Hazard Estimation With Bivariate Survival Data and Copula Density Estimation

Chong Gu

Bivariate survival function can be expressed as the composition of marginal survival functions and a bivariate copula and, consequently, one may estimate bivariate hazard functions via marginal hazard estimation and copula density estimation. Leveraging on earlier developments on penalized likelihood density and hazard estimation, a nonparametric approach to bivariate hazard estimation is being explored in this article. The new ingredient here is the nonparametric estimation of copula density, a subject of interest by itself, and to accommodate survival data one needs to allow for censoring and truncation in the setting. A simple copularization process is implemented to convert density estimates into copula densities, and a cross-validation scheme is devised for density estimation under censoring and truncation. Empirical performances of the techniques are investigated through simulation studies, and potential applications are illustrated using real-data examples and open-source software.

Key Words: Copularization; Cross-validation; Penalized likelihood; Smoothing parameter.

1. INTRODUCTION

Hazard estimation using censored lifetime data is among routine tasks in survival analysis. For standard univariate right-censored lifetime data with possible left-truncation and covariate, one may employ, among others, the penalized likelihood approach to the task (Gu 1994, 1996). In this article, we explore an approach to hazard estimation using paired lifetime data. The approach is fully nonparametric, leveraging on earlier developments on univariate hazard estimation and multivariate density estimation.

For the estimation of bivariate survival function, one has the Kaplan-Meier estimate that does not even assume continuity (Dabrowska 1988), and one has the frailty models (see Hougaard 1986; Oakes 1989) that are largely parametric. The approach we explore here sits somewhere in between, in that we do assume the existence of density with sufficient smoothness, but we do not assume any parametric form for the entities involved. A bivariate survival function can be expressed as the composition of univariate survival functions and a bivariate copula, and its estimation naturally decomposes into two phases: (i) univariate
hazard estimation, a solved problem; and (ii) copula density estimation, a problem needing new treatment. The extra information carried in bivariate survival models beyond marginal models is the association between the two time axes, which is characterized by the copula density in our setting.

Copula density estimation is by itself a subject of interest, and has seen treatments in the literature via kernel density estimation (Gijbels and Mielniczuk 1990), Bernstein polynomials (Sancetta and Satchell 2004), penalized hierarchical B-splines (Kauermann, Schellhase, and Ruppert 2013), etc. In the setting of bivariate hazard estimation, one needs to accommodate censored/truncated samples, for which none of the existing methods in the literature appears easily amendable. A routine adaptation of penalized likelihood density estimation (Gu and Qiu 1993; Gu and Wang 2003) can be used in the setting, but the resulting density estimates are not copula densities. To make things work, we devise a cross-validation scheme for smoothing parameter selection in density estimation with censored/truncated samples, and we implement a copularization process to convert the initial density estimates into copula densities. Absent censoring/truncation, the method is numerically feasible up to dimensions 4 or 5. In higher dimensions, one may explore conditional independence structures among variables using tools developed by Gu, Jeon, and Lin (2013), and if such structure exists, a copula density in high dimension can often be decomposed into product of copula densities in lower dimensions.

The rest of the article is organized as follows. In Section 2, aspects of bivariate hazard estimation are laid out in some detail, which include likelihoods of censored/truncated lifetime data, the decomposition of likelihood for phased estimation, and the estimation of marginal hazards possibly with covariate. Copula density estimation is discussed in Section 3, concerning the general method on unit cubes $[0, 1]^d$, the handling of censoring/truncation on the unit square $[0, 1]^2$, and how one may enforce symmetry when the marginals are interchangeable; some of the treatments require technical details of tensor product cubic splines, which are summarized in an appendix. Simulation studies are presented in Section 4 to assess the empirical performances of the techniques being developed, such as the effectiveness of cross-validation, the effect of copularization on estimation precision, etc. Real-data examples are given in Section 5 to illustrate potential applications of the methods using open-source software embodied in an R package gsscopu. A few remarks in Section 6 conclude the article.

2. HAZARD ESTIMATION WITH BIVARIATE SURVIVAL DATA

In this section, we explore hazard estimation using bivariate survival data. Expressing a bivariate survival function as the composition of univariate hazards and a bivariate copula, the task can be decomposed into the estimation of marginal hazard functions and that of the copula density.

2.1 BIVARIATE SURVIVAL AND HAZARD FUNCTIONS

For a pair of lifetimes $(T_1, T_2) \in (0, \infty)^2$, consider a survival function of form

$$S(t_1, t_2) = P(T_1 > t_1, T_2 > t_2) = C(S_1(t_1), S_2(t_2)).$$

(2.1)
where $S_1(t_1) = P(T_1 > t_1)$, $S_2(t_2) = P(T_2 > t_2)$ are marginal survival functions and $C(u_1, u_2)$ is a bivariate copula satisfying $C(0, u) = C(u, 0) = 0$, $C(1, u) = C(u, 1) = u$, and $C(u_1, u_2) - C(u_1, v_2) - C(v_1, u_2) + C(v_1, v_2) \geq 0$, $\forall [v_1, u_1] \times [v_2, u_2] \subseteq [0, 1]^2$.

The bivariate hazard function is given by

$$\lambda(t_1, t_2) = \lim_{\Delta t_1, \Delta t_2 \to 0} \frac{P(t_1 < T_1 \leq t_1 + \Delta t_1, t_2 < T_2 \leq t_2 + \Delta t_2 | T_1 > t_1, T_2 > t_2)}{\Delta t_1 \Delta t_2} = \frac{1}{S(t_1, t_2)} \frac{\partial^2}{\partial t_1 \partial t_2} S(t_1, t_2).$$

The marginal survival functions $S_1(t_1), S_2(t_2)$ can be expressed as

$$S(t) = \exp\{- \int_0^t e^{\eta(u)} du\},$$

where

$$e^{\eta(t)} = \lambda(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} = - \frac{d}{dt} \log S(t)$$

is the marginal hazard. However, there is no simple expression of $S(t_1, t_2)$ in terms of $\lambda(t_1, t_2)$.

### 2.2 Likelihood of Right-Censored Data With Possible Left Truncation

When both $T_1$ and $T_2$ are observed, the likelihood is given by

$$\frac{\partial^2}{\partial t_1 \partial t_2} S(t_1, t_2) \bigg|_{t_1 = T_1, t_2 = T_2} = C_{12}(S_1(T_1), S_2(T_2)) S_1(T_1)e^{\eta_1(T_1)} S_2(T_2)e^{\eta_2(T_2)}$$

$$- \frac{\partial}{\partial t_1} S(t_1, t_2) \bigg|_{t_1 = T_1, t_2 = C_2} = \hat{C}_1(S_1(T_1), S_2(C_2)) S_1(T_1)e^{\eta_1(T_1)} S_2(C_2),$$

where $C_{12}(u_1, u_2) = \hat{C}_1(u_1, u_2) = d^2 C(u_1, u_2)/du_1 du_2 = f(u_1, u_2)$ is the copula density.

When $T_1$ is observed and $T_2 > C_2$ is censored, the likelihood is given by

$$- \frac{\partial}{\partial t_1} S(t_1, t_2) \bigg|_{t_1 = T_1, t_2 = C_2} = \hat{C}_1(S_1(T_1), S_2(C_2)) S_1(T_1)e^{\eta_1(T_1)} S_2(C_2),$$

where $\hat{C}_1(u_1, u_2) = \partial C(u_1, u_2)/\partial u_1$, $G_1(u_1, u_2) = \hat{C}_1(u_1, u_2)/u_2 = \int_0^{u_2} f(u_1, v_2) dv_2 / u_2$.

Likewise, the likelihood of $(T_1 > C_1, T_2)$ is given by

$$G_2(S_1(C_1), S_2(T_2)) S_1(C_1) S_2(T_2) e^{\eta_2(T_2)},$$

where $G_2(u_1, u_2) = \hat{C}_2(u_1, u_2)/u_1 = \int_0^{u_1} f(v_1, u_2) dv_1 / u_1$.

When both $T_1$ and $T_2$ are censored, $T_1 > C_1, T_2 > C_2$, the likelihood is simply

$$S(C_1, C_2) = C(S_1(C_1), S_2(C_2)) = G_{\theta}(S_1(C_1), S_2(C_2)) S_1(C_1) S_2(C_2),$$

where $G_{\theta}(u_1, u_2) = C(u_1, u_2)/u_1 u_2 = \int_0^{u_1} \int_0^{u_2} f(v_1, v_2) dv_1 dv_2 / u_1 u_2$.

Accommodating the left truncation point $(T_1, T_2)$, one works with $\tilde{S}(t_1, t_2) = S(t_1, t_2)/S(Z_1, Z_2)$ on $(Z_1, \infty) \times (Z_2, \infty)$. The likelihoods become

$$\frac{\partial^2}{\partial t_1 \partial t_2} \tilde{S}(t_1, t_2) \bigg|_{t_1 = T_1, t_2 = T_2} = \frac{G_{12}(S_1(T_1), S_2(T_2))}{G_{\theta}(S_1(T_1), S_2(T_2))} \tilde{S}(T_1) e^{\eta_1(T_1)} \tilde{S}(T_2) e^{\eta_2(T_2)},$$

$$- \frac{\partial}{\partial t_1} \tilde{S}(t_1, t_2) \bigg|_{t_1 = T_1, t_2 = C_2} = \frac{G_1(S_1(T_1), S_2(C_2))}{G_{\theta}(S_1(T_1), S_2(C_2))} \tilde{S}(T_1) e^{\eta_1(T_1)} \tilde{S}(C_2).$$
\[-\frac{\partial}{\partial t_2} \tilde{S}(t_1, t_2) \bigg|_{t_1=C_1, t_2=T_2} = \frac{G_2(S_1(C_1), S_2(T_2))}{G_\theta(S_1(Z_1), S_2(Z_2))} \tilde{S}_1(C_1) \tilde{S}_2(T_2)e^{\eta_2(T_2)}, \]

\[\tilde{S}(C_1, C_2) = \frac{G_\theta(S_1(C_1), S_2(C_2))}{G_\theta(S_1(Z_1), S_2(Z_2))} \tilde{S}_1(C_1) \tilde{S}_2(C_2),\]

where \(\tilde{S}(t) = S(t)/S(Z).\)

Observing \((X_{ij}, \delta_{ij}, Z_{ij}), i = 1, \ldots, n, j = 1, 2,\) where \(X_{ij} = \min(T_{ij}, C_{ij}), \delta_{ij} = I_{[X_{ij} \leq C_{ij}]},\) and \(X_{ij} > Z_{ij},\) the minus log-likelihood is seen to be

\[-\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_{ij} \eta_1(X_{ij}) - \int_{Z_{ij}}^{X_{ij}} e^{\eta_1(s)} ds \right\} - \frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_{ij} \eta_2(X_{ij}) - \int_{Z_{ij}}^{X_{ij}} e^{\eta_2(s)} ds \right\} \]

\[-\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_{ij} \delta_{ij} \log G_{12}(u_{i1}, u_{i2}) + \delta_{ij}(1 - \delta_{ij}) \log G_{1}(u_{i1}, u_{i2}) \right. \]

\[+ (1 - \delta_{ij}) \delta_{ij} \log G_{2}(u_{i1}, u_{i2}) + (1 - \delta_{ij})(1 - \delta_{ij}) \log G_{\theta}(u_{i1}, u_{i2}) \]

\[- \log G_{\theta}(z_{i1}, z_{i2}) \right\}, \tag{2.2}\]

where \(u_{ij} = S_j(X_{ij})\) and \(z_{ij} = S_j(Z_{ij});\) absent left-truncation, one simply sets \(Z_{ij} = 0\) and \(z_{ij} = 1.\) The log-likelihood naturally decomposes into three terms; the first two terms are the familiar log-likelihood for univariate hazard estimation, and the third term is the only one involving the copula. The marginal hazards can thus be estimated using existing methods with the respective marginal data, and given \(u_{ij}\) and \(z_{ij}\) evaluated using the estimated \(\eta_j,\) one may employ the third term in (2.2) to estimate the copula density \(f(u_1, u_2).\) Marginal hazard estimation is discussed briefly below. The estimation of copula density will be addressed in Section 3.

### 2.3 Marginal Hazard Estimation

To estimate a univariate log hazard \(\eta(t),\) one may minimize a penalized likelihood functional,

\[-\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_i \eta(X_i) - \int_{Z_i}^{X_i} e^{\eta(s)} ds \right\} + \frac{\lambda}{2} J(\eta), \tag{2.3}\]

where \(J(\eta)\) is a roughness functional such as \(\int (\eta''(t))^2 dt,\) and \(\lambda\) is the smoothing parameter to be selected by a cross-validation score; see, for example, Gu (1994). The marginal survival function may also depend on some covariate, say \(V,\) thus is of form \(S(t, v) = P(T > t|V = v) = \exp \left\{ - \int_0^t e^{\eta(s, v)} ds \right\}.\) The hazard \(e^{\eta(t, v)} = -\partial S(t, v)/\partial t\) can then be estimated through a routine extension of (2.3),

\[-\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_i \eta(X_i, V_i) - \int_{Z_i}^{X_i} e^{\eta(s, V_i)} ds \right\} + \frac{\lambda}{2} J(\eta); \tag{2.4}\]

see Gu (1996). One then may use \(u_{ij} = S_j(X_{ij}, V_{ij})\) and \(z_{ij} = S_j(Z_{ij}, V_{ij})\) in the third term of (2.2) to estimate the copula density.

With continuous covariate \(V_i,\) hazard estimation via the minimization of (2.4) is numerically time-consuming due to the repeated evaluations of integrals \(\int_{Z_i}^{X_i} e^{\eta(s, V_i)} ds\) in iterations.
As an alternative, one may minimize a penalized pseudo-likelihood functional

$$\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_i e^{-\eta(X_i, V_i)} \rho(X_i, V_i) + \int_{Z_i}^{X_i} \eta(s, V_i) \rho(s, V_i) ds \right\} + \frac{\lambda}{2} J(\eta), \quad (2.5)$$

where $\rho(s, v)$ is a known positive function and integrals $\int_{Z_i}^{X_i} \eta(s, V_i) \rho(s, V_i) ds$ can largely be precomputed; see Du and Gu (2009) and Gu (2013), Section 10.4. Among good choices of $\rho(s, v)$ is $\rho(s) = e^{\tilde{\eta}(s)}$, where $\tilde{\eta}$ minimizes (2.3).

When it can be assumed that $S_1(t) = S_2(t)$, the data on the two margins shall be combined for the estimation of the marginal hazard.

### 3. COPULA DENSITY ESTIMATION

As seen in Section 2, the bivariate hazard function $\lambda(t_1, t_2)$ involves the copula density $f(u_1, u_2)$, which is to be estimated using the third term in (2.2). Copula density estimation is a subject of interest by itself, and we now discuss a nonparametric approach to the task.

#### 3.1 Penalized Likelihood Density Estimation and Copularization

Absent censoring and truncation, density estimation on $[0, 1]^2$ using samples $(u_{i1}, u_{i2}) \sim f(u_1, u_2), i = 1, \ldots, n$ can be conducted through the minimization of a penalized likelihood functional

$$-\frac{1}{n} \sum_{i=1}^{n} \eta(u_{i1}, u_{i2}) + \log \int_{0}^{1} \int_{0}^{1} e^{\eta(u_1, u_2)} du_1 du_2 + \frac{\lambda}{2} J(\eta), \quad (3.1)$$

yielding $f(u_1, u_2) = e^{\tilde{\eta}(u_1, u_2)} / \int_{0}^{1} \int_{0}^{1} e^{\tilde{\eta}(v_1, v_2)} dv_1 dv_2$; see, for example, Gu and Qiu (1993) and Gu and Wang (2003). The resulting distribution function,

$$F(u_1, u_2) = \int_{0}^{u_1} \int_{0}^{u_2} f(v_1, v_2) dv_1 dv_2 = \frac{\int_{0}^{u_1} \int_{0}^{u_2} e^{\tilde{\eta}(v_1, v_2)} dv_1 dv_2}{\int_{0}^{1} \int_{0}^{1} e^{\tilde{\eta}(v_1, v_2)} dv_1 dv_2},$$

is in general not a copula, however, as there is no assurance that $F(u_1, 1) = u_1$ and $F(1, u_2) = u_2$.

Using the estimated marginal density $f_1(u_1) = \int_{0}^{1} e^{\tilde{\eta}(u_1, v_2)} dv_2 / \int_{0}^{1} \int_{0}^{1} e^{\tilde{\eta}(v_1, v_2)} dv_1 dv_2$ and the marginal distribution function $F_1(u_1) = \int_{0}^{u_1} f_1(v_1) dv_1$, one may transform the domain via $\tilde{u}_1 = F_1(u_1)$; likewise, one has $f_2(u_2)$ and $\tilde{u}_2 = F_2(u_2)$. The joint distribution function of $(\tilde{u}_1, \tilde{u}_2)$, $\tilde{F}(\tilde{u}_1, \tilde{u}_2) = F(F_1^{-1}(\tilde{u}_1), F_2^{-1}(\tilde{u}_2))$, is a copula, and its density is seen to be

$$\tilde{f}(\tilde{u}_1, \tilde{u}_2) = \frac{f(F_1^{-1}(\tilde{u}_1), F_2^{-1}(\tilde{u}_2))}{f_1(F_1^{-1}(\tilde{u}_1)) f_2(F_2^{-1}(\tilde{u}_2))}. \quad (3.2)$$

One may use $\tilde{f}(u_1, u_2)$, the “copularized” version of $f(u_1, u_2)$, to estimate the targeted copula density. As an added benefit, copularization also seems to improve, often substantially, the estimation accuracy in terms of Kullback-Leibler discrepancy; empirical results are to be found in Section 4.2.

In general, copula density estimation can be done on unit cubes $[0, 1]^d$ for $d \geq 2$; forms of (3.1) and (3.2) for general $[0, 1]^d$ are straightforward. To evaluate the marginal
density $f_j(u_j)$ needed in copularization, one has to integrate out the rest of the coordinates from the joint density $f(u_1, \ldots, u_d)$, a numerically costly undertaking, and the need for $F_j^{-1}(p)$ further complicates the matter. An efficiently solution to the problem is to compute and store the marginal densities on a grid at fitting time, to be used later in the evaluation of $f_j(u)$ and $F_j^{-1}(p)$ at arbitrary points.

The grid $\{\tilde{u}_i\} \subset [0, 1]$ on which a marginal density $f(u)$ is to be computed is taken as the nodes of a Gauss-Legendre quadrature with associated weights $w_i$. For a Gauss-Legendre quadrature $\{\tilde{u}_i\}$ on $[0, 1]$, one can partition $[0, 1] = \bigcup_i \mathcal{I}_i$ with consecutive intervals $\mathcal{I}_i$ such that $\mathcal{I}_i \ni \tilde{u}_i$, and with $\tilde{u}_i$ at the center of $\mathcal{I}_i$ for all $\mathcal{I}_i$ but the two at the ends. Spreading $w_i$ evenly over $\mathcal{I}_i$ and taking $f(u) = f(\tilde{u}_i)$, $\forall u \in \mathcal{I}_i$, one may approximate the marginal distribution function $F(u) = \int_0^u f(v)dv$ by $\tilde{F}(u) = \sum_i f(\tilde{u}_i)\tilde{w}_i$, where $\tilde{w}_i = p_i(u)w_i$ for $p_i(u) = \text{length}((0, u] \cap \mathcal{I}_i)/\text{length}(\mathcal{I}_i)$; $\tilde{F}(u)$ is piece-wise linear, easy to invert on the fly. To evaluate $f(u)$ at arbitrary point $u$, one may take four points $\{v_{j}\}_{j=1}^4 \subset \{\tilde{u}_i\}$ from the grid that are the closest to $u$, and interpolate by fitting a cubic polynomial to $(v_j, f(v_j))$. In practice, a Gauss-Legendre quadrature of size 200 appears sufficient for the purpose.

The marginal densities $f_j(u)$ resulting from (3.1) should be well-behaving as the samples are marginally $U(0, 1)$, and instead of repeating the costly numerical integration 200 times on the grid points $\{\tilde{u}_i\}$, one may employ Chebyshev interpolation to approximate the marginal densities. In our implementation, $f_j(u)$ are evaluated using numerical integration at 10 interpolation points $u_k = \cos((2k+1)\pi/20)/2 + 1/2$, $k = 0, \ldots, 9$, and a polynomial approximation $f_j(u) = \sum_{k=0}^9 c_k T_k(u)$ is fitted to $(u_k, f_j(u_k))$, where $T_0(u) = 1$, $T_1(u) = 2u - 1$, $T_k(u) = (4u - 2)T_{k-1}(u) - T_{k-2}(u)$, $k > 1$ are Chebyshev polynomials.

The feasibility of the approach hinges on that of numerical integration on $[0, 1]^d$, which may stretch up to $d = 5$. When conditional independence structures exist among the marginals, however, a high-dimensional copula density can often be decomposed into a product of lower dimensional ones, as will be seen in the example of Section 5.1.

### 3.2 Density Estimation Under Censoring and Truncation on $[0, 1]^2$

Denoting $\gamma_0 = \delta_1 \delta_2$, $\gamma_1 = (1 - \delta_1) \delta_2$, $\gamma_2 = \delta_1 (1 - \delta_2)$, $\gamma_3 = (1 - \delta_1)(1 - \delta_2)$ in the third term of (2.2), and dropping entries that do not involve $f(u_1, u_2)$, (3.1) becomes

$$\frac{1}{n} \sum_{i=1}^n \left\{ \gamma_0 \eta(u_{i1}, u_{i2}) + \gamma_1 \log \int_0^{u_{i1}} e^{\eta(u_{i1}, u_{i2})} du_1 + \gamma_2 \log \int_0^{u_{i2}} e^{\eta(u_{i1}, u_{i2})} du_2 ight. \\
+ \left. \gamma_3 \log \int_0^{\alpha_{i1}} \int_0^{\alpha_{i2}} e^{\eta(u_{i1}, u_{i2})} du_{11}du_{12} - \log \int_0^{\alpha_{i1}} \int_0^{\alpha_{i2}} e^{\eta(u_{i1}, u_{i2})} du_{11}du_{12} \right\} + \frac{\lambda}{2} J(\eta). \quad (3.3)$$

Tensor product cubic splines will be used for $\eta$ in (3.3), and we shall discuss key issues in its implementation. Pertinent technical details concerning tensor product cubic splines can be found in Appendix A, which we will quote where needed.

The minimization of (3.1) or (3.3) is implicitly over $\eta$ in a reproducing kernel Hilbert space $\mathcal{H} \subseteq \{ \eta : J(\eta) < \infty \}$. One has a tensor sum decomposition $\mathcal{H} = \mathcal{N}_f \oplus \mathcal{H}_f$, where $\mathcal{N}_f = \{ \eta : J(\eta) = 0 \}$ and $\mathcal{H}_f$ has $J(\eta)$ as its square norm. The reproducing kernel $R_f(x, y)$ of $\mathcal{H}_f$ satisfies $J(R_f(x, \cdot, \eta(\cdot)) = \eta(x), \forall \eta \in \mathcal{H}_f.$
The minimizer $\hat{\eta}$ of (3.1) in $\mathcal{H}$ is numerically intractable. Instead, one may calculate the minimizer $\hat{\eta}^*$ in a finite dimensional space

$$\mathcal{H}^* = \mathcal{N}_J \oplus \text{span}\{R_j(v_j, \cdot), j = 1, \ldots, q\},$$

(3.4)

where $\{v_j\}$ is a random subset of $\{(u_{i1}, u_{i2})\}$. It can be shown that, for $q \asymp n^{2(\gamma + \epsilon)}, \forall \epsilon > 0$ with tensor product cubic splines, $\hat{\eta}^*$ shares the same asymptotic convergence rates as $\hat{\eta}$; see, for example, Gu and Qiu (1993) and Gu and Wang (2003). Convergence rates are yet to be established for the minimizer of (3.3), but it appears adequate to adopt the same procedure for the numerical implementation of (3.3). The computation of $\hat{\eta}^*$ is of order $O(q^2)$.

### 3.2.1 Newton Iteration.

A function in $\mathcal{H}^*$ can be expressed as $\eta(x) = \sum_v d_v \phi_v(x) + \sum_j c_j R_j(v_j, x)$, where $\{\phi_v\}$ form a basis of $\mathcal{N}_J$. For notational simplicity, we shall denote $\phi_v, R_j(v_j, \cdot)$ collectively as $\xi_j$, and write $\eta(x) = \sum_j c_j \xi_j(x) = \mathbf{c}^T \mathbf{\xi}(x)$. Substituting this into (3.3), one minimizes

$$-\frac{1}{n} \sum_{i=1}^n \left\{ \gamma_{0i} \mathbf{e}^T \mathbf{\xi}(u_i) + \gamma_{1i} \log \int_0^{u_{i1}} e^{\mathbf{e}^T \mathbf{\xi}(u_{i1}, u_{i2})} du_{i1} + \gamma_{2i} \log \int_0^{u_{i2}} e^{\mathbf{e}^T \mathbf{\xi}(u_{i1}, u_{i2})} du_{i2} 
\right\} + \frac{1}{2} \mathbf{c}^T \mathbf{Q} \mathbf{c}$$

(3.5)

with respect to $\mathbf{c}$, where $u_i = (u_{i1}, u_{i2})$, $v = (u_1, u_2)$, and $Q = J(\mathbf{\xi}, \mathbf{\xi}^T)$ comprises $J(\phi_v, \phi_v) = J(\phi_v, R_j(v_j, \cdot)) = 0$ and $J(R_j(v_j, \cdot), R_j(v_{j'k}, \cdot)) = R_j(v_j, v_{j'k})$.

Differentiating with respect to $\mathbf{c}$, one has the gradient $\mathbf{g}$ and Hessian $\mathbf{H}$,

$$\mathbf{g} = -\frac{1}{n} \sum_{i=1}^n \left\{ \gamma_{0i} \mathbf{e}^T \mathbf{\xi}(u_i) + \gamma_{1i} \mu_{1i}(\mathbf{\xi}) + \gamma_{1i} \mu_{2i}(\mathbf{\xi}) + \gamma_{1i} \mu_{3i}(\mathbf{\xi}) - \mu_i(\mathbf{\xi}) \right\} + \lambda \mathbf{Q} \mathbf{c}$$

$$= -\frac{1}{n} \sum_{i=1}^n \mu_i + \lambda \mathbf{Q} \mathbf{c},$$

$$\mathbf{H} = -\frac{1}{n} \sum_{i=1}^n \left\{ \gamma_{1i} v_{1i}(\mathbf{\xi}, \mathbf{\xi}^T) + \gamma_{1i} v_{2i}(\mathbf{\xi}, \mathbf{\xi}^T) + \gamma_{1i} v_{3i}(\mathbf{\xi}, \mathbf{\xi}^T) - v_i(\mathbf{\xi}, \mathbf{\xi}^T) \right\} + \lambda \mathbf{Q}$$

$$= \mathbf{V} + \lambda \mathbf{Q},$$

(3.6)

where

$$\mu_{1i}(g) = \frac{\int_0^{u_{i1}} g(u_{i1}, u_{i2}) e^{\eta(u_{i1}, u_{i2})} du_{i1}}{\int_0^{u_{i1}} e^{\eta(u_{i1}, u_{i2})} du_{i1}}, \quad \mu_{2i}(g) = \frac{\int_0^{u_{i2}} g(u_{i1}, u_{i2}) e^{\eta(u_{i1}, u_{i2})} du_{i2}}{\int_0^{u_{i2}} e^{\eta(u_{i1}, u_{i2})} du_{i2}},$$

$$\mu_{3i}(g) = \frac{\int_0^{u_{i1}} \int_0^{u_{i2}} g(u) e^{\eta(u)} du_{i2} du_{i1}}{\int_0^{u_{i1}} \int_0^{u_{i2}} e^{\eta(u)} du_{i2} du_{i1}}, \quad \mu_{i}(g) = \frac{\int_0^{u_{i1}} \int_0^{u_{i2}} g(u) e^{\eta(u)} du_{i2} du_{i1}}{\int_0^{u_{i1}} \int_0^{u_{i2}} e^{\eta(u)} du_{i2} du_{i1}},$$

and $v_{1i}(g, h) = \mu_{1i}(gh) - \mu_{1i}(g)\mu_{1i}(h), v_{2i}(g, h) = \mu_{1i}(gh) - \mu_{1i}(g)\mu_{i2}(h), v_{3i}(g, h) = \mu_{1i}(gh) - \mu_{1i}(g)\mu_{i3}(h), v_i(g, h) = \mu_{1i}(gh) - \mu_{1i}(g)\mu_{i}(h)$. Fixing the smoothing parameters consisting of $\lambda$ and the $\theta$’s hidden in $R_j$ (see (A.1)), one may employ a standard Newton iteration to minimize (3.3).
3.2.2 Cross-Validation. With varying smoothing parameters, the minimizer $\hat{\eta}^* = \eta_*$ of (3.3), where $\lambda$ represents both the $\lambda$ in front of $f(\eta)$ and the $\theta$’s hidden therein, provides a collection of estimates to choose from, and the proper selection of smoothing parameters is crucial in practical estimation. A cross-validation scheme was developed earlier for use with (3.1), which aimed to minimize $\text{KL}(\eta, \lambda_\eta) = E_f \log(f/f_\lambda) = \mu_\eta(\eta - \eta_\lambda) - \log \int_X e^{\eta(x)} dx + \log \int_X e^{f(\eta)} dx$, where $f = e^{\eta}/\int_X e^{\xi}$ is the true density, $f_\lambda = e^{\eta}/\int_X e^{\eta}$ is the estimate, and $\mu_\eta(g) = \int_X g e^{\eta}/\int_X e^{\eta}$; see, for example, Gu (1993) and Gu and Wang (2003). We shall now adapt the scheme for use with (3.3).

Write $X_i = [0, z_{i1}] \times [0, z_{i2}]$. With observation $(u_{i1}, u_{i2}) \in X_i$, one naturally looks for a small Kullback-Leibler discrepancy $E_f \log(f/f_\lambda) = -\mu_{\eta,i}(\eta_{\lambda}) + \log \int_X e^{\eta_{\lambda}} + C$, where $f_i \propto e^\eta I_{X_i}$, $f_{\lambda,i} \propto e^{\eta_{\lambda}} I_{X_i}$, $\mu_{\eta,i}(g) = \int_{X_i} g e^{\eta}/\int_{X_i} e^{\eta}$, and $C$ does not involve $\eta_\lambda$. For a deterministic $g$, $E_f g(u_{i1}, u_{i2}) = \mu_{\eta,i}(g)$, but $\eta_\lambda$ involves $(u_{i1}, u_{i2})$. As a crude approximation, one may substitute $\eta_{\lambda}^{\theta_i}(u_{i1}, u_{i2})$ for $\mu_{\eta,i}(\eta_{\lambda})$ coupled with $\log \int_X e^{\eta_{\lambda}}$ for $\log \int_X e^{\eta_{\lambda}}$, where $\eta_{\lambda}^{\theta_i}$ minimizes some “delete-one” version of (3.3) in which the impact of $(u_{i1}, u_{i2}) \in X_i \subseteq X$ is diminished.

With censored observation $\Delta_i \subset X_i$, where $\Delta_i = [0, u_{i1}] \times [0, u_{i2}]$, or $[0, u_{i1}] \times [0, u_{i2}]$, one may look for a small Kullback-Leibler discrepancy

$$
p_i \log \frac{p_i}{p_{\lambda,i}} + (1 - p_i) \log \frac{1 - p_i}{1 - p_{\lambda,i}} = -p_i \log \int_{\Delta_i} e^{\eta_{\lambda}} - (1 - p_i) \log \left(\int_{\Delta_i} e^{\eta_{\lambda}} - \int_{\Delta_i} e^{\eta_{\lambda}}\right) + \int_{X_i} e^{\eta_{\lambda}} + C
$$

where $p_i = \int_{\Delta_i} e^{\eta}/\int_X e^{\eta}$, $p_{\lambda,i} = \int_{\Delta_i} e^{\eta_{\lambda}}/\int_X e^{\eta_{\lambda}}$, and $C$ does not involve $\eta_{\lambda}$. Conditioning on $\Delta_i$, $I_{\Delta_i} \sim \text{Bin}(1, p_i)$, so for a deterministic $g$, $E[I_{\Delta_i} \log \int_{\Delta_i} e^{\eta} + (1 - I_{\Delta_i}) \log(\int_{X_i} e^{\eta - \int_{\Delta_i} e^{\eta}}) \mid \Delta_i] = p_i \log \int_{\Delta_i} e^{\eta} + (1 - p_i) \log(\int_{X_i} e^{\eta} - \int_{\Delta_i} e^{\eta})$. As a crude approximation, one may substitute $\log \int_{\Delta_i} e^{\eta_{\lambda}}$ for $p_i \log \int_{\Delta_i} e^{\eta_{\lambda}} + (1 - p_i) \log(\int_{X_i} e^{\eta_{\lambda}} - \int_{\Delta_i} e^{\eta_{\lambda}})$ coupled with $\log \int_{X_i} e^{\eta_{\lambda}}$ for $\log \int_X e^{\eta_{\lambda}}$.

Putting thing together, one may select smoothing parameters via the minimization of a cross-validation score of the following form,

$$
- \frac{1}{n} \sum_{i=1}^n \left\{ \gamma_0 \eta_{\lambda i}^{\theta_i}(u_{i1}) + \gamma_1 \log \int_0^{u_{i1}} e^{\eta_{\lambda i}^{\theta_i}(u_{i1}, u_{i2})} du_1 + \gamma_2 \log \int_0^{u_{i2}} e^{\eta_{\lambda i}^{\theta_i}(u_{i1}, u_{i2})} du_2 \\
+ \gamma_3 \log \int_0^{u_{i1}} \int_0^{u_{i2}} e^{\eta_{\lambda i}^{\theta_i}(u_{i1}, u_{i2})} du_1 du_2 - \log \int_0^{z_{i1}} \int_0^{z_{i2}} e^{\eta_{\lambda i}^{\theta_i}(u_{i1}, u_{i2})} du_1 du_2 \right\}. \tag{3.7}
$$

To make this work, one needs some $\eta_{\lambda i}^{\theta_i}$ that are analytically tractable, and to this end, let us consider the quadratic approximation of (3.5) at $\tilde{\eta} = \eta_*$,

$$
- \frac{1}{n} \sum_{i=1}^n \tilde{\mu}_i^T c + \frac{1}{2} (c - \tilde{c})^T \tilde{V}(c - \tilde{c}) + \frac{\lambda}{2} c^T Q c, \tag{3.8}
$$

where $\tilde{\mu}_i$ and $\tilde{V}$ are $\mu_i$ and $V$ appearing in (3.6) evaluated at $\tilde{\eta} = \tilde{c}^T \tilde{\xi}$. The minimizer of (3.8) is clearly $\tilde{c}$, which satisfies $\tilde{c} = \tilde{H}^{-1}(n^{-1} \sum_{i=1}^n \tilde{\mu}_i + \tilde{V} \tilde{c})$, where $\tilde{H} = \tilde{V} + \lambda Q$. Now
consider a “delete-one” version of (3.8),

$$- \frac{1}{n-1} \sum_{j \neq i} \tilde{\mu}_j^T e + \frac{1}{2} (e - \bar{e})^T \tilde{V} (e - \bar{e}) + \frac{\lambda}{2} e^T Q e, \quad (3.9)$$

whose minimizer is given by

$$e^{(l)} = \tilde{H}^{-1} \left( \frac{\sum_{j=1}^n \tilde{\mu}_j - \tilde{\mu}_i}{n-1} + \bar{V} \bar{e} \right) = \bar{e} + \frac{\tilde{H}^{-1} \sum_{j=1}^n \tilde{\mu}_j}{n(n-1)} - \frac{\tilde{H}^{-1} \tilde{\mu}_i}{n-1},$$

and one may take $\tilde{\xi}^T e^{(l)}$ as $n_\lambda^{(l)}$ in (3.7). With observation $u_i = (u_{i1}, u_{i2}) \in \mathcal{X}_i$,

$$n_\lambda^{(l)}(u_i) - n_\lambda(u_i) = \tilde{\xi}^T (u_i)(e^{(l)} - \bar{e}) = \frac{\tilde{\xi}^T (u_i) \tilde{H}^{-1} \sum_{j=1}^n \tilde{\mu}_j}{n(n-1)} - \frac{\tilde{\xi}^T (u_i) \tilde{H}^{-1} \tilde{\mu}_i}{n-1},$$

$$\log \int_{\mathcal{X}_i} e^{T e^{(l)}} \log \int_{\mathcal{X}_i} e^{T e^{(l)}} \approx \tilde{\mu}_i (\tilde{\xi}^T e^{(l)} - \bar{e}) = \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \sum_{j=1}^n \tilde{\mu}_j)}{n(n-1)} - \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \tilde{\mu}_i)}{n-1},$$

where $\tilde{\mu}_i(g) = \int_{\mathcal{X}_i} g e^0 / \int_{\mathcal{X}_i} e^0$; remember that $\tilde{\mu}_i = \tilde{\xi} (u_i) - \tilde{\mu}_i (\tilde{\xi})$, so

$$n_\lambda^{(l)}(u_i) - \log \int_{\mathcal{X}_i} e^{(l)} \approx n_\lambda(u_i) - \log \int_{\mathcal{X}_i} e^{(l)} + \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \sum_{j=1}^n \tilde{\mu}_j)}{n(n-1)} - \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \tilde{\mu}_i)}{n-1}.$$

Likewise, forensored observation $\Delta_i \subset \mathcal{X}_i$, one has

$$\log \int_{\Delta_i} e^{(l)} \log \int_{\Delta_i} e^{(l)} \approx \log \int_{\Delta_i} e^{(l)} - \log \int_{\mathcal{X}_i} e^{(l)} + \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \sum_{j=1}^n \tilde{\mu}_j)}{n(n-1)} - \frac{\tilde{\mu}_i (\tilde{\xi}^T \tilde{H}^{-1} \tilde{\mu}_i)}{n-1}.$$

Summing up, one may select smoothing parameters via the minimization of

$$V(\lambda) = -\frac{1}{n} \sum_{j=1}^n \left\{ \gamma_0 n_\lambda(u_i) + \gamma_1 \log \int_0^{u_{i1}} e^{\tilde{\eta}_i(u_{i1}, u_{i2})} du_{i1} + \gamma_2 \log \int_0^{u_{i2}} e^{\tilde{\eta}_i(u_{i1}, u_{i2})} du_{i2} \right\}

+ \gamma_3 \log \int_0^{\tilde{u}_{i1}} \int_0^{\tilde{u}_{i2}} e^{\tilde{\eta}_i(u_{i1}, u_{i2})} du_{i1} du_{i2} - \log \int_0^{\tilde{z}_{i1}} \int_0^{\tilde{z}_{i2}} e^{\tilde{\eta}_i(u_{i1}, u_{i2})} du_{i1} du_{i2} \right\}

+ \frac{\alpha}{n(n-1)} \sum_{i=1}^n \tilde{\mu}_i^T \tilde{H}^{-1} \left( \tilde{\mu}_i - \frac{1}{n} \tilde{\mu}_j \right) \quad (3.10)$$

for $\alpha = 1$. $V(\lambda)$ is to be evaluated at the convergence of Newton iteration for $\tilde{\eta} = n_\lambda$.

Absent censoring and truncation, (3.10) reduces to the cross-validation score derived earlier for use with (3.1), to be found in, for example, Gu (1993) and Gu and Wang (2003). The empirical performance of (3.10) is assessed in Section 4.1.

3.2.3 Numerical Integration. As seen above, integrals of form $\int_A h e^0$ need to be computed repeatedly for the minimization of (3.3). Absent censoring and truncation, all integrals are over $A = [0, 1]^d$, and one may use methods such as Smolyak cubatures that are more efficient than product quadratures; see Gu and Wang (2003) and Section 3.4 for discussions. With the varying $A = \mathcal{X}_i$, $\Delta_i$ under censoring and truncation on $[0, 1]^2$, however, a product quadrature allows one to use the same set of nodes for all $A$, and hence is a more efficient choice.
For the 2-D integrals involved in (3.3) with varying $\mathcal{A} \subset [0, 1]^2$, one may use the product of a Gauss-Legendre quadrature $\{\tilde{u}_i\} \subset [0, 1]$ with weights $w_i$. The same set of nodes $\{\tilde{u}_i, \tilde{u}_j\} \subset [0, 1]^2$ are used for all 2-D integrals, on which $\xi_j(u_1, u_2)$ are to be computed and stored, but the associated weights $w_{ij}$ vary with $\mathcal{A}$. Using the partition $[0, 1] = \cup_i \mathcal{I}_i$ discussed in Section 3.1, one has $[0, 1]^2 = \cup_{i,j}(\mathcal{I}_i \times \mathcal{I}_j)$, and the cubature weights for $\int_{\mathcal{A}} h e^\eta$ depend on the amount of overlap $\mathcal{A}$ has with $\mathcal{R}_{ij} = \mathcal{I}_i \times \mathcal{I}_j$, $w_{ij} = p_{ij}(\mathcal{A})w_i w_j$, where $p_{ij}(\mathcal{A}) = \text{area}(\mathcal{A} \cap \mathcal{R}_{ij})/\text{area}(\mathcal{R}_{ij})$. Integrals over 1-D $\Delta_i$ can be handled similarly.

3.3 Estimation of Symmetric Density

When the marginals are exchangeable, the copula density should be symmetric, or invariant under permutations of the coordinates. The standard construction of tensor product cubic splines needs to be modified to enforce symmetry.

First consider $[0, 1]^2$. The standard construction has $\mathcal{N}_J = \text{span}\{k_1(x_1), k_1(x_2), k_1(x_1)k_1(x_2)\}$. To ensure $\eta(x_1, x_2) = \eta(x_2, x_1)$, $k_1(x_1)$ and $k_1(x_2)$ must have the same coefficient, so we use $\hat{\mathcal{N}}_J = \text{span}\{k_1(x_1) + k_1(x_2), k_1(x_1)k_1(x_2)\}$.

For $R_J$, one obviously needs $\theta_{0,0,1} = \theta_{1,0,0}$ and $\theta_{0,1,1} = \theta_{1,0,1}$, but these alone are not quite enough. Given $R_{\mu,\nu}(x, y) = R_{\mu}^{(1)}(x_1, y_1)R_{\nu}^{(2)}(x_2, y_2)$ one needs to use

$$\hat{R}(x, y) = R_{\mu}^{(1)}(x_1, y_1)R_{\nu}^{(2)}(x_2, y_2) + R_{\mu}^{(1)}(x_1, y_2)R_{\nu}^{(2)}(x_2, y_1)$$

to ensure the symmetry of $\hat{R}_s(y) = \hat{R}(x, y)$ as function of $y = (y_1, y_2)$. Summing up, one has $\theta_{0,0,1} = \theta_{1,0,0}$ attached to a total of four terms, $\theta_{0,1,1} = \theta_{1,0,1}$ attached to four terms, and $\theta_{1,1}$ attached to two terms.

On $[0, 1]^d$ for $d > 2$, consider main effects plus interactions up to order $m \leq d$. One has $\mathcal{N}_J = \{\phi_v\}_{v=1}^m$, with $\phi_v$ being the sum of $C^d_v$ terms of $v$-way interaction. In $R_J$, the $k$-way interaction will have $k \theta$’s, attached to kernels involving $j R_1$’s, $(k - j) R_0$’s, and $(d - k) R_0$’s, $j = 1, \ldots, k$, which are, respectively, sums of $d!/(j!(k-j)!(d-k)!)$ terms; the extra $d!$ enumerates the number of permutations of $y = (y_1, \ldots, y_d)$ to pair with $x = (x_1, \ldots, x_d)$. Due to the large number of permutations on top of the increased number of terms, symmetry is computationally impractical to enforce in dimensions beyond $d = 3$.

3.4 Miscellaneous

Apart from the copularization afterward, copula density estimation via a general form of (3.1) is simply the method developed by Gu and Qiu (1993) and Gu and Wang (2003) applied on domain $\mathcal{X} = [0, 1]^d$. Special features of copula density warrant customized implementation, however, which we discuss below.

As noted by Gu and Wang (2003), efficient numerical integration on $[0, 1]^d$ can be facilitated by Smolyak cubatures, which achieve efficiency by thinning out nodes from product quadratures. Distributional data are often dense in the middle and sparse on the edges of the domain, but nodes of Smolyak cubatures are dense near the edges and sparse in the middle. To make things work in the stock implementation, univariate marginal density estimates were used to rescale the axes to spread out the data more evenly (Gu and
Wang 2003). With samples from copula density, the marginal distributions are known to be \( \mathcal{U}(0, 1) \), so axis scaling becomes unnecessary.

Using tensor product splines in the general form of (3.1), one has a functional ANOVA decomposition built in in the log density \( \eta \). Selective elimination of ANOVA terms may imply conditional independence structures, and the exclusion of higher order terms may help to ease the curse of dimensionality. For copula densities, the marginals are more homogeneous, and instead of the model formulas used in the stock implementation, model complexity can be controlled globally by the highest order interactions allowed; to enforce conditional independence when necessary, one simply excludes interactions involving selected pairs of marginals.

To minimize the cross-validation score with respect to \( \lambda \) in front of \( J(\eta) \) and \( \theta \)’s hidden therein, one may perform two passes of fixed-\( \theta \) optimization to obtain initial values of \( \theta \)’s, then use quasi-Newton iteration to update \( \theta \)’s; see Gu (2013), Appendix A. The initial values of \( \theta \)’s prove to be effective, often leaving only “20% of performance” to be picked up by the time-consuming quasi-Newton iteration. One typically has a large number of \( \theta \)’s to select when \( d > 2 \), and it is prudent to skip quasi-Newton iteration by default in the situation.

4. SIMULATION STUDIES

We now present simulation studies of limited scales to assess the empirical performances of the techniques developed in earlier sections. Copula samples were generated from the Frank copula, which has a distribution function of form

\[
C(u_1, \ldots, u_d) = -\frac{1}{\theta} \log \left\{ 1 + \prod_{j=1}^{d} \left( e^{-\theta u_j} - 1 \right)^{d-1} \right\}.
\]

Random number generation and the evaluation of true copula density are performed using utilities supplied in the R package \texttt{copula} (Hofert et al. 2014).

4.1 CROSS-VALIDATION FOR DENSITY ESTIMATION UNDER CENSORING/TRUNCATION

To assess the effectiveness of \( V(\lambda) \) in (3.10) for use with (3.3), samples \((x_{i1}, x_{i2})\) were generated from the 2-D Frank copula with \( \theta = 4 \). The samples are subject to left censoring at \( c_{ij} \sim \text{Beta}(1, 9) \) and right truncation at \( z_{ij} = 0.9 \), \( j = 1, 2 \), with \( u_{ij} = \max(x_{ij}, c_{ij}) < z_{ij} \) and \( \delta_{ij} = I[x_{ij} \geq c_{ij}] \) to be used in (3.3); note that \( u = S(t) \) turns right-censoring/left-truncation on the \( t \) axis into left-censoring/right-truncation on the \( u \) axis. Despite censoring and truncation, we shall still use the standard Kullback-Leibler loss to measure the performance of \( f_\lambda \propto e^{\eta_\lambda} \) as an estimate of \( f \), \( L(\lambda) = KL(f, f_\lambda) = E_f[\log(f/f_\lambda)] \), which is a function of smoothing parameters given data.

Samples of size \( n = 100 \) were used to estimate the density \( f(u_1, u_2) \propto e^{\eta_\lambda(u_1, u_2)} \), with \( X_i = [0, z_{i1}] \times [0, z_{i2}] = [0, 0.9]^2 \), \( [0, 0.9] \times [0, 1] \), \( [0, 1] \times [0, 0.9] \), \( [0, 1]^2 \), 25 each. Estimates \( \eta_\lambda \) were calculated using smoothing parameters \( \lambda_v \) that minimize \( V(\lambda) \) for \( \alpha = 1, 1.4 \), and the Kullback-Leibler loss \( L(\lambda_v) \) was evaluated for each of the estimates. Also calculated was the optimal estimate \( \eta_{\lambda_o} \) with \( \lambda_o \) minimizing \( L(\lambda) \). Enforcing symmetry
as described in Section 3.3, we also calculated symmetric fits and evaluated the respective losses $L(\lambda_v)$ or $L(\lambda_o)$.

Results from 100 replicates are summarized in Figure 1. In the left frame, $L(\lambda_v)$ with $\alpha = 1$ in (3.10) are compared with $L(\lambda_v)$ with $\alpha = 1.4$, and it is clear that $\alpha = 1.4$ is the better choice; this is consistent with similar empirical results for density estimation without censoring and truncation, found in, for example, Gu and Wang (2003). The relative efficacy of cross-validation is depicted as box plots of $L(\lambda_o)/L(\lambda_v)$ in the center frame. The test density is symmetric, and as seen in the right frame, enforcing symmetry does seem to deliver better performance in general.

In the 100 replicates represented in Figure 1, the number of exact data $\sum_i \gamma_{10} = \sum_i \delta_i(1-\delta_i)$ was in the range [56, 80] with mean 67.49, the numbers of singly censored data $\sum_i \gamma_{11} = \sum_i \delta_i(1-\delta_i)$, $\sum_i \gamma_{12} = \sum_i \delta_i(1-\delta_i)$ were in ranges [6, 21], [4, 20], respectively, with means 12.31, 12.58, and the number of doubly censored data $\sum_i \gamma_{13} = \sum_i (1-\delta_i)(1-\delta_i)$ was in the range [1, 16] with mean 7.62.

Note that the estimates involved in the above discussion are the “raw” minimizers of (3.3), not the copularized version. Cross-validation is designed to assist the execution of (3.3), but is unable to foresee the effect of copularization.

### 4.2 Copularization

To see how copularization may affect the accuracy of density estimation, we now compare the performance of the “raw” minimizers of (3.1) or (3.3) with that of their copularized version. Simulations were conducted in dimensions $d = 2, 3, 4$, with samples generated from the Frank copula with $\theta = 4$, and with the default $\alpha = 1.4$ in cross-validation.

For $d = 2$, results are obtained from the same set of replicates represented in Figure 1. The vertical axis in the left frame of Figure 1 are reproduced in the horizontal axis in the left frame of Figure 2: each “raw” estimate $\hat{f}$ was copularized to obtain $\tilde{f}$, with $KL(f, \tilde{f})$ on the vertical axis pairing with $KL(f, \tilde{f})$ on the horizontal axis. For $d = 3$, exact samples of size $n = 200$ were used to calculate estimates with full interactions, and results from 100 replicates are shown in the center frame. For $d = 4$, samples of size $n = 500$ were used to calculate estimates with full interactions and results from one hundred replicates are
shown in the right frame, but the symmetric fit was not calculated; as noted in Section 3.3, symmetric fit is computationally impractical for $d > 3$.

It is clear that copularization improves the accuracy of density estimation in our experiments, where the true density is a copula density. There, however, are a few points above the 45° line in Figure 2, albeit barely, thus absolute improvement is not guaranteed.

### 4.3 Model Complexity

We now take a look at the effect of model complexity on estimation accuracy and execution time. Calculating estimates without the three-way interaction on $[0, 1]^3$ using the replicates represented in the center frame of Figure 2, we can compare the performance of order 2 fits with that of full order fits as in the left frame of Figure 3; the fits being compared are copularized so the horizontal axis in the left frame of Figure 3 reproduces the vertical axis in the center frame of Figure 2. Similarly, copularized fits with only two-way interactions were calculated on $[0, 1]^4$ and their performance is compared in the center frame of Figure 3 with that of fits with full interactions, using the replicates represented in the right frame of Figure 2. The right frame of Figure 3 casts the results of the left frame from a different perspective, comparing the asymmetric fits with the same-data symmetric fits. The true copula density contains full interactions and it is symmetric, and it is clear that full interactions and symmetry both yield more estimation accuracy.
Most of the numerical load is on the evaluation of $R_J(v_j, x)$’s (see (3.4)) on quadrature nodes, in the fitting step and in the copularization step, so the execution time is largely determined by the quadrature sizes and the number of terms involved in $R_J(v_j, x)$. For asymmetric fits, the number of terms involved in $R_J(v_j, x)$ is simply the number of $\theta$’s; for symmetric fits, fewer $\theta$’s are attached to combined terms but the work load for each evaluation remains the same. Listed in Table 1 are the quadrature sizes and the number of terms in $R_J(v_j, x)$ for the fits represented in Figure 3, along with the respective total execution times for 100 replicates. Note that for the symmetric fits on $[0, 1]^3$, the number of terms of $R_J(v_j, x)$ are in parentheses which is different from the number of $\theta$’s, and each term needs to be evaluated 3! = 6 times. Also, the sizes of $\{v_j\}$ used in the simulations are $q = 33$ on $[0, 1]^3$ and $q = 40$ on $[0, 1]^4$, the default values given by $q = 10n^{2/9}$, which are in turn based on the simulation results of Gu and Wang (2003).

### 4.4 Bivariate Hazard Estimation

Finally let us take a look at the performance of bivariate hazard estimation as discussed in Section 2. Given covariate $V \sim \mathcal{U}(0, 1)$, $(T_1, T_2)|V$ were sampled from $S(t_1, t_2|V = v) = C(S_1(t_1, v), S_2(t_2, v))$, where $C(u_1, u_2)$ is the Frank copula with $\theta = 4$ and $S_1(t, v) = S_2(t, v)$ are specified via the marginal hazard $e^{\gamma(t, v)} = -\partial \log S(t, v)/\partial t = (24(t - 0.35)^2 + 2)(3(v - 0.5)^2 + 0.5)$. Censoring and truncation times $(C_1, C_2)$, $(Z_1, Z_2)$ follow distributions with $P(C > c) = I_{[c \leq 1]}e^{-c}$, $P(Z > z) = e^{-z^2}$, independent of $(T_1, T_2, V)$ and of each other; the data are of form $(X_{i1}, \delta_{i1}, Z_{i1}, X_{i2}, \delta_{i2}, Z_{i2}, V_i)$.

On $(t_1, t_1 + dt_1) \times (t_2, t_2 + dt_2)$, one may measure the estimation accuracy of infinitesimal failure probability $p = \lambda(t_1, t_2, v)dt_1dt_2 = P(T_1 \in (t_1, t_1 + dt_1), T_2 \in (t_2, t_2 + dt_2)|T_1 > t_1, T_2 > t_2, V = v)$ by $\hat{p} = \hat{\lambda}(t_1, t_2, v)dt_1dt_2$ via the Kullback-Leibler discrepancy in a Bernoulli setting,

$$p \log(p/\hat{p}) + (1 - p) \log((1 - p)/(1 - \hat{p})) = \{\lambda \log(\lambda/\hat{\lambda}) - \lambda + \hat{\lambda}\}dt_1dt_2 + O((dt_1dt_2)^2).$$

Accumulating over the empirical at-risk processes $(I_{Z_{i1} < t_1 \leq X_{i1}}I_{Z_{i2} < t_2 \leq X_{i2}}, V_i)$, one has

$$\text{KL}(\lambda, \hat{\lambda}) = \frac{1}{n} \sum_{i=1}^{n} \int_{Z_{i1}}^{X_{i1}} \int_{Z_{i2}}^{X_{i2}} \left\{\lambda(t_1, t_2, V_i) \log \frac{\lambda(t_1, t_2, V_i)}{\hat{\lambda}(t_1, t_2, V_i)} - \lambda(t_1, t_2, V_i) + \hat{\lambda}(t_1, t_2, V_i)\right\} dt_1dt_2,$$

which will be used to measure the estimation accuracy of $\lambda(t_1, t_2, v)$ by $\hat{\lambda}(t_1, t_2, v)$.
HAZARD ESTIMATION WITH BIVARIATE SURVIVAL DATA

Samples of size \( n = 150 \) were generated, with which four cross-validated estimates were calculated. Two of the fits assumed \( S_1(t) = S_2(t) \) and \( C(u_1, u_2) = C(u_2, u_1) \), with the marginal hazard estimated via (2.4) or (2.5), respectively; the other two were parallel but did not assume symmetry. Due to the phased estimation of marginal hazards and copula density, the optimal bivariate hazard estimate is not available even in simulations, but one may locate the next best thing, minimizing the univariate version of \( KL(\hat{\lambda}, \hat{\lambda}) \) for smoothing parameter selection in marginal hazard estimation, then, based on the resulting \((u_{ij}, z_{ij})\), calculating the optimal copula density estimate as in Section 4.1; this yielded four pseudo-optimal estimates, against which one may assess the relative efficacy of the cross-validated fits.

Results from 100 replicates are summarized in Figure 4: the left frame compares the penalized likelihood of (2.4) with the penalized pseudo-likelihood of (2.5), the center frame compares symmetric fits with asymmetric ones, and the right frame depicts the relative efficacy \( KL(\hat{\lambda}, \hat{\lambda}_v)/KL(\hat{\lambda}, \hat{\lambda}_o) \) in boxplots, where \( \hat{\lambda}_o \) denotes cross-validated hazard estimate and \( \hat{\lambda}_v \) denotes the pseudo-optimal one. In theory, (2.4) should deliver better performance in general, but in our particular simulation setting, the comparison is more of a toss-up or even slightly in favor of (2.5). The total execution times over the 100 replicates were 41,635 versus 33,753 CPU sec for symmetric fits and 47,272 versus 41,112 CPU sec for asymmetric fits, all in favor of (2.5) as expected; the relative saving of (2.5) over (2.4) is, however, much less dramatic than in univariate hazard estimation, due to the dominant numerical “overhead” of copula density estimation.

Over replicates represented in Figure 4, the number of observed lifetime pairs, \( \sum_i \delta_{i1}\delta_{i2} \), was in the range \([50, 81]\) with mean 63.01, the numbers of singly censored, \( \sum_i \delta_{i1}(1 - \delta_{i2}) \) and \( \sum_i (1 - \delta_{i1})\delta_{i2} \), were in ranges \([19, 46]\), \([21, 44]\), respectively, with means 31.73, 31.34, and the number of doubly censored, \( \sum_i (1 - \delta_{i1})(1 - \delta_{i2}) \), was in the range \([11, 38]\) with mean 23.92.

5. EXAMPLES

The techniques developed in Sections 2 and 3 have been implemented in an R package \texttt{gsscopu}, which we shall now employ to illustrate potential applications using real-data
examples; as a niche extension of the gss package, gsscopu uses some gss functions as building blocks and inherits some utilities and syntax.

5.1 Major World Stock Indices

As an example of copula density estimation, consider the worldindices data frame in the R package CDVine (Brechmann and Schepsmeier 2013). Daily log returns of major world stock indices in 2009 and 2010 were filtered using time series models, and standardized residuals were transformed into copula data; further details are to be found in Brechmann and Schepsmeier (2013).

```r
library(CDVine); data(worldindices); w.idx <- worldindices;
names(w.idx) <- c('GSPC','N225','SSEC','GDAXI','FCHI', 'FTSE')
```

The data include six indices, American S&P 500 (GSPC), Japanese Nikkei 225 (N225), Chinese SSE Composite (SSEC), German DAX (GDAXI), French CAC 40 (FCHI), and British FTSE 100 (FTSE).

We first explore conditional independence structures using the sden1 suite in the gss package, which operates on a variant of (3.1) that avoids numerical integration in high dimensions; see Gu, Jeon, and Lin (2013). Fitting a log density containing main effects and two-way interactions,

```r
domain <- data.frame(GSPC=c(0,1),N225=c(0,1),SSEC=c(0,1),
                     GDAXI=c(0,1),FCHI=c(0,1),FTSE=c(0,1))
fit <- sden1(~(GSPC+N225+SSEC+GDAXI+FCHI+FTSE)^2,data=w.idx,
               domain=domain)
```

one may check the strengths of the interactions, where the top five all involve FCHI.

```r
lab <- fit$terms$label[-(1:6)]
rat <- project(fit,lab,drop1=TRUE)$ratio
lab[order(-rat)]
```

Projecting the fit to a space containing only interactions involving FCHI,

```r
project(fit,lab[order(-rat)[1:5]])$ratio
```

one loses only 1% of “entropy;” see Gu, Jeon, and Lin (2013) for further methodological details. We are led to the same graphical model as in Brechmann and Schepsmeier (2013): given FCHI, the other five indices are conditionally independent.

For a copula density, \( f(u_1,u_2) = f(u_2|u_1) = f(u_1|u_2) \), so when \( U_2 \perp U_3 | U_1 \), \( f(u_1,u_2,u_3) = f(u_1,u_2)f(u_1,u_3) \). In our case here, the joint density of the indices is simply the product of bivariate copula densities with FCHI as one of the marginals, which can be estimated using facilities in the gsscopu package. For example, a copula density can be fitted to the pair GSPC-FCHI via

```r
w15 <- as.matrix(w.idx[,c(1,5)])
fit15 <- sscopu(w15);
fit15.2 <- sscopu2(w15,id.basis=fit15$id.basis)
```
Repeated calls to sscopu/sscopu2 generally yield different fits due to the random selection of \(\{v_j\}\) in (3.4), but one may ensure the same selection by passing along \(\text{id.basis}\). Note that sscopu/sscopu2 takes as input a matrix instead of a model formula with variables in a data frame, and the marginal domain \([0, 1]\) needs no specification; sscopu2 implements the techniques of Section 3.2, using a product quadrature instead of Smolyak cubatures used in sscopu. For bivariate fits, one may use method \text{summary} to obtain Kendall’s \(\tau\) and Spearman’s \(\rho\).

\[
\text{summary(fit15)}; \text{summary(fit15.2)}
\]

In fitting copula densities to pairs GDAXI-FCHI and FTSE-FCHI, sscopu2 operates smoothly while sscopu does not. Negative weights are introduced in Smolyak cubatures as nodes from product quadratures being thinned out, and this could cause numerical problems in our setting when the true density has high peaks/deep valleys. For \(d = 2\), one may safely use sscopu2, but for \(d > 2\), one may have to increase quadrature sizes via larger \(qdsz.depth\) values in the call to sscopu.

### 5.2 Diabetic Retinopathy

For an example of bivariate hazard estimation, consider the diabetic retinopathy data analyzed by Huster, Brookmeyer, and Self (1989). The data involve 197 patients, of whom one eye was randomly selected for treatment and both eyes were followed up until blindness or censoring. The data are found in the diabetes data frame in the R package timereg (Scheike and Zhang 2011), and are reformatted in gsscopu as data frame DiaRet. An initial model is fitted to the data,

\[
\begin{align*}
\text{data(DiaRet)} \quad \text{fit0} & \leftarrow \text{sshzd2d(Surv(time1,status1)\sim time1*trt1*type,} \\
& \quad \text{Surv(time2,status2)\sim time2*trt2*type,} \\
& \quad \text{data=DiaRet,symmetry=TRUE)}
\end{align*}
\]

where \(\text{time1/time2}, \text{status1/status2}\) are the follow-up time and censoring status of the left/right eye, \(\text{trt1/trt2}\) is treatment indicator with levels 0 and 1, and \(\text{type}\) is a factor with levels “adult” and “juvenile” indicating patient’s age at the onset of diabetes. The eyes are interchangeable in the context so symmetry is enforced, and the two model formulas must match each other but with generally different variable names. The fitted marginal hazards are in \text{fit0$hzd1} and \text{fit0$hzd2}, which can be accessed through utility functions of the sshzd suite in the gss package. With \text{symmetry=TRUE} in the call to sshzd2d, the common marginal hazard is estimated using combined marginal data, and our marginal log hazard here is of form

\[
\eta(t, u, v) = \eta_0 + \eta_t(t) + \eta_u(u) + \eta_v(v) + \eta_{tu}(t, u) + \eta_{tv}(t, v) + \eta_{uv}(u, v) + \eta_{tuv}(t, u, v),
\]

where \(u, v\) represent \(\text{trt1/trt2}\) and \(\text{type}\). Projecting into a space excluding \(\eta_{tu}, \eta_{tv},\) and \(\eta_{tuv},\)

\[
\text{project(fit0$hzd1, inc=c(‘time1’, ’trt1’, ‘type’, ’trt1:type’))}
\]
one loses only 4% of “entropy,” so a proportional hazard model on the margins is in order,

```r
fit <- sshzd2d(Surv(time1,status1)~time1+trt1*type,
               Surv(time2,status2)~time2+trt2*type,
               data=DiaRet,symmetry=TRUE,id.basis=fit0$id)
```

see Gu (2004) for further details concerning Kullback-Leibler projection. To evaluate the estimated survival function \( S(t_1,t_2) \) and hazard function \( \lambda(t_1,t_2) \), which in our example also depend on covariates \((u_1,v_1)\) and \((u_2,v_2)\), one may use something like

```r
time <- cbind(c(50,70),c(70,70))
cova <- data.frame(trt1=as.factor(c(1,1)),trt2=as.factor(c(1,0)),
                   type=as.factor(c("juvenile\{del","\}ins","")"adult")))
survexp.sshzd2d(fit,time,cov=cova)
hzdrate.sshzd2d(fit,time,cov=cova)
```

The association between the two eyes, which is characterized via the copula estimate in `fit$copu`, appears to be moderate, with Kendall’s \( \tau \) at 0.25 and Spearman’s \( \rho \) at 0.37.

```r
summary(fit$copu)
```

To evaluate the fitted copula density on a grid, say \((u_i,u_j)\), \( u_i = (i-0.5)/10, \ i = 1,\ldots,10 \), use

```r
u <- ((1:10)-.5)/10; uu <- cbind(rep(u,10),rep(u,rep(10,10)))
dd <- matrix(dsscopu(fit$copu,uu),10,10)
```

Instead of the left and right eyes, one may alternatively take the treated and untreated eyes as the margins. In such a setting, treatment cannot be used as a separate covariate, and symmetry no longer holds between the two margins.

```r
fit1 <- sshzd2d(Surv(time.t,status.t)~time.t+type,
                Surv(time.u,status.u)~time.u+type,
                data=DiaRet,id.basis=fit$id)
```

Given the random selection of eyes for treatment, the association between the two margins should remain the same, which is indeed the case.

```r
summary(fit1$copu)
```

### 6. DISCUSSION

Explored in this article is a nonparametric approach to bivariate hazard estimation and copula density estimation, which are technically related but serve different applications. The techniques are implemented in an R package ready for use by practitioners.

For the direct nonparametric estimation of multivariate density, working with copula data does not seem to offer any benefit yet the constraint a copula density is subject to could be a computational annoyance. Given conditional independence structures, however, a copula density can often be decomposed into product of lower dimensional copula densities, facilitating a divide-and-conquer solution not available on other scales.
For the handling of censoring/truncation, one must use a product quadrature in copula density estimation, thus our approach to bivariate hazard estimation is not extensible to trivariate or beyond computationally. On the other hand, the covariate in the marginal hazards may include random effects (Du and Ma 2010; Gu 2013, sect. 8.3.3), so what we have here seems more than just an alternative to frailty models for multivariate survival.

**APPENDIX A: TENSOR PRODUCT CUBIC SPLINES**

In this appendix, we review some technical facts concerning tensor product cubic splines. Further details can be found in, for example, (Gu 2013, chap. 2).

The minimization of (3.1) or (3.3) is implicitly in a Hilbert space $\mathcal{H} \subseteq \{\eta : J(\eta) < \infty\}$ in which $J(\eta)$ is a square semi-norm with a finite dimensional null space $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$. A Hilbert space has a metric and a geometry that facilitate analysis and computation, and a finite dimensional $\mathcal{N}_J$ prevents interpolation. Function evaluations appear in the log-likelihood, so one also needs the evaluation functional $\{x|\eta = \eta(x)\}$ to be continuous in $\eta \in \mathcal{H}$, $\forall x \in \mathcal{X}$, $\mathcal{X} = [0, 1]^2$ in (3.1) and (3.3).

When evaluation functional is continuous in a Hilbert space, the space is known as a reproducing kernel Hilbert space possessing a reproducing kernel $\eta$ on $\mathcal{X}$, with nine tensor-sum terms $\mathcal{X}$ may include random effects (Du and Ma 2010; Gu 2013, sect. 8.3.3), so what we have here seems more than just an alternative to frailty models for multivariate survival.

As a tutorial of its use in penalized regression can be found in Nosedal-Sanchez et al. (2012). On $\mathcal{X} = [0, 1]$ with $J(\eta) = \int_0^1 (\eta''(x))^2 dx$, one may use

$$
\langle f, g \rangle = \left( \int_0^1 f(x) dx \right) \left( \int_0^1 g(x) dx \right) + \left( \int_0^1 f'(x) dx \right) \left( \int_0^1 g'(x) dx \right) + \int_0^1 f''(x)g''(x) dx,
$$

where the three terms $J_0(f, g)$, $J_0(f, g)$, and $J_1(f, g) = J(f, g)$, in order, are inner products in $\mathcal{H}_0 = \text{span}(1)$, $\mathcal{H}_1 = \text{span}(k_1(x))$, and $\mathcal{H}_1 = \{\eta : J(\eta) < \infty, \int_0^1 \eta(x) dx = \int_0^1 \eta'(x) dx = 0\}$, where $k_1(x) = x - 0.5$. In fact, $\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}_1$ form a tensor sum decomposition of $\mathcal{H} = \{\eta : J(\eta) < \infty\}$, $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1 \oplus \mathcal{H}_1$, with reproducing kernels $R_{00}(x, y) = 1$, $R_{00}(x, y) = k_1(x)k_1(y)$, and $R_{00}(x, y) = k_2(x)k_2(y) - k_2(x - y)$, where $k_2 = B_2/v!$ are scaled Bernoulli polynomials. This is commonly known as a cubic spline construction: the minimizer of a penalized likelihood least squares functional, $n^{-1} \sum_{i=1}^n (Y_i - \eta(x_i))^2 + \lambda \int_0^1 (\eta''(x))^2 dx$, is a piece-wise cubic polynomial, although the minimizer of the univariate version of (3.1), $n^{-1} \sum_{i=1}^n (Y_i - \log \int_0^1 e^{\eta(x)} dx)^2 + \lambda \int_0^1 (\eta''(x))^2 dx$, is not.

On $\mathcal{X} = [0, 1]^2$, one may construct tensor-product cubic splines using the marginal construction given above, with nine tensor-sum terms $\mathcal{H}_{\mu, \nu} = \mathcal{H}_{\mu}^{(1)} \otimes \mathcal{H}_{\nu}^{(2)}$, $\mu, \nu = 00, 01, 1$ generated from reproducing kernels $R_{\mu, \nu}(x, y) = R_{\mu}(x_1, y_1)R_{\nu}(x_2, y_2)$, where $x = (x_1, x_2)$, $y = (y_1, y_2)$; expressions of the respective inner products in these spaces, $J_{\nu, \nu}(f, g)$, can be found in (Gu 2013), Table 2.3. The four subspaces with $\mu, \nu = 00, 01$ are of one-dimension each, and can be lumped together as $\mathcal{N}_J$. The other five subspaces can be put together as $\mathcal{H}_J = \mathcal{H} \otimes \mathcal{N}_J$ with a reproducing kernel

$$
R_J = \theta_{00,1} R_{00,1} + \theta_{1,0} R_{1,0} + \theta_{0,1} R_{0,1} + \theta_{1,0} R_{1,0} + \theta_{1,1} R_{1,1}
$$

and the corresponding square norm of $\mathcal{H}_J$, $J(\eta) = \sum_{\mu, \nu} \theta_{\mu, \nu}^{-1} J_{\mu, \nu}(\eta_{\mu, \nu})$, for $\eta = \sum_{\mu, \nu} \eta_{\mu, \nu}$ and $\eta_{\mu, \nu} \in \mathcal{H}_{\mu, \nu}$, is to be used in (3.1) and (3.3); $\theta_{\mu, \nu} > 0$ are a set of smoothing parameters adjusting the
relative weights of the roughness of different components. The construction also induces a functional
ANOVA decomposition of \( \eta \in \mathcal{H}, \eta(x) = \eta_0 + \eta_1(x_1) + \eta_2(x_2) + \eta_{12}(x_1, x_2), \) with \( \eta_0 \in \mathcal{H}_{00,00}, \eta_1 \in \mathcal{H}_{01,00} \oplus \mathcal{H}_{10,00}, \eta_2 \in \mathcal{H}_{00,01} \oplus \mathcal{H}_{00,10}, \) and \( \eta_{12} \in \mathcal{H}_{01,10} \oplus \mathcal{H}_{00,11}. \)

For a one-to-one mapping \( f(u_1, u_2) \leftrightarrow \eta(u_1, u_2)/\int_0^1 \int_0^1 \exp(\eta(u_1, u_2)) du_1 du_2 \) in (3.1) and (3.3), one may set \( \eta_0 = 0 \) to remove \( \mathcal{H}_{00,00} = \text{span}\{1\} \) from \( \mathcal{H}, \) leaving only three components in \( \mathcal{N}_f. \)

Parallel constructions on \([0, 1]^d\) for \( d > 2 \) follow the same lines.

[Received February 2014. Revised June 2014.]

REFERENCES

Aronszajn, N. (1950), “Theory of Reproducing Kernels,” Transactions of the American Mathematical Society, 68, 337–404. [1071]

Brechmann, E. C., and Schepsmeier, U. (2013), “Modeling Dependence With C- and D-Vine Copulas: The R Package CDVine,” Journal of Statistical Software, 52, 1–27. [1068]

Dabrowska, D. M. (1988), “Kaplan-Meier Estimate on the Plane,” The Annals of Statistics, 16, 1475–1486. [1053]

Du, P., and Gu, C. (2009), “Penalized Pseudo-Likelihood Hazard Estimation: A Fast Alternative to Penalized Likelihood,” The Journal of Statistical Planning and Inference, 139, 891–899. [1057]

Du, P., and Ma, S. (2010), “Frailty Model With Spline Estimated Nonparametric Hazard Function,” Statistica Sinica, 20, 561–580. [1071]

Gijbels, I., and Mielniczuk, J. (1990), “Estimating the Density of a Copula Function,” Communication in Statistics—Theory and Methods, 19, 445–464. [1054]

Gu, C. (1993), “Smoothing Spline Density Estimation: A Dimensionless Automatic Algorithm,” Journal of the American Statistical Association, 88, 495–504. [1060,1061]

——— (1994), “Penalized Likelihood Hazard Estimation: Algorithm and Examples,” in Statistical Decision Theory and Related Topics V, eds. S. S. Gupta and J. O. Berger, New York: Springer-Verlag, pp. 61–72. [1053,1056]

——— (1996), “Penalized Likelihood Hazard Estimation: A General Procedure,” Statistica Sinica, 6, 861–876. [1053,1056]

——— (2004), “Model Diagnostics for Smoothing Spline ANOVA Models,” Canadian Journal of Statistics, 32, 347–358. [1070]

——— (2013), Smoothing Spline ANOVA Models (2nd ed.), New York: Springer-Verlag. [1057,1063,1071]

Gu, C., Jeon, Y., and Lin, Y. (2013), “Nonparametric Density Estimation in High Dimensions,” Statistica Sinica, 23, 1131–1153. [1054,1068]

Gu, C., and Qiu, C. (1993), “Smoothing Spline Density Estimation: Theory,” The Annals of Statistics, 21, 217–234. [1054,1057,1059,1062]

Gu, C., and Wang, J. (2003), “Penalized Likelihood Density Estimation: Direct Cross-Validation and Scalable Approximation,” Statistica Sinica, 13, 811–826. [1054,1057,1059,1061,1062,1064,1066]

Hofert, M., Kojadinovic, I., Maechler, M., and Yan, J. (2014), Copula: Multivariate Dependence With Copulas. R package version 0.999-11. [1063]

Hougaard, P. (1986), “A Class of Multivariate Failure Time Distributions,” Biometrika, 73, 671–678. [1053]

Huster, W. J., Brookmeyer, R., and Self, S. G. (1989), “Modeling Paired Survival Data With Covariates,” Biometrics, 45, 145–156. [1069]

Kauermann, G., Schellhase, C., and Ruppert, D. (2013), “Flexible Copula Density Estimation With Penalized Hierarchical B-splines,” Scandinavian Journal of Statistics, 40, 685–705. [1054]

Noseda-Sanchez, A., Storlie, C. B., Lee, T. C. M., and Christensen, R. (2012), “Reproducing Kernel Hilbert Spaces for Penalized Regression: A Tutorial,” The American Statistician, 66, 50–60. [1071]
Oakes, D. (1989), “Bivariate Survival Models Induced by Frailties,” *Journal of the American Statistical Association*, 84, 487–493. [1053]

Sancetta, A., and Satchell, S. (2004), “The Bernstein Copula and Its Applications to Modeling and Approximating of Multivariate Distributions,” *Economic Times*, 20, 535–562. [1054]

Scheike, T. H., and Zhang, M.-J. (2011), “Analyzing Competing Risk Data Using The R Timereg Package,” *The Journal of Statistical Software*, 38, 1–15. [1069]