
1 0.5 -17.5 36.5 63
2 4.0 -18.5 35.5  1
3 4.0 -18.5 35.5 29
4 4.0 -33.5 20.5 46

> attach(AIDS.DT)
> res <- shen(X, U, V, boot = F, display.F = T,
+            display.UV = T)
n.iterations 23
S0 6.280252e-07
events 295

> plot(res$time, res$biasf, type = "s", ylim = c(0, 1))
```
DTDA in practice

Shen estimator

Survival

Marginal U

Marginal V
DTDA in practice: AIDS (Kalbfleisch and Lawless 1989)
> #hazard function:
> res.h <- hazardDT(X, U, V, bw = 5, from = min(X),
+ to = max(X), n = 500)
> plot(res.h$x, res.h$y, type = "l", ylim = c(0, .14))

> b <- seq(3, 7, length = 14)
> for (i in 1:k){
+ res hb <- hazardDT(X, U, V, bw = b[i], from = min(X),
+ to = max(X), n = 500)
+ lines(res hb$x, res hb$y, col = "gray")
+ }
> lines(res.h$x, res.h$y, col = 2, lwd = 2)
DTDA in practice: AIDS (Kalbfleisch and Lawless 1989)
DTDA in practice: simulating double truncation

- Simulating interval sampling:
  \[ X \sim U(0, 1) \]
  \[ U \sim U(-\tau, 1), \ V = U + \tau \text{ in model 1 (no sampling bias)} \]
  \[ U \sim \text{uniform squared (Beta)} \text{ in model 2 (sampling bias)} \]

```r
> set.seed(1234)
> rsim.DT(500, tau = 1/3, model = 2) -> mysim
> str(mysim)
'data.frame': 500 obs. of 3 variables:
$ X: num 0.11 0.622 0.145 0.443 0.58 ...
$ U: num -0.1687 0.5222 -0.0967 0.3343 0.5005 ...
$ V: num 0.165 0.856 0.237 0.668 0.834 ...
> ressim <- shen(mysim$X, mysim$U, mysim$V, boot = F)
> plot(ressim$time, ressim$biasf, type = "s", ylim = c(0, 1))
```
DTDA in practice: simulating double truncation
Further use of the sampling probabilities $W$

- **Linear (multiple, polynomial) regression:**
  
  `lm(formula, data, subset, weights = W, na.action, ...)`

- **Nonparametric regression (local linear smoothers):**
  
  `sm.regression(x, y, h, design.mat = NA, model = "none", weights = W, group = NA, ...)`

- **Proportional hazards (Cox) regression:** add `+offset(-log(W))` as an extra covariate in `formula`:
  
  `coxph(formula, data=, weights, subset, na.action, ...)`

- **Remark:** standard errors must be updated (use e.g. bootstrapping)
Further use of the sampling probabilities $W$ (cont.)

- Local linear smoothing:

```r
> attach(AIDS.DT)
> W <- res$biasf^-1
> W <- W / sum(W)

> library(sm)
> res.LL <- sm.regression(AGE, X, h = 20, weights = W)
> res.LL2 <- sm.regression(AGE, X, h = 7.59, weights = W)

#naive estimator:
> res.LL2n <- sm.regression(AGE, X, h = 7.59)
```
Further use of the sampling probabilities $\tilde{W}$ (cont.)
Further use of the sampling probabilities $W$ (cont.)

- Proportional hazards (Cox) regression:

```r
> library(survival)
> coxph(Surv(X) ~ AGE + I(AGE^2) + offset(-log(W)),
+ data = AIDS.DT) -> rescox
> coef(rescox)
   AGE   I(AGE^2)
-0.0290770914 0.0002863366
```

- Same for proportional cause-specific hazards model (competing risks)

- Linear regression:

```r
> lm(X ~ AGE + I(AGE^2), data = AIDS.DT,
+ weights = W) -> reslm
> coef(reslm)
(Intercept)     AGE   I(AGE^2)
 36.11955323 1.48368927 -0.01785602
```
Three iterative algorithms implemented: efron.petrosian(), lynden(), shen()

Other packages:
- double.truncation by Takeshi Emura
- SurvTrunc by Lior Rennert
- DTDA.cif, DTDA.ni by José Carlos Soage

Advantages of DTDA:
- Sampling probabilities $G_n(X_i)$ returned (shen())
- Simple and obvious bootstrap available
- Confidence intervals for the truncation cdf, automatic plots
- Faster computational times
- Smoothing methods
Thanks for your attention!

Forthcoming (January 2022):

Contact: jacobouvn@gmail.com
jacobowebs.uvigo.es