

A fast and stable algorithm for non-parametric maximum likelihood estimation of survival functions for left-truncated and interval-censored data

Zachary Waller

zwaller01@qub.ac.uk

Queen's University Belfast

Adele H. Marshall

Queen's University Belfast

Frank Kee

Queen's University Belfast

Felicity Lamrock

Queen's University Belfast

Research Article

Keywords:

Posted Date: August 9th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-5684468/v2>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A fast and stable algorithm for non-parametric
maximum likelihood estimation of survival
functions for left-truncated and interval-censored
data

Zachary Waller^{1*}, Adele H. Marshall^{1, 3}, Frank Kee²,
Felicity Lamrock¹

^{1*}School of Maths and Physics, Queen's University Belfast, Belfast, BT7
1NN, UK.

²Centre for Public Health in the School of Medicine, Dentistry and
Biomedical Sciences, Queen's University Belfast, Belfast, BT1 7NN, UK.

³Faculty of Business and IT, Ontario Tech University, 2000 Simcoe
Street North, Oshawa, Ontario, L1G 0C5, Canada.

*Corresponding author(s). E-mail(s): zwaller01@qub.ac.uk;

Abstract

We present a fast and conceptually appealing product-limit style EM algorithm for calculating the non-parametric maximum likelihood estimator (NPMLE) for left-truncated and interval-censored (LTIC) data. The reparameterization as a product-limit allows left-truncation and right-censoring to be accounted for analytically, while interval-censoring is still handled algorithmically. This reparameterization leads to a significant reduction in the number of iterations required to converge when applied to LTIC data compared to other EM algorithms. Combining this novel EM algorithm with a modified iterative convex minorant (ICM) step (Pan, 1999) allows the estimation of survival functions for larger and more complex LTIC data than was previously practical. We evaluate the performance of this algorithm through simulation and apply it to data from the Massachusetts Health Care Panel Study, demonstrating its effectiveness over other algorithms.

Keywords: Non-parametric maximum likelihood estimation, Survival analysis, Interval censoring, Product-limit, Left truncation, Expectation Maximization

1 Introduction

When events are only known to have occurred between assessment times, the resulting event-time data is interval-censored. This is common in the study of asymptomatic diseases, for example. Left-truncation occurs when the observation of an event is conditional on the event happening after some initial assessment time. Left-truncation may occur when age or calendar-time are of interest for assessing disease onset and only disease-free participants are enrolled in a study. Age may be appropriate in studies of, for example, dementia, while calendar-time more natural when tracking infectious diseases such as HIV. The combination of interval-censoring and left-truncation presents a computational challenge to investigators ([Pan and Chappell, 1998](#)).

The non-parametric maximum likelihood estimator (NPMLE) of survival functions for left-truncated and interval-censored (LTIC) data generalizes the well known [Kaplan and Meier \(1958\)](#) (KM) estimator beyond right-censored observations. However, unlike for right-censored data, the NPMLE generally has no analytical solution for LTIC data due to the more complex form of the likelihood and solutions must be found algorithmically.

[Turnbull \(1976\)](#) was the first to consider truncated and interval-censored data and calculated the NPMLE with a self-consistency algorithm. Self-consistency is a concept introduced by [Efron \(1967\)](#) and is a form of expectation-maximization (EM) algorithm ([Dempster et al., 1977](#)). Turnbull's EM algorithm ([Turnbull, 1976](#)) is simple to implement but can be very slow to converge ([Groeneboom and Wellner, 1992](#)) and so other algorithms have been proposed. The iterative-convex-minorant (ICM) ([Aragón and Eberly, 1992](#); [Groeneboom and Wellner, 1992](#)) algorithm and a hybrid

expectation-maximization iterative-convex-minorant (EM-ICM) (Wellner and Zhan, 1997) both converge more quickly than Turnbull’s algorithm for interval-censored data without truncation (IC data). For LTIC data, Pan and Chappell (1998) observed extremely slow convergence of Turnbull’s algorithm and developed gradient projection and modified ICM algorithms to improve performance. Hudgens (2005) instead recommended using a generic quasi-Newton optimization technique for LTIC data to avoid the slow convergence of Turnbull’s EM algorithm. More recently, Yu (2023) and Shen (2020) proposed new EM algorithms for LTIC data but gave no performance details or comparisons. Gao and Chan (2018) and Li et al. (2020) developed EM algorithms for semi-parametric estimation of regression models for LTIC data, using data-augmentation with Poisson variables to simplify computation. Applying these methods without covariates would allow the NPMLE to be calculated but this was not explored by either Gao and Chan (2018) or Li et al. (2020). A recent advance in the efficient computation of the NPMLE for IC data is a linear time-per-iteration implementation of Turnbull’s EM algorithm Anderson-Bergman (2017a). To our knowledge, this has not been extended to LTIC data.

Fast calculation of the NPMLE is important as there exists no closed form method for calculating confidence intervals for interval-censored data (with or without truncation). Re-sampling methods (Banerjee and Wellner, 2005) or likelihood-ratio tests (Sen and Xu, 2015) are necessary instead. These each require recalculating the NPMLE a many times, increasing the computational cost of handling interval-censored data. However, to our knowledge, no study has compared the currently existing methods of calculating the NPMLE for LTIC data and no freely available software implementation exists of any of them. This suggests a need for a systematic comparison of efficient implementations of the existing algorithms.

Interestingly, when the exact time of failure is known or right-censored, left-truncation can be accounted for trivially – the KM estimator (Kaplan and Meier, 1958)

is easily modified in this case and an analytic solution can be found (see [Lynden-Bell \(1971\)](#) and [Woodroffe \(1985\)](#), among others). The standard approach to left-truncated interval-censored data, Turnbull’s algorithm, handles truncation in a fundamentally different way to the KM estimator. Turnbull’s algorithm adds “ghost” observations to the starting sample to account for the fact that only some participants survived until their entry time. [Pan and Chappell \(1998\)](#) showed that these ghosts can have a detrimental effect on convergence times.

In this paper, we propose a new EM algorithm that we call the product-limit (PL) algorithm. This relies on a reparameterization of the likelihood for LTIC data that is similar to the KM estimator. Left-truncation and right-censoring times are known exactly and can be handled as with the KM estimator while interval-censoring must still be accounted for algorithmically. To our knowledge, a PL-style estimator for the interval-censored NPMLE has only been mentioned briefly by [Finkelstein \(1986\)](#) to calculate the baseline hazard of a proportional hazards model in the absence of truncation, but its computational use was not explored.

We also combine the PL algorithm with a modified ICM algorithm ([Pan and Chappell, 1998](#); [Pan, 1999](#)) to improve performance. This improvement contrasts the findings of [Pan and Chappell \(1998\)](#) who observed that, while a hybrid Turnbull EM-ICM algorithm is fast and accurate for IC data ([Anderson-Bergman, 2017a](#)), a similar hybrid algorithm for LTIC data actually increased convergence times beyond that of the modified ICM algorithm alone.

The PL-style estimator also avoids instances where the maximizer, as parameterized by Turnbull ([Turnbull, 1976](#)), does not exist. [Hudgens \(2005\)](#) showed that, for some LTIC datasets, the NPMLE of the survival function can prematurely drop to zero, leading to a maximizer that only exists in the limit. Reparameterized as a product-limit, the survival function can approach zero for periods of time without numerical issues, allowing a survival estimate conditioned on a later time to still be calculated.

The remainder of this paper is organized as follows. Section 2 introduces the LTIC model setup, data and notation. Section 3 derives the maximum likelihood estimation procedure and PL algorithm. Section 4 proposes a hybrid algorithm, combining the PL algorithm with a modified ICM step to improve performance. Section 5 explores the behavior of the novel methods in a simulation study and compares them with efficiently implemented algorithms from the literature. Section 6 applies the method to data from the Massachusetts Health Care Panel Study. Section 7 concludes with a discussion.

2 Notation and Likelihood

2.1 Likelihood and Inner-intervals

Let X be the survival time and T the time of truncation, where $X, T \geq 0$. Because of interval-censoring, the value of X is only known to lie within the interval $A = (L, R]$ where $L < R$. Due to truncation, X is only observed if $X \geq T$. Letting $B = [T, \infty)$ and $A \cap B = (L \vee T-, R]$ then $X \in A \cap B$.

Let the survival function be $S(t) = P(X > t)$ where $t \geq 0$. Conditional on the assessment and truncation times, the likelihood for observations $i = 1, 2, \dots, n$ is

$$\mathcal{L} = \prod_{i=1}^n \frac{P(X \in A_i \cap B_i)}{P(X \in B_i)} = \prod_{i=1}^n \frac{S(L_i \vee T_i-) - S(R_i)}{S(T_i-)} . \quad (1)$$

In general, the end-points of each A_i are taken from a set of $K_i > 0$ inspection times $0 < Y_{i,1} < Y_{i,2} < \dots < Y_{i,K_i}$. We assume that, given K , each of X , $(Y_1, \dots, Y_K)$ and T are independent. This is the mixed-case LTIC approach from Yu (2023) and, in the absence of truncation, reduces to the mixed-case interval-censoring scheme given by

Schick and Yu (2000). The observations (T, L, R) are then

$$(T, L, R) = \begin{cases} (T, -\infty, Y_1) & \text{if } T \leq X \leq Y_1 \\ (T, Y_{k-1}, Y_k) & \text{if } Y_{k-1} < X \leq Y_k, k = 2, \dots, K \\ (T, Y_K, \infty) & \text{if } X > Y_K. \end{cases}$$

Let $\hat{S}(t)$ be the NPMLE of $S(t)$. It is well known that $\hat{S}(t)$ is piece-wise constant and only assigns weight to a subset of the intervals present in the data (Peto, 1973; Turnbull, 1976; Frydman, 1994; Alioum and Commenges, 1996). For LTIC data, Yu (2023) established the correct form of these intervals (so-called inner-intervals, but also referred to as Turnbull-intervals or maximal cliques) and we briefly characterize them here.

The set of inner-intervals on which $\hat{S}(t)$ changes can be induced from $A_i \cap B_i$ and $[T_{(1)}, T_i]$ for $1 \leq i \leq n$ where $T_{(1)} = \min_i(T_i)$ (see Lemma 2.1 in Yu (2023)). Let $\mathcal{L} = \bigcup_{i=1}^n \{W_i\} \cup \{T_{(1)}-\}$ and $\mathcal{R} = \bigcup_{i=1}^n \{R_i\} \cup \bigcup_{i=1}^n \{T_i-\}$, where $W_i = L_i \vee T_i-$. The set of inner-intervals $\mathcal{Q}$ is the union of the disjoint intervals whose left end-points lie in $\mathcal{L}$ and right end-points in $\mathcal{R}$ with no other elements of $\mathcal{L}$ or $\mathcal{R}$ in between. Let $\mathcal{Q}$ be the set of inner-intervals, ordered by increasing left end-point and m the number of such inner-intervals. Let $\mathcal{Q}_j$ be the j th inner-interval, with left end-point l_j and right end-point r_j such that $\mathcal{Q}_j = (l_j, r_j]$ for $j = 1, \dots, m$. For convenience we also add a zeroth inner-interval $\mathcal{Q}_0 = [-\infty, -\infty]$.

The likelihood is independent of the behavior of $S(\cdot)$ within the inner-intervals and so, to simplify notation and plotting, we let $\hat{S}(t)$ change only at the right end-points of each inner-interval, r_j for $j = 1, \dots, m$.

Letting $s_j = S(l_j) - S(r_j)$, the likelihood in Equation (1) can then be written as in Turnbull (1976) as

$$\mathcal{L} = \prod_{i=1}^n \frac{\sum_{j=1}^m \alpha_{ij} s_j}{\sum_{j=1}^m \beta_{ij} s_j} \quad (2)$$

where $\alpha_{ij} = I(\mathcal{Q}_j \subseteq A_i \cap B_i)$, $\beta_{ij} = I(\mathcal{Q}_j \subseteq B_i)$ and $I(\cdot)$ is the usual indicator function. From this Turnbull's algorithm can be derived (see Section 1.1 of the Supplementary Material for details on the algorithm).

2.2 Reparameterization

Reparameterizing the likelihood in Equation (2) allows left-truncation and right-censoring to be accounted for easily. Let $h_j = \frac{S(l_j) - S(r_j)}{S(l_j)} = P(X \in \mathcal{Q}_j | X > l_j)$ for $j = 1, \dots, m$, $h_0 = 0$, and $\mathbf{h} = (h_0, h_1, \dots, h_m)$. We call $\mathbf{h}$ the “interval hazard” because it is the probability of an event occurring within an inner-interval conditional on survival up to that inner-interval. This is analogous to the interpretation of the usual hazard rate, despite $\mathbf{h}$ not being strictly a hazard function. The unconditional probability of an event within $\mathcal{Q}_j$ is $S(l_j) - S(r_j) = S(l_j)h_j$.

The estimator for $S(r_j)$ can be written in terms of $\mathbf{h}$ as a product similar to the KM estimator for right-censored data

$$S(r_j) = \prod_{k=0}^j (1 - h_k).$$

To simplify notation in the sequel, we introduce indices for the observed times L, R, T with respect to the inner-intervals: $L_i^* = \max\{j | r_j \leq L_i\}$, $R_i^* = \max\{j | r_j \leq R_i\}$, $T_i^* = \max\{j | r_j \leq T_i\}$ and $W_i^* = \max(L_i^*, T_i^*)$. The likelihood $\mathcal{L}$ from Equation (1) can then be written in terms of $\mathbf{h}$ as

$$\mathcal{L} = \prod_{i=1}^n \frac{\prod_{j=0}^{L_i^*} (1 - h_j) - \prod_{j=0}^{R_i^*} (1 - h_j)}{\prod_{j=0}^{T_i^*} (1 - h_j)}$$

which simplifies to

$$\mathcal{L} = \prod_{i=1}^n \left(\prod_{j=T_i^*}^{W_i^*} (1 - h_j) - \prod_{k=T_i^*}^{R_i^*} (1 - h_k) \right).$$

The difference of products in the large parentheses can be rewritten as a sum such that the log-likelihood becomes

$$\log \mathcal{L} = \sum_{i=1}^n \log \left(\sum_{j=W_i^*}^{R_i^*} h_j \prod_{k=T_i^*}^{j-1} (1 - h_k) \right). \quad (3)$$

For right-censored observations, the likelihood contribution is just $S(L_i)/S(T_i) = \prod_{j=T_i^*}^{L_i^*} (1 - h_j)$. Ordering the observations such that the first n_0 are right-censored gives the final log-likelihood

$$\log \mathcal{L} = \sum_{i=1}^{n_0} \sum_{j=T_i^*}^{L_i^*} \log(1 - h_j) + \sum_{i=n_0+1}^n \log \sum_{j=W_i^*}^{R_i^*} h_j \prod_{k=T_i^*}^{j-1} (1 - h_k). \quad (4)$$

Remark 1. If all events are either right-censored or known to have occurred in only one inner-interval then the sum over j in Equation (3) disappears and the likelihood is almost identical to that used to derive the familiar KM estimator (under left-truncation) and has an explicit solution. This could happen if, for example, all assessment times for all participants aligned.

Remark 2. Note that, for an arbitrary inner-interval $\mathcal{Q}_j = (l_j, r_j]$, h_j should not be confused with the change in cumulative hazard over the same time interval. The usual (instantaneous) hazard is defined as $\lambda(t) \equiv \lim_{\Delta t \rightarrow 0} \frac{P(t \leq X < t + \Delta t | X \geq t)}{\Delta t}$ and the change in cumulative hazard $\Delta\Lambda(l_j, r_j)$ over $\mathcal{Q}_j$ is $\Delta\Lambda(l_j, r_j) \equiv \int_{l_j}^{r_j} \lambda(t) dt$. By writing h_j as

$$h_j = \frac{1}{S(l_j)} \int_{l_j}^{r_j} S(t) \lambda(t) dt$$

it is clear that h_j will underestimate $\Delta\Lambda(l_j, r_j)$ because $S(t)/S(l_j) \leq 1$. Despite this, [Sun \(2006, Chap. 3.5\)](#) and [Frydman \(1992\)](#) attempted to estimate $\Delta\Lambda(l_j, r_j)$ in this way (though without using the “interval hazard” concept). [Frydman and Liu \(2013\)](#) later noticed that this approach severely underestimates the change in cumulative hazard and instead derived (under a different parameterization) the NPMLE $\Delta\hat{\Lambda}(l_j, r_j)$ as

$$\Delta\hat{\Lambda}(l_j, r_j) = -\ln(1 - \hat{h}_j)$$

where $\hat{h}_j$ is the NPMLE of h_j .

3 Maximization

Our goal is to maximize Equation (4) with respect to $\mathbf{h}$, subject to the constraints $0 \leq h_j \leq 1$ for $1 \leq j \leq m$. The Lagrangian $\mathcal{H}$ for this maximization problem is

$$\mathcal{H} = \log \mathcal{L} + \sum_{j=1}^m a_j h_j + \sum_{j=1}^m b_j (1 - h_j) \quad (5)$$

where $a_j, b_j \geq 0$, $a_j h_j = 0$ and $b_j (1 - h_j) = 0$.

Letting $\hat{\mathbf{h}}$ denote the NPMLE of $\mathbf{h}$, $\hat{S}(t)$ can be calculated from

$$\hat{S}(t) = \prod_{j=0}^{R_t^*} (1 - \hat{h}_j) \quad (6)$$

where $R_t^* = \max\{j | r_j \leq t\}$.

Lemma 1. *The maximum likelihood estimate of h_m is $\hat{h}_m = 1$.*

Proof. From inspection of Equation (3) it is clear that h_m will only appear in the likelihood as h_m and never as $(1 - h_m)$. Therefore, to maximize the likelihood, h_m should be made as large as possible subject to the constraints on Equation (5), i.e. $\hat{h}_m = 1$. $\square$

Theorem 1. *The values of $\hat{\mathbf{h}}$ that maximize Equation (5) are the solutions to the set of equations*

$$\hat{h}_j = \frac{d_j}{\mathcal{R}_j} \quad (7)$$

for $1 \leq j \leq m$, where

$$d_j = \sum_{i=n_0+1}^n I(W_i^* \leq r_j \leq R_i^*) \frac{h_j \prod_{k=T_i^*}^{j-1} (1 - h_k)}{\sum_{k=W_i^*}^{R_i^*} h_k \prod_{l=T_i^*}^{k-1} (1 - h_l)} \quad (8)$$

and

$$\mathcal{R}_j = \sum_{i=1}^{n_0} I(T_i^* \leq r_j \leq L_i^*) + \sum_{k \geq j}^m d_k. \quad (9)$$

Proof. For the proof see Appendix A where it is shown that Equations (8) and (9) correspond to the score equations for h_m of the Lagrangian in Equation (5). $\square$

Both d_j and $\mathcal{R}_j$ have direct interpretations: d_j is the expected number of events within the time interval $\mathcal{Q}_j$ and $\mathcal{R}_j$ is the expected number of events occurring after time l_j , where the expectations are taken with respect to the observed data. In this sense, $\mathcal{R}_j$ acts as a risk-set that observations enter after their truncation time and leave after their right-censoring or event times.

3.1 Algorithm

Equation (7) suggests an iterative algorithm for calculating $\hat{\mathbf{h}}$ and therefore $\hat{S}(t)$. We call this the product-limit (PL) algorithm.

At any iteration $g = 1, 2, \dots$, the current estimates of $h_j^{(g)}$, for $1 \leq j < m - 1$, are updated by

$$h_j^{(g+1)} = \frac{d_j^{(g)}}{\mathcal{R}_j^0 - D_{j-1}^{(g)}} \quad (10)$$

where

$$d_j^{(g)} = h_j^{(g)} S_{j-1}^{(g)} \sum_{i=n_0+1}^n \frac{I(W_i^* \leq r_j \leq R_i^*)}{S_{W_i^*}^{(g)} - S_{R_i^*}^{(g)}}, \quad (11)$$

with $d_0^{(g)} = 0$,

$$D_j^{(g)} = \sum_{k=0}^j d_k^{(g)}, \quad (12)$$

$$S_j^{(g)} = \prod_{k=1}^j (1 - h_k^{(g)}) \quad (13)$$

with $S_0 = 1$ and

$$\mathcal{R}_j^0 = \sum_{i=1}^{n_0} I(T_i^* \leq r_j \leq L_i^*) + \sum_{i=n_0+1}^n I(T_i^* \leq r_j) \quad (14)$$

which is only calculated once, at the start of the algorithm.

The algorithm is halted when $|\log \mathcal{L}^{(g+1)} - \log \mathcal{L}^{(g)}| < \epsilon$ where ϵ is the desired tolerance.

The algorithm is summarized in Algorithm 1 and the derivation is given in Appendix B.

3.2 Implementation Details

Notice that $\mathcal{R}_j^0$ in Equation (14) acts as an initial risk-set estimate that only accounts for left-truncation and right-censoring. After the initial calculation of $\mathcal{R}_j^0$, the rest of the algorithm handles the effects of interval-censoring only.

Anderson-Bergman (2017a) gave an efficient implementations of Turnbull's EM algorithm for IC that requires $O(n)$ operations per iteration. The same approach can be applied here by noticing that each observation i contributes $1/(S_{W_i^*} - S_{R_i^*})$ to Equation (11) for all j where $W_i^* \leq r_j \leq R_i^*$. This is a contiguous block of inner-intervals and a recursive sum can be calculated where each observation contributes at most twice – once adding to it and again taking from it.

Algorithm 1 The PL-EM algorithm

Inputs:

the right end-points of the inner-inters $\mathbf{r} = (r_0, \dots, r_m)$,
 observation indices T_i^*, L_i^*, R_i^* for $i = 1, \dots, n$, initial estimate of
 $\mathbf{h}$, tolerance ϵ , maximum iterations $g_{\max}$

Outputs:

NPMLE estimate of $\mathbf{h}$

Initialize:

$h_0 \leftarrow 0, h_m \leftarrow 1, g \leftarrow 0, \log \mathcal{L}^{(1)} \leftarrow 0, \log \mathcal{L}^{(0)} \leftarrow -\infty$

For $j = 1, \dots, m$ calculate $S_j^{(0)}$ from Equation (13) and $\mathcal{R}_j^0$ from
 Equation (14)

while $|\log \mathcal{L}^{(g+1)} - \log \mathcal{L}^{(g)}| \geq \epsilon$ & $g \leq g_{\max}$ **do**

for $j = 1, \dots, m - 1$ **do**

 Calculate $d_j^{(g+1)}$ by Equation (11)

 Calculate $D_j^{(g+1)}$ by Equation (12)

 Calculate $h_j^{(g+1)}$ by Equation (10)

end for

 Calculate $\log \mathcal{L}^{(g+1)}$ by Equation (1)

$g \leftarrow g + 1$

end while

In the presence of heavy truncation, $\mathcal{R}_j^0$ may be small at early times, and it is possible that $\hat{h}_j = 1$ for $j < m$. This results in the estimate of the survival function to dropping to zero and leading to non-identifiable values of d_k for $k > j$ from Equation (11). To avoid this, we suggest introducing a small buffer η , such, for any j , if $h_j = 1$ it is set to $h_j < 1 - \eta$. In the simulations below we have found a buffer of 1×10^{-10} to give suitable accuracy and stability. A robust approach that avoids the need for a buffer is to take

$$d_j^{(g)} = \sum_{i=n_0+1}^n \frac{C_{ij}^{(g)}}{\sum_{k=1}^m C_{ik}^{(g)}} \quad (15)$$

where

$$C_{ij}^{(g)} = I(W_i^* \leq r_j \leq R_i^*) h_j^{(g)} \prod_{k=T_i^*}^{j-1} (1 - h_k^{(g)}). \quad (16)$$

However, this requires far more operations per iteration and leads to, at worst $O(nm^2)$ running time. Aside from the reduced performance, we also found the difference in accuracy to be very minor when a small enough buffer was chosen.

Hudgens (2005) gave conditions for times when the risk-set reduces to zero for LTIC data, but assessing these conditions isn't straightforward. Our algorithm is generally fast enough that it is simple to estimate the NPMLE and find any instances where some $\hat{h}_j = 1$. In these cases, while there is a maximizer to Equation (4), Hudgens (2005) showed that, under Turnbull's parameterization, a maximizer to Equation (2) does not exist.

In data with unusual truncation patterns, the risk-set may become very small multiple times over the whole observation period, and using the above buffer method may experience numerical issues arising from very low survival function estimates at late times. One method to avoid calculating Equations (15) and (16) in these cases would be to split the survival function any time $\hat{h}_j = 1$, for some j , and calculate a conditional survival for $k > j$. This conditional survival function would include only observations where $W_i^* \geq j$ and all observations with $W_i^* < j$ would be removed from $\mathcal{R}_k^0$. We did not experiment much with this technique as the issue was never encountered in our simulation or data illustration studies.

If the initial values of $h_j^{(0)}$ for $1 \leq j < m$ are set too high, Equation (13) may become zero at $j < m$ when m is large, due to the limits of floating-point precision. Because the EM algorithm cannot change a parameter once it is 0, the algorithm would never reach the maximum likelihood in this case. To avoid this, we suggest initial values of either $h_j^{(0)} = 1/m$ for $j = 1, \dots, m$ or setting $h_j^{(0)}$ such that $S_{j-1}^{(0)} - S_j^{(0)} = 1/m$.

3.3 Convergence Criteria

In the following simulations and examples we have used the difference in log-likelihood between two iterations to check convergence – this is simple to implement and gives acceptable performance in our experience.

Other authors (for example Yu (2023), Shen (2020) and, in a related problem for semi-parametric regression, Zeng et al. (2016)) have suggested using the change in

parameter estimates, which is very simple to implement but may lead to premature halting and makes comparisons between different methods more difficult.

A more complete approach is to use the Karush-Kuhn-Tucker (KKT) conditions (Kuhn and Tucker, 1951) which have been applied to the interval censoring problem (without truncation) by Gentleman and Geyer (1994). These conditions are given in Section 2 of the Supplementary Material for our parameterization, but we found similar performance to using the difference in log-likelihood.

4 Hybrid Algorithm

For IC data it is well known that a hybrid algorithm that alternates between steps of the Turnbull EM algorithm with the ICM algorithm improves performance compared to each algorithm alone (Wellner and Zhan, 1997). In this section, we propose a hybrid PL-ICM algorithm that alternates PL algorithm steps and with those of an ICM algorithm modified to work with LTIC data. Pan and Chappell (1998) (see also Pan (1999)) were the first to extend the ICM algorithm to LTIC data, but found that a hybrid EM-ICM algorithm (with Turnbull’s algorithm for the EM step) was still subject to slow convergence. As we will show in Sections 5 and 6, the hybrid PL-ICM algorithm performs better than either of the PL or ICM algorithms individually.

Our modified ICM algorithm is based on that of Pan (1999) and, because it reduces to the usual ICM algorithm when applied to IC data, we simply refer to the ICM algorithm.

The ICM algorithm works as follows. First, the survival function is written in terms of the cumulative hazard function $\Lambda(t)$ as $S(t) = e^{-\Lambda(t)}$ where $\Lambda(t)$ is positive and monotonic. The likelihood in Equation (1) can then be written as

$$\mathcal{L} = \prod_{i=1}^n \left[e^{-\Lambda(L_i \vee T_i-) + \Lambda(T_i-)} - e^{-\Lambda(R_i) + \Lambda(T_i-)} \right]. \quad (17)$$

This is slightly different to the likelihood considered by [Pan \(1999\)](#) who accounted for truncation by including $e^{-\Lambda(T_i-)}$ in the denominator. By including $\Lambda(T_i-)$ within the exponent, this likelihood is more stable in the presence of heavy truncation where $\Lambda(t)$ can become very large. Let $\Lambda_j = \Lambda(r_j)$ and $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1, \dots, \Lambda_m)$ with $\Lambda_0 = 0$, $\Lambda_m = \infty$, and $\mathbf{\Lambda}^{(g)}$ be the estimate of $\mathbf{\Lambda}$ at iteration g .

The ICM algorithm first takes a Newton-like step using the gradient and inverse Hessian of the likelihood. Importantly, the Hessian of the likelihood in Equation (17) can be well approximated by its diagonal ([Groeneboom and Wellner, 1992](#); [Pan and Chappell, 1998](#)). The j th element of the inverse of the Hessian diagonal is $1/\frac{\partial^2 \log \mathcal{L}}{\partial \Lambda_j^2}$. For LTIC data, the Hessian may not be negative-definite because $\frac{\partial^2 \log \mathcal{L}}{\partial \Lambda(T_i)^2} = 0$. [Pan \(1999\)](#) added a Levenberg-Marquardt adjustment ([Fletcher, 2000](#), Sec. 5.2) to the usual ICM algorithm ensure that finite steps are taken in the Newton-like step. To do so, any non-negative elements of the Hessian diagonal are replaced with a small negative number ν , where $\nu = -1 \times 10^{-9}$ in our simulation studies below. For $j = 1, \dots, m-1$, let

$$H_j^{(g)} = \min \left(\nu, \frac{\partial^2 \log \mathcal{L}}{\partial \Lambda_j^{(g)2}} \right) \quad (18)$$

and calculate $\tilde{\mathbf{\Lambda}} = (\tilde{\Lambda}_0, \dots, \tilde{\Lambda}_m)$, an update of $\mathbf{\Lambda}^{(g)}$, in a direction that increases the likelihood using

$$\tilde{\Lambda}_j = \Lambda_j^{(g)} - \frac{\partial \log \mathcal{L}}{\partial \Lambda_j^{(g)}} / H_j. \quad (19)$$

After this update, $\tilde{\mathbf{\Lambda}}$ need not be monotonic and so a weighted pool-adjacent-violators algorithm (WPAVA) ([Busing, 2022](#)) is used to calculate a further update $\mathbf{\Lambda}^+ = (\Lambda_0^+, \dots, \Lambda_m^+)$. The WPAVA calculates the $\mathbf{\Lambda}^+$ that minimizes

$$\sum_{j=1}^m w_j (\Lambda_j^+ - \tilde{\Lambda}_j)^2$$

subject to $\Lambda_0^+ \leq \Lambda_1^+ \leq \dots \leq \Lambda_m^+$ where

$$w_j = \frac{1}{2}H_j. \quad (20)$$

After the WPAVA step if, for any j , $\Lambda_j^+ < 0$ then set $\Lambda_j^+ = 0$.

Finally, half-stepping is employed to ensure that the likelihood increases at each step. To do this, $\Lambda^{(g+1)}$ is calculated iteratively from

$$\Lambda_j^{(g+1)} = \Lambda_j^{(g)} + 2^{-\gamma} \left(\Lambda_j^+ - \Lambda_j^{(g)} \right) \quad (21)$$

for $j = 1, \dots, m-1$, where $\gamma = 0, 1, 2, \dots$ is incrementally increased by one until $\log \mathcal{L}^{(g+1)} \geq \log \mathcal{L}^{(g)}$. In our implementation, we limit γ to four after which another iteration of the whole ICM algorithm is taken. The algorithm is halted when $|\log \mathcal{L}^{(g+1)} - \log \mathcal{L}^{(g)}| < \epsilon$ where ϵ is the desired tolerance.

The ICM algorithm is summarized in Algorithm 2 and further details of the log-likelihood derivatives are given in Appendix C.

The hybrid PL-ICM algorithm alternates between the inner-loops of Algorithms 1 and 2. This requires converting between $\mathbf{h}$ and Λ when switching between algorithms using

$$h_j = 1 - \exp(\Lambda_j - \Lambda_{j+1}).$$

Following Anderson-Bergman (2017a) who implemented an efficient EM-ICM algorithm for IC data, we take 10 PL steps to each ICM step due to the comparatively low computational cost of PL steps.

5 Simulation study

To evaluate the performance of our proposed PL and PL-ICM algorithms, we compared them to nine other algorithms from the literature. Three are self-consistent

Algorithm 2 The ICM algorithm

Inputs:

Observations indices T_i^*, L_i^*, R_i^* for $i = 1, \dots, n$, initial estimates of Λ , tolerance ϵ , maximum allowed iterations $g_{\max}$

Outputs:

NPMLE estimate of Λ

Initialize:

$\Lambda_0 \leftarrow 0, \Lambda_m \leftarrow \infty, g \leftarrow 0, \log \mathcal{L}^{(1)} \leftarrow 0, \log \mathcal{L}^{(0)} \leftarrow -\infty$

while $|\log \mathcal{L}^{(g+1)} - \log \mathcal{L}^{(g)}| \geq \epsilon$ & $g \leq g_{\max}$ **do**

for $j = 1, \dots, m$ **do**

 Calculate $\tilde{\Lambda}_j$ by Equation (19)

 Calculate w_j by Equation (20)

end for

 Use a WPAVA to calculate Λ_j^+ for $j = 1, \dots, m$ such that $\sum_{j=1}^m w_j (\Lambda_j^+ - \tilde{\Lambda}_j)^2$ is minimized and $\Lambda_0^+ \leq \Lambda_1^+ \leq \dots \leq \Lambda_m^+$.

$\gamma \leftarrow 0, \log \mathcal{L}^{(g+1)} \leftarrow \log \mathcal{L}^+$

while $\log \mathcal{L}^{(g+1)} < \log \mathcal{L}^{(g)}$ & $\gamma < 5$ **do**

for $j = 1, \dots, m$ **do**

$\Lambda_j^{(g+1)} \leftarrow \Lambda_j^{(g)} + 2^{-\gamma} (\Lambda_j^+ - \Lambda_j^{(g)})$

end for

$\gamma \leftarrow \gamma + 1$

 Calculate $\log \mathcal{L}^{(g+1)}$

end while

$g \leftarrow g + 1$

end while

EM algorithms, from [Turnbull \(1976\)](#), [Shen \(2020\)](#), [Yu \(2023\)](#) (which we refer to as the Turnbull, Shen, Yu algorithms respectively). A fourth EM algorithm was adapted from a semi-parametric estimator for LTIC data from [Gao and Chan \(2018\)](#) and [Li et al. \(2020\)](#) by ignoring the covariates. This is reminiscent of the estimator for calculating the baseline survival suggested by Breslow ([Cox, 1972](#)) so we refer to it as the Breslow estimator. Next is a quasi-Newton algorithm based on a suggestion by [Hudgens \(2005\)](#) (although their exact procedure isn't given). We also implemented an ICM algorithm from Section 4, and hybrid EM-ICM algorithms using the Yu, Turnbull and Breslow algorithms in the EM step (referred to as Yu-ICM, Turnbull-ICM and Breslow-ICM respectively). To our knowledge, the Turnbull-ICM algorithm is the only hybrid-algorithm to have been considered in the literature.

Algorithm Type	Method
EM	PL
	Breslow (Gao and Chan, 2018 ; Li et al., 2020)
	Shen (Shen, 2020)
	Turnbull (Turnbull, 1976)
	Yu (Yu, 2023)
ICM	ICM (Groeneboom and Wellner, 1992 ; Pan, 1999)
Hybrid	PL-ICM
	Breslow-ICM
	Turnbull-ICM
	Yu-ICM
Other	Quasi-Newton (Hudgens, 2005)

Table 1 Summary of algorithms used in the simulation study.

To summarize, we tested eleven algorithms: the PL, PL-ICM, Breslow, Breslow-ICM, Shen, Turnbull, Turnbull-ICM, Yu, Yu-ICM, ICM, and quasi-Newton. A summary of the algorithms is given in Table 5 and details of each are given in Section 1 in the Supplementary Material.

Computations were performed on a 2020 M1 Apple MacBook Pro with pre-processing carried out in the R programming language ([R Core Team, 2024](#)) using R version 4.3.3 (2024-02-29) and all algorithms except the quasi-Newton written from scratch in C++. The quasi-Newton method used the in-built `optim()` function in R.

We applied the $O(n)$ per-iteration implementation trick described in Section 3.2 and used by [Anderson-Bergman \(2017a\)](#) to each of the EM algorithms. To our knowledge, this is the first time $O(n)$ per-iteration EM algorithms have been implemented for calculating the NPMLE of the survival function for LTIC data.

Timings were taken from the start of each algorithm after pre-processing to calculate the inner-intervals. The difference in timings needed to pre-calculate $\mathcal{R}^0$ for the PL and PL-ICM algorithms compared to the other methods were too small to affect the results. For the hybrid-algorithms, we counted both EM and ICM iterations (this contrasts with, for example, the R package `icenReg` ([Anderson-Bergman, 2017b](#)) which classes one iteration as 10 EM steps and one ICM step). Each algorithm was stopped when the change in log-likelihood estimate was less than 10^{-5} or when the number

of iterations exceeded 100,000, whichever was sooner. Algorithms were also prematurely stopped if they resulted in nonsense estimates with a log-likelihood of $-\infty$ (e.g. non-monotonically decreasing survival functions, or survival estimates of greater than one or less than zero). The results from these runs contributed log-likelihood results recorded, but not to the run time or number of iterations.

The initial parameter values for each algorithm were set to such that $S_j^{(0)} - S_{j+1}^{(0)} = 1/m$ for $j = 0, \dots, m-1$. This ensures that all methods start at the same log-likelihood. Aside from the potentially problematic initial values mentioned in Section 3.2, we found that most initial parameter values gave very similar convergence behavior for each algorithm.

5.1 Progressively heavier left-truncation

To highlight the effect of truncation on convergence time and stability, we simulated three experiments with progressively heavier truncation, but with the same number of observations.

Each experiment consisted of 100 repetitions of the following procedure. First, an event time was drawn from an exponential distribution $X \sim \text{Exp}(1)$ and then two assessment times generated, the first from a uniform distribution on $Y_1 \sim U(0, 4)$ and the second from $Y_2 = Y_1 + 0.5$. Next, a truncation time was drawn from a uniform distribution on $T = U(0, w)$ with w fixed for each experiment. All observations where $X < T$ were discarded, resulting in a random sample size. The inner-intervals were determined using the procedure outlined in Section 2, and survival curves for each simulated dataset were estimated with all eleven algorithms.

The values of w chosen were 0, 0.6 and 1.6 to give approximately 0%, 25% and 50% of observations truncated respectively. To reduce the possibility of sample size affecting convergence time, the sample sizes for each experiment were chosen such that after truncation around 1000 observations remained. In Section 2 of the Supplementary

Material we repeat these simulations with sample sizes after truncation of 500 and 2000. The conclusions from these are the same as for the simulation presented here.

Method	log-likelihood	Iterations	CPU time (s)	% converged
Breslow	-825.00(25.75)	3,696(597)	0.237(0.042)	100%
PL	-824.99(25.76)	2,534(429)	0.108(0.020)	100%
Shen	-824.99(25.75)	2,794(446)	0.146(0.024)	100%
Turnbull	-824.99(25.75)	2,794(446)	0.158(0.027)	100%
Yu	-824.99(25.75)	2,794(446)	0.128(0.021)	100%
Breslow-ICM	-824.98(25.76)	88(11)	0.004(0.001)	100%
PL-ICM	-824.98(25.76)	88(22)	0.004(0.001)	100%
Turnbull-ICM	-824.98(25.76)	99(22)	0.005(0.001)	100%
Yu-ICM	-824.98(25.76)	99(22)	0.005(0.001)	100%
ICM	-824.98(25.76)	26(4)	0.004(0.001)	100%
Quasi-Newton	-825.16(25.76)	111(3)	0.623(0.052)	100%

Table 2 Median and inter-quartile range (in parentheses) of log-likelihood, run times (in iterations and seconds) and percentage converged before 100,000 iterations or with finite log-likelihood for the 0% truncated experiment. The proposed PL and PL-ICM methods are highlighted in bold.

The results for 0% truncation are shown in Table 2. For this experiment, the ICM and all of the EM-ICM algorithms are similar in terms of speed and accuracy. Each of the EM-ICM algorithms took fewer than 10 ICM steps, with the rest being EM iterations. The low computational cost of the EM steps explains why the EM-ICM algorithms matched the ICM algorithm in terms of speed while taking more iterations.

The EM algorithms perform similarly in terms of accuracy, but the PL algorithm is the fastest in terms of both iterations and CPU time. Removing right-censored observations appears to reduce the number of iterations slightly compared to the other EM algorithms. As expected, without truncation, the Turnbull, Shen and Yu methods are identical with minor differences in run-time. The Breslow estimator generally required more iterations to converge than the other EM algorithms. A possible explanation for this is that, because it uses latent Poisson variables, the Breslow algorithm cannot generally converge in a single step even if there is no interval-censoring (i.e. $\sum_j I(W_i^* \leq r_j \leq R_i^*) = 1$ for all i) which contrasts with all the other EM algorithms. The quasi-Newton method is easy to implement but is the slowest algorithm gives the

poorest maximum likelihood estimates in general. The calculation of the likelihood and likelihood-gradient were calculated in **R** for the quasi-Newton method, so a faster implementation may improve convergence time, but probably not accuracy.

Method	log-likelihood	Iterations	CPU time (s)	% converged
Breslow	-812.44(29.87)	4,122(538)	0.280(0.040)	100%
PL	-812.44(29.87)	3,269(503)	0.163(0.027)	100%
Shen	-812.55(29.97)	8,090(9,372)	0.517(0.541)	100%
Turnbull	-812.55(29.97)	8,090(9,372)	0.563(0.591)	100%
Yu	-819.58(46.58)	4,548(2,489)	0.282(0.155)	78%
Breslow-ICM	-812.42(29.87)	588(652)	0.029(0.031)	100%
PL-ICM	-812.42(29.87)	99(33)	0.005(0.002)	100%
Turnbull-ICM	-812.44(29.89)	34,463(33,874)	2.193(2.203)	100%
Yu-ICM	-812.49(29.84)	2,827(2,703)	0.155(0.164)	100%
ICM	-812.47(29.92)	2,552(9,806)	0.388(1.460)	100%
Quasi-Newton	-814.19(29.72)	110(1)	1.793(0.298)	100%

Table 3 Median and inter-quartile range (in parentheses) of log-likelihood, run times (in iterations and seconds) and percentage converged before 100,000 iterations or with finite log-likelihood for the 25% truncated experiment. The proposed PL and PL-ICM methods are highlighted in bold.

The results for the 25% and 50% truncation experiments are shown in Tables 3 and 4 respectively. The PL-ICM and PL algorithms were the most consistently fast, accurate and stable when truncation was introduced, with the PL-ICM achieving the fastest average time and the joint highest average log-likelihood across experiment repetitions. The presence of truncation seems to increase convergence times across all methods, but the PL and PL-ICM algorithms are the least affected. Notice that, in the presence of truncation, the PL algorithm is the only one that can maximize the likelihood analytically if there is no interval-censoring.

From these experiments it is clear that the Turnbull and Shen methods are just minor variants of the same algorithm and converge in the same number of iterations for each simulation.

In the presence of truncation, Yu’s algorithm was susceptible to leaving the parameter space, leading to situations where the survival function no longer monotonically decreased. We attempted to fix this such that negative step changes in the survival

Method	log-likelihood	Iterations	CPU time (s)	% converged
Breslow	-702.98(35.96)	3,315(459)	0.208(0.032)	100%
PL	-702.97(35.96)	2,536(350)	0.122(0.018)	100%
Shen	-703.03(36.07)	8,967(10,639)	0.564(0.626)	100%
Turnbull	-703.03(36.07)	8,967(10,639)	0.618(0.687)	100%
Yu	-716.41(∞)	4,622(5,105)	0.268(0.300)	74%
Breslow-ICM	-702.96(35.96)	572(638)	0.026(0.027)	100%
PL-ICM	-702.96(35.96)	88(36)	0.005(0.002)	100%
Turnbull-ICM	-702.96(35.97)	39,072(39,726)	2.402(2.500)	100%
Yu-ICM	-707.70(37.13)	3,074(7,381)	0.171(0.399)	96%
ICM	-702.97(35.96)	4,092(12,367)	0.589(1.831)	100%
Quasi-Newton	-704.82(36.25)	109(3)	1.734(0.193)	100%

Table 4 Median and inter-quartile range (in parentheses) of log-likelihood, run times (in iterations and seconds) and percentage converged before 100,000 iterations or with finite log-likelihood for the 50% truncated experiment. The proposed PL and PL-ICM methods are highlighted in bold.

function were reset to a small positive number and the parameters recalculated accordingly. This didn’t fix the problem for all cases, however, as reflected in the % converged column of Tables 3 and 4. In the cases of non-convergence, the algorithm left the parameter space and never returned, resulting in some nonsense results with a log-likelihood of $-\infty$. Shen and Turnbull’s algorithms never encountered these issues. The Yu-ICM algorithm appeared more stable.

The quasi-Newton method was consistent across experiments in terms of iterations, but did not reach the maximum likelihood reliably. The quasi-Newton method required tolerances added to the boundaries (see Section 1.5 in the Supplementary Material) to avoid issues with numerical stability. Any sufficiently small value (less than around 1×10^{-7}) seemed to give the same results. Removing these resulted in non-finite likelihoods a large proportion of the time.

After the PL method, the Breslow algorithm was the second most consistently fast, accurate and stable EM algorithm, and the addition of the modified ICM step appears to improve convergence to be almost as fast as the PL-ICM method.

6 Illustration

Here we apply our proposed method to data from the Massachusetts Health Care Protection Study (MHCPS), which has been extensively explored in the context of calculating the NPMLE for LTIC data (Pan and Chappell, 1998; Hudgens, 2005; Gao and Chan, 2018).

The MHCPS, described fully by Chappell (1991), followed a population of non-institutionalized people over the age of 65 and aimed to estimate the age at which older adults lose functional independence. Here functional independence is defined as being able to carry out everyday tasks such as bathing or dressing. A baseline survey was conducted in 1974, with examinations 15 months, 6 and 10 years later. Age was taken as the time-scale and only those who were classed as functionally independent at enrollment were included in the study, leading to left-truncation. The time of change in functional independence was interval-censored between examinations. The study included 1030 participants, 609 female and 421 male, enrolled between the ages of 65 and 97.3. Since the youngest participant was 65, $\hat{S}(t)$ estimates the conditional probability of functional independence past age 65.

Each of the eleven algorithms from the simulation study were used to estimate survival for males and females separately. Given that estimating the NPMLE for this dataset has been historically difficult (Pan and Chappell, 1998; Hudgens, 2005), the maximum number of iterations was increased to one million and the convergence criteria set to a change in log-likelihood of 1×10^{-8} .

The run times and estimated log-likelihoods at convergence are illustrated in Tables 5 and 6 for the male and female sub-groups respectively. None of the Yu, Yu-ICM or Turnbull-ICM algorithms were stable enough to converge on a result and are omitted from the results.

The PL and PL-ICM algorithms converge quickly and stably to estimates with higher log-likelihoods than reported by either Pan and Chappell (1998) or Hudgens

Method	log-likelihood	Iterations	CPU time (s)
Breslow	-597.41	9,572	0.271
PL	-597.41	2,438	0.043
Shen	-603.03	820,879	23.624
Turnbull	-603.04	783,767	24.184
Breslow-ICM	-597.41	11,352	0.221
PL-ICM	-597.41	99	0.003
ICM	-597.43	1,000,000	75.900
Quasi-Newton	-597.78	105	0.098

Table 5 Run time and log-likelihoods for the unconditional NPMLE for females in the MHCPS. The proposed PL and PL-ICM methods are highlighted in bold.

Method	log-likelihood	Iterations	CPU time (s)
Breslow	-435.67	8,531	0.179
PL	-435.67	913	0.025
Shen	-439.17	1,000,000	20.634
Turnbull	-439.17	1,000,000	22.097
Breslow-ICM	-435.67	11,319	0.151
PL-ICM	-435.67	77	0.002
ICM	-435.68	797,917	40.329
Quasi-Newton	-435.82	105	0.076

Table 6 Run time and log-likelihoods for the unconditional NPMLE for males in the MHCPS. The proposed PL and PL-ICM methods are highlighted in bold.

(2005) for the same problem. The Breslow and Breslow-ICM algorithms reached the same estimate but were each at least five times slower than the PL algorithm and 70 times slower than the PL-ICM algorithm.

For females the ICM algorithm did not converge after 1 million iterations, but did for males after nearly 800,000 iterations. Its final log-likelihood estimate was close to, but below, that of the PL and PL-ICM algorithms. The quasi-Newton method converged faster than the ICM algorithm (in terms of both iterations and run time), but did not reach as high a likelihood and was still not as fast as either the PL or PL-ICM algorithms. We tried the quasi-Newton method with multiple tolerances but these made very little difference to the resulting estimate and run time.

The Turnbull and Shen methods were very slow, taking over 20 seconds for each estimate. For males, both methods satisfied the convergence criteria after over 700,000

iterations but, by comparing the log-likelihoods with those from other methods, it is clear that they stopped prematurely. For females neither method converged after reaching one million iterations and were still far from the log-likelihood achieved by any of the other algorithms.

Figure 1 gives the estimated survival from the PL-ICM, quasi-Newton and Turnbull methods (the Turnbull and Shen estimates were identical to each other while the PL-ICM, PL, ICM, Breslow and Breslow-ICM were mutually indistinguishable). An early precipitous drop can clearly be seen in the PL-ICM and quasi-Newton estimates, suggesting that conditioning on survival past this time would be fruitful. Conversely, the Turnbull algorithm gives almost no indication that a conditional survival estimate would be necessary. The issue of this large drop in survival has been highlighted previously by [Hudgens \(2005\)](#) who showed there is no solution to the maximization problem for either males or females (using the parameterization in Equation (2) at least). Using the likelihood in Equation (4) allows for a solution and we see that $\mathcal{R}_1^{(0)} = 1$ and $\hat{h}_1 = 1$ for both males and females, where $\mathcal{Q}_1 = (65, 65.3)$. Given $\hat{h}_1 = 1$, it appears principled to condition on survival past 65.3, which essentially removes only one observation in each data set.

When calculating the NPMLE for LTIC data, [Hudgens \(2005\)](#) proposed checking whether a solution exists by inspecting the form of the inner-intervals manually or checking the strong connected-ness of an intersection graph formed by the intervals. Neither approach is straightforward compared to simply running the proposed PL-ICM algorithm and searching for a j where $\hat{h}_j = 1$. The PL method, and methods parameterized in terms of the cumulative hazard, can maximize the full likelihood and allow conditioning to be carried out even in the presence of this drop.

The futility of the Turnbull method for this data is highlighted by inspecting Equation 1 in the Supplementary Materials, where $M = \sum_{j=1}^m M_j$ represents the sample size augmented with virtual “ghost” observations to account for truncation

(Turnbull, 1976). Due to truncation, we expect M to be larger than the real sample sizes. Pan and Chappell (1998) showed that Turnbull’s algorithm updates the parameters with steps inversely proportional to M . For this dataset, by the time it has “converged”, Turnbull’s algorithm gives an M of over 10^8 for males and females. The resulting updates are tiny and, given the estimate is so far from the true NPMLE, we don’t expect Turnbull’s algorithm (or Shen’s, which appears to behave almost identically) to ever realistically converge.

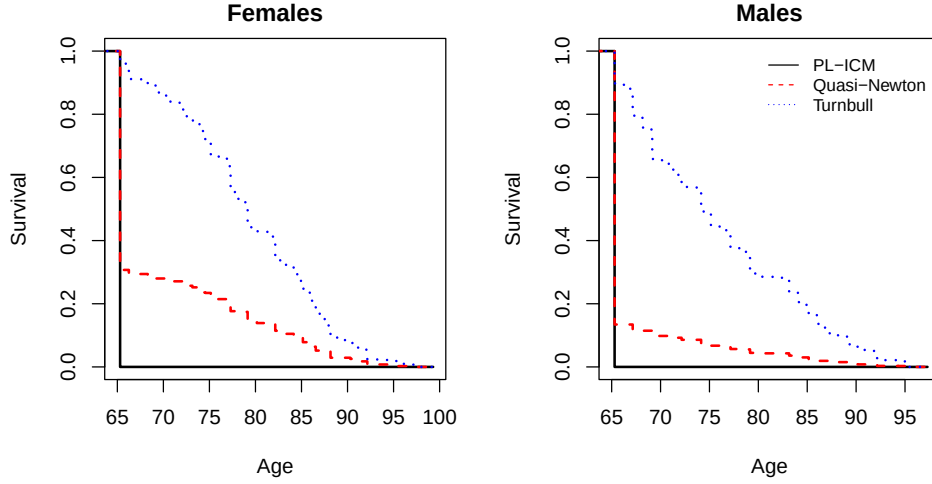


Fig. 1 Survival estimates for the MHCPS dataset for females and males. The estimate from the Turnbull algorithm are in dotted blue, quasi-Newton method in dashed red and PL-ICM method in solid black.

Figure 2 shows the conditional survival estimates given survival past age 65.3. This was calculated directly from the PL-ICM estimate used to generate Figure 1 (i.e. the algorithm was not re-run). This shows a slight difference in survival probabilities between females and males, with females losing functional independence later than males. Further investigation to extend the work of Sen and Xu (2015) to allow

bootstrapped confidence intervals for LTIC data would be warranted to better quantify this difference. In a semi-parametric analysis [Pan and Chappell \(1998\)](#) and [Gao and Chan \(2018\)](#) found a small though non-significant difference (at a 5% significance level) difference between males and females for this same dataset.

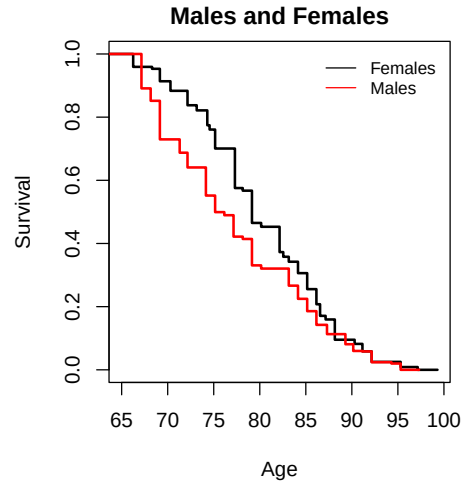


Fig. 2 Conditional survival estimates for the MHCPS dataset for females and males for age greater than 65.1. This was calculated from the outputs of the PL-ICM algorithm used in [Figure 1](#).

The Turnbull estimate of the NPMLE from [Figure 1](#) is quite close to the conditional estimate of the true NPMLE, as shown in [Figure 3](#). This does little to bolster the case for using the Turnbull estimator, however, as it is much slower than the PL algorithm and is still far from the true maximum likelihood. We would argue that a very clear instance of needing to use a conditional survival estimate on is more justifiable than a seemingly sensible estimate that is actually far from the maximum likelihood with behavior that is difficult to reason about.

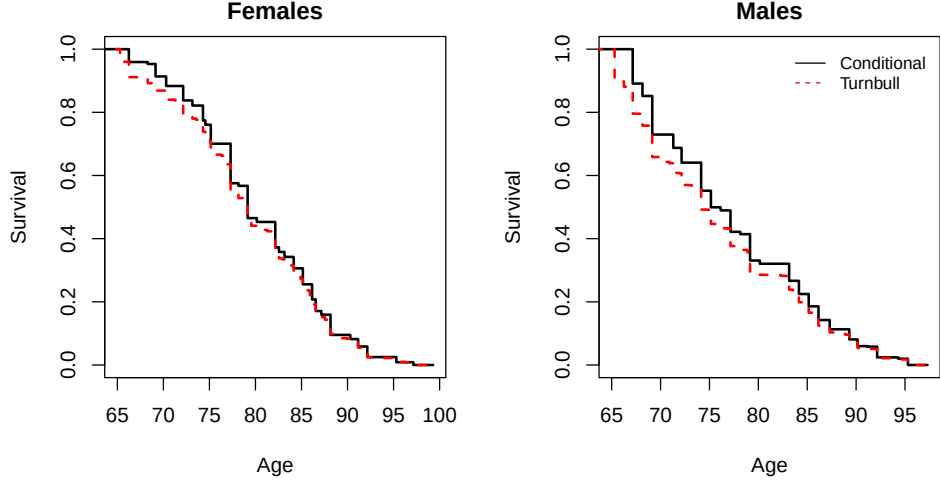


Fig. 3 Comparison of the survival estimate given by the Turnbull algorithm and the conditional survival estimate past 65.3 from the PL-ICM algorithm for the MHCPS dataset for females and males.

7 Discussion

Here we have presented a fast and stable PL-ICM algorithm for calculating the NPMLE for LTIC data. Our efficient implementation allows for the estimation of the NPMLE in scenarios that are impractical to accurately estimate with other algorithms.

The PL method is appealing in its similarity to the well-known KM method for right-censored and left-truncated data. The advantage of this approach is that right-censoring and left-truncation are handled in one step (the calculation of $\mathcal{R}^0$) and that, in the absence of interval-censoring, the NPMLE can be solved analytically. This contrasts with the usual Turnbull estimator, where extra weight is iteratively added to observations to account for truncation. The product-limit formulation also allows for stable estimation in the presence of heavy truncation, as seen in the simulation study and MHCPS illustration.

Wang (2008) suggested a dimension-reduction algorithm could be applied to truncated and interval-censored data but, to our knowledge, this has not yet been achieved. However, echoing the argument of Anderson-Bergman (2017a): given the strength of the dimension-reduction algorithm is to reduce iteration time from $O(nm)$ to $O(nm')$ (where m' is some reduced number of inner-intervals), an improvement over the $O(n)$ iteration time of the PL-ICM algorithm is unlikely.

Given the high incompleteness of LTIC data, cases of non-existent and non-unique NPMLEs are well known (Hudgens, 2005), particularly when truncation is heavy or sample sizes are small. However, the PL method does avoid some of the non-existence issues. One such instance is Example 1 in Hudgens (2005) which gives three observations (T_i, L_i, R_i) : $(1, 2, 4)$, $(3, 5, 6)$, $(3, 7, 8)$. This results in three inner-intervals $(2, 3)$, $(5, 6]$, $(7, 8]$. Using Equation (2) the likelihood is

$$\frac{s_1}{s_1 + s_2 + s_3} \frac{s_2}{s_2 + s_3} \frac{s_3}{s_2 + s_3}$$

where $s_1 + s_2 + s_3 = 1$. The maximum likelihood estimate of this is then the limit of $s_1 = 1 - \eta$, $s_2 = \eta/2$, $s_3 = \eta/2$ as $\eta \rightarrow 0$, giving a limiting likelihood of $\frac{1}{4}$. The equivalent PL likelihood using Equation (4) is

$$h_1 h_2 (1 - h_2) h_3$$

with a solution of $h_1 = 1$, $h_2 = 1/2$, $h_3 = 1$ and a likelihood of exactly $\frac{1}{4}$. Indeed, here the solution can be found explicitly. Although this gives a survival function that drops almost immediately to 0 it is still possible to calculate a conditional survival estimate $P(X > t | X > 4)$.

Due to the unique way of dealing with left-truncation, the PL method finds instances where $\hat{h}_j = 1$ for $j < m$ far quicker than all other algorithms. In these cases it seems principled to condition on survival past these times. However, aside from instances when $\hat{h}_j = 1$, there appears to be no consistent method for picking a time to

condition on. For left-truncated and right-censored processes [Woodroffe \(1985\)](#) and others avoid the issue of conditioning altogether by simply adding a small, non-zero number to (the equivalent of) $\mathcal{R}_j$ in Equation (9). This does little to attenuate very large drops, however. A more fruitful approach may be to estimate survival from the full likelihood involving $P(X \in A | X > T)$ and by making assumptions about the distribution of T . Assuming T follows a uniform distribution, for example, gives rise to length-biased data ([Vardi, 1982](#)). [Gao and Chan \(2018\)](#) applied the length-biased assumption to semi-parametric regression in an interval-censored context.

In our simulation study, to highlight the difference in convergence due to algorithm choice alone, we chose to time each run starting from after calculating the inner-intervals. For most of our simulations, the time taken to calculate the inner-intervals actually took more time than the PL-ICM algorithm took to converge due to a fairly naive pre-processing method. A faster algorithm, potentially based on HeightMap algorithm ([Maathuis, 2005](#)) and extended to LTIC data, would be fruitful. This could make a significant difference if bootstrapping estimates from the NPMLE were needed, where the inner-intervals would need to be recalculated with each bootstrap iteration.

Acknowledgments

The authors are grateful to Professor Qiqing Yu for some helpful correspondence regarding left-truncation, and to Professor Rick Chappell for help with the MHCPS data. The linear-time implementation of the EM algorithms were guided by the source code of the `icenReg` ([Anderson-Bergman, 2017b](#)) package and the `monotone` ([Busing, 2022](#)) package was used for the WPAVA algorithm.

Funding

This research is funded by the Department for the Economy Northern Ireland (DfE NI).

Data and Code Availability

The code used for all the simulation and illustration studies in this paper can be found here: <https://github.com/ZacharyWaller/ltic>. The Massachusetts Health Care Panel Study data used in Section 6 can be accessed here: <https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fj.0006-341X.2002.00064.x&file=BIOM.64.sm.010423.txt>.

Author Contributions

Conceptualization, methodology, software implementation, data acquisition and analysis were performed by ZW. The original manuscript was written by ZW and all authors reviewed and revised each version. All authors read and approved the final manuscript. FL, AHM and FK supervised the project.

Declarations

The authors declare no conflicts of interest.

Appendix A Derivation of the NPMLE

From Equation (5) the solution to $\frac{\partial H}{\partial h_j} = \frac{\partial \log \mathcal{L}}{\partial h_j} + a_j - b_j = 0$ for $1 \leq j \leq m$ maximizes the likelihood. For the first term, from Equation (4),

$$\begin{aligned} \frac{\partial \log \mathcal{L}}{\partial h_j} = & \sum_{i=1}^{n_0} \frac{-I(T_i^* \leq r_{ij} \leq L_i^*)}{(1 - h_j)} + \\ & \sum_{i=n_0+1}^n \left[\frac{I(W_i^* \leq r_j \leq R_i^*) \prod_{k=T_i^*}^{m-1} (1 - h_k)}{\mathcal{L}_i} - \sum_{l>j, l \geq L_i^*}^{R_i^*} \frac{I(r_j \leq R_i^*) h_l \prod_{k=T_i^*, k \neq j}^{l-1} (1 - h_k)}{\mathcal{L}_i} \right] \end{aligned} \quad (\text{A1})$$

where

$$\mathcal{L}_i = \sum_{j=W_i^*}^{R_i^*} h_j \prod_{k=T_i^*}^{j-1} (1 - h_k). \quad (\text{A2})$$

Remembering that $a_j h_j = 0$ and $b_j(1 - h_j) = 0$ then at the maximum

$$\frac{\partial \log \mathcal{L}}{\partial h_j} h_j (1 - h_j) = 0,$$

that is

$$\begin{aligned} & h_j \sum_{i=1}^{n_0} -I(T_i^* \leq r_{ij} \leq L_i^*) + \\ & \sum_{i=n_0+1}^n \left[\frac{I(W_i^* \leq r_j \leq R_i^*) h_j \prod_{k=T_i^*}^{m-1} (1 - h_k)}{\mathcal{L}_i} - h_j \sum_{l \geq j, l \geq L_i^*}^{R_i^*} \frac{I(r_j \leq R_i^*) h_l \prod_{k=T_i^*}^{l-1} (1 - h_k)}{\mathcal{L}_i} \right] = 0 \end{aligned} \quad (\text{A3})$$

Rearranging Equation (A3) gives

$$\begin{aligned} h_j = & \sum_{i=n_0+1}^n \frac{I(W_i^* \leq r_j \leq R_i^*) h_j \prod_{k=T_i^*}^{m-1} (1 - h_k)}{\mathcal{L}_i} / \\ & \left[\sum_{i=1}^{n_0} I(T_i^* \leq r_{ij} \leq L_i^*) + \sum_{l \geq j, l \geq L_i^*}^{R_i^*} \frac{I(r_j \leq R_i^*) h_l \prod_{k=T_i^*}^{l-1} (1 - h_k)}{\mathcal{L}_i} \right] \end{aligned} \quad (\text{A4})$$

which, upon substituting in Equation (A2) and using definitions (8) and (9), gives Equation (7).

Appendix B Derivation of the PL algorithm

Using Equation (13), $\mathcal{L}_i$ in Equation (A2) can be rewritten $\mathcal{L}_i = \frac{S_{L_i^*} - S_{R_i^*}}{S_{T_i^*}}$, and

$$I(W_i^* \leq r_j \leq R_i^*) h_j \prod_{k=T_i^*}^{j-1} (1 - h_k) = I(W_i^* \leq r_j \leq R_i^*) h_j \frac{S_{j-1}}{S_{T_i^*}}$$

meaning Equation (8) can be simplifies to

$$d_{ij} = h_j S_{j-1} \frac{I(W_i^* \leq r_j \leq R_i^*)}{S_{L_i^*} - S_{R_i^*}}.$$

Next, notice that $\sum_{k=j}^m d_{ik} = 1 - \sum_{k'=1}^{j-1} d_{ik'}$, meaning Equation (9) can be rewritten

as

$$\mathcal{R}_j = \sum_{i=1}^{n_0} I(T_i^* \leq r_j \leq L_i^*) + \sum_{i=n_0+1}^n I(T_i^* \leq r_j) - \sum_{i=n_0+1}^n \sum_{k=0}^{j-1} d_{ik}.$$

which, using Equation (14), is just

$$\Phi_j = \mathcal{R}_j^0 - \sum_{i=n_0+1}^n \sum_{k=0}^{j-1} d_{ik}$$

and therefore

$$h_j = \frac{\sum_{i=n_0+1}^n d_{ij}}{\mathcal{R}_j^0 - \sum_{i=n_0+1}^n \sum_{k=0}^{j-1} d_{ik}}.$$

Given $d_j = \sum_{i=n_0+1}^n d_{ij}$ and the definition of D_j in Equation (12), this is just Equation (10).

Appendix C ICM Derivatives

Assuming all of L_i^* , R_i^* and T_i^* are distinct, the derivatives of the log-likelihood contribution of participant i are

$$\frac{\partial \mathcal{L}_i}{\partial \Lambda_{L_i^*}} = \frac{-e^{-\Lambda_{L_i^*} + \Lambda_{T_i^*}}}{e^{-\Lambda_{L_i^*} + \Lambda_{T_i^*}} - e^{-\Lambda_{R_i^*} + \Lambda_{T_i^*}}}, \quad \frac{\partial \mathcal{L}_i}{\partial \Lambda_{R_i^*}} = \frac{e^{-\Lambda_{R_i^*} + \Lambda_{T_i^*}}}{e^{-\Lambda_{L_i^*} + \Lambda_{T_i^*}} - e^{-\Lambda_{R_i^*} + \Lambda_{T_i^*}}} \quad \text{and} \quad \frac{\partial \mathcal{L}_i}{\partial \Lambda_{T_i^*}} = 1.$$

The diagonal elements of the Hessian are then

$$\frac{\partial^2 \mathcal{L}_i}{\partial \Lambda_{L_i^*}^2} = \frac{\partial^2 \mathcal{L}_i}{\partial \Lambda_{R_i^*}^2} = \frac{-e^{-\Lambda_{L_i^*} - \Lambda_{R_i^*} + 2\Lambda_{T_i^*}}}{(e^{-\Lambda_{L_i^*} + \Lambda_{T_i^*}} - e^{-\Lambda_{R_i^*} + \Lambda_{T_i^*}})^2}, \quad \frac{\partial^2 \mathcal{L}_i}{\partial \Lambda_{T_i^*}^2} = 0.$$

These can be combined whenever any of L_i^* , R_i^* and T_i^* are equal. Whenever $\frac{\partial^2 \mathcal{L}_i}{\partial \Lambda_j^2} = 0$ set it to some small negative value, ν .

The WPAVA algorithm from the R package `monotone` (Busing, 2022) was used to ensure monotonicity after the Newton-like step.

References

- Anderson-Bergman, C.: An Efficient Implementation of the EMICM Algorithm for the Interval Censored NPMLE. *Journal of Computational and Graphical Statistics* **26**(2), 463–467 (2017) <https://doi.org/10.1080/10618600.2016.1208616> . Accessed 2022-11-18
- Anderson-Bergman, C.: **icenReg** : Regression Models for Interval Censored Data in *R*. *Journal of Statistical Software* **81**(12) (2017) <https://doi.org/10.18637/jss.v081.i12>
- Alioum, A., Commenges, D.: A Proportional Hazards Model for Arbitrarily Censored and Truncated Data. *Biometrics* **52**(2), 512 (1996) <https://doi.org/10.2307/2532891>

- Aragón, J., Eberly, D.: On convergence of convex minorant algorithms for distribution estimation with interval-censored data. *Journal of Computational and Graphical Statistics* **1**(2), 129–140 (1992). Publisher: Taylor & Francis
- Busing, F.M.T.A.: Monotone Regression: A Simple and Fast $O(n)$ PAVA Implementation. *Journal of Statistical Software* **102**(Code Snippet 1) (2022) <https://doi.org/10.18637/jss.v102.c01>
- Banerjee, M., Wellner, J.A.: Confidence Intervals for Current Status Data. *Scandinavian Journal of Statistics* **32**(3), 405–424 (2005) <https://doi.org/10.1111/j.1467-9469.2005.00454.x>
- Chappell, R.: Sampling design of multiwave studies with an application to the Massachusetts health care panel study. *Statistics in Medicine* **10**(12), 1945–1958 (1991) <https://doi.org/10.1002/sim.4780101209>
- Cox, D.R.: Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **34**(2), 187–202 (1972). Publisher: Wiley Online Library
- Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data Via the *EM* Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1), 1–22 (1977) <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>
- Efron, B.: The two sample problem with censored data. *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability* **4**, 831–853 (1967)
- Finkelstein, D.M.: A Proportional Hazards Model for Interval-Censored Failure Time Data. *Biometrics* **42**(4), 845 (1986) <https://doi.org/10.2307/2530698>
- Frydman, H., Liu, J.: Nonparametric estimation of the cumulative intensities in an

- interval censored competing risks model. *Lifetime Data Analysis* **19**(1), 79–99 (2013) <https://doi.org/10.1007/s10985-012-9232-6>
- Fletcher, R.: *Practical Methods of Optimization*. John Wiley & Sons, ??? (2000)
- Frydman, H.: A Nonparametric Estimation Procedure for a Periodically Observed Three-State Markov Process, with Application to Aids. *Journal of the Royal Statistical Society: Series B (Methodological)* **54**(3), 853–866 (1992) <https://doi.org/10.1111/j.2517-6161.1992.tb01457.x>
- Frydman, H.: A Note on Nonparametric Estimation of the Distribution Function from Interval-Censored and Truncated Observations. *Journal of the Royal Statistical Society: Series B (Methodological)* **56**(1), 71–74 (1994) <https://doi.org/10.1111/j.2517-6161.1994.tb01960.x>
- Gao, F., Chan, K.C.G.: Semiparametric Regression Analysis of Length-Biased Interval-Censored Data. *Biometrics* **75**(1), 121–132 (2018) <https://doi.org/10.1111/biom.12970>
- Gentleman, R., Geyer, C.J.: Maximum likelihood for interval censored data: Consistency and computation. *Biometrika* **81**(3), 618–623 (1994). Publisher: Oxford University Press
- Groeneboom, P., Wellner, J.A.: *Information Bounds and Nonparametric Maximum Likelihood Estimation*. Birkhäuser Basel, Basel (1992). <https://doi.org/10.1007/978-3-0348-8621-5>
- Hudgens, M.G.: On Nonparametric Maximum Likelihood Estimation with Interval Censoring and Left Truncation. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **67**(4), 573–587 (2005) <https://doi.org/10.1111/j.1467-9868.2005.00516.x>

- Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. *Journal of the American statistical association* **53**(282), 457–481 (1958). Publisher: Taylor & Francis
- Kuhn, H.W., Tucker, A.W.: Nonlinear programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1950, 481–492 (1951)
- Lynden-Bell, D.: A method of allowing for known observational selection in small samples applied to 3CR quasars. *Monthly Notices of the Royal Astronomical Society* **155**(1), 95–118 (1971). Publisher: Oxford University Press Oxford, UK
- Li, C., Pak, D., Todem, D.: Adaptive lasso for the Cox regression with interval censored and possibly left truncated data. *Statistical Methods in Medical Research* **29**(4), 1243–1255 (2020) <https://doi.org/10.1177/0962280219856238>
- Maathuis, M.H.: Reduction Algorithm for the NPMLE for the Distribution Function of Bivariate Interval-Censored Data. *Journal of Computational and Graphical Statistics* **14**(2), 352–362 (2005) <https://doi.org/10.1198/106186005X48470> . Accessed 2024-12-06
- Pan, W.: Extending the Iterative Convex Minorant Algorithm to the Cox Model for Interval-Censored Data. *Journal of Computational and Graphical Statistics* **8**(1), 109–120 (1999) <https://doi.org/10.1080/10618600.1999.10474804>
- Pan, W., Chappell, R.: Computation of the NPMLE of distribution functions for interval censored and truncated data with applications to the Cox model. *Computational Statistics & Data Analysis* **28**(1), 33–50 (1998) [https://doi.org/10.1016/S0167-9473\(98\)00026-7](https://doi.org/10.1016/S0167-9473(98)00026-7)
- Peto, R.: Experimental Survival Curves for Interval-Censored Data. *Applied Statistics*

22(1), 86 (1973) <https://doi.org/10.2307/2346307>

R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2024). <https://www.R-project.org/>

Shen, P.-S.: Nonparametric estimators of survival function under the mixed case interval-censored model with left truncation. *Lifetime Data Analysis* **26**(3), 624–637 (2020) <https://doi.org/10.1007/s10985-020-09493-2>

Sun, J.: The Statistical Analysis of Interval-censored Failure Time Data. *Statistics for biology and health*. Springer, New York (2006)

Sen, B., Xu, G.: Model based bootstrap methods for interval censored data. *Computational Statistics & Data Analysis* **81**, 121–129 (2015) <https://doi.org/10.1016/j.csda.2014.07.007>

Schick, A., Yu, Q.: Consistency of the GMLE with Mixed Case Interval-Censored Data. *Scandinavian Journal of Statistics* **27**(1), 45–55 (2000) <https://doi.org/10.1111/1467-9469.00177>

Turnbull, B.W.: The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data. *Journal of the Royal Statistical Society: Series B (Methodological)* **38**(3), 290–295 (1976) <https://doi.org/10.1111/j.2517-6161.1976.tb01597.x>

Vardi, Y.: Nonparametric estimation in the presence of length bias. *The Annals of Statistics* **10**(2), 616–620 (1982). Publisher: Institute of Mathematical Statistics

Wang, Y.: Dimension-reduced nonparametric maximum likelihood computation for interval-censored data. *Computational Statistics & Data Analysis* **52**(5), 2388–2402

(2008) <https://doi.org/10.1016/j.csda.2007.10.018> . Accessed 2024-10-16

Woodroffe, M.: Estimating a distribution function with truncated data. *The Annals of Statistics* **13**(1), 163–177 (1985). Publisher: Institute of Mathematical Statistics

Wellner, J.A., Zhan, Y.: A Hybrid Algorithm for Computation of the Nonparametric Maximum Likelihood Estimator from Censored Data. *Journal of the American Statistical Association* **92**(439), 945–959 (1997) <https://doi.org/10.1080/01621459.1997.10474049>

Yu, Q.: The generalised MLE with truncated interval-censored data. *Journal of Nonparametric Statistics* **35**(2), 266–282 (2023) <https://doi.org/10.1080/10485252.2022.2147173>

Zeng, D., Mao, L., Lin, D.Y.: Maximum likelihood estimation for semiparametric transformation models with interval-censored data. *Biometrika* **103**(2), 253–271 (2016) <https://doi.org/10.1093/biomet/asw013>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [IntervalCensoringLeftTruncationSupplementary1.pdf](#)