Journal of Data Science **7**(2009), 365-380

Estimating Bivariate Survival Function by Volterra Estimator Using Dynamic Programming Techniques

Jiantian Wang and Pablo Zafra
Kean University

Abstract: For estimating bivariate survival function under random censorship, it is commonly believed that the Dabrowska estimator is among the best ones while the Volterra estimator is far from being computational efficiency. As we will see, the Volterra estimator is a natural extension of the Kaplan-Meier estimator to bivariate data setting. We believe that the computational 'inefficiency' of the Volterra estimator is largely due to the formidable computational complexity of the traditional recursion method. In this paper, we show by numerical study as well as theoretical analysis that the Volterra estimator, once computed by dynamic programming technique, is more computationally efficient than the Dabrowska estimator. Therefore, the Volterra estimator with dynamic programming would be quite recommendable in applications owing to its significant computational advantages.

Key words: Bivariate survival function, computational complexity, Dabrowska estimator, dynamic programming, Kaplan-Meier estimator, Volterra estimator.

1. Introduction

Bivariate failure time data arise when a survival study is done on paired units, such as twins, both eyes of one patient, or paired organs of the same individual. Nonparametric estimation of bivariate survival function under the presence of censoring is of great importance in applications (Prentice and Cai, 1992).

Let $T = (T_1, T_2)$ be a pair of nonnegative random variables representing the failure times with joint survival function $S(t_1, t_2) = P(T_1 \geq t_1, T_2 \geq t_2)$, and $C = (C_1, C_2)$ be a pair of censoring times which is independent of T. Under usual right censoring, the observable variables are,

$$X = (X_1, X_2) = (T_1 \wedge C_1, T_2 \wedge C_2), \text{ and } \Delta = (\Delta_1, \Delta_2) = \{I(T_1 \leq C_1), I(T_2 \leq C_2)\},$$

where $a \wedge b = \min(a, b)$ and $I(\cdot)$ is the indicator function. A primary objective in analyzing bivariate failure data is to estimate the bivariate survival

function $S(t_1, t_2)$ based on the observed data $X_i = (X_{1i}, X_{2i})$, $\Delta_i = (\Delta_{1i}, \Delta_{2i})$, $i = 1, \cdots, n$.

This estimating problem turns out to be surprisingly 'difficult' (Gill, 1992; Oakes, 2001), even though the corresponding problem in univariate case is well solved by the classical Kaplan-Meier estimator (Kaplan and Meier, 1958). Many statisticians have made their efforts in solving this problem. For example, Bickel (personal communication) suggested an estimator based on the Volterra representation of the survival function. Langberg and Shaked (1982) advocated an estimator by conditioning technique. Campbell and Földes (1982), Lo and Wang (1989), among others, investigated the properties of the Langberg-Shaked estimator. Muñoz (1980), Campbell (1981), Hanley and Parnes (1983) and van der Laan (1996) studied the nonparametric maximum likelihood estimation (NPMLE) of bivariate survival function by using Efron's (1967) self-consistency algorithm. Lin and Ying (1993), Wang and Wells (1997) and Tsai and Crowley (1998) considered this estimation problem under some special censoring mechanisms. Based on a product representation of bivariate survival function in terms of its conditional bivariate hazard measures, Dabrowska (1988) proposed her bivariate product-limit estimator. Prentice and Cai (1992) represented bivariate survival function by its marginal survival functions and its covariance function and proposed their estimator in accordance with such representation.

Extensive comparisons for these proposed estimators have been carried out by some researchers. For instance, Bakker (1990) compared the Dabrowska estimator and the Volterra estimator. van der Laan (1997) compared the Dabrowska estimator, the Prentice-Cai estimator, and the NPMLE of van der Laan. These researchers, among others, claimed that the Dabrowska estimator and the Prentice-Cai estimator are among the best ones while the Volterra estimator is far from being computational efficiency.

It has been shown that those estimators are all consistent under independent censoring. In comparing the performance of consistent estimators, we believe that the computational complexity could be a determining factor. We thus speculate that the 'inefficiency' of the Volterra estimator may be due to the unfavorable computational complexity of the traditional recursion method. If this estimator is computed by some more efficient algorithms, say, the dynamic programming technique, its computational complexity could be reduced, and thus the performance of this estimator would be improved significantly.

In this paper, we investigate the performances of the Volterra estimator, the Dabrowska estimator (DB), among others. For our convenience, we refer to the Volterra estimator computed by dynamic programming technique as the Dynamic Volterra estimator (DV), that computed by traditional recursion method as the Recursive Volterra estimator (RV). We show that the computational complexity

of the DB is $O(n^4)$, the complexity of the RV is NP or at least exponential, whereas that of the DV, is only $O(n^2)$. Therefore, the DV is of the least computationally complicated among these estimators and therefore, the performance of the DV would be better than even the DB.

To confirm our theoretical analysis, we conduct a number of numerical studies for comparing the performances and computational time costs of the DV, the DB, the RV, and the Lin-Ying estimator and the Independence estimator, which are nearly-optimal under some special data settings. Our simulation results show that the DV is quite efficient in accuracy and cost, its performance could even better than those nearly-optimal estimators. Therefore the DV would be the most recommendable one to applications due to its computational efficiency and the ease for implementation.

The following section describes the Dabrowska estimator, the Volterra estimator, the Lin-Ying estimator, and the Independence estimator, along with some short comments. Section 3 discusses the computational complexities of these estimators, while numerical studies are followed in Section 4. A short conclusion is then presented in Section 5.

2. Description of the Estimators

2.1. Dabrowska estimator

For a bivariate step function $F(t_1, t_2)$ satisfying $F(0, 0) = 1$, one has the following product representation,

$$\begin{aligned} F(t_1,t_2) &= F(0,t_2)F(t_1,0)\frac{F(t_1,t_2)}{F(0,t_2)}\frac{F(0,0)}{F(t_1,0)} \\ &= \prod_{0<v\le t_2}\frac{F(0,v)}{F(0,v^-)}\prod_{0<u\le t_1}\frac{F(u,0)}{F(u^-,0)}\prod_{\substack{0<u\le t_1\\0<v\le t_2}}\frac{F(u,v)}{F(u,v^-)}\frac{F(u^-,v^-)}{F(u^-,v)}. \end{aligned}$$

Based on this representation, Dabrowska (1988) proposed the following estimator,

$$\hat{S}_D(t_1,t_2) = \hat{S}(t_1,0)\hat{S}(0,t_2)\prod_{\substack{u\le t_1\\v\le t_2}}\Big[1-\frac{\hat{\Lambda}_{10}(du,v^-)\hat{\Lambda}_{01}(u^-,dv)-\hat{\Lambda}_{11}(du,dv)}{\{1-\hat{\Lambda}_{10}(du,v^-)\}\{1-\hat{\Lambda}_{01}(u^-,dv)\}}\Big], \tag{2.1}$$

where $\hat{S}(t_1, 0)$, $\hat{S}(0, t_2)$ are marginal Kaplan-Meier estimators for $S(t_1, 0)$, $S(0, t_2)$, and $\hat{\Lambda}_{11}(du, dv)$, $\hat{\Lambda}_{10}(du, v^-)$, $\hat{\Lambda}_{01}(u^-, dv)$ are empirical estimates of the respective probabilities that $T_1 = u$ and $T_2 = v$, that $T_1 = u$, and that $T_2 = v$ among pairs in the risk set $Y(u, v) = \sum_{j=1}^{n} I(X_{1j} \ge u, X_{2j} \le v)$, (see Dabrowska (1988) and Prentice and Cai (1992) for more details).

2.2. Volterra estimator

In order to show that the Volterra estimator is a natural extension of the Kaplan-Meier estimator in high dimension, we consider general multivariate data settings first. Define a partial-order in $\mathcal{R}^m = \{(x_1, x_2, \cdots, x_m)\}$ as follows: For two points $p = (p_1, p_2, \cdots, p_m)$, $q = (q_1, q_2, \cdots, q_m)$, we write $p \geq q$, $p \leq q$, $p < q$ and $p > q$ if these hold componentwise. Denote $H(t)$ as the bivariate Heaviside function defined as $H(t) = 1$ for $t \geq 0$ and $H(t) = 0$ otherwise.

For a function $f(x)$ in $\mathcal{R}^m$, denote

$$f(x_1, \ldots, \tilde{x}_i, \ldots, x_m) = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_m),$$

that is, replace the ith component by 0. Define the following projective operators:

$$\begin{aligned}
\Gamma^i f(x_1, x_2, \ldots, x_m) &= f(x_1, \ldots, \tilde{x}_i, \ldots, x_m), \\
\Gamma^{ij} f(x_1, x_2, \ldots, x_m) &= f(x_1, \ldots, \tilde{x}_i, \ldots, \tilde{x}_j, \ldots, x_m), \\
&\cdots \quad \cdots \quad \cdots \\
\Gamma^{12\cdots m} f(x_1, x_2, \ldots, x_m) &= f(\tilde{x}_1, \cdots, \tilde{x}_m) = f(0, \cdots, 0).
\end{aligned}$$

Let $\Gamma_i = \sum_{|I|=i} \Gamma^I$ for $I \subseteq \{1, 2, \cdots, m\}$ and Γ_0 be the identity operator.

Let $T_i = (T_{1i}, T_{2i}, \cdots, T_{mi})$ and $C_i = (C_{1i}, C_{2i}, \cdots, C_{mi})$ be multivariate variables in $\mathcal{R}^m$. Correspondingly, $t = (t_1, \cdots, t_m) \in \mathcal{R}^m$, is the multivariate time variable. Denote $N(t) = \sum_{i=1}^n I(T_i \leq t, T_i \leq C_i)$ and $Y(t) = \sum_{i=1}^n I(T_i \geq t, C_i \geq t)$. By Glivenko-Cantelli theorem, $n^{-1}N(t) \to \omega(t) = \Pr(T \leq t, T \leq C)$ and $n^{-1}Y(t) \to \pi(t) = \Pr(T \geq t, C \geq t)$. The independence assumption between C and T leads to an identity $dS(t)/S(t) = (-1)^m d\omega(t)/\pi(t)$, here $dS(t) = dS(t_1, \cdots, t_m) = \partial^m_{t_1, \cdots, t_m} S(t_1, \cdots, t_m)$. Consequently the estimating equation is

$$d\hat{S}(t)/\hat{S}(t^-) = (-1)^m dN(t)/Y(t),$$

or equivalently, $d\hat{S}(t) = (-1)^m \hat{S}(t^-) dN(t)/Y(t)$. Solving this multivariate difference equation, one obtains

$$\hat{S}(t) = (-1)^m \sum_{i=1}^n \hat{S}(T_i^-) \frac{H(t - T_i) I(T_i \leq C_i)}{Y(T_i)} + \sum_{k=1}^m (-1)^{k-1} \Gamma_k \hat{S}(t). \tag{2.2}$$

In univariate case, the estimator (2.2) reduces to

$$\hat{S}(t) = 1 - \sum_{i=1}^n \hat{S}(T_i^-) \frac{H(t - T_i) I(T_i \leq C_i)}{Y(T_i)} = \prod_{i=1}^n \Big\{ 1 - \frac{H(t - T_i) I(T_i \leq C_i)}{Y(T_i)} \Big\},$$

which is just the classical Kaplan-Meier product limit estimator (KME). Hence, we can regard (2.2) as a natural generalization of the KME in higher dimension.

In bivariate case, the estimator (2.2) becomes

$$\hat{S}(t) = \sum_{i=1}^{n} \hat{S}(T_i^-) \frac{H(t-T_i)I(T_i \le C_i)}{Y(T_i)} + \hat{S}(t_1, 0) + \hat{S}(0, t_2) - 1, \tag{2.3}$$

where $\hat{S}(t_1, 0), \hat{S}(0, t_2)$ are corresponding marginal Kaplan-Meier estimators of $S(t_1, 0)$, $S(0, t_2)$, respectively. This estimator is similar to the one proposed by Bickel (unpublished communication) who obtained it by solving a Volterra type equation. We thus refer to the estimator (2.2) as to multivariate Volterra estimator and the estimator (2.3) as Volterra estimator.

Conceive the value of $W_i = \hat{S}(T_i^-)H(t-T_i)I(T_i \le C_i)/Y(T_i)$ as the jump of $\hat{S}(t)$ at point T_i and regard $\sum_{k=1}^{m}(-1)^{k-1}\Gamma_k \hat{S}(t)$ as the boundary value of $\hat{S}(t)$, we can understand that the multivariate Volterra estimator simply evaluates the estimated value as the boundary values add or subtract the summation of preceding jumps as compensation.

2.3 Lin-Ying estimator

When censoring has a known simple structure, one can have a simple estimator for $S(t)$. For example, under univariate censoring, Lin and Ying (1993) proposed the following estimator for bivariate survival function,

$$\hat{S}_{LY}(t_1, t_2) = \frac{1}{n} \sum_{i=1}^{n} \frac{I(X_{1i} \ge t_1, X_{2i} \ge t_2)}{\hat{K}(t_1 \vee t_2)}, \tag{2.4}$$

where $t_1 \vee t_2 = \max(t_1, t_2)$ and $\hat{K}(\cdot)$ is the Kaplan-Meier estimator for $K(t) = P(C \ge t)$ applying to univariate data $\{X_i = \min(C_i, T_i), \Delta = I(C_i \le T_i)\}$ with $T_i = T_{1i} \vee T_{2i}$, $i = 1, \cdots, n$.

2.4 Independence estimator

When failure times T_1 and T_2 are independent, the bivariate survival function becomes $S(t_1, t_2) = P(T_1 \ge t_1, T_2 \ge t_2) = P(T_1 \ge t_1)P(T_2 \ge t_2) = S_1(t_1)S_2(t_2)$. Thus, a natural candidate for estimating the bivariate survival function would be the 'Independence' estimator

$$\hat{S}(t_1, t_2) = \hat{S}_1(t_1)\hat{S}_2(t_2), \tag{2.5}$$

where $\hat{S}_1(t_1) = \hat{S}(t_1, 0)$ and $\hat{S}_2(t_2) = \hat{S}(0, t_2)$ are corresponding marginal Kaplan-Meier estimators of $S(t_1, 0)$, $S(0, t_2)$, respectively.

2.5 Some comments

A common misunderstanding regarding on the Volterra estimator is that it might not use data efficiently. Suppose that a sample consists of 25% doubly uncensored observations, 25% doubly censored observations, 25% of singly censored observations along the x coordinate and 25% along the y coordinate. The Volterra estimator uses the marginal Kaplan-Meier estimators, and the jump points of the estimate coincide with the location of the doubly uncensored observations (25% of the sample). It seems to some researchers that the Volterra estimator use only of 25% of data. While the Dabrowska estimators use all doubly uncensored and singly censored observations, and thus it uses 75% of the data so it has to be more efficient.

However, this is not true. Those singly uncensored observations and even doubly censored observations are used in the estimation of the marginal Kaplan-Meier estimators in Volterra estimator.

When data size is small or at the points where the size of 'at risk' is small, the Volterra estimator could be negative while the Dabrowska estimator is always positive. This is also a reason that many researchers believe that the Dabrowska estimator is superior to the Volterra estimator. However, when sample size is small, none of them can give a good estimation.

A basic feature for survival function is its monotonicity. Dabrowska (1988) revealed that the DB can not guarantee its monotonicity. We find such anomaly is inevitable. Since from observed data, essentially only the hazard functions (and its functionals) are identifiable. As we all know, for univariate function, its monotonicity can be determined by its first derivatives. However, for multivariate function, its monotonicity cannot be determined by its partial derivatives. Thus, for multivariate survival function, its monotonicity cannot be determined by its hazard functions, and therefore, it is not surprising that the Dabrowska estimator, the Prentice-Cai estimator, the Lin-Ying estimator, as well as the Volterra estimator, all can not guarantee the monotonicity of the estimated multivariate survival function.

The Lin-Ying estimator and the Independence estimator are almost the optimal ones under the corresponding data structures. Unfortunately, unlike the Volterra estimator, they are not applicable to general random censored data. If the Volterra estimator is competitive to these estimators under the special data settings, then, it strongly suggests that the Volterra estimato could be the optimal one.

When sample size is large enough, these estimators will converge to the true survival function since all are consistent under independent censoring. For a consistent estimator, its performance largely depends on how fast it converges. In

comparing the performance of consistent estimators, the computational complexity could be a determining factor. In the next section, we give out the algorithms and analyze the complexity for these estimators.

3. Algorithms and Complexity Analysis

The algorithm for the Dabrowska estimator was described in Bakker (1990) and in van der Laan (1997). For completeness sake, we state it here again.

Step 1. Compute $\hat{S}(t_1, 0)$ and $\hat{S}(0, t_2)$ by the Kaplan-Meier estimator;

Step 2. For each lattice point (X_{1i}, X_{2j}), compute $R(i, j)$ which is defined as $R(i, j) = A/B$ where

$$\begin{aligned} A &= Y(X_{1i}, X_{2j})\{Y(X_{1i}, X_{2j}) - N_{10}(\delta X_{1i}, X_{2j}) - N_{01}(X_{1i}, \delta X_{2j}) \\ &\quad + N_{11}(\delta X_{1i}, \delta X_{2j}) \\ B &= Y(X_{1i}, X_{2j}) - N_{10}(\delta X_{1i}, X_{2j})\}\{Y(X_{1i}, X_{2j}) - N_{01}(X_{1i}, \delta X_{2j}) \end{aligned}$$

and $N(\delta x, y) = N(x^-, y) - N(x, y)$.

Step 3. Compute the estimate $\hat{S}_D(t_1, t_2)$ by the formula

$$\hat{S}_D(t_1, t_2) = \hat{S}(t_1, 0)\hat{S}(0, t_2) \prod_{(X_{1i}, X_{2j}) \le (t_1, t_2)} R(i, j).$$

It is clear from the formula $\hat{S}(t_1, 0) = \prod_{i=1}^{n} \{1 - \delta_{1i}/Y(X_{1i}, 0)\}$ that the computational complexity for step 1 is $O(n^2)$. The computational complexity for computing $R(i, j)$, on the average, is $O(n^2)$ since to evaluate the functions N_{10}, N_{01}, and N_{11} is $O(n^2)$. Consequently, the complexity for computing the Dabrowska estimator is $O(n^4)$.

Traditionally, the Volterra estimator is computed by recursive algorithm according to the recursion scheme (2.2). The algorithm can be simply described as:

RecursionVolterra(t_1, t_2)
if (t_1 or t_2 is 0), return $\hat{S}(t_1, 0)$ or $\hat{S}(0, t_2)$
else for $i = 1, \cdots, n$, if $T_i \le C_i$, result=result+RecursionVolterra$(T_i^-)/Y(T_i)$
result=result+ $\hat{S}(t_1, 0) + \hat{S}(0, t_2) - 1$, return result

Recursive method is desirable for programming but its computational complexity is usually terrible. Clearly, the complexity for the above algorithm is NP

or non-polynomial. One way to reduce the complexity of recursive scheme is using the so-called dynamic programming technique (see, Sahni, 1997, Chapter 15). The algorithm for computing the Volterra estimator by applying dynamic programming technique can be described as follows:

Step 1. Compute $\hat{S}(t_1, 0)$ and $\hat{S}(0, t_2)$ by the Kaplan-Meier estimator;

Step 2. Sort $\{X_i; i = 1, 2, \ldots, n\}$ lexicographically according to the first or second component and denote the sorted data by $\{X_{(1)}, \ldots, X_{(n)}\}$;

Step 3. By the recursive formula (2.3), compute the estimated value $\hat{S}(X_{(1)})$, $\hat{S}(X_{(2)}), \cdots$, until for all $X_{(i)} \leq t$, $i = 1, \ldots n$;

Step 4. Compute $\hat{S}(t)$ by (2.3).

The computational complexity for step 1 is $O(n^2)$, for step 2 is at most $O(n^2)$, for step 3 and 4 is also $O(n^2)$. Thus, the complexity for computing the Volterra estimator by dynamic programming technique is $O(n^2)$.

The computational complexities for the Lin-Ying estimator and the 'Independence' estimator are clearly $O(n^2)$.

4. Numerical Studies

To investigate the behavior of the Volterra estimators, we conduct a number of simulations with different survival and censoring distributions, like the Clayton distributions and the Pareto distributions. We compare the DV with DB and also with the Lin-Ying estimator and the Independence Estimator. We refer to the probability $P(\Delta_1 = 1, \Delta_2 = 1)$ as the probability for 'full observation'.

4.1 The Clayton model

Assume the pairs of failure times are distributed according to the Clayton (1978) bivariate exponential distribution with bivariate survival function as

$$S_\theta(t_1, t_2) = (e^{t_1/\theta} + e^{t_2/\theta} - 1)^{-\theta}.$$

Assume the censoring C follows an independent bivariate exponential model whose components are all exponential with mean 2. The failure times (T_1, T_2) in Clayton model can be obtained from uniform $(0, 1)$ variables using the transformation

$$T_2 = -\log(1 - U_2), \quad T_1 = \theta \log\{(1 - a) + a(1 - U_1)^{-(1+\theta)^{-1}}\},$$

where $a = (1 - U_2)^{-\theta^{-1}}$ and U_1, U_2 are independent uniform (0,1) variables (see, Prentice and Cai (1992)). In this simulation we compare the estimators DV, DB and RV, 500 iterations were carried out with each sample size 100. The results are summarized in Table 1.

From Table 1 we can see that all these three estimators perform quite well even though they all underestimate the true values slightly. Since they are all consistent estimators, they are asymptotically equivalent in theory. The difference among them is due to the roundoff errors and computational complexity. The computational time costs for the these estimators could be quite different. As we recorded in this simulation, the computational time costs for the DV, DB, and RV are 1, 7703, and 489 seconds respectively. Therefore, the DV is the most computationally efficient.

Table 1: Comparison between the Dynamic Volterra estimator (DV), the Dabrowska estimator (DB), and the Recursive Volterra estimator (RV) based on Clayton's exponential model with $\theta = 0.25$. The probability of 'full observation' is about 51.13%.

t_1	$t_2 = 0$	$t_2 = 0.2231$	$t_2 = 0.5108$	$t_2 = 0.9163$
0	1.00000	0.80003	0.60001	0.39999
	1.00000(0.0000)	0.79214(0.0421)	0.58543(0.0550)	0.38993(0.0590)
	1.00000(0.0000)	0.79126(0.0421)	0.58459(0.0552)	0.38912(0.0592)
	1.00000(0.0000)	0.79197(0.0421)	0.58518(0.0551)	0.38962(0.0591)
0.2231	0.80003	0.71242	0.57487	0.39639
	0.78808(0.0423)	0.70010(0.0468)	0.56156(0.0532)	0.38892(0.0577)
	0.78655(0.0430)	0.69687(0.0488)	0.55799(0.0549)	0.38542(0.0581)
	0.78775(0.0425)	0.70208(0.0458)	0.56456(0.0518)	0.39226(0.0568)
0.5108	0.60001	0.57487	0.51307	0.38444
	0.58446(0.0552)	0.55935(0.0554)	0.49819(0.0550)	0.37549(0.0569)
	0.58308(0.0557)	0.55667(0.0568)	0.49542(0.0550)	0.37149(0.0563)
	0.58401(0.0553)	0.56228(0.0540)	0.50347(0.0525)	0.38219(0.0550)
0.9163	0.39999	0.39639	0.38444	0.33744
	0.38724(0.0583)	0.38267(0.0582)	0.37280(0.0577)	0.32682(0.0608)
	0.38634(0.0582)	0.38110(0.0583)	0.36941(0.0568)	0.32343(0.0544)
	0.38678(0.0583)	0.38591(0.0570)	0.37938(0.0556)	0.33695(0.0583)

The first row in each cell is the true value; the second, third and fourth rows are the mean (standard deviation) of the estimates by DV, DB, and RV respectively, 500 samples, each with size 100.

4.2 Univariate censoring

In this subsection, we consider an interesting censoring situation, that is, univariate censoring. In univariate censoring, both failure times may be censored by one censoring variable. Under univariate censoring, the Lin-Ying estimator is of the simplest form and is of least computational complexity.

In the followed simulations, we still use the Clayton model for failure times. We assume $\theta = 0.50$ and the univariate censoring follows an exponential distribution with mean 2. Lin and Ying (1993) conducted a simulation to compare their estimator with the Dabrowska estimator with sample size 60. In this simulation study, we also choose sample size 60. 1000 iterations are performed and Table 2 summarizes the results.

Table 2: Comparison among the DV, LY and DB under univariate censoring based on Clayton's exponential model with $\theta = 0.50$. Under such model, the probability for 'full observation' is about 61.11%.

t_1	$t_2 = 0$	$t_2 = 0.2231$	$t_2 = 0.5108$	$t_2 = 0.9163$
0	1.00000	0.80003	0.60001	0.39999
	1.00000(0.0000)	0.80142(0.0524)	0.59859(0.0666)	0.39527(0.0758)
	1.00000(0.0000)	0.80081(0.0536)	0.59759(0.0700)	0.39335(0.0790)
	1.00000(0.0000)	0.80160(0.0523)	0.59858(0.0666)	0.39495(0.0757)
0.2231	0.80003	0.68603	0.54717	0.38313
	0.79800(0.0544)	0.68547(0.0625)	0.54602(0.0671)	0.37944(0.0741)
	0.79597(0.0557)	0.68477(0.0646)	0.54498(0.0715)	0.37754(0.0786)
	0.79767(0.0546)	0.68420(0.0629)	0.54369(0.0679)	0.37603(0.0745)
0.5108	0.60001	0.54717	0.46853	0.35294
	0.59351(0.0684)	0.54284(0.0689)	0.46388(0.0680)	0.34765(0.0729)
	0.59213(0.0701)	0.54277(0.0734)	0.46319(0.0719)	0.34592(0.0779)
	0.59283(0.0686)	0.54126(0.0696)	0.46205(0.0696)	0.34445(0.0736)
0.9163	0.39999	0.38313	0.35294	0.29488
	0.39088(0.0749)	0.37429(0.0741)	0.34654(0.0711)	0.28858(0.0712)
	0.39052(0.0786)	0.37530(0.0790)	0.34628(0.0758)	0.28712(0.0734)
	0.39010(0.0751)	0.37238(0.0744)	0.34291(0.0711)	0.28317(0.0705)

The first row in each cell is the true value; the second, third and fourth rows are the mean (standard deviation) of the estimates by the DV, LY, and DB respectively, 1000 samples, each with size 60.

Under univariate censoring, the Lin-Ying (LY) estimator is of the simplest form and is of least computational complexity, and thus can be expected to perform the best. However, Table 2 shows that the DV outperforms the Lin-Ying estimator almost everywhere, even at those marginal points. We notice that $\hat{S}_{LY}(t_1, 0)$ is not the marginal Kaplan-Meier estimator for $S(t_1, 0)$, but the DV is. Since the DV is a natural extension of the Kaplan-Meier estimator, and from the optimality of the Kaplan-Meier estimator, we can expect the DV would be the optimal one in bivariate data settings. This simulation supports such speculation. Lin and Ying (1993) compared the Lin-Ying estimator and the DB and concluded that Lin-Ying estimator performs better than the DB. Thus, the DV performs better than DB consequently. The Table also indicates that the Lin-Ying estimator outperforms the DB excluding those marginal points on which DB reduces to the Kaplan-Meier estimator. Roughly, we can say that in this simulation, the performance order for these three estimators is: the DV, the

Lin-Ying, and the DB. The computational time costs for these estimators are 2, 1, and 2078 seconds, respectively, for the DV, LY, and DB.

4.3 Independent failure times

When failure times T_1 and T_2 are independent, the bivariate survival function is $S(t_1, t_2) = S_1(t_1)S_2(t_2)$. Thus, a 'good' estimator would be the 'Independence' estimator $\hat{S}(t_1, t_2) = \hat{S}_1(t_1)\hat{S}_2(t_2)$, where $\hat{S}_i(t_i)$ is the corresponding Kaplan-Meier marginal estimator for $S_i(t_i)$, $(i = 1, 2)$. In the simulation study below, we compare the performances of the 'Independence' estimator, the DV and DB when failure times are independent. The results are displayed in Table 3.

Table 3: Comparison among the 'Independence' estimator, the DV, and the DB under independent failure times model. The probability for 'full observation' is about 54.08%.

t_1	$t_2 = 0$	$t_2 = 0.2231$	$t_2 = 0.5108$	$t_2 = 0.9163$
0	1.00000	0.80003	0.60001	0.39999
	1.00000(0.0000)	0.79666(0.0558)	0.59681(0.0682)	0.39298(0.0690)
	1.00000(0.0000)	0.79789(0.0557)	0.59826(0.0683)	0.39555(0.0692)
	1.00000(0.0000)	0.79670(0.0558)	0.59694(0.0681)	0.39320(0.0691)
0.2231	0.80003	0.64005	0.48003	0.32001
	0.79780(0.0561)	0.63718(0.0663)	0.47738(0.0658)	0.31408(0.0590)
	0.79859(0.0561)	0.63813(0.0687)	0.47816(0.0716)	0.31521(0.0676)
	0.79796(0.0561)	0.63735(0.0688)	0.47719(0.0717)	0.31285(0.0669)
0.5108	0.60001	0.48003	0.36001	0.24000
	0.59437(0.0697)	0.47477(0.0669)	0.35563(0.0594)	0.23397(0.0493)
	0.59504(0.0697)	0.47510(0.0710)	0.35623(0.0687)	0.23417(0.0629)
	0.59446(0.0696)	0.47504(0.0705)	0.35764(0.0684)	0.23371(0.0611)
0.9163	0.39999	0.32001	0.24000	0.15999
	0.39199(0.0689)	0.31309(0.0605)	0.23430(0.0490)	0.15420(0.0383)
	0.39358(0.0686)	0.31660(0.0679)	0.23886(0.0626)	0.15715(0.0611)
	0.39210(0.0689)	0.31473(0.0676)	0.23717(0.0619)	0.15405(0.0553)

The first row in each cell is the true value; the second, third and fourth rows are the mean (standard deviation) of the estimates by the 'Independence' estimator, the DV, and DB respectively. 500 samples, each with size 60.

In this simulation, the pairs of failure times are unit exponential distributed, while censoring times also follow independent exponential distributions with mean 2. 500 simulations were carried out with sample size 60 in each.

From Table 3 we can see that, except those marginal points, the standard deviation of the 'Independence' estimator is less than those of the DV and the DB. Therefore, the performance of the 'Independence' estimator would be more stable.

Also, as we can see from the table that the estimated value from the DV is closer to the true value than that from the 'Independence' estimator. This

strongly suggests that the DV is likely the optimal one, since the 'Independence' estimator is almost the best one from the optimality of the Kaplan-Meier estimator (Wellner, 1982). However, the product of optimal estimators may not always be optimal. Clearly, when there is no censoring, $\hat{S}_1(t_1)$ and $\hat{S}_2(t_2)$ become empirical, but their product is not necessary to be empirical for $S(t_1, t_2)$. For computational time cost, the 'Independence' estimator is 1 second, while those for the DV and the DB are 2, and 1021 seconds respectively.

4.4 The Pareto model

Denote G as the distribution function of the censoring variable $C = (C_1, C_2)$. Assume that both the functions $S(t_1, t_2)$ and $G(c_1, c_2)$ are member of the family of Pareto distribution, defined as $S_a(t_1, t_2) = ((1-t_1)^{-1/a} + (1-t_2)^{-1/a} - 1)^{-a}$ and $G_b(c_1, c_2) = (c_1^{-1/b} + c_2^{-1/b} - 1)^{-b}$. The generation for these bivariate Pareto distributions is described in Johnson(1987). Specifically, the generation for the Pareto distribution with parameter 1 can be easily fulfilled as: Let Y_1, Y_2 and Y be independent and exponentially distributed variables. Set $T_1 = Y/(Y + Y_1)$, $T_2 = Y/(Y + Y_2)$, then (T_1, T_2) follows the Pareto distribution with parameter 1. When $a \to \infty$ and $b \to \infty$, $S_a(t_1, t_2) = (1-t_1)(1-t_2)$ and $G_b(c_1, c_2) = c_1 c_2$. In such a case, the components are independent. For conciseness, we just check the Pareto distribution with $(a, b) = (1, 1)$. Bakker (1990) compared the RV and DB under Pareto models. He used sample size $n = 100$ and iteration 100. Here we use the same sample size and the iteration number as Bakker (1990). The results are displayed in Table 4.

Table 4: The estimated values and the mean standard deviation for the Dynamic Volterra (DV), the Dabrowska (DB), and the Recursive Volterra (RV) estimator, based on 100 replicates with each sample size $n = 100$, both the failure times and the censoring times follow Pareto(1).

Point	True	DV	DB	RV	SD(DV)	SD(DB)	SD(RV)
(0.1,0.1)	.818181	.820823	.738927	.823226	.041451	.262883	.041118
(0.25,0.25)	.599999	.607002	.546726	.610586	.053955	.197185	.053952
(0.25,0.5)	.428571	.434905	.393033	.441395	.057905	.147253	.059430
(0.25,0.75)	.230769	.246639	.217276	.254100	.077464	.103619	.079034
(0.5,0.5)	.333333	.342007	.307738	.354070	.060356	.118329	.062130
(0.5,0.75)	.199999	.214823	.181800	.231701	.076353	.094270	.081113
(0.75,0.75)	.142857	.144711	.110214	.173648	.113205	.094905	.116535
(0.1,0.9)	.098901	.091908	.084291	.099271	.078897	.078039	.077617
(0.8,0.8)	.111111	.048398	.100316	.161381	.079890	.047083	.056394
(0.9,0.9)	.052631	-.002688	.003960	.039868	.170432	.051097	.204041

The column DV, DB, and RV are the estimate values from the Dynamic Volterra estimator, the Dabrowska estimator and the Recursive Volterra estimator respectively. The SD columns are the values of sample standard deviations for the corresponding estimators.

Table 4 reveals that in general, a strict performance order among the three estimators is elusive. We notice that in this model, censoring is quite heavy. Actually, the probability of full observation is only about 30%. Under such heavy censoring, desirable performance from these estimators may not be so likely, even some anomaly (such as non-monotonicity, negativeness) may occur. It is not so proper to make a conclusion that the Dabrowska estimator is better than the Volterra estimator simply the former is positive while the latter could be negative.

Caution should be taken in the situation of 'heavy censoring'. Fortunately, in most applications, censoring may not be so heavy (see, McGilchrist and Aisbett (1991)). The computational time costs for these estimators are < 1, 2587, and 23 seconds for the DV, DB and RV respectively. Thus, again, from the point of view of computational time cost, the DV is the best among these three estimators in term of computational time cost.

4.5 Computational time costs

As claimed, the computational complexity of the Dynamic Volterra estimator, the Dabrowska estimator and the Recursive Volterra estimator are respectively $O(n^2)$, $O(n^4)$, and NP. To confirm such a conclusion, we carry out the following simulation, in which we record the time costs for executing these estimator when T follows the bivariate Clayton distribution with parameter 0.25 and C follows the bivariate independent exponential distribution with both means 2. The computation is based on 100 iterations with different sample size n. The following table lists the results.

Table 5: Computational time costs (in second, < 1 is recoreded as 0)

n	20	40	60	80	100	120	140	160	180	200
DV	0	0	0	0	0	1	2	4	4	5
DB	3	37	187	569	1353	2783	5240	8587	13598	21078
RV	0	0	5	39	182	797	3478	10175	39574	104397

From this table, we can see that the DV is extremely computationally efficient. When simple size n is about 200, the computational cost for this estimator is only 0.023% of that for the DB and 0.005% of that for the RV. When n is less than about 150, the DB is more computational time cost than the RV, whereas when n become larger than about 150, the RV is more computational time cost than the DB. When n increase from 180 to 200, the computational cost for the DB increase from 13598 to 21078. This fact is well agreeable with the claim that the computational complexity of the DB is $O(n^4)$, since $(200/180)^4 \times 13598 = 20725$, which is quite close to 21078.

5. Conclusion

Estimating multivariate survival function under random censorship is a crucial problem in multivariate analysis. The Volterra estimator has long been ignored somehow. One reason, we believe, is due to its formidable computational complexity by using the traditional recursive method. In this paper we show that by using dynamic programming technique, the Volterra estimator (DV as we refer to) turns out to be much computationally simpler even than the Dabrowska estimator (DB), which has won the highest praise so far. Therefore, when assessing an estimator, the computational complexity and the efficiency of the algorithm also should be take into consideration.

As we can see from (2.2), the Volterra estimator is a natural generalization of the Kaplan-Meier estimator to high dimension, and thus it would be optimal as the Kaplan-Meier estimator. Our numerical studies show that the DV outperforms DB and even the Lin-Ying estimator and the 'Independence' estimator under special censoring settings. These observations strongly suggest the 'optimality' of the DV. However, the theoretical proof for the optimality of the DV is not so straightforward.

When censoring is heavy or data set is small, desirable estimation is not easily attainable. But with moderate censoring, as shown in our numerical simulations, the Volterra estimator with dynamic programming would be the most recommendable to applications due to its clear computational advantage over the Dabrowska estimator, the Prentice-Cai estimator, and other proposed estimators.

Acknowledgment

The authors are very grateful to the referee for his/her valuable comments. Also, the authors wish to thank the Editor for bringing a paper of Professor Lo and Professor Wang (Lo and Wang (1989)) to our attention. This research was supported by the UFRI award and the SpF award from Kean University.

References

Bakker, D. M. (1990). Two nonparametric estimators of the survival function of bivariate right censored observations, report BS-R9035. Center for Mathematics and Computer Science, Amsterdam.

Campbell, G. (1981). Nonparametric bivariate estimation with randomly censored data. *Biometrika* **68**, 417-422.

Campbell, G. and Földer, A. (1982). Large-sample properties of nonparametric bivariate estimators with censored data. In *Nonparametric Statistical Inference* (Edited by B. V. Gnedenko, M. L. Puri and I. Vincze, eds.) 103-122. North-Holland.

Clayton, D. G. (1978). A model for association in bivariate life table and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika* **65**, 141-51.

Dabrowska, D. M. (1988). Kaplan-Meier estimate on the plane. *Ann. Statist.* **16**, 1475-89.

Efron, B. (1967). The two-sample problem with censored data. *proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **4**, 831-853.

Gill, R. D. (1992). Multivariate survival analysis. *Theory Prob. Applic.* **37**, 18-31.

Gill, R. D., van der Laan, Mark J. and Wellner, J. A. (1995). Inefficient estimators of the bivariate survival function for three models. *Annales de l'Institut Henri Poincare, Section B, Calcul des Probabilities et Statistique* **31**, 545-597

Hanley, J. A. and Parnes. M. N. (1983). Nonparametric estimation of a multivariate distribution in the presence of censoring. *Biometrics* **39**, 129-139.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *J. Am. Statist. Assoc.* **53**, 457-481.

Langberg, N. A. and Shaked. M. (1982). On the identifiability of multivariate life distribution functions. *Ann. Statist.* **10**, 773-779.

Lin, D, Y. and Ying, Z. (1993). A simple nonparametric estimator of the bivariate survival function under univariate censoring. *Biometrika* **80**, 573-581.

Lo, S. H. and Wang, J. L. (1989). I.I.D representations for the bivariate product limit estimators and the bootstrap versions. *J. Multivariate Analysis* **28**, 211-226.

McGilchrist, C. A. and Aisbett, C. W. (1991). Regression with frailty in survival analysis. *Biometrics* **47**, 461-466.

Muñoz, A. (1980). Nonparametric estimation from censored bivariate observations. Technical Report, Dept. Statistics, Stanford Univ.

Oakes, D. (2001). Biometrika Centenary: Survival analysis. *Biometrika* **88**, 90-142.

Prentice, R. L. and Cai, J. (1992). Covariance and survival function estimation using censored multivariate failure time data. *Biometrika* **79**, 495-512.

Sahni, S. (1997). *Data Structures, Algorithms, and Applications in C++.* McGraw-Hill.

Tsai, W-Y. and Crowley, J. (1998). A note on nonparametric estimators of the bivariate survival function under univariate censoring. *Biometrika* **85**, 573-580.

van der Laan, M. J. (1996). Efficient estimation in the bivariate censoring model and repairing NPMLE. *Ann. Statist.* **24**, 596-627.

Wang, W. and Wells, M. (1997). Nonparametric estimators of the bivariate survival function under simplified censoring conditions. *Biometrika* **84**, 863-880.

Wellner, J. A. (1982). Asymptotic optimality of the product limit estimator. *Ann. Statist.* **10,** 595-602.

Received November 21, 2007; accepted December 19, 2008.

Jiantian Wang
Mathematics Department
Kean University
Union, NJ 07083, USA
jwang@kean.edu

Pablo Zafra
Mathematics Department
Kean University
Union, NJ 07083, USA
pzafra@kean.edu