Statistics and Computing manuscript No.
(will be inserted by the editor)

Efficient Computation of Nonparametric Survival Functions via a Hierarchical Mixture Formulation

Yong Wang [1] **and Stephen M. Taylor** [2]

Received: June 2011 / Accepted: date

Abstract We propose a new algorithm for computing the maximum likelihood estimate of a nonparametric survival function for interval-censored data, by extending the recently-proposed constrained Newton method in a hierarchical fashion. The new algorithm makes use of the fact that a mixture distribution can be recursively written as a mixture of mixtures, and takes a divide-and-conquer approach to break down a large-scale constrained optimization problem into many small-scale ones, which can be solved rapidly. During the course of optimization, the new algorithm, which we call the hierarchical constrained Newton method, can efficiently reallocate the probability mass, both locally and globally, among potential support intervals. Its convergence is theoretically established based on an equilibrium analysis. Numerical study results suggest that the new algorithm is the best choice for data sets of any size and for solutions with any number of support intervals.

Keywords: Nonparametric maximum likelihood; survival function; interval censoring; clinical trial; constrained Newton method; disease-free survival

1 Introduction

Interval-censored data can easily arise in fields like epidemiological studies and clinical trials. For example, HIV/AIDS studies may yield various types of interval-censored data (Siegfried et al., 2005; Chen et al., 2007; Kumwenda et al., 2008). To investigate the influence of risk factors or the effectiveness of treatments, researchers often need to estimate and compare survival functions for different groups of subjects. One can choose to use parametric models, but parametric assumptions may fail in the real world and lead to significantly biased estimates or even incorrect conclusions. In contrast, the assumption-free, nonparametric maximum likelihood approach is advantageous in this regard and is widely adopted in practice. However, the lack of fast algorithms for finding the nonparametric maximum likelihood estimate (NPMLE) of a survival function hinders the application of this methodology, e.g., when a large-scale problem is studied or a bootstrap or other resampling procedure needs to be performed.

Owing to the difficulty of computing the NPMLE, a common, practical approach to dealing with general interval-censored data is imputation (Sun, 2006, section 2.4). By replacing an interval-censored observation with one or more points from that interval, the nonparametric Kaplan-Meier survival curve can be used. However this throws away some of the information in the data, and therefore introduces bias and reduces the power of statistical tests applied to the data.

For computing the NPMLE of a survival function, a number of algorithms have been proposed in the past, e.g., the expectation-maximization (EM) algorithm (Turnbull, 1974, 1976; Dempster et al., 1977), the iterative convex minorant (ICM) algorithm (Groeneboom, 1991; Groeneboom and Wellner, 1992; Jongbloed, 1998), the hybrid ICM-EM algorithm (Wellner and Zhan, 1997), the subspace-based Newton (SBN) method (Dümbgen et al., 2006), the constrained Newton method (CNM) (Wang, 2007, 2008), and the support reduction (SR) algorithm (Groeneboom et al., 2008). A general dimension reduction technique was also proposed by Wang

1. Department of Statistics, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand. Tel.: +64-9-9234700. Fax: +64-9-3737018. E-mail: yongwang@auckland.ac.nz
2. Auckland University of Technology, Private Bag 92006, Auckland 1142, New Zealand. E-mail: steve.taylor@aut.ac.nz

(2008) that can help improve the performance of an algorithm. Numerical studies conducted by Wang (2008) showed that an algorithm may perform well in some scenarios but unsatisfactorily in others, and no algorithm can efficiently handle high-dimensional problems with many support intervals.

Our new algorithm presented below is an extension to the CNM and is based on the fact that the nonparametric survival function for interval-censored data has a mixture structure (Böhning et al., 1996). The CNM has a quadratic order of convergence and requires in practice only a small number of iterations to converge. Nevertheless, when the maximum likelihood estimate has a large number of support intervals, as may arise in large-scale survival studies, its computation cost can be extremely high due to solving a large-scale constrained linear regression problem at each iteration. In such cases, its overall performance is understandably very poor. To overcome this difficulty, the new algorithm makes use of the fact that a mixture distribution can be recursively written as a mixture of mixtures and be represented in a hierarchical form. Using such a hierarchy breaks down a large-scale optimization problem into many small-scale ones that are all of the same nature and can be solved quickly by the constrained Newton method. Furthermore it allows an efficient flow of probability mass at all levels among the support intervals and hence rapid convergence can be expected.

This hierarchical structure was also used by Pilla and Lindsay (2001) to propose several alternative EM methods, which are hybrids of the EM algorithm and the univariate Newton-Raphson method. Even with their significant performance improvement upon the conventional EM algorithm, we suggest that the potential of the hierarchical structure has not been fully realized. The EM algorithm is known to have a slow convergence, and the pairing of support points (intervals), as needed inevitably by the univariate Newton-Raphson method, can be inefficient at redistributing probability mass even in a local area. These two drawbacks, however, can be readily resolved simultaneously by the CNM, which has a rapid convergence and can easily handle higher-dimensional problems. Evidence shows that combining the CNM with a hierarchical structure brings about remarkable performance gain, both numerically and analytically. Since our new hierarchical algorithm is developed around a single algorithm, namely the CNM, it also has a conceptually simpler formulation.

The remainder of the paper is organized as follows. In Section 2, the nonparametric maximum likelihood estimation and computation of a survival function are briefly reviewed. The hierarchical constrained Newton method is described in Section 3, with its convergence established in Section 4, using an equilibrium approach. Section 5 presents results of numerical studies that compare the performance of the new algorithm and competitors. Concluding remarks are given in Section 6.

2 Estimation of a nonparametric survival function

2.1 Nonparametric maximum likelihood estimation

Let $O_1, \ldots, O_n \subset [0, \infty)$ denote the censoring intervals in which, respectively, n independent times to event are known to have fallen. Each censoring interval may, e.g., result from follow-up visits, which also need to be independent of the event time. The aim here is to estimate the distribution of the time to event nonparametrically by the maximum likelihood approach based on these interval-valued observations. Generally, depending on the type of censoring, each interval O_i can be open, semi-open or closed. As pointed out by Peto (1973) and Turnbull (1976), the NPMLE of the survival function could only have positive masses on a subset of $O_1 \cup \cdots \cup O_n$, namely the set of maximal intersection intervals. An intersection interval is a nonempty intersection of any combination of $O_1, \ldots, O_n$ and a maximal intersection interval is an intersection interval that contains no other intersection interval (Wong and Yu, 1999; Gentleman and Vandal, 2001; Maathuis, 2005)

Denote by $I_1, \ldots, I_m$ the maximal intersection intervals after being sorted from left to right according their positions on the real line. Then the NPMLE must have all of its support on a subset of $\{I_1, \ldots, I_m\}$. For any I_j allocated with a positive mass, it is impossible to determine the distribution of mass within it, and one therefore focuses on estimating its overall mass p_j. Let $\mathcal{J} = \{1, \ldots, m\}$, the index set of these intervals, and $\mathbf{p} = (p_1, \ldots, p_m)^\top$, a point in the $(m-1)$-dimensional probability simplex

$$\mathcal{P} \equiv \{\mathbf{p} : \mathbf{p}^\top \mathbf{1} = 1, \mathbf{p} \geq \mathbf{0}\},$$

where $\mathbf{0} = (0, \ldots, 0)^\top$ and $\mathbf{1} = (1, \ldots, 1)^\top$. Let $\delta_{ij} = 1$ if $I_j \subseteq O_i$, and $\delta_{ij} = 0$ otherwise. Given $\mathbf{p}$, the probability for the event time to be in O_i is

$$f_i = f_i(\mathbf{p}) = \sum_{j=1}^{m} p_j \delta_{ij}, \tag{1}$$

and the log-likelihood function of $\mathbf{p}$ is

$$\ell(\mathbf{p}) = \sum_{i=1}^{n} \log\left(\sum_{j=1}^{m} p_j \delta_{ij} \right). \tag{2}$$

The NPMLE $\hat{\mathbf{p}}$ maximizes $\ell(\mathbf{p})$ among all $\mathbf{p} \in \mathcal{P}$, which is characterized by the vertex directional derivative (Lindsay, 1995; Böhning et al., 1996). The gradient vector of the log-likelihood function is given by

$$\mathbf{g} = \mathbf{g}(\mathbf{p}) = \frac{\partial \ell(\mathbf{p})}{\partial \mathbf{p}},$$

whose largest element is

$$g^* = g^*(\mathbf{p}) = \max_{1 \le j \le m} \{g_j(\mathbf{p})\}.$$

Denoting by $\mathbf{e}_j$ the jth vertex of $\mathcal{P}$, the jth vertex directional derivative from $\mathbf{p}$ is defined as

$$\begin{aligned} d_j(\mathbf{p}) &\equiv \left. \frac{\partial \ell\{(1-\epsilon)\mathbf{p} + \epsilon \mathbf{e}_j\}}{\partial \epsilon} \right|_{\epsilon=0} \\ &= g_j(\mathbf{p}) - n. \end{aligned} \tag{3}$$

At $\hat{\mathbf{p}}$, we have

$$\begin{cases} d_j(\hat{\mathbf{p}}) = 0, & \text{if } \hat{p}_j > 0, \\ d_j(\hat{\mathbf{p}}) \le 0, & \text{if } \hat{p}_j = 0. \end{cases} \tag{4}$$

For any $\mathbf{p} \in \mathcal{P}$, it holds that

$$\max_{1 \le j \le m} \{d_j(\mathbf{p})\} \ge \ell(\hat{\mathbf{p}}) - \ell(\mathbf{p}), \tag{5}$$

due to the concavity of the log-likelihood.

For notational simplicity, we will frequently write h for the value of a function $h(\mathbf{p})$ that is evaluated at a generic estimate $\mathbf{p} \in \mathcal{P}$, when no ambiguity about $\mathbf{p}$ should arise.

2.2 The constrained Newton method

The constrained Newton method for computing the NPMLE is briefly described here. Let

$$\begin{aligned} &\mathbf{s}_i = (\delta_{i1}, \ldots, \delta_{im})^\top / f_i, \quad i = 1, \ldots, n, \\ &\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)^\top. \end{aligned}$$

Then the log-likelihood function $\ell(\mathbf{p})$ has the following gradient vector and Hessian matrix:

$$\begin{aligned} &\frac{\partial \ell(\mathbf{p})}{\partial \mathbf{p}} = \mathbf{S}^\top \mathbf{1}, \\ &\frac{\partial^2 \ell(\mathbf{p})}{\partial \mathbf{p} \partial \mathbf{p}^\top} = -\mathbf{S}^\top \mathbf{S}. \end{aligned}$$

Using the Taylor series expansion about $\mathbf{p}$ and letting $\boldsymbol{\eta} = \mathbf{p}' - \mathbf{p}$, one obtains

$$\begin{aligned} \ell(\mathbf{p}') - \ell(\mathbf{p}) &= \mathbf{1}^\top \mathbf{S}\boldsymbol{\eta} - \frac{1}{2}\boldsymbol{\eta}^\top \mathbf{S}^\top \mathbf{S}\boldsymbol{\eta} + o(||\mathbf{S}\boldsymbol{\eta}||^2) \\ &= -\frac{1}{2}||\mathbf{S}\mathbf{p}' - \mathbf{2}||^2 + \frac{n}{2} + o(||\mathbf{S}\boldsymbol{\eta}||^2), \end{aligned} \tag{6}$$

where $\mathbf{S}\mathbf{p} = \mathbf{1}$ and $\mathbf{2} = (2, \ldots, 2)^\top$. Therefore, maximizing $\ell(\mathbf{p}')$ in the neighbourhood of $\mathbf{p}$ can be approximated by solving the least squares linear regression problem with equality and non-negativity constraints:

$$\min_{\mathbf{p}'} \quad ||\mathbf{S}\mathbf{p}' - \mathbf{2}||^2, \quad \text{s.t.} \quad \mathbf{p}'^\top \mathbf{1} = 1, \mathbf{p}' \ge \mathbf{0}. \tag{7}$$

This problem can be efficiently solved by the NNLS algorithm of Lawson and Hanson (1974) (for solving a least squares problem with non-negativity constraint), after a transformation suggested by Dax (1990); see Appendix A.

To ensure monotone increase and global convergence, a line search is needed. Let $\mathbf{p}' = \mathbf{p} + \boldsymbol{\eta}$ be the solution to problem (7). The new vector is $\mathbf{p} + \sigma^u \boldsymbol{\eta}$, using the smallest $u \in \{0, 1, 2, \ldots\}$ that satisfies the inequality

$$\ell(\mathbf{p} + \sigma^u \boldsymbol{\eta}) \ge \ell(\mathbf{p}) + \alpha \sigma^u \mathbf{g}^\top \boldsymbol{\eta}, \quad 0 < \alpha < \frac{1}{2}. \tag{8}$$

In our implementation, the values $\sigma = \frac{1}{2}$ and $\alpha = \frac{1}{3}$ were used.

2.3 Dimension-reduced computation

Solving problem (7) on the full dimensional $(m-1)$-simplex of $\mathcal{P}$ can be computationally very expensive, and is not necessary. For interval-censored data, the NPMLE $\hat{\mathbf{p}}$ may have a much smaller number of positive elements and it is hence possible to restrict the optimization in dimension-reduced simplexes, ideally the one that corresponds to only the positive elements of $\hat{\mathbf{p}}$. A strategy for expanding the support set progressively is proposed in Wang (2008). Its idea is to start with a small support set and expand it rapidly. Specifically, in each iteration it adds to the support set the candidate support intervals that have the maximum gradient values between inclusively every two neighbouring support intervals that are currently in the support set. The dimension-reduced CNM thus expands the support set at an exponential rate, if necessary. Once near the NPMLE, it maintains virtually the same support set as the NPMLE, as desired.

Let us denote by $j_{s1} < \cdots < j_{sm_s}$ the ordered indexes of the m_s positive elements of the sth iterate $\mathbf{p}_s \in \mathcal{P}$. Always we have $j_{s1} = 1$ and $j_{sm_s} = m$, due to Lemma 1 of Wang (2008).

Algorithm 1 (CNM) Choose a small $\tau > 0$ and set $s = 0$. From an initial estimate $\mathbf{p}_0$ with $\ell(\mathbf{p}_0) > -\infty$, repeat the following steps.

Step 1: set $\mathcal{J}_s = \{j_{s1}, \cdots, j_{sm_s}\}$, the index set of the positive elements of $\mathbf{p}_s$.

Step 2: compute $d_1(\mathbf{p}_s), \ldots, d_m(\mathbf{p}_s)$. If $\max_{1\le j\le m}\{d_j(\mathbf{p}_s)\} \le \tau$, stop.
Step 3: find $\mathcal{J}_s^* \equiv \{j_{s1}^*, \ldots, j_{s,m_s-1}^*\}$, where $j_{sl}^* = \arg\max_{j_{sl}\le j\le j_{s,l+1}}\{d_j(\mathbf{p}_s)\}$.
Step 4: compute $\mathbf{p}_{s+1}$ by solving problem (7), using only the elements indexed by $\mathcal{J}_s^+ \equiv \mathcal{J}_s \cup \mathcal{J}_s^*$, and by performing the line search (8).
Step 5: set $s = s + 1$.

The above dimension reduction technique can also be used in combination with other algorithms that can effectively find redundant support intervals, i.e., set their probability masses to zero. The dimension-reduced versions of the iterative convex minorant algorithm (Groeneboom, 1991; Groeneboom and Wellner, 1992; Jongbloed, 1998), the hybrid ICM-EM algorithm (Wellner and Zhan, 1997) and the subspace-based Newton method (Dümbgen et al., 2006) are also considered by Wang (2008), and all give enhanced performance.

3 The hierarchical constrained Newton method

3.1 A hierarchical formulation

Owing to the additivity of probability mass, the mixture model (1) can be rewritten in a hierarchical form as follows. First let us partition the index set $\mathcal{J}$ into disjoint subsets $J_1, \ldots, J_b$, and reformulate model (1) as follows:

$$f_i = \sum_{j=1}^{m} p_j \delta_{ij} = \sum_{k=1}^{b} \pi_k \left(\sum_{j\in J_k} \frac{p_j}{\pi_k} \delta_{ij} \right) = \sum_{k=1}^{b} \pi_k f_{ik},$$

where $\pi_k = \sum_{j\in J_k} p_j$ and $f_{ik} = \sum_{j\in J_k} \frac{p_j}{\pi_k}\delta_{ij}$. This turns the original mixture model with m components into one with only b components. Each new component f_{ik} is itself a mixture, so the model becomes a mixture of mixtures. Clearly we can carry on this process of reformulation for each f_{ik}, and for each of its new components, and so on, until there are insufficient elements left for further partitioning. Thus a hierarchy of mixture models is constructed.

With this hierarchical structure, a large-sized mixture model is rewritten into many much smaller-sized mixture models. This helps to break down the original optimization problem of dimension m into many that are identical in nature but of smaller dimension b. We can apply the CNM to each mixture model in the hierarchy, in a way described below.

Note that throughout the paper we always treat $\mathcal{J}$ as an ordered set and preserve the order in all groupings and partitions. This does not affect the eventual convergence of the proposed algorithm, but it is helpful for its speed since a maximal intersection interval is mostly correlated with its neighboring ones.

3.2 Allocation within a block

Consider an arbitrary subset (or block) B of $\mathcal{J}$. Write the mass allocated to it according to $\mathbf{p} \in \mathcal{P}$ as

$$\pi_B = \pi_B(\mathbf{p}) = \sum_{j\in B} p_j$$

and define the average gradient of B with $\pi_B > 0$ as

$$\bar{g}_B = \bar{g}_B(\mathbf{p}) = \frac{\sum_{j\in B} p_j g_j}{\pi_B}.$$

There are two special cases. For an atomic block $B = \{j\}$, $j \in \mathcal{J}$, we always define $\bar{g}_B = g_j$ even if $p_j = 0$; and for the entire set $\mathcal{J}$, it always holds that $\bar{g}_{\mathcal{J}} = n$ for any $\mathbf{p}$ with $\ell(\mathbf{p}) > -\infty$, because $\mathbf{p}^\top \mathbf{g} = \mathbf{p}^\top \mathbf{S}^\top \mathbf{1}$ and $\mathbf{S}\mathbf{p} = \mathbf{1}$.

Let B, with $\pi_B(\mathbf{p}) > 0$, be composite (as opposed to atomic) and have partition $P = \{B_1, \ldots, B_t\}$, where $B = \cup_{k=1}^t B_k$ and each subblock B_k is either atomic, or composite with a positive mass. In what follows, we restrict our attention to the non-overlapping case, namely $B_j \cap B_k = \emptyset$ if $j \ne k$. One may also allow the subblocks to overlap one another by dividing the mass of any shared component among overlapping subblocks, but tentative numerical studies suggest that this is computationally less efficient. With a (non-overlapping) partition P, we also have π_{B_k} and $\bar{g}_{B_k}$ for each subblock B_k, due to the above definitions. Let

$$\boldsymbol{\pi}_P = \boldsymbol{\pi}_P(\mathbf{p}) = (\pi_{B_1}, \ldots, \pi_{B_t})^\top,$$
$$\mathbf{g}_P = \mathbf{g}_P(\mathbf{p}) = (\bar{g}_{B_1}, \ldots, \bar{g}_{B_t})^\top,$$

and denote the maximum gradient of partition P by

$$g_P^* = g_P^*(\mathbf{p}) = \max_{1\le k\le t}\{\bar{g}_{B_k}\}.$$

One could increase the value of $\ell(\mathbf{p})$ by increasing the value of the blockwise log-likelihood

$$\ell_P(\boldsymbol{\pi}; \mathbf{p}) \equiv \sum_{i=1}^{n} \log\left\{ \sum_{k=1}^{t} \pi_k f_{ik} + \sum_{j\notin B} p_j \delta_{ij} \right\},$$

s.t. $\boldsymbol{\pi} \ge \mathbf{0}$ and $\boldsymbol{\pi}^\top \mathbf{1} = \pi_B$, where $f_{ik} = \sum_{j\in B_k} p_j \delta_{ij}/\pi_{B_k}$. The probability for observation O_i is now a function of $\boldsymbol{\pi}$:

$$f_i = f_i(\boldsymbol{\pi}) = \sum_{k=1}^{t} \pi_k f_{ik} + \sum_{j\notin B} p_j \delta_{ij}.$$

Let $f_{+k}(j) = p_j/\pi_{B_k}$, for $j \in B_k$. Then f_{+k} has unit mass and is a probability density (or mass) function defined on B_k given $\mathbf{p}$. These blockwise densities constitute part of the entire mixture distribution, with mixing proportions $\boldsymbol{\pi}$.

The constrained Newton method can be applied to maximizing $\ell_P(\boldsymbol{\pi}; \mathbf{p})$ in almost an identical manner as to maximizing $\ell(\mathbf{p})$. Here the unknown is $\boldsymbol{\pi}$, with the f_{ik}'s being held fixed. Denoting

$$\mathbf{s}_{iP} = (f_{i1}, \ldots, f_{it})^\top / f_i, \quad i = 1, \ldots, n,$$
$$\mathbf{S}_P = (\mathbf{s}_{1P}, \ldots, \mathbf{s}_{nP})^\top,$$

we have

$$\frac{\partial \ell_P(\boldsymbol{\pi}; \mathbf{p})}{\partial \boldsymbol{\pi}} = \mathbf{S}_P^\top \mathbf{1},$$
$$\frac{\partial^2 \ell_P(\boldsymbol{\pi}; \mathbf{p})}{\partial \boldsymbol{\pi} \partial \boldsymbol{\pi}^\top} = -\mathbf{S}_P^\top \mathbf{S}_P.$$

Similar to the Taylor series expansion (6) that leads to the least squares problem (7), we derive that maximizing $\ell_P(\boldsymbol{\pi}; \mathbf{p})$ over $\boldsymbol{\pi}$ can be achieved by solving the following least squares problem iteratively:

$$\min_{\boldsymbol{\pi}'} \quad ||\mathbf{S}_P \boldsymbol{\pi}' - \mathbf{S}_P \boldsymbol{\pi} - \mathbf{1}||^2, \quad \text{s.t.} \quad \boldsymbol{\pi}'^\top \mathbf{1} = \pi_B, \boldsymbol{\pi}' \geq \mathbf{0}. \tag{9}$$

This is a problem that can also be solved by Dax's (1990) method (see Appendix A).

Note that when $\boldsymbol{\pi} = \boldsymbol{\pi}_P$ $(= \boldsymbol{\pi}_P(\mathbf{p}))$, it always holds that

$$\mathbf{g}_P = \mathbf{S}_P^\top \mathbf{1}.$$

For a new estimate $\boldsymbol{\pi}'$ that replaces $\boldsymbol{\pi}_P$, each p_j for $j \in B_k$ is updated to

$$p'_j = \frac{\pi'_k p_j}{\pi_{B_k}}. \tag{10}$$

As a result, $\pi'_k = \pi'_{B_k}$ $(\equiv \pi_{B_k}(\mathbf{p}') = \sum_{j \in B_k} p'_j)$, which is desirable. For $\boldsymbol{\pi}'$ to be an ascent direction from $\boldsymbol{\pi}_P$, it must satisfy

$$\mathbf{g}_P^\top (\boldsymbol{\pi}' - \boldsymbol{\pi}_P) > 0,$$

or equivalently

$$\frac{\mathbf{g}_P^\top \boldsymbol{\pi}'}{\pi_B} - \bar{g}_B > 0.$$

Note that $\mathbf{g}_P^\top \boldsymbol{\pi}'/\pi_B$ is the average gradient with respect to $\boldsymbol{\pi}'$. Therefore, $\ell_P(\boldsymbol{\pi}; \mathbf{p})$ is maximal at $\boldsymbol{\pi}_P$, if and only if

$$g_P^* = \bar{g}_B;$$

i.e., the maximum gradient equals the average gradient.

3.3 Hierarchical allocation

With the above method for blockwise updating of mixing proportions, one can maximize $\ell(\mathbf{p})$ by redistributing the probability mass among blocks or, indeed, blocks of blocks. The performance of the resulting algorithm depends on how efficiently the probability mass is transferred. Of many possible ways of partitioning $\mathcal{J}$, it appears that using a hierarchical structure of partitions is most efficient at transferring probability mass, both locally and globally.

The building of a hierarchy starts with a given index set $\mathcal{J}_s^+ \subset \mathcal{J}$, such as that defined in step 4 of Algorithm 1, where $\pi_{\mathcal{J}_s^+} = 1$ and each $j \in \mathcal{J}_s^+$ corresponds to an atomic block. In the bottom layer of the hierarchy, the atomic blocks are grouped according to their neighborhood into, say, $t_{(1)}$ roughly equal-sized blocks. In the layer above it, neighboring blocks are further grouped into $t_{(2)}$ superblocks. One continues this grouping process, until a tree-like hierarchical structure of blocks is fully constructed, with a single superblock in the top layer. All blocks in each layer of the hierarchy have the total mass 1.

The constrained Newton method can be used to update the mixing proportions for the subblocks of each block in each layer. Let $\mathcal{J}_s^+$ have partition P with t blocks in a particular layer, and its kth block B_k has partition P_k, for $k = 1, \ldots, t$. We hence need to solve the following least squares problem:

$$\min_{\boldsymbol{\pi}'_{P_1}, \cdots, \boldsymbol{\pi}'_{P_t}} \quad \sum_{k=1}^{t} ||\mathbf{S}_{P_k} \boldsymbol{\pi}'_{P_k} - \mathbf{S}_{P_k} \boldsymbol{\pi}_{P_k} - \mathbf{1}||^2, \tag{11}$$

s.t. $\boldsymbol{\pi}'^\top_{P_k} \mathbf{1} = \pi_{B_k}$ and $\boldsymbol{\pi}'_{P_k} \geq \mathbf{0}$, for $k = 1, \ldots, t$. This problem breaks down to t individual problems of form (9), each of which can be solved quickly. While the order of updating the layers does not affect the convergence of the algorithm, in our implementation we chose to traverse all the layers from the bottom up and thus a recursive function call can be used. Certainly one does not have to find the exact solution that maximizes each $\ell_{P_k}(\boldsymbol{\pi}; \mathbf{p})$. Solving problem (11) approximately for each layer, using just one update step per traverse of the hierarchy, or perhaps two for higher-level layers, is overall more cost-effective.

For algorithmic convergence, one must allow sufficient mass transfer among the blocks during the computation. Hereby we enforce the following condition on all composite blocks B_k in the hierarchy:

$$\pi_{B_k}(\mathbf{p}) \geq \gamma, \tag{12}$$

for some pre-given $\gamma > 0$. This is a necessary condition for establishing convergence; see the proof of Theorem 1. Note that the $\mathbf{p}$ in condition (12) needs not be

$\hat{\mathbf{p}}$ but instead can be $\mathbf{p}_s$, the sth iterate for estimating $\mathbf{p}$. This makes the practical satisfaction of the condition very easy, e.g., by changing dynamically the block size or collapsing nearby ones. Since γ can be arbitrarily small, it does not make much difference in practice if one only restricts each composite block to having a positive mass.

Instead of performing a line search after updating each $\boldsymbol{\pi}_{P_k}$, which is costly, one could perform a single line search for all the blocks in a layer. Let each $\boldsymbol{\pi}_{P_k}$ be updated to $\boldsymbol{\pi}'_{P_k} = \boldsymbol{\pi}_{P_k} + \boldsymbol{\eta}_{P_k}$ by solving the least squares problem (11), and, accordingly, $\mathbf{p}$ to $\mathbf{p}' = \mathbf{p} + \boldsymbol{\eta}$ by using formula (10). One then chooses the smallest $u \in \{0, 1, 2, \dots\}$ that satisfies

$$\ell(\mathbf{p} + \sigma^u \boldsymbol{\eta}) - \ell(\mathbf{p}) \geq \alpha \sigma^u \mathbf{g}^\top \boldsymbol{\eta} = \alpha \sigma^u \sum_{k=1}^{t} \mathbf{g}_{P_k}^\top \boldsymbol{\eta}_{P_k}. \quad (13)$$

Since every $\boldsymbol{\eta}_{P_k}$ is an ascent direction due to solving problem (11), so is $\boldsymbol{\eta}$, a convex combination of ascent directions.

With such a hierarchy, the constrained Newton method is thus able to redistribute the probability mass, both locally via the blocks at the bottom, and among increasingly larger regions by climbing up the hierarchy. We summarize the above developments for the hierarchical constrained Newton method as follows.

Algorithm 2 (HCNM) The initialization and the first three steps are the same as in Algorithm 1.

Step 4: let partition $P^{(s,0)} = \{\{j\} : j \in \mathcal{J}_s^+\}$ and $l = 1$. Repeat the following substeps:

Step 4.1: build partition $P^{(s,l)}$ by grouping the neighboring blocks in $P^{(s,l-1)}$. (Make sure $\pi_B(\mathbf{p}_s) \geq \gamma$ for every $B \in P^{(s,1)}$.)

Step 4.2: for layer l, update $\mathbf{p}_{sl}$ (once or twice) by solving problem (11) and by performing line search (13).

Step 4.3: if $P^{(s,l)}$ has only one block, break; otherwise, set $l = l + 1$.

Step 5: let $\mathbf{p}_{s+1}$ be the final solution of step 4. Set $s = s + 1$.

From bottom up, the number of blocks in each layer decreases at an exponential rate and hence the vast majority of computation cost is for the bottom layer. For each traverse of the hierarchy, we choose to update $\mathbf{p}_s$ once for the bottom layer and twice for each of the other layers in Step 5 of Algorithm 2. This helps reduce the total number of iterations and improve overall performance, as compared with one update per layer per traverse.

3.4 Block size formula

The size of a block, namely the number of its subblocks, is important for the performance of the algorithm. Using larger blocks results in a smaller total number of blocks, which increases the efficiency of mass transfer per iteration, but at a higher cost for the constrained Newton method to update mixing proportions each time; and using smaller blocks does the opposite. In the numerical studies given in Section 5, the block size ranges roughly between 20 and 60, as determined by the following simple formula that only depends on $m_s^+ = |\mathcal{J}_s^+|$, the size of $\mathcal{J}_s^+$.

Motivated by a time complexity consideration (see Section 6), we determine the block size on the logarithmic scale of m_s^+. With a bit of rounding done afterwards for even partitioning, the block size for the entire hierarchy is determined by the formula

$$b = \max\{a_1, a_2 \log_2(m_s^+ / a_3)\}, \quad (14)$$

and a blockwise computation is only carried out when a subset has more than $1.5a_1$ elements. In our implementation, the default values are set to $a_1 = 20$, $a_2 = 10$ and $a_3 = 100$, which appears to work quite well for a range of problems that we studied. Table 1 gives a few block sizes computed from this formula with default settings. Note that the CNM can easily handle a problem of dimension up to 200, suggesting that the blockwise computation remains effective even if the size of the original problem is extremely large.

m_s^+	400	1600	6400	25600	$\cdots$	$\sim 10^8$
b	20	40	60	80	$\cdots$	200

Table 1 Block sizes determined by (14), with $a_1 = 20$, $a_2 = 10$ and $a_3 = 100$

4 Convergence analysis: an equilibrium approach

In this section, we establish the convergence of Algorithm 2 for computing the NPMLE. The proof below can be easily extended to other similar situations, e.g., when the support space is continuous or when blocks overlap one another. Moving probability mass between blocks can be compared to liquid flow under gravity between containers or to gas transfer from high concentration to low concentration, both of which will eventually settle down to their respective global equilibrium. Here in particular, probability mass is transferred from low average gradient areas to high average gradient areas,

in a discrete fashion by an algorithm, until the global equilibrium of uniform average gradient, which is n, is achieved.

With this analogy in mind, let us first define the equilibrium that characterizes a blockwise NPMLE. Recall that a partition P is always so chosen that each $B_k \in P$ is either atomic, or composite with $\pi_{B_k}(\mathbf{p}) > 0$. This ensures that $\bar{g}_{B_k}(\mathbf{p})$ is always defined. With some $\mathbf{p} \in \mathcal{P}$, block $B \subset \mathcal{J}$ is said to be at equilibrium for partition $P = \{B_1, \ldots, B_t\}$, if

$$\begin{cases} \bar{g}_{B_k}(\mathbf{p}) = \bar{g}_B(\mathbf{p}), & \text{if } \pi_{B_k}(\mathbf{p}) > 0; \\ \bar{g}_{B_k}(\mathbf{p}) \le \bar{g}_B(\mathbf{p}), & \text{otherwise,} \end{cases} \tag{15}$$

for $k = 1, \ldots, t$. This implies that there exists no ascent direction from $\boldsymbol{\pi}_P(\mathbf{p})$, so $\ell_P(\boldsymbol{\pi}; \mathbf{p})$ is maximized at $\boldsymbol{\pi}_P(\mathbf{p})$. It is basically the same NPMLE characterization condition (4) but applied to a block. As a result, B is at equilibrium for partition P, if and only if $g^*_P(\mathbf{p}) = \bar{g}_B(\mathbf{p})$.

We will also say that B is at global equilibrium, if it is at equilibrium for every partition of B or, equivalently, for the partition with all atomic blocks. Therefore, by applying this to $\mathcal{J}$, the following three statements are equivalent:

(a) $\hat{\mathbf{p}}$ maximizes $\ell(\mathbf{p})$;
(b) $g^*(\hat{\mathbf{p}}) = n$;
(c) With $\hat{\mathbf{p}}$, $\mathcal{J}$ is at global equilibrium.

In other words, the NPMLE is also characterized by the global equilibrium.

In the following, we establish the global convergence of any sequence created by Algorithm 2, by showing its convergence to each blockwise equilibrium and thus, due to sufficient mass transfer among the blocks, to the global equilibrium. The proofs are omitted for the next two lemmas, which are blockwise extensions of Lemmas 1 and 2 in Wang (2007) and can be established in virtually the same way. Denote $\mathcal{P}_0 = \{\mathbf{p} \in \mathcal{P} : \ell(\mathbf{p}) \ge \ell(\mathbf{p}_0) > -\infty\}$.

Lemma 1 *There exists an upper bound $U > 0$ such that, for any $B \subset \mathcal{J}$ with any partition P, and any direction $\boldsymbol{\eta}_P = \boldsymbol{\pi}_P(\mathbf{p}') - \boldsymbol{\pi}_P(\mathbf{p})$ with $\mathbf{p}' \in \mathcal{P}$ and $\mathbf{p} \in \mathcal{P}_0$,*

$$\boldsymbol{\eta}_P^\top \mathbf{S}_P^\top \mathbf{S}_P \boldsymbol{\eta}_P \le \pi_B^2 U \le U.$$

Lemma 2 *The backtracking search (13) always succeeds within a finite number of steps independent of any $\mathbf{p} \in \mathcal{P}_0$ and any partitions.*

Now let us define the convergence to a blockwise equilibrium, as specified by condition (15).

Definition 1 With a sequence $\{\mathbf{p}_s : \mathbf{p}_s \in \mathcal{P}_0\}$, $B \subset \mathcal{J}$ with $\pi_B(\mathbf{p}_s) > 0$ is said to approach the equilibrium for partition P, if for any $B_k \in P$ with mass bounded away from zero,

$$\lim_{s\to\infty} \{\bar{g}_{B_k}(\mathbf{p}_s) - \bar{g}_B(\mathbf{p}_s)\} = 0 \tag{16}$$

and for any $B_k \in P$ with mass approaching zero,

$$\lim_{s\to\infty} \max_k \{\bar{g}_{B_k}(\mathbf{p}_s) - \bar{g}_B(\mathbf{p}_s)\} \le 0. \tag{17}$$

Lemma 3 *With $\{\mathbf{p}_s : \mathbf{p}_s \in \mathcal{P}_0\}$, $B \subset \mathcal{J}$ approaches the equilibrium for partition P, if and only if*

$$\lim_{s\to\infty} \{g^*_P(\mathbf{p}_s) - \bar{g}_B(\mathbf{p}_s)\} = 0. \tag{18}$$

Proof First, let $g^*_P(\mathbf{p}_s) - \bar{g}_B(\mathbf{p}_s) \to 0$. Because

$$\begin{aligned} g^*_P(\mathbf{p}_s) - \bar{g}_B(\mathbf{p}_s) &= g^*_P(\mathbf{p}_s) - \sum_{k=1}^t \frac{\pi_{B_k}(\mathbf{p}_s)\bar{g}_{B_k}(\mathbf{p}_s)}{\pi_B(\mathbf{p}_s)} \\ &= \sum_{k=1}^t \frac{\pi_{B_k}(\mathbf{p}_s)}{\pi_B(\mathbf{p}_s)} \{g^*_P(\mathbf{p}_s) - \bar{g}_{B_k}(\mathbf{p}_s)\} \end{aligned}$$

and every term of the last expression has to be nonnegative, we have $g^*_P(\mathbf{p}_s) - \bar{g}_{B_k}(\mathbf{p}_s) \to 0$ for every $B_k \in P$ with mass bounded away from 0. Hence, condition (16) holds. Condition (17) also holds, because $g^*_P(\mathbf{p}_s)$ is maximal.

The converse is clearly true, by combining conditions (16) and (17).

The following theorem shows that the global equilibrium is achieved by achieving all blockwise equilibria. Note that hierarchies may differ from iteration to iteration. Also, the proof only requires the satisfication of the Armijo rule, and thus no exact optimization is needed for any partial problem.

Theorem 1 *Let $\hat{\mathbf{p}}$ maximize $\ell(\mathbf{p})$ and $\{\mathbf{p}_s\}$ be any sequence created by Algorithm 2. Then as $s \to \infty$,*

(a) every block in a hierarchy constructed by the algorithm approaches the equilibrium for its partition;
(b) $g^*(\mathbf{p}_s) \to n$;
(c) $\ell(\mathbf{p}_s) \to \ell(\hat{\mathbf{p}})$.

Proof By construction, $\ell(\mathbf{p}_s)$ increases monotonically and must converge to a finite value no greater than $\ell(\hat{\mathbf{p}})$. Let us denote by $\mathbf{p}_{sl}$ the iterate prior to updating $\mathbf{p}_s$ for layer l. Due to line search (13) and Lemma 2,

$$\ell(\mathbf{p}_{s,l+1}) - \ell(\mathbf{p}_{sl}) \ge \alpha\sigma^{\bar{u}} \sum_{k=1}^t \mathbf{g}_{P_k}(\mathbf{p}_{sl})^\top \boldsymbol{\eta}_{P_k}(\mathbf{p}_{sl}), \tag{19}$$

where $\bar{u}$ is an upper bound on the number of backtracking steps, which is independent of s and l.

To establish statement (a), we only need to establish limit (18) for every composite block B_k in the hierarchy which, by restriction (12), has $\pi_{B_k}(\mathbf{p}_{sl}) \geq \gamma > 0$. Assume that $\max_{k,l}\{g^*_{P_k}(\mathbf{p}_{sl}) - \bar{g}_{B_k}(\mathbf{p}_{sl})\}$ does not approach 0 as $s \to \infty$ and there must be infinitely many s such that $\max_{B_k,l}\{g^*_{P_k}(\mathbf{p}_{sl}) - \bar{g}_{B_k}(\mathbf{p}_{sl})\} \geq \tau$ for some $\tau > 0$. Then for such a block, it holds that, for any $0 \leq \epsilon \leq 1$,

$$\ell(\mathbf{p}_{s+1}) - \ell(\mathbf{p}_s) \geq \ell(\mathbf{p}_{s,l+1}) - \ell(\mathbf{p}_{sl}) \geq \alpha\sigma^{\bar{u}}(\epsilon\gamma\tau - \frac{\epsilon^2 U}{2}),$$

due to the optimality of $\boldsymbol{\eta}_{P_k}$ for solving problem (11) and Lemma 1; see the proof of Theorem 1 in Wang (2007) for a similar reasoning. Without loss of generality, assume $\gamma\tau/U \leq 1$ and let $\epsilon = \gamma\tau/U$. Then,

$$\ell(\mathbf{p}_{s+1}) - \ell(\mathbf{p}_s) \geq \frac{\alpha\sigma^{\bar{u}}\gamma^2\tau^2}{2U}, \tag{20}$$

which is positive and independent of s. This contradicts the Cauchy property for a convergent sequence. Therefore, from Lemma 3, the proof of statement (a) is completed.

Due to Definition 1, the average gradient of every block with a positive mass converges to the average gradient of its immediate superblock. Since $\bar{g}_{\mathcal{J}}(\mathbf{p}_s) = n$ for every $\mathbf{p}_s$ and there is a finite number of blocks, all of these average gradients must converge to n as $s \to \infty$. Because the atomic block that has $g^*(\mathbf{p}_s)$ is always included in the hierarchy, the proof of statement (b) is completed.

Statement (c) follows from inequality (5).

5 Numerical studies

This section presents the results of our numerical studies that compare the proposed HCNM with other algorithms available in the literature. Table 2 lists the algorithms included in the studies, all in Example 1 and some in Example 2. We did not implement the hybrid EM and univariate Newton-Raphson methods of Pilla and Lindsay (2001) that also use the hierarchical structure of a mixture model, since the EM algorithm is well-known for its slow convergence and we observe that the pairing of support points (i.e., block size = 2) is not an efficient way of exchanging probability mass. Pilla and Lindsay (2001) only numerically studied problems with up to 64 potential support points, whereas the problems studied below can have thousands of true support points. The studies in the following focus on relatively large-sized problems. It should be noted that for small-sized problems, these algorithms may have different relative performance and should all perform reasonably well to meet practical needs.

In our studies, each algorithm was terminated when the following condition was satisfied:

$$\frac{\max_{1\leq j\leq m}\{d_j(\mathbf{p}_s)\}}{|\ell(\mathbf{p}_s)|} \leq \tau. \tag{21}$$

We used $\tau = 10^{-5}$ for the faster algorithms, including SR, SBN, SBNDR, ICM-EM, ICMDR-EM, CNM and HCNM, and larger values of τ for the others. For a single problem, this stopping criterion is practically the same as that described in step 2 of Algorithms 1 and 2. The scaling of d_j here is to allow for the magnitude of the log-likelihood. Note that always $\ell(\mathbf{p}) < 0$, because $\prod_{i=1}^n f_i(\mathbf{p}) < 1$. This ensures the number of accurate digits in the log-likelihood and that the performance of an algorithm on different problems can be compared fairly. An extremely slow algorithm was also stopped if the number of iterations exceeded 10000. By the number of iterations, here and below, we mean that of the outermost loop of an algorithm. Having the number of iterations recorded helps understand the performance of an algorithm; for comparison of different algorithms, we prefer running times.

All computations were performed in R. In particular, we used the R packages "Icens" (Gentleman and Vandal, 2009) for VEM and GPM, and "MLEcens" (Maathuis, 2007) for the SR method, which was implemented in the C language with an R interface function. Slight modifications were made to their original code to produce and extract information needed for Table 3, as well as to enforce the same stopping criterion (21). "Icens" also provides an implementation of the intra-simplex direction method (ISDM) (Lesperance and Kalbfleisch, 1992). However, it uses VEM to update the probability vector that defines an intra-simplex, which is not what is described in the original paper. We found, without including the results here, that this version of ISDM had a very slow convergence. We wrote the R code for the other algorithms, except that the NNLS algorithm, needed internally by CNM and HCNM for solving a least squares problem with non-negativity constraint, was given in FORTRAN (Lawson and Hanson, 1974). The computer used for the studies has a 2.40GHz Intel Core 2 Duo processor.

Example 1 The study reported by Kumwenda et al. (2008) looked at the infection-free survival of infants being breast-fed by mothers with the HIV-1 virus in Malawi. Infants who were unfortunate enough to be already infected at birth were excluded from the study. More than 3000 infants were randomized into three groups and were monitored for up to two years. The three groups were a control group (receiving standard treatment) and two types of extended treatment. The

Abbreviation	*Method*	*References*
EM	Expectation-maximization	Turnbull (1974, 1976); Dempster et al. (1977)
GPM	Gradient projection	Wu (1978)
VEM	Vertex exchange	Böhning (1986)
ICM	Iterative convex minorant	Groeneboom (1991); Jongbloed (1998)
ICM-EM	Hybrid ICM and EM	Wellner and Zhan (1997)
SBN	Subspace-based Newton	Dümbgen et al. (2006)
SR	Support reduction	Groeneboom et al. (2008)
ICMDR	Dimension-reduced ICM	Wang (2008)
ICMDR-EM	Dimension-reduced ICM-EM	Wang (2008)
SBNDR	Dimension-reduced SBN	Wang (2008)
CNM	Dimension-reduced constrained Newton	Wang (2007, 2008)
HCNM	Hierarchical CNM	This paper

Table 2 Algorithms included in the studies

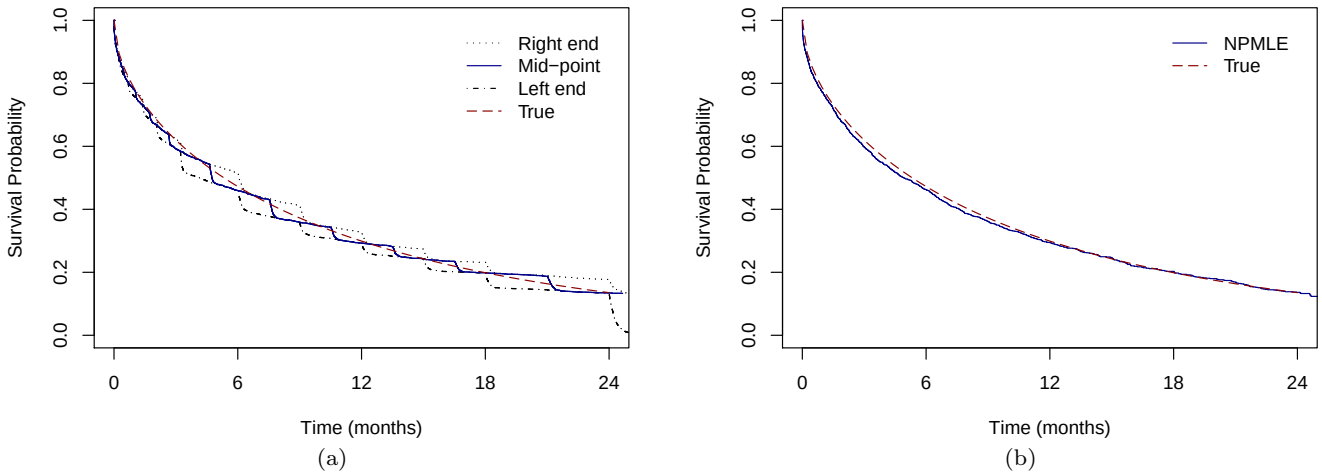

Fig. 1 The true and estimated survival functions: (a) Kaplan-Meier estimates (based on imputed data); (b) NPMLE.

Algorithm	s	$\frac{\ell(\hat{\mathbf{p}})-\ell(\mathbf{p}_s)}{\lvert\ell(\hat{\mathbf{p}})\rvert}$	$\frac{\max_j\{d_j(\mathbf{p}_s)\}}{\lvert\ell(\hat{\mathbf{p}})\rvert}$	*Time (s)*
ICM	10000	2.54×10^{-2}	3.05×10^{-1}	7518.5
ICMDR	10000	2.54×10^{-2}	3.05×10^{-1}	6770.3
VEM	2057	5.26×10^{-4}	1.00×10^{-2}	3602.1
GPM	1218	1.28×10^{-4}	9.96×10^{-3}	1875.9
EM	901	8.11×10^{-8}	1.06×10^{-4}	657.0
SR	151	1.61×10^{-10}	1.39×10^{-7}	390.5
CNM	13	1.74×10^{-15}	2.99×10^{-8}	287.5
SBN	43	3.14×10^{-8}	1.55×10^{-6}	127.2
SBNDR	19	1.39×10^{-9}	7.18×10^{-6}	32.8
ICM-EM	43	2.18×10^{-10}	8.65×10^{-6}	29.0
ICMDR-EM	43	1.50×10^{-10}	8.65×10^{-6}	27.4
HCNM	12	7.40×10^{-12}	7.01×10^{-6}	13.7

Table 3 Performance of algorithms on the data set in Example 1.

aim of the study was to test whether these extended treatments were more effective in preventing the transmission of the virus from the mother to the infant in the breast milk.

Fig. 1(a) illustrates an undesirable characteristic of imputation in this context. The illustration is based on simulated data with 3000 observations intended to follow the methods used by Kumwenda et al. (2008). In particular, two Weibull distributions, with shape parameters 0.5 and 0.9 respectively and the same scale parameter 730, were used to generate, respectively, the HIV infection times, which were further censored by the intervals between follow-up visits, and the exact times of unfortunate deaths. It was either the exact death

time or the censoring interval of an HIV infection that was recorded, in days, for each subject, whichever appeared first. To be more realistic, we also introduced a small amount of random delay to each of the follow-up inspection dates that were scheduled at 1, 3, 6, 9 and 14 weeks and at 6, 9, 12, 15, 18 and 24 months after birth. Each random delay had an exponential distribution with a rate equal to 5% of the length of the time interval between the current and the next follow-up. The three most common types of imputation were used: choosing the right end, midpoint and left end of the censoring intervals, respectively. Clearly the Kaplan-Meier survival curves do not model well the underlying survival, since their shapes exhibit an artifact of the study design. The scolloped patterns arise from the series of scheduled follow-up inspections, which determine the endpoints of the censoring intervals. In comparison, the NPMLE, shown in Fig. 1(b), makes more efficient use of the data and provides a smoother and more accurate estimate of the survival curve. Note that because of the existence of exact observations or the imputation that replaces intervel-censored observations with their exact endpoints, all maximal intersection intervals turn out to points precisely. Therefore, despite that we use rectangles to represent the nonuniqueness on I_j, they reduce to points and each estimated survival function is simply a curve.

The performance of the algorithms on the data set generated above is given in Table 3, in descending order of computation time, as well as roughly in ascending order of solution accuracy. The "exact" NPMLE $\hat{\mathbf{p}}$ used in the table was produced by the HCNM, and confirmed by other fast algorithms, through iterating indefinitely until the log-likelihood value failed to numerically increase, an indication that the machine precision was reached. Most algorithms took long times to converge, and some were so slow that larger values of τ had to be used to terminate them earlier. Although the CNM used nearly the least number of iterations, owing to its quadratic order of convergence, its computation time was not short, owing to the size of the problem. By contrast, the HCNM even used one less iteration and reduced the computation time dramatically, by 20-fold, which made it the fastest of all.

Example 2 In this study, we used the same scheme for data generation as described in Wang (2008). To generate a data set, n exact event times are first drawn independently from the exponential distribution with mean 1, and $r \times n$ $(0 \le r < 1)$ of them, randomly chosen, will remain uncensored. For each censored observation, a random sample of size 10 is drawn from the same exponential distribution, which divides $[0, \infty)$ into 11 disjoint subintervals. The subinterval that contains the exact event time replaces it. Clearly, when $r = 0$, all observations are purely interval-censored; and when $0 < r < 1$, there are both exact and interval-valued observations. This data generation scheme is perhaps less realistic than that used in Example 1, but it helps compare systematically the performance of the algorithms in different scenarios, which is the focus of the study here. From our computing experience, the general conclusions drawn here remain consistent with those when other data generation schemes are used.

Only ICMDR-EM, SBNDR, CNM and HCNM were included in this study. Except in a few cases where the computational cost is too high, we investigated the performance of the four algorithms in these situations: $(r, n) \in \{0\%, 10\%, 30\%, 50\%, 70\%, 90\%\} \times \{400, 1600, 6400\}$. Based on 20 replications in each situation, the experimental results are displayed in Table 4, including the mean and standard deviation (in parentheses) of both the number of iterations and computation time that were needed by each algorithm. Fig. 2 also shows the mean computation times of the algorithms plotted against r for each size of data set.

In all situations, the performance of the HCNM was pleasing. It was practically no worse than the best of its competitors in every situation, and clearly the fastest algorithm for large data sets and in situations with large proportions of exact observations. It was also clear that the relative performance gaps between the HCNM and other algorithms increased with n. Results for the CNM deteriorated rapidly as n or r increased. The SBNDR algorithm showed a similar, though less extreme, deterioration in these situations. Although the ICMDR-EM algorithm performed consistently worse than the HCNM, it is interesting to note that its mean computation time curve has a concave shape, even with a decreasing trend for large values of r. This can be explained by its use of a diagonal Hessian approximation that is increasingly accurate as r increases (Wang, 2008).

6 Concluding remarks

A new algorithm, called the hierarchical constrained Newton method (HCNM), is presented and studied for computing the NPMLE of a survival function. It uses a divide-and-conquer approach to break down the problem into small ones in a hierarchical structure, which are of the same nature as the original problem and can be solved quickly. The algorithm makes use of the "mixture of mixtures" structure of the problem and can be readily implemented as a recursive function. Its performance was consistently among the best in every case we studied.

Method	*#Iterations*	*Time (s)*	*#Iterations*	*Time (s)*	*#Iterations*	*Time (s)*
	$n = 400$					
	$r = 0\%$		$r = 10\%$		$r = 30\%$	
ICMDR-EM	26.6 (5.2)	0.102 (0.016)	36.5 (5.3)	0.180 (0.024)	34.5 (3.0)	0.303 (0.029)
SBNDR	19.9 (3.2)	0.094 (0.016)	14.6 (2.0)	0.090 (0.013)	9.6 (2.0)	0.110 (0.027)
CNM	7.3 (1.5)	0.044 (0.008)	7.0 (2.1)	0.063 (0.014)	5.5 (0.5)	0.134 (0.013)
HCNM	7.5 (1.1)	0.056 (0.009)	7.9 (1.6)	0.075 (0.013)	6.5 (0.6)	0.101 (0.010)
	$r = 50\%$		$r = 70\%$		$r = 90\%$	
ICMDR-EM	25.6 (1.5)	0.347 (0.019)	17.4 (1.5)	0.326 (0.027)	10.4 (1.2)	0.242 (0.030)
SBNDR	5.4 (0.7)	0.122 (0.013)	4.4 (0.7)	0.166 (0.034)	3.8 (0.4)	0.218 (0.047)
CNM	5.0 (0.2)	0.252 (0.014)	4.5 (0.6)	0.375 (0.060)	4.0 (0.5)	0.479 (0.070)
HCNM	5.7 (0.5)	0.127 (0.011)	4.7 (0.5)	0.138 (0.015)	4.7 (0.6)	0.175 (0.024)
	$n = 1600$					
	$r = 0\%$		$r = 10\%$		$r = 30\%$	
ICMDR-EM	26.8 (3.7)	1.03 (0.13)	59.4 (4.2)	3.44 (0.27)	43.5 (1.9)	5.21 (0.24)
SBNDR	32.0 (4.3)	1.51 (0.21)	16.8 (2.2)	1.43 (0.19)	6.6 (1.1)	1.85 (0.35)
CNM	7.4 (0.8)	0.59 (0.04)	5.8 (0.5)	1.19 (0.10)	5.0 (0.2)	5.38 (0.28)
HCNM	8.8 (1.3)	0.69 (0.07)	8.7 (1.4)	0.85 (0.12)	5.5 (0.5)	1.06 (0.10)
	$r = 50\%$		$r = 70\%$		$r = 90\%$	
ICMDR-EM	25.8 (1.6)	4.78 (0.26)	16.1 (1.4)	3.99 (0.33)	9.0 (0.6)	2.88 (0.21)
SBNDR	5.0 (0.7)	4.32 (0.92)	4.0 (0.2)	7.45 (0.79)	3.2 (0.4)	9.24 (3.90)
CNM	4.7 (0.5)	13.52 (1.61)	4.0 (0.2)	21.37 (1.53)	3.5 (0.5)	27.85 (5.66)
HCNM	4.5 (0.5)	1.34 (0.16)	4.1 (0.3)	1.64 (0.14)	4.0 (0.0)	2.07 (0.01)
	$n = 6400$					
	$r = 0\%$		$r = 10\%$		$r = 30\%$	
ICMDR-EM	27.3 (4.8)	14.5 (1.8)	94.2 (3.6)	78.7 (3.2)	44.5 (1.8)	126.7 (5.5)
SBNDR	49.9 (5.2)	32.6 (4.2)	13.2 (1.9)	28.3 (5.3)	5.3 (0.8)	105.4 (32.1)
CNM	8.5 (1.5)	10.7 (1.1)	5.7 (1.2)	60.2 (14.2)	4.4 (0.5)	327.5 (46.5)
HCNM	9.2 (0.9)	10.2 (0.7)	6.8 (0.9)	10.6 (1.2)	4.6 (0.5)	15.4 (1.8)
	$r = 50\%$		$r = 70\%$		$r = 90\%$	
ICMDR-EM	23.2 (0.6)	122.8 (3.9)	14.4 (1.2)	109.1 (9.2)	7.4 (0.8)	69.9 (8.1)
SBNDR	4.0 (0.2)	245.6 (25.7)	—	—	—	—
CNM	4.1 (0.3)	768.2 (75.9)	—	—	—	—
HCNM	4.2 (0.4)	23.4 (2.7)	3.5 (0.6)	27.9 (5.6)	3.0 (0.0)	32.9 (0.1)

Table 4 Results for Example 2: mean (standard deviation) number of iterations and computation time.

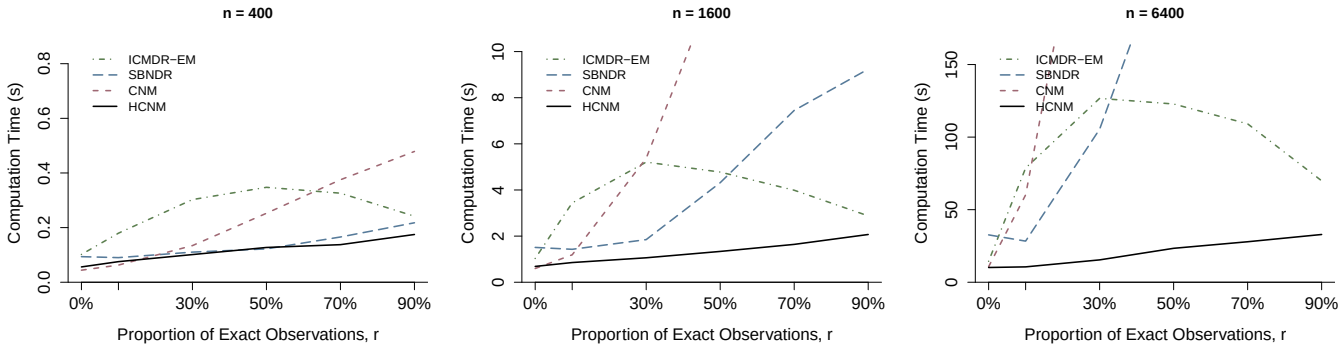

Fig. 2 Mean computation time, comparing HCNM with competing algorithms.

While specific details still need to be worked out, the HCNM can also be applied to the bivariate and, more generally, multivariate cases; see also Bogaerts and Lesaffre (2004) and Maathuis (2005) for fast algorithms for finding the maximal intersection regions in the multivariate case. In such cases, one only needs to partition the set of potential support regions (as opposed to intervals discussed in this paper) into a hierarchy of blocks, based on their neighborhood information, and possibly in a dynamic manner so that condition (12) is guaranteed. Then the rest of the computation can proceed very much in the same way.

The time complexities of CNM and HCNM depend mainly on that of the NNLS algorithm. While the worst-case time complexity of NNLS is perhaps much higher, in our extensive numerical studies it consistently exhibits $O(nm^2)$, where n is the number of rows and m ($\leq n$) the number of columns of the design matrix that is provided to NNLS as input. This is of the same order as solving an ordinary least squares problem. If this holds true, and since NNLS dominates the computational cost of CNM, the full-dimensional and the dimension-reduced CNM have, respectively, time complexities $O(nm^2)$ and $O(n\hat{m}^2 + nm)$ per iteration, m

being the number of maximal intersection intervals and $\hat{m}$ the number of support intervals of the NPMLE. By using the block size formula (14), it is easy to derive that HCNM has time complexity $O(n\hat{m}\log(\hat{m}) + nm)$ per iteration. If, in addition, HCNM does not take many more iterations than CNM, with supporting evidence given in Tables 3 and 4, it gives a dramatic reduction in time complexity over CNM. As a result, one can expect a remarkable performance gain in practice, especially when $\hat{m}$ is large. This, and the fact that for small $\hat{m}$, HCNM reduces to CNM which is already fast in this case, made it the only algorithm that performed best in all circumstances we studied. It is thus the best choice in practice where $\hat{m}$ is inevitably unknown beforehand.

Acknowledgments

The authors thank the two anonymous reviewers for constructive comments and Bruce Lindsay for helpful suggestions. This research was supported by a Marsden grant of the Royal Society of New Zealand (9145/3608546).

A Linear regression over a simplex

Consider the constrained least squares problem:

$$\min_{\mathbf{x}} \quad ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2, \qquad \text{s.t. } \mathbf{x}^\top \mathbf{1} = \delta, \mathbf{x} \geq \mathbf{0},$$

for $\delta > 0$. It can be solved by the NNLS algorithm of Lawson and Hanson (1974), after a transformation suggested by Dax (1990). Letting $\mathbf{y} = \mathbf{x}/\delta$ and $\mathbf{c} = \mathbf{b}/\delta$, it is apparent that the problem is equivalent to

$$\min_{\mathbf{y}} \quad ||\mathbf{A}\mathbf{y} - \mathbf{c}||^2, \qquad \text{s.t. } \mathbf{y}^\top \mathbf{1} = 1, \mathbf{y} \geq \mathbf{0},$$

which is further equivalent to

$$\min_{\mathbf{y}} \quad ||\mathbf{P}\mathbf{y}||^2, \qquad \text{s.t. } \mathbf{y}^\top \mathbf{1} = 1, \mathbf{y} \geq \mathbf{0}, \tag{22}$$

where $\mathbf{P} = \mathbf{A} - (\mathbf{c}, \ldots, \mathbf{c})$. The solution to problem (22) can be found by solving the following least squares problem with only non-negativity constraints:

$$\min_{\mathbf{y}} \quad ||\mathbf{P}\mathbf{y}||^2 + |\mathbf{y}^\top \mathbf{1} - 1|^2, \qquad \text{s.t. } \mathbf{y} \geq \mathbf{0}. \tag{23}$$

By relating the Karush-Kuhn-Tucker conditions for both problems, Dax established that if $\tilde{\mathbf{y}}$ solves problem (23), then $\tilde{\mathbf{y}}/\tilde{\mathbf{y}}^\top \mathbf{1}$ solves problem (22).

Problem (23) can be solved by the NNLS algorithm of Lawson and Hanson (1974).

References

Bogaerts, K. and Lesaffre, E. (2004). A new, fast algorithm to find the regions of possible support for bivariate interval-censored data. *Journal of Computational & Graphical Statistics* **13**, 330–340.

Böhning, D. (1986). A vertex-exchange-method in D-optimal design theory. *Metrika* **33**, 337–347.

Böhning, D., Schlattmann, P., and Dietz, E. (1996). Interval censored data: A note on the nonparametric maximum likelihood estimator of the distribution function. *Biometrika* **83**, 462–466.

Chen, L., Jha, P., Sirling, B., Sgaier, S. K., Daid, T., Kaul, R., and Nagelkerke, N. (2007). Sexual risk factors for HIV infection in early and advanced HIV epidemics in Sub-Saharan Africa: systematic overview of 68 epidemiological studies. *PLoS ONE* **2**, e1001.

Dax, A. (1990). The smallest point of a polytope. *Journal of Optimization Theory and Applications* **64**, 429–432.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Ser. B* **39**, 1–22.

Dümbgen, L., Freitag-Wolf, S., and Jongbloed, G. (2006). Estimating a unimodal distribution from interval-censored data. *Journal of the American Statistical Association* **101**, 1094–1106.

Gentleman, R. and Vandal, A. C. (2001). Computational algorithms for censored-data problems using intersection graphs. *Journal of Computational & Graphical Statistics* **10**, 403–421.

Gentleman, R. and Vandal, A. C. (2009). Icens: NPMLE for censored and truncated data. R package version 1.18.0.

Groeneboom, P. (1991). Nonparametric maximum likelihood estimators for interval censoring and deconvolution. Technical Report 378, Department of Statistics, Stanford University.

Groeneboom, P., Jongbloed, G., and Wellner, J. A. (2008). The support reduction algorithm for computing nonparametric function estimates in mixture models. *Scandinavian Journal of Statistics* **35**, 385–399.

Groeneboom, P. and Wellner, J. A. (1992). *Information Bounds and Nonparametric Maximum Likelihood Estimation*. Basel: Birkhäuser.

Jongbloed, G. (1998). The iterative convex minorant algorithm for nonparametric estimation. *Journal of Computational & Graphical Statistics* **7**, 301–321.

Kumwenda, N. I., Hoover, D. R., Mofenson, L. M., Thigpen, M. C., Kafulafula, G., Li, Q., Mipando, L., Nkanaunena, K., Mebrahtu, T., Bulterys, M., Fowler, M. G., and Taha, T. E. (2008). Extended antiretroviral prophylaxis to reduce breast-milk HIV-1 transmission. *New England Journal of Medicine* **359**, 119–129.

Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*. Prentice-Hall, Inc.

Lesperance, M. L. and Kalbfleisch, J. D. (1992). An algorithm for computing the nonparametric MLE of a mixing distribution. *Journal of the American Statistical Association* **87**, 120–126.

Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry and Applications*, Volume 5 of *NSF-CBMS Regional Conference Series in Probability and Statistics*. Institute for Mathematical Statistics: Hayward, CA.

Maathuis, M. H. (2005). Reduction algorithm for the NPMLE for the distribution of bivariate interval-censored data. *Journal of Computational & Graphical Statistics* **14**, 352–362.

Maathuis, M. H. (2007). MLEcens: Computation of the MLE for bivariate (interval) censored data. R package version 0.1-2.

Peto, R. (1973). Experimental survival curves for interval-censored data. *Applied Statistics* **22**, 86–91.

Pilla, R. S. and Lindsay, B. G. (2001). Alternative EM methods for nonparametric finite mixture models. *Biometrika* **88**, 535–550.

Siegfried, N., Clarke, M., and Volmink, J. (2005). Randomised controlled trials in Africa of HIV and AIDS: descriptive study and spatial distribution. *BMJ* **331**, 742.

Sun, J. (2006). *The Statistical Analysis of Interval-censored Failure Time Data*. Springer.

Turnbull, B. W. (1974). Nonparametric estimation of a survivorship function with doubly censored data. *Journal of the American Statistical Association* **69**, 169–173.

Turnbull, B. W. (1976). The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society, Ser. B* **38**, 290–295.

Wang, Y. (2007). On fast computation of the non-parametric maximum likelihood estimate of a mixing distribution. *Journal of the Royal Statistical Society, Ser. B* **69**, 185–198.

Wang, Y. (2008). Dimension-reduced nonparametric maximum likelihood computation for interval-censored data. *Computational Statistics & Data Analysis* **52**, 2388–2402.

Wellner, J. A. and Zhan, Y. (1997). A hybrid algorithm for computation of the nonparametric maximum likelihood estimator from censored data. *Journal of the American Statistical Association* **92**, 945–959.

Wong, G. Y. and Yu, Q. (1999). Generalized MLE of a joint distribution function with multivariate interval-censored data. *Journal of Multivariate Analysis* **69**, 155–166.

Wu, C. F. (1978). Some algorithmic aspects of the theory of optimal designs. *Annals of Statistics* **6**, 1286–1301.