iclogcondist: Nonparametric Estimation for a Log-concave Distribution Function with Interval-censored Data

Chi Wing Chu, Hok Kan Ling and Chaoyu Yuan

2024-11-26

1 Introduction

This manual aims to provide an introduction to the `iclogcondist` R package. The main function `ic_LCMLE` implements the algorithm in Chu, Ling & Yuan (2025+) for computing the maximum likelihood estimator (MLE) of a log-concave distribution function for the underlying failure time given mixed-case interval-censored data.

```
library(iclogcondist)
library(knitr) #for the kable() command
```

This package includes an interval-censored dataset `lgnm` for illustration, which contains 100 data points simulated from a lognormal distribution.

```
data(lgnm)
head(lgnm)
```

```
##     left  right
## 1 0.9551 2.6442
## 2 0.0000 1.1719
## 3 0.1898 1.6624
## 4 0.0000 1.8821
## 5 1.4852 6.7888
## 6 0.4558 4.2442
```

```
X <- lgnm
```

2 Nonparametric estimation for interval-censored data

2.1 Log-concave MLE of the underlying distribution function

The function `ic_LCMLE` computes the nonparametric MLE for a log-concave distribution function using an active set algorithm coupled with the iterative convex minorant (ICM) algorithm. The interval-censored data should be organized in the standard format, namely, a two-column matrix with each row containing the data interval $(L, R]$ for each observation. To evaluate the fitted MLE at any points, use `get_F_at_x`.

```
# Log-concave MLE
fit_LCMLE <- ic_LCMLE(X)

# Evaluate log-concave MLE at time = (0.1, 0.5, 1)
get_F_at_x(fit_LCMLE, x = c(0.1, 0.5, 1))
```

```
## [1] 0.0000000 0.1587697 0.4022400
```

2.2 Unconstrained MLE of the underlying distribution function

The unconstrained nonparametric MLE based on the function `ic_np()` in the R package `icenReg` is incorporated as `ic_UMLE()`.

```
# Unconstrained MLE
fit_UMLE <- ic_UMLE(X)

# Evaluate unconstrained MLE at time = (0.1, 0.5, 1)
get_F_at_x(fit_UMLE, x = c(0.1, 0.5, 1))
```

```
## [1] 0.0000000 0.1165246 0.4728116
```

2.3 Least concave majorant (LCM) of the unconstrained MLE

The `ic_LCM_UMLE` command computes the LCM of the unconstrained MLE.

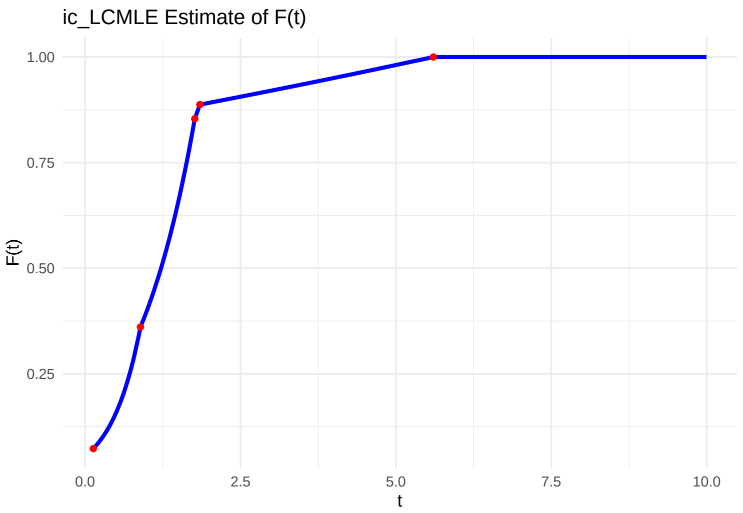
```
# LCM of the unconstrained MLE
fit_LCM_UMLE <- ic_LCM_UMLE(X)

# Evaluate the LCM of the unconstrained MLE at time = (0.1, 0.5, 1)
get_F_at_x(fit_LCM_UMLE, x = c(0.1, 0.5, 1))
```

```
## [1] 0.0000000 0.2302002 0.5317400
```

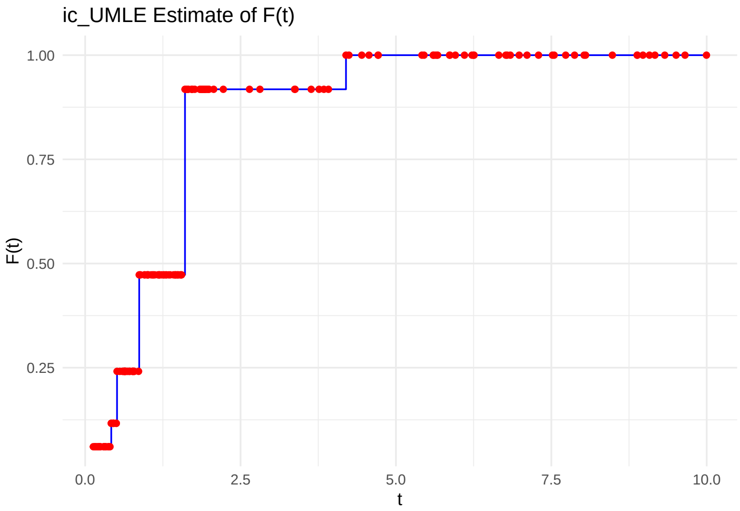
3 Visualization

3.1 Individual plot

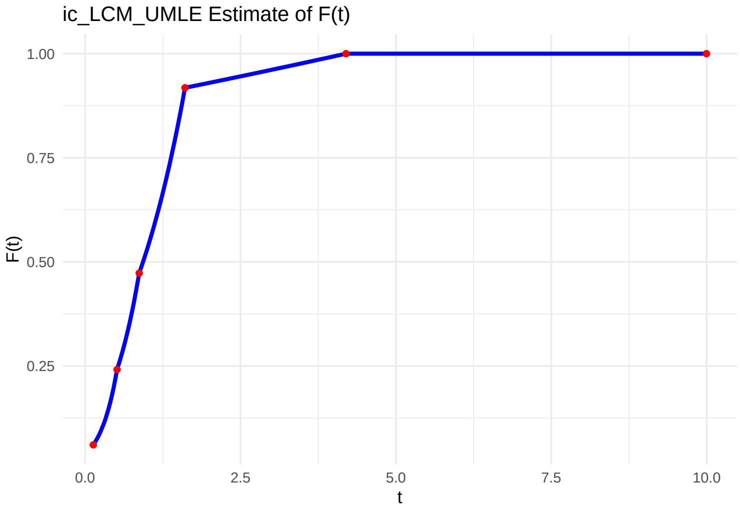
```
plot(fit_LCMLE)
```



```
plot(fit_UMLE)
```



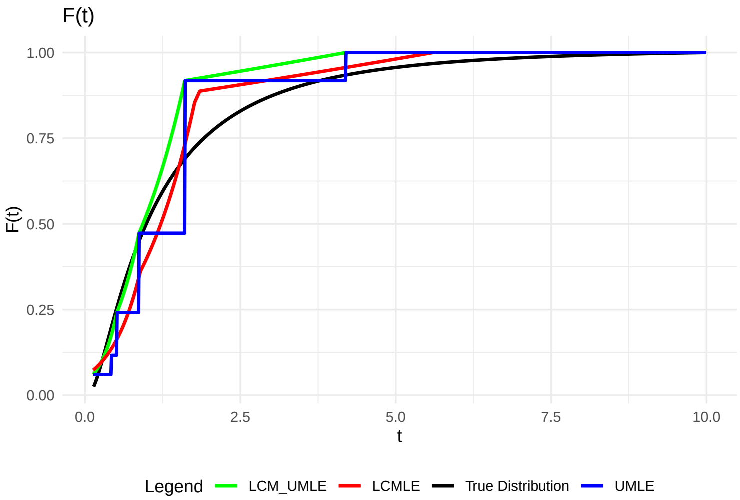
```
plot(fit_LCM_UMLE)
```



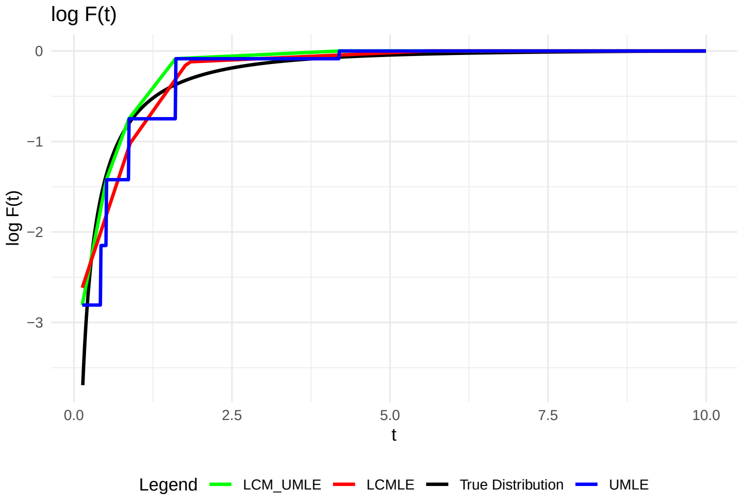
3.2 Overlaying plots for comparison

The `iclogcondist_visualization` command admits a list of estimates and overlays their plot for a better comparison, with an optional argument `true_dist` to add a reference distribution.

```
fit_list <- list(
  "UMLE" = fit_UMLE,
  "LCM_UMLE" = fit_LCM_UMLE,
  "LCMLE" = fit_LCMLE
)

true_dist_x <- seq(0.01, 10, 0.01)
true_dist_y <- plnorm(true_dist_x, 0, 1) / plnorm(10, 0, 1)
true_dist <- data.frame(x = true_dist_x, y = true_dist_y)

plots <- iclogcondist_visualization(X, fit_list = fit_list, true_dist = true_dist)

print(plots$F_plot)
```



```r
print(plots$logF_plot)
```



4 Simulation

The package provides a function `simulate_ic_data` to simulate interval-censored data; see `help(simulate_ic_data)` for details.

```r
# Simulate from a truncated Weibull distribution with shape parameter = 0.5 and
# scale parameter = 1 on [0, 5] under case 2 interval censoring
X <- simulate_ic_data(n = 1000, dist = "weibull", para1 = 0.5, para2 = 1,
                      upper_bound = 5, case = 2)
fit_LCMLE <- ic_LCMLE(X)
```

5 Real data illustration

This section illustrates the application of the nonparametric estimators to two real data examples.

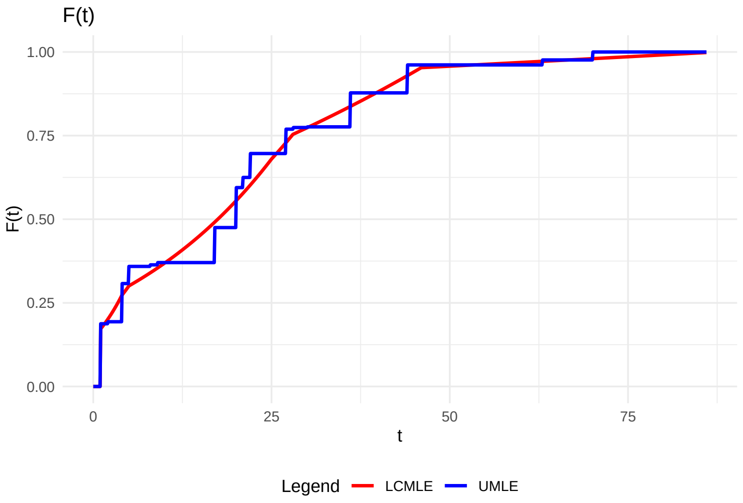
5.1 Hepatitis A Data

The hepatitisA dataset is available in the curstatCI R package; see help(hepatitisA) for more details. It is a special case of interval censored data, known as the current status data. The data is first organized into the standard format before fitting.

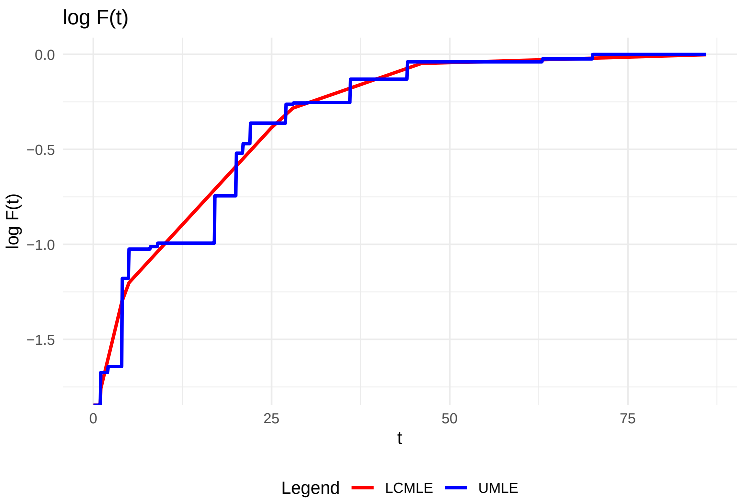
```r
library(curstatCI)
library(dplyr)
data(hepatitisA)
# organize the data into the standard format
X <- hepatitisA %>%
  rowwise() %>%
  do(data.frame(
    value = c(rep(0, .$freq1), rep(.$t, .$freq2 - .$freq1)),
    t = c(rep(.$t, .$freq1), rep(Inf, .$freq2 - .$freq1))
  )) %>%
  ungroup()
X <- as.matrix(X)

# Log-concave MLE
fit_LCMLE <- ic_LCMLE(X)

# Unconstrained MLE for reference
fit_UMLE <- ic_UMLE(X)
```

The unconstrained MLE and its logarithm appear to have a concave shape. The log-concave MLE closely resembles the shape of the unconstrained MLE while offering a smoother estimate.

```r
fit_list <- list(
  "UMLE" = fit_UMLE,
  "LCMLE" = fit_LCMLE
)

plots <- iclogcondist_visualization(X, range = c(0.01, 86), fit_list = fit_list)

# Plot of the estimated distribution function
plots$F_plot
```



```
# Plot of the logarithm of the estimates
plots$logF_plot
```



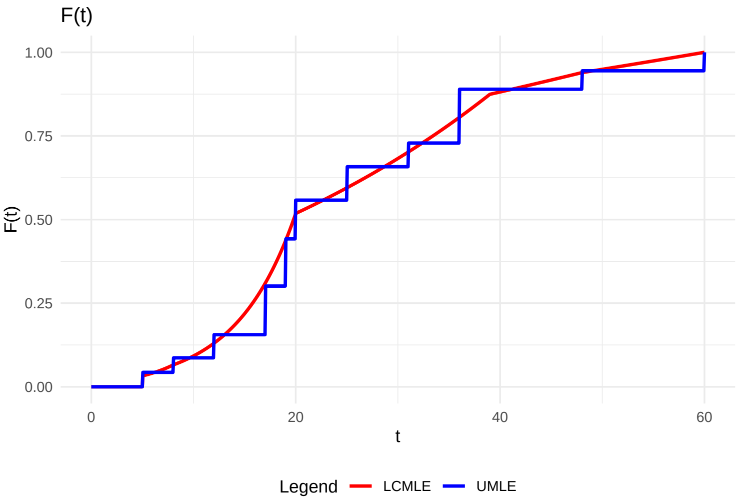
5.2 Breast Cosmesis Data

The bcos dataset is available in the dynsurv R package; see help(bcos) for more details. Here, we examine only the subset of patients receiving both radiotherapy and chemotherapy.

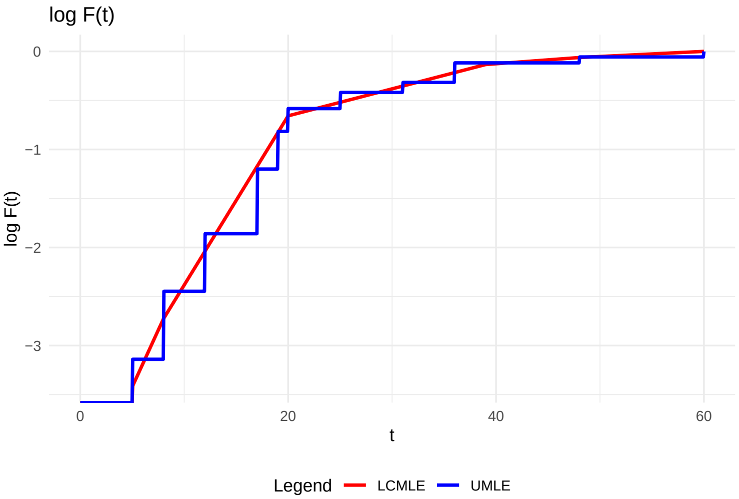
```
library(dynsurv)
data(bcos)
X <- bcos[bcos[,3] == "RadChem",c(1,2)]

# Log-concave MLE
fit_LCMLE <- ic_LCMLE(X)

# Unconstrained MLE for reference
fit_UMLE <- ic_UMLE(X)
```

Again, the log-concave MLE seems to fit well as it closely resembles the shape of the unconstrained MLE while providing a smoother estimate of the underlying distribution function.

```
fit_list <- list(
  "UMLE" = fit_UMLE,
  "LCMLE" = fit_LCMLE
)

plots <- iclogcondist_visualization(X, range = c(0, 60), fit_list = fit_list)

# Plot of the estimated distribution function
plots$F_plot
```



```
# Plot of the logarithm of the estimates
plots$logF_plot
```


log F(t)
log F(t)
t
Legend
LCMLE
UMLE