2-1-2012

Copula Density Estimation by Total Variation Penalized Likelihood with Linear Equality Constraints

Leming Qu
Boise State University

Wotao Yin
Rice University

Copula density estimation by total variation penalized likelihood with linear equality constraints

Leming Qu, Wotao Yin

Abstract

A copula density is the joint probability density function (PDF) of a random vector with uniform marginals. An approach to bivariate copula density estimation is introduced that is based on a maximum penalized likelihood estimation (MPLE) with a total variation (TV) penalty term. The marginal unity and symmetry constraints for copula density are enforced by linear equality constraints. The TV-MPLE subject to linear equality constraints is solved by an augmented Lagrangian and operator-splitting algorithm. It offers an order of magnitude improvement in computational efficiency over another TV-MPLE method without constraints solved by log-barrier method for second order cone program. A data-driven selection of the regularization parameter is through K-fold cross-validation (CV). Simulation and real data application show the effectiveness of the proposed approach. The MATLAB code implementing the methodology is available online.

Keywords: Copula density estimation; Total variation; Maximum penalized likelihood estimation; Augmented Lagrangian method

1. Introduction

Dependence modeling consists of finding a model that describes dependencies between variables, which is a fundamental task of multivariate statistics (Cox and Wermuth (1996)). Statistical approaches to dependence modeling describes an underlying random process in terms of a multivariate distribution. Multivariate probability density estimation based on observed data from a random process is a long standing and active research area in statistics (Scott (1992)). In a linear, Gaussian world stochastic dependencies are captured by correlations. In more general settings, copula (otherwise known as dependence function) has emerged as a useful tool for modeling stochastic dependence. Some recent review papers on copulas include Embrechts et al. (2003); Kolve et al. (2006); Mikosch (2006); Embrechts (2009) and Patton (2009). Some recent books on copulas include Cherubini et al. (2004); Nelsen (2006) and Trivedi and Zimmer (2007). In essence, a copula is a multivariate probability distribution with uniform marginals. One of the main advantages of copula over full probability function is that copula allows the separation of dependence modeling from the marginal distributions. The last decade or so has witnessed an explosion of papers on the application of copulas, especially to financial problems (Patton (2009), Genest et al. (2009), Haug et al. (2011)). In the introduction to their book, Cherubini et al. (2004)

*Leming Qu is with Department of Mathematics, Boise State University, Boise, Idaho 83725-1555, USA; Tel: 1-208-426-2803; Fax: 1-208-426-1356; lqu@boisestate.edu. Wotao Yin is with Department of of Computational and Applied Mathematics, Rice University, Houston, Texas 77005, USA; wotao.yin@rice.edu

state that "the copula function methodology has become the most significant new technique to handle the co-movement between markets and risk factors in a flexible way". Balakrishnan and Lai (2009) lists many applications of copulas in several categories. Specifically in finance, copulas have attracted much attention in the analysis of contagion between financial markets (Rodriguez (2007); Chen and Poon (2007)), the analysis of risky portfolios of stocks (Malevergne and Sornette (2003); Junker and May (2005)) or the modeling of credit default (Li (2000)). For a statistical introduction to copula, see Nelsen (2006).

The copula density estimation has been mostly studied in a parametric framework, whereby a bivariate copula density $c(u, v)$ is assumed to be a member of a copula family determined by a few parameters (for example, Shih and Louis (1995)). The parametric copula density estimation problem is then essentially reduced to estimate the few parameters that determine the copula. Choros et al. (2010) provide a brief survey of parametric, semiparametric and nonparametric estimation procedures for copula models. We propose here to estimate the bivariate copula density nonparametrically. For practitioners, nonparametric estimates could be used as the first step toward selecting the right parametric family.

Nonparametric estimation of copula and its density does not assume a specific parametric form for the copula and the marginals and thus provides great flexibility and generality. Nonparametric estimators of a bivariate copula density using kernels have been suggested by Gijbels and Mielniczuk (1990) and Fermanian and Scaillet (2003). The advantage of kernel based copula density estimation is that it provides a smooth (differentiable) reconstruction of the copula function without putting any particular parametric a priori on the dependence structure between margins and without losing the usual parametric rate of convergence (Fermanian and Scaillet (2003)). Kernel estimators have a severe drawback as they require a very large amount of data (page 195, Malevergne and Sornette (2006)) and suffer from a corner bias. Nonparametric estimator of a copula using splines was proposed in Shen et al. (2008) for a new class of copulas called linear B-spline copulas. The linear B-spline copula estimation can be regarded as a semiparametric approach for copula estimation. While it is still defined in terms of a parametric form, it shares the same flexibility as that exhibited by a nonparametric approach. For a sub-family of copulas named Archimedean, Lambert (2007) proposed to use B-splines for a ratio approximation of the generator and of its first derivative, and estimated the associated parameters using Markov chains Monte Carlo methods. Sancetta and Satchell (2004) employed techniques based on Bernstein polynomials. Bernstein copula family belongs to the family of polynomial copulas (Nelsen (2006)) and can be used as an approximation to any copula. Nonparametric estimators of a copula density using wavelets were proposed in Hall and Neumeyer (2006) and Autin et al. (2010). These wavelet methods can better adapt to nonsmooth regions such as corners of a copula density.

What does a copula density $c(u, v)$ look like? In one extreme, when U and V are independent of each other, $c(u, v) = 1$. When U and V are dependent, $c(u, v)$ can be smooth, have sharp boundaries, or even be unbounded. It is reasonable to assume that the total variation (TV) of $c(u, v)$, or at least its discrete version, is bounded. In practice, we often estimate and display the density in a finite grid. We propose a maximum penalized likelihood estimation (MPLE) with TV penalty method. This method is capable of capturing sharp changes in the target copula density, suffering less from edge effects when the copula density can be unbounded at boundaries in some statistically important cases, whereas conventional kernel or spline techniques have difficulties in nonsmooth regions.

The TV penalty based MPLE for copula density was proposed in Qu et al. (2009), where

the penalty term is the TV of the log density, and the unity requirement for a density function is imposed. However, the marginal unity and symmetrical properties for a copula density are not enforced. In fact, we are not aware of any method that explicitly imposes all the essential properties for a copula density. The main reason behind this is probably related to the difficulty of the induced estimation or optimization procedure. In this paper, we enforce the properties of marginal unity and symmetry as linear equality constraints for the discretized copula density. We solve the problem of minimizing penalized negative log likelihood with TV penalty subject to linear equality constraints by an augmented Lagrangian and operator-splitting algorithm. The effectiveness of our method is illustrated through numerical experiments.

Density estimation by TV penalized likelihood has been proposed by several groups of researchers. Koenker and Mizera (2007) used the TV of the derivative of the log density as the penalty in the univariate case and TV of the log density defined in a triogram in the bivariate case. Sardy and Tseng (2010) and Mohler et al. (2010) used the TV of the density itself as the penalty. Mohler et al. (2010) presented a fast and accurate numerical method, based upon the Split Bregman L1 minimization technique (Goldstein and Osher (2009)).

The rest of the paper is organized as follows: In Section 2, we formulate the problem. In Section 3, we present the Augmented Lagrangian and operator–splitting algorithm. In Section 4, we discuss the regularization parameter selection, and in Section 5 show the experimental results. We apply the method to a real data set in Section 6. Finally, Section 7 concludes the paper.

2. Problem Formulation

A bivariate copula density $c(u, v)$, $[u, v] \in [0, 1]^2$ can be regarded as the joint probability density function (PDF) of a bivariate standard uniform random variable (U, V). Most copulas are exchangeable, thus implying $c(u, v)$ is symmetric. The $c(u, v)$ must satisfy the following four properties:

(P1) $c(u, v) \geq 0$, for $[u, v] \in [0, 1]^2$;

(P2) $\int_0^1 c(u, v)du = 1$, for $0 \leq v \leq 1$;

(P3) $\int_0^1 c(u, v)dv = 1$, for $0 \leq u \leq 1$;

(P4) $c(u, v) = c(v, u)$.

Note that (P2) and (P4) implies (P3), so (P3) is redundant.

A bivariate copula $C(u, v)$ defined on the unit square $[0, 1]^2$ is a bivariate cumulative distribution function (CDF) with univariate standard uniform margins:

$$C(u, v) = \int_0^u \int_0^v c(s, t)\mathrm{d}s\mathrm{d}t.$$

Sklar's Theorem (Sklar (1959)) states that the joint CDF $F(x, y)$ of a bivariate random variable (X, Y) with marginal CDF $F_X(x)$ and $F_Y(y)$ can be written as $F(x, y) = C(F_X(x), F_Y(y))$, where copula C is the joint CDF of $(U, V) = (F_X(X), F_Y(Y))$. This indicates a copula connects the marginal distributions to the joint distribution and justifies the use of copulas for building bivariate distributions.

Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be a random sample from the unknown distribution F of (X, Y). We wish to estimate aspects of the joint distribution of X and Y, in particular, the copula density function $c(u, v)$.

When the two marginal distributions are continuous, the copula density $c(u, v)$ is the unique bivariate density of $(U, V) = (F_X(X), F_Y(Y))$ as implied by Sklar's theorem. As copulas are not directly observable, a nonparametric copula density estimator has to be formed in two stages: obtaining the observations for (U, V) first and then estimating the copula density based on these observations.

In the first stage, the original data set (X_i, Y_i) for $i = 1, \ldots, n$ is converted to $(\hat{U}_i, \hat{V}_i) = (\hat{F}_X(X_i), \hat{F}_Y(Y_i))$, where $\hat{F}_X$ and $\hat{F}_Y$ are conventional estimators of F_X and F_Y. If models are available for the marginal distributions of X and Y but not for the joint distribution, one can use a technique such as maximum likelihood to estimate the marginal distribution functions. Otherwise, some nonparametric univariate distribution estimation methods or simply the following empirical CDFs (ECDFs) can be used:

$$\hat{F}_X(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x), \quad \hat{F}_Y(y) = \frac{1}{n}\sum_{i=1}^{n} I(Y_i \le y), \tag{1}$$

where $I(\cdot)$ is the indicator function. When ECDFs are used as the marginal CDF estimators (e.g., in Autin et al. (2010)), $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$ is nothing but the standardized ranks. In the second stage, we estimate the copula density $c(u, v)$ based on the observations $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$.

Here we do not assume any parametric form for $c(u, v)$ and instead, obtain an estimate of it that satisfies properties (P1-P4) and is defined on a unit rectangle grid. The grid is made by equally dividing domain of $c(u, v)$, $[0, 1]^2$, into $N = m^2$ rectangle cells with cell size $(1/m) \times (1/m)$. A reasonable grid size is 64×64 (i.e., $m = 64$) for sample size $n = 2000$ and $m = 32$ for $n = 500$. A much finer discretization will slow down computation unnecessarily. In most numerical scheme, one fixes a grid resolution of $1/m$ much smaller than 2^{-J_n} with $J_n = \lfloor \frac{1}{2}\log_2(\frac{n}{\log n}) \rfloor$ (page 207 of Autin et al. (2010)).

Let us use $i, j = 1, \ldots, m$ to index all the N cells of this grid. On each cell (i, j), $i, j = 1, \ldots, m$, let x_{ij} denote the constant estimate of $c(u, v)$ over the cell and set p_{ij} to the number of observations $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$ falling in this cell.

The marginal integral of $c(u, v)$ can be approximated by the Riemann sum

$$\int_0^1 c(u, v)du \approx \frac{1}{m}\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, \ldots, m$$

and

$$\int_0^1 c(u, v)dv \approx \frac{1}{m}\sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, \ldots, m.$$

TV of $\boldsymbol{x}$ is defined as

$$\mathrm{TV}(\boldsymbol{x}) = \sum_{i=1}^{m}\sum_{j=1}^{m} \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} \approx \int\int \left\| D\left(c(u, v)\right) \right\|_2 ,$$

where we set the Nuemann boundary conditions for TV, namely, $x_{m+1,j} \equiv x_{m,j}$, $j = 1, \ldots m$, and $x_{i,m+1} \equiv x_{i,m}$, $i = 1, \ldots, m$.

In Qu et al. (2009), by defining $z_{ij} = \log x_{ij}$, the TV-MPLE is to solve :

$$\min_{\boldsymbol{z}} \ T_\lambda(\mathbf{z}) = -\sum_{i=1}^{m}\sum_{j=1}^{m} p_{ij} z_{ij} + \lambda \mathrm{TV}(\mathbf{z}), \quad \text{s.t.} \quad \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{m} \exp(z_{ij}) = 1,$$

where λ is a smoothing parameter controlling the smoothness of the estimate. The above constrained minimization problem is equivalent to the following unconstrained minimization problem:

$$\min_{\boldsymbol{z}} \ T_\lambda(\mathbf{z}) = \sum_{i,j}\left[-p_{ij} z_{ij} + \frac{n}{N}\exp(z_{ij})\right] + \lambda \mathrm{TV}(\mathbf{z}).$$

Even though this unconstrained minimization formulation is attractive, it does not impose the properties (P2) and (P4).

In terms of $\mathbf{z} = \log \boldsymbol{x}$, the property (P2) requires the nonlinear constraints

$$\sum_{i=1}^{m} \exp(z_{ij}) = m, \quad j = 1, \ldots, m.$$

Nonlinear constraints are more difficult to work with than linear constraints, so it is preferable to minimize with respect to $\boldsymbol{x}$ instead of $\mathbf{z}$ if properties (P2) and (P4) are to be imposed.

Imposing the marginal unity (P2) and symmetry (P4) properties, we estimate a copula density as a $m \times m$ digital image by solving:

$$\begin{aligned} \min_{\boldsymbol{x}} \ T_\lambda(\boldsymbol{x}) &= -\sum_{i,j} p_{ij} \log x_{ij} + \lambda \mathrm{TV}(\boldsymbol{x}), \\ \text{s.t.} \quad \sum_{i=1}^{m} x_{ij} &= m, \quad j = 1, \ldots, m, \text{ and} \\ x_{ij} &= x_{ji}, \quad i, \ j = 1, \ldots, m. \end{aligned}$$

The linear equality constraints in the above minimization problem can be written in the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by forming the $m(m+1)/2 \times N$ matrix $\boldsymbol{A}$ and $m(m+1)/2$-vector $\boldsymbol{b}$ as follows:

$$\begin{aligned} A(i,j) &= 1, \quad i = 1, \ldots, m, \quad j = (i-1)m + 1, \ldots, im; \\ b(i) &= m, \quad i = 1, \ldots, m; \end{aligned}$$

and

$$\begin{aligned} A(m + (i-1)(i-2)/2 + j, \ (j-1)m + i) &= 1, \quad i = 2, \ldots, m, \quad j = 1, \ldots, i-1; \\ A(m + (i-1)(i-2)/2 + j, \ (i-1)m + j) &= -1, \quad i = 2, \ldots, m, \quad j = 1, \ldots, i-1; \\ b(i) &= 0, \quad i = m+1, \ldots, m(m+1)/2; \end{aligned}$$

and

$$A(i,j) = 0, \text{ otherwise.}$$

The matrix A is very sparse.

Let $f(\boldsymbol{x}) = -\sum_{i,j} p_{ij} \log x_{ij}$, then $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = -\boldsymbol{p}./\boldsymbol{x}$, where $\nabla_{\boldsymbol{x}}$ denotes the gradient operator with respect to $\boldsymbol{x}$ and ./ denotes element-wise division. This gradient will be used in the optimization algorithm discussed in the next section.

Our proposed copula density estimate solves:

$$\min_{\boldsymbol{x}} \ T_\lambda(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda \mathrm{TV}(\boldsymbol{x}), \text{ s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{2}$$

3. Augmented Lagrangian and operator–splitting algorithm

This section describes how to efficiently solve problem (2) by the augmented Lagrangian and operator–splitting techniques with modifications. The so-called augmented Lagrangian of (2) is

$$L(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \lambda \mathrm{TV}(\boldsymbol{x}) + \frac{\alpha}{2}||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 - < \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} >,$$

where $\boldsymbol{y}$ contains the Lagrange multipliers. The traditional augmented Lagrangian algorithm is the iteration of

(1) $\boldsymbol{x} \leftarrow \arg\min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{y})$;

(2) $\boldsymbol{y} \leftarrow \boldsymbol{y} + \alpha(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$,

which avoids solving the original constrained problem and only requires a moderate penalty parameter α for quick convergence. However, because of the nonsmooth TV term and matrix $\boldsymbol{A}$ in (2), it is a time consuming task to complete step (1) above. A good way to get around the computational complexity of step (1) above is through linearization, which is related to the classical work of augmented Lagrangian and alternating direction methods in Glowinski and Tallec (1989).

Introducing the gradient vector

$$g(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \left(f(\boldsymbol{x}) + \frac{\alpha}{2}||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 - < \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} > \right).$$

For the problem formulated in the last section, $g(\boldsymbol{x}) = -\boldsymbol{p}./\boldsymbol{x} + \alpha \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) - \boldsymbol{A}^T\boldsymbol{y}$. The one-step operator-splitting iteration is

$$\boldsymbol{x} \leftarrow R(\boldsymbol{x} - \beta g(\boldsymbol{x})), \tag{3}$$

where $R(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}||\boldsymbol{z} - \boldsymbol{x}||^2 + \lambda \mathrm{TV}(\boldsymbol{x})$ ("arg min" is well-defined because the strong convexity of $\|\boldsymbol{z} - \boldsymbol{x}\|^2$ guarantees solution existence and uniqueness). This step together with the update to $\boldsymbol{y}$ can be written as

Step 1: $\boldsymbol{x}^{k+1} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}||(\boldsymbol{x}^k - \beta g(\boldsymbol{x}^k)) - \boldsymbol{x}||^2 + \lambda \mathrm{TV}(\boldsymbol{x})$.

Step 2: $\boldsymbol{y}^{k+1} \leftarrow \boldsymbol{y}^k + \gamma\alpha(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{k+1})$.

The algorithm starts with $\boldsymbol{y}^0 = \boldsymbol{0}$ and an initial $\boldsymbol{x}^0$, then iterates through steps 1 and 2 until certain convergence criteria are met. As pointed out by a reviewer, step 1 makes a compromise between a steepest descent update (that aims to minimize the augmented Lagrangian) and sparsity (through the $\mathrm{TV}(\boldsymbol{x})$ term).

Before discussing parameter selection and the implementation of Step 1, we note that Step 1 above is different from the so-called alternating direction method (ADM, Glowinski and Tallec (1989)) or the recent algorithm TVAL3 in Li et al. (2009). To minimize a function in the form of $a(B\boldsymbol{x}) + b(\boldsymbol{x})$ where B is a certain operator, ADM introduces an unknown vector $\boldsymbol{z}$ and constraints $B\boldsymbol{x} = \boldsymbol{z}$ and uses an augmented Lagrangian $L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ to relax these constraints. However, ADM has a different Step 1. In Step 1, ADM computes $x \leftarrow \arg\min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ and uses the updated $\boldsymbol{x}$ to obtain $\boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z}} L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$. TVAL3 is similar to ADM as it splits $\mathrm{TV}(\boldsymbol{x}) = \sum_{ij} \|(D\boldsymbol{x})_{ij}\|_2$

into $\sum_{ij} \|\boldsymbol{z}_{ij}\|_2$ and constraints $\boldsymbol{z} = D\boldsymbol{x}$, where $\boldsymbol{z}$ is an unknown vector. In Step 1, however, TVAL3 does not exactly minimize with respect to $\boldsymbol{x}$ but updates $\boldsymbol{x}$ by one or more gradient descents. Different from ADM and TVAL3, our approach does not split TV($\boldsymbol{x}$) or exactly minimizes any term involving $f(\boldsymbol{x})$ in Step 1.

It is sometimes tricky to set appropriate penalty parameter α and step length β. One usually has a bound beforehand and tries different values in practice. An excessive large α overweighs the penalty term $(1/2)\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$, causing a slowly convergent or even non-convergent algorithm. We found that $\alpha = 0.05$ worked well and fixed it throughout our simulations. According to Glowinski and Tallec (1989), γ must be strictly less than $(\sqrt{5}+1)/2$ for ADM to converge, but one can simply try with different values. We set γ as 1 in our simulations.

The step length β should be smaller than $2/||\boldsymbol{J}(g(\boldsymbol{x}))||$, where $\boldsymbol{J}(g(\boldsymbol{x}))$ denotes the Jacobian of $g(\boldsymbol{x})$, to essentially guarantee that update (3) and thus Step 1 above are non-expansive. Loosely speaking, $||\boldsymbol{J}(g(\boldsymbol{x}))||$ is basically the max curvature of the graph of $f(\boldsymbol{x}) + \alpha/2||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 - <\boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}>$ around the current $\boldsymbol{x}$. The larger the curvature, the smaller a step should be because the gradient information is accurate in a smaller perimeter. The log function has unlimited curvature but locally, it is bounded. In our numerical study, we fix β as 0.1 and it works well for all our experiments. Line search techniques can be applied to automate β choice.

The problem in Step 1 above is a so-called ROF/TV-L2 denoising problem, for which a few efficient algorithms exist in the literature. They include the latest graph-cut/max-flow algorithms (Chambolle (2005), Darbon and Sigelle (2006), Goldfarb and Yin (2009)). We use the parametric max-flow code from Yin (2010). Next, we discuss how to choose the regularization parameter λ in a data adaptive way.

4. Data-Driven Selection of $\boldsymbol{\lambda}$

It is well known that the choice of the regularization parameter λ is one of the most important steps of the MPLE. It has to be tuned for practical applications. In this work we use the popular K-fold cross-validation (CV) method for density estimation to select the tuning parameter. Liu et al. (2009) gave a brief review of this method in its section 5.1. For completeness, we give details of K-fold CV below in the context of copula density estimation.

We randomly divide all the samples $\{(U_i, V_i)\}_{i=1}^n$ into K disjoint subsets (folds) of approximately the same size. Let S_k be the index set of the kth subset, $k = 1, ..., K$, $\hat{c}_\lambda(u, v)$ be the copula density estimate based on the entire data set, and $\hat{c}_{\lambda,-k}(u, v)$ be the the copula density estimate based on all data points except those in the kth subset.

The quality of a copula density estimator $\hat{c}_\lambda(u, v)$ is measured by $E(\text{Loss}(\hat{c}_\lambda, c))$ where $\text{Loss}(\hat{c}_\lambda, c)$ is a Loss function or distance measure between $\hat{c}_\lambda(u, v)$ and the true copula density estimator $c(u, v)$. Two commonly used distance measure between two densities are integrated squared error

$$\text{ISE}(\hat{c}_\lambda, c) = \int_0^1 \int_0^1 (\hat{c}_\lambda(u, v) - c(u, v))^2 \, dudv,$$

and Kullback-Leibler distance

$$\text{KLD}(\hat{c}_\lambda, c) = \int_0^1 \int_0^1 \log\left(\frac{c(u, v)}{\hat{c}_\lambda(u, v)}\right) c(u, v) dudv. \tag{4}$$

Given a data set $\{(U_i, V_i)\}_{i=1}^n$ generated from $c(u, v)$, we aim to find the λ which minimizes $\text{Loss}(\hat{c}_\lambda, c)$.

Least squares CV represents a data-driven attempt at constructing $\hat{c}_\lambda(u,v)$ so as to minimize $\mathrm{ISE}(\hat{c}_\lambda, c)$. By expanding ISE, we have

$$\mathrm{ISE}(\hat{c}_\lambda, c) = \int_0^1 \int_0^1 \hat{c}_\lambda(u,v)^2 dudv - 2\int_0^1 \int_0^1 \hat{c}_\lambda(u,v)c(u,v)dudv + \int_0^1 \int_0^1 c(u,v)^2 dudv.$$

The term $\int_0^1 \int_0^1 c(u,v)^2 dudv$ does not depend on λ, so it can be dropped for the purpose of searching for λ. The term

$$\int_0^1 \int_0^1 \hat{c}_\lambda(u,v)c(u,v)dudv = E(\hat{c}_\lambda(U,V))$$

may be estimated approximately by

$$\frac{1}{K}\sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i \in S_k} \hat{c}_{\lambda,-k}(U_i, V_i),$$

where $|S_k|$ is the cardinality of S_k. Hence, the least squares CV score $\mathrm{LS}(\lambda)$ is defined as

$$\mathrm{LS}(\lambda) = \int_0^1 \int_0^1 \hat{c}_\lambda(u,v)^2 dudv - \frac{2}{K}\sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i \in S_k} \hat{c}_{\lambda,-k}(U_i, V_i). \tag{5}$$

Likelihood CV represents a data-driven attempt at constructing $\hat{c}_\lambda(u,v)$ so as to minimize $\mathrm{KLD}(\hat{c}_\lambda, c)$ (Hall (1987)). By expanding KLD, we have

$$\mathrm{KLD}(\hat{c}_\lambda, c) = \int_0^1 \int_0^1 (\log c(u,v))\, c(u,v)dudv - \int_0^1 \int_0^1 (\log \hat{c}_\lambda(u,v))\, c(u,v)dudv.$$

The first term on the right hand side above can be dropped for the purpose of searching for λ. The term

$$\int_0^1 \int_0^1 (\log \hat{c}_\lambda(u,v))\, c(u,v)dudv = E\left(\log \hat{c}_\lambda(U,V)\right)$$

may be approximated by

$$\frac{1}{K}\sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i \in S_k} \log \hat{c}_{\lambda,-k}(U_i, V_i).$$

Hence, the likelihood CV score $\mathrm{KL}(\lambda)$ is defined as

$$\mathrm{KL}(\lambda) = -\frac{1}{K}\sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i \in S_k} \log \hat{c}_{\lambda,-k}(U_i, V_i). \tag{6}$$

Then, we choose $\lambda_{CV} = \arg\min \mathrm{CV}_{\lambda \in G}(\lambda)$ as the best tuning parameter, where G is a pre-specified discrete or continuous set in which λ is searched over, and CV score is either LS score or KL score. For simplicity, one usually pre-specifies G as a fine finite grid, where λ_{CV} is found by a simple grid search. For $\mathrm{CV}_{\lambda \in G}(\lambda)$ over a continuous region G, λ_{CV} may be found by some simple single variable minimization methods such as bisection method or golden section search method. One should make sure that λ_{CV} is not at the boundaries of the set G. In case λ_{CV} is located at the boundaries of the set G, one needs to enlarge the G and includes the added portion into the search.

Van der Laan et al. (2004) studied the choice of K. They established asymptotic optimality of K-fold CV, in the sense that the CV selector performs asymptotically as well (w.r.t. to the Kullback-Leibler distance to the true density) as an optimal benchmark model selector which depends on the true density. Crucial conditions of their theorem are that the size of the validation sample n/K goes to infinity, which excludes leave-one-out CV, and that the candidate density estimates are bounded away from zero and infinity. Some copula densities may not be bounded away from infinity, but it is not a concern for finite sample studies.

5. Simulations

We report results from simulation studies which were designed to demonstrate the effectiveness of the MPLE with TV penalty subject to linear equality constraints for copula density estimation and the K-fold CV regularization parameter selector.

The stopping criteria of our augmented Lagrangian and operator-splitting algorithm were $||\boldsymbol{x}^{k+1}-\boldsymbol{x}^k||/||x^k|| <= 10^{-5}$ or total number of iterations reaching 20, where each iteration includes going through steps 1 and 2 once.

In the simulation, the marginal distributions F_X and F_Y were estimated by ECDFs (1). This amounts to use the standardized ranks of the sample $\{(X_i, Y_i)\}_{i=1}^n$ as estimates of $\{(U_i, V_i)\}_{i=1}^n$ (remind that $U_i = F_X(X_i)$ and $V_i = F_Y(Y_i)$). The CDF of a continuous random variable is continuous and increasing within its domain, which implies that the ranks of X_i's are the same as the ranks of U_i's, so are the ranks of Y_i's and those of V_i's. Therefore it is unnecessary to explicitly specify the F_X and F_Y in our simulation for copula density estimation. One can first generate $\{(U_i, V_i)\}_{i=1}^n$ from an underlying copula density $c(u, v)$, then use their standardized ranks as their estimates.

The setting of our simulation study is mostly the same as the one in Autin et al. (2010) as we intend to make a comparison with their wavelet thresholding estimates. We tested five parametric families of copulas: Gaussian, Student, Clayton, Frank and the Gumbel families. For each copula model, independent and identically distributed (i.i.d.) standard uniform bivariate random variables $\{(U_i, V_i)\}_{i=1}^n$ were generated from the specified copula with parameter θ using MATLAB's *copularnd*() function. That was, $\{U_i\}_{i=1}^n$ was a sample from a Uniform(0,1) distribution, and so was the $\{V_i\}_{i=1}^n$. The joint pdf of (U, V) was the specified copula density $c(u, v)$ with parameter θ. The sample sizes considered were $n = 125$, $n = 500$ and $n = 2000$. The grid sizes used were $m = 16$ for $n = 125$, $m = 32$ for $n = 500$ and $m = 64$ for $n = 2000$. In Autin et al. (2010), the rule for setting the grid size m in terms of the sample size n is $m = 4 \times 2^{J_n}$.

Various error measures were evaluated over the equally spaced grid points within $[0, 1]^2$ where the copula densities were estimated. For one data set, the quality of an estimate $\hat{c}_\lambda(u, v)$ of the true copula density $c(u, v)$ was measured by an error measure $\text{Loss}(\hat{c}_\lambda, c)$, which can be either relative errors

$$RE_q(\lambda) = \frac{||\hat{c}_\lambda - c||_{N,q}}{||c||_{N,q}}, \quad \text{for } q = 1,\ 2,\ \infty, \tag{7}$$

or the KLD (4). The sample average of an error measure $\text{Loss}(\hat{c}_\lambda, c)$ over replications of random data set approximates the population mean of the error measure $E(\text{Loss}(\hat{c}_\lambda, c))$ for the proposed estimator $\hat{c}_\lambda(u, v)$. We replicated 100 times for each experiment setting and report the sample average, the associated standard errors (in parentheses) and boxplots of these 100 error measures in Tables 2, 3, 4 and Figs. 5, 6, respectively.

The regularization parameter λ was chosen from the grid $G = \{0.01 \times 10^{2(i-1)/27}\}_{i=1}^{28}$, i.e., 28 equally spaced numbers in [0.01, 1] in a log10 scale. All the best regularization parameters were found near the central portion of this G. For the error measure $\text{Loss}(\hat{c}_\lambda, c)$, the best regularization parameter $\lambda_{Loss} = \arg\min_{\lambda \in G} \text{Loss}(\hat{c}_\lambda, c)$ and the best estimate is Loss-best=$\hat{c}_{\lambda_{Loss}}$. The CV data driven regularization parameter $\lambda_{CV} = \arg\min_{\lambda \in G} \text{CV}(\hat{c}_\lambda, c)$ and the data adaptive estimate TV-CV = $\hat{c}_{\lambda_{CV}}$. The closer the λ_{CV} is to λ_{Loss}, the better a TV-CV is in terms of $\text{Loss}(\hat{c}_\lambda, c)$.

For the number of folders in CV, we used $K = 10$. To see the effectiveness of the 10-fold CV regularization parameter selector, in Fig. 1, we plotted the curves of some $\text{Loss}(\hat{c}_\lambda, c)$ and $\text{CV}(\lambda)$ vs. λ respectively for a typical run of the case: Gaussian copula with $\theta = 0.5$, sample size $n = 2000$, and grid size $m = 64$. In this specific case, the λ_{RE2} which minimized $RE_2(\hat{c}_\lambda, c)$ for $\lambda \in G$ coincided with λ_{KLD} which minimized $\text{KLD}(\hat{c}_\lambda, c)$. The λ_{LS} which minimized $\text{LS}(\lambda)$ was 1 grid below λ_{RE2}; The λ_{KL} which minimized $\text{KL}(\lambda)$ was 1 grid above λ_{KLD}. Fig. 2 shows the scatter plots of (a)the original data $\{(X_i, Y_i)\}_{i=1}^n$ with standard Gaussian marginals; (b)the original data $\{(U_i, V_i)\}_{i=1}^n$ with standard Uniform marginals; and (c)the standardized ranks $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$. Note the close similarity of (b) and (c). Table 1 summarizes the shorthand notations for different estimates. Fig. 3 display the surface plots on the left column and contour plots on the right column of the true and estimated copula densities. For comparison, we computed a 2D kernel density estimate using the kde2D program (Botev et al. (2010); Botev (2011)). Obviously, there is an oversmoothing by KDE. The RE2-best catches the two peaks in the front and back corners well. Both the TV-LS and TV-KL are close to RE2-best.

Table 1: Shorthand notations for different estimates

RE2-best	estimate with the λ chosen to minimize the RE2(λ) [equation (7) with $q = 2$]
TV-LS	estimate with the λ chosen to minimize the LS(λ) [equation (5)]
TV-KL	estimate with the λ chosen to minimize the KL(λ) [equation (6)]
KDE	kernel density estimate by the *kde2D* MATLAB program [Botev (2011)]

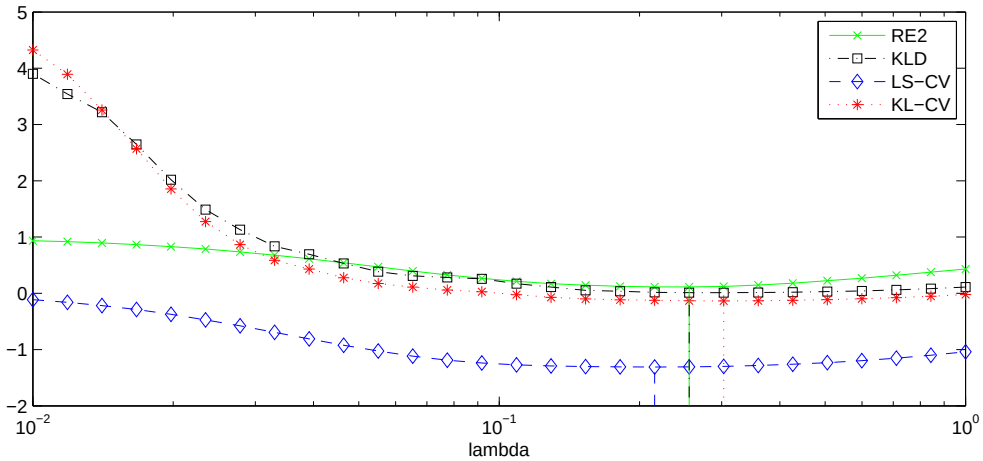

Figure 1: Loss and CV curves in a typical run of the case: Gaussian copula with $\theta = 0.5$, sample size $n = 2000$, grid size $m = 64$.

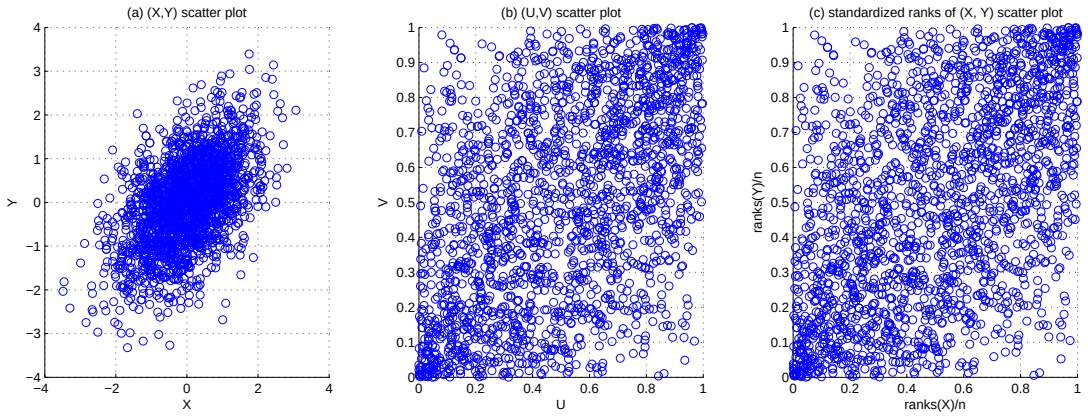

Figure 2: scatter plots of (a) Y vs. X; (b) V vs. U; and (c)the standardized ranks of Y vs. X in a typical run of the case: Gaussian copula with $\theta = 0.5$, sample size $n = 2000$, and grid size $m = 64$.

To have a sense of the speed of the algorithm, for the data set used for Fig. 1, Fig. 4 plot the times in seconds needed to obtain the solution $\hat{c}_\lambda$ from the full data set for a sequence of λ and the times in seconds needed to obtain both the LS(λ) and KL(λ) for the 10-fold CV for a sequence of λ. For a fixed λ, 10-fold CV took 3.42 seconds on average to finish, while it took only 0.4 seconds on average to obtain $\hat{c}_\lambda$ for the full data set. This computational efficiency is an order of magnitude improvement over another TV-MPLE method in Qu et al. (2009) solved by log-barrier method for second order cone program (SOCP) which took 2 minutes to solve the similar problem. We did not compare our proposed method in this paper with the method in Qu et al. (2009) because of the low computational efficiency of the latter.

The side-by-side boxplots in Figs. 5 and 6 show that both TV-LS and TV-KL are close to TV-best. In general, TV-LS is closer to RE_q-best than TV-KL; and TV-KL is closer to KL-best than TV-LS which is what we should expect because the goal of TV-LS is to minimize RE_2 and the goal of TV-KL is to minimize KLD. All the TV-estimators outperform the conventional kernel density estimator.

Tables 2, 3 and 4 list Monte Carlo approximations to $E(\text{Loss}(\hat{c}_\lambda, c))$ over 100 replications for (1) $n = 125$, $m = 16$; (2) $n = 500$, $m = 32$; and (3) $n = 2000$, $m = 64$ respectively. Comparing the mean RE_q of our TV-LS with those of WaveThresh-Local in Table A.3 and A.4 of Autin et al. (2010), we observe that (1) the mean RE_1 of TV-LS is mostly smaller than those by WaveThresh-Local; (2) the mean RE_2 of TV-LS is all larger than those by WaveThresh-Local except for the Gumbel copula with $\theta = 8.3$; (3) the mean RE_∞ of TV-LS is all smaller than those by WaveThresh-Local except for the Frank copula with $\theta = 4.0$.

5.1. Selecting a Parametric Copula from Several Parametric Families

Our MPLE-TV copula density estimate can serve the purpose to select a parametric copula from several parametric families. A parametric copula c_θ is wholly determined by its parameter θ. The parameter θ can be estimated by classical parameter estimation methods such as maximum likelihood. We measure the distance between our nonparametric estimate $\hat{c}_\lambda$ and the parametric estimate $c_{\hat{\theta}}$ by their relative errors

$$RE_q(\hat{\theta}) = \frac{||\hat{c}_\lambda - c_{\hat{\theta}}||_{N,q}}{||c_{\hat{\theta}}||_{N,q}}, \quad \text{for } q = 1,\ 2,\ \infty.$$

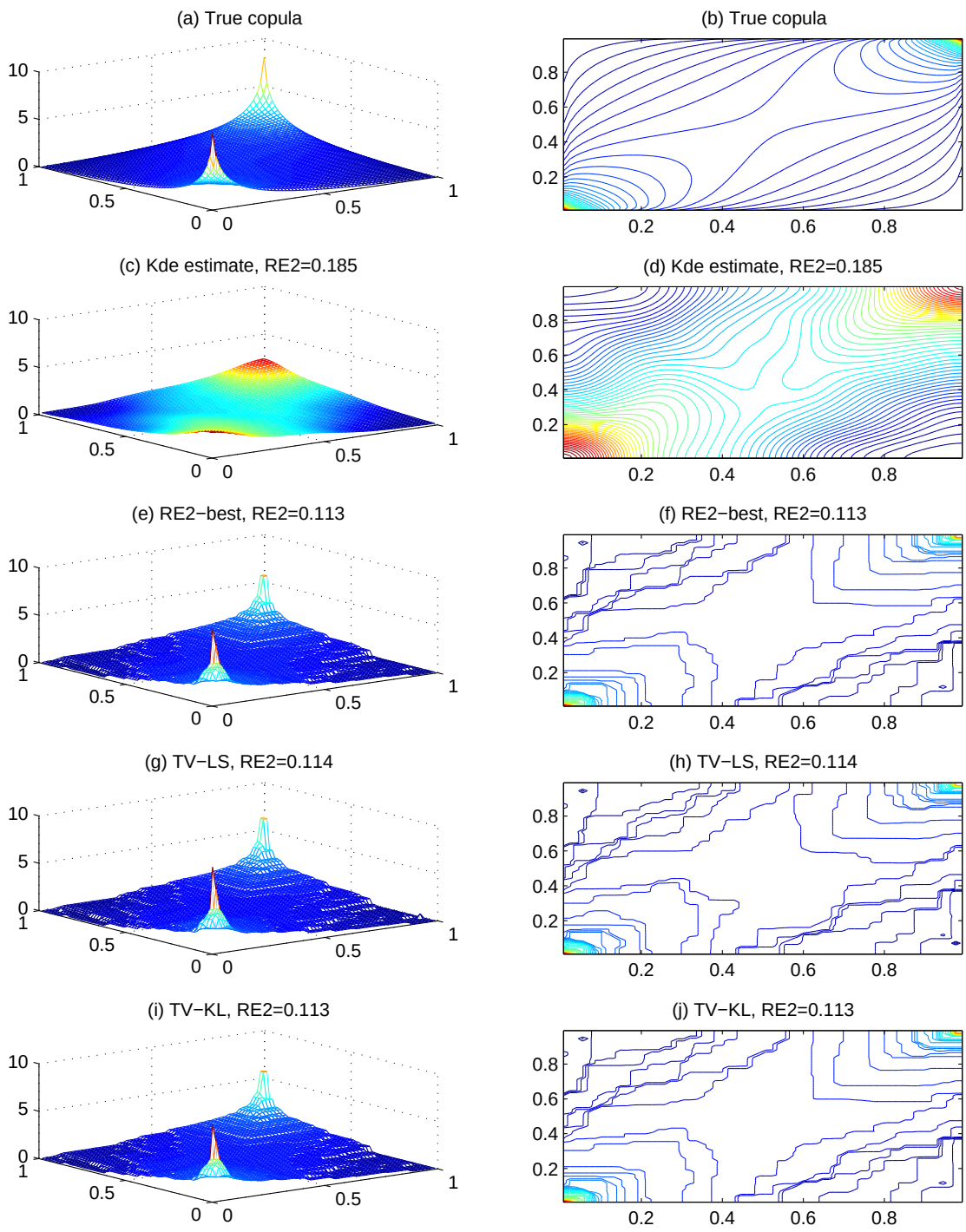

Figure 3: True and estimated copula densities in a typical run of the case: Gaussian copula with $\theta = 0.5$, sample size $n = 2000$, grid size $m = 64$.

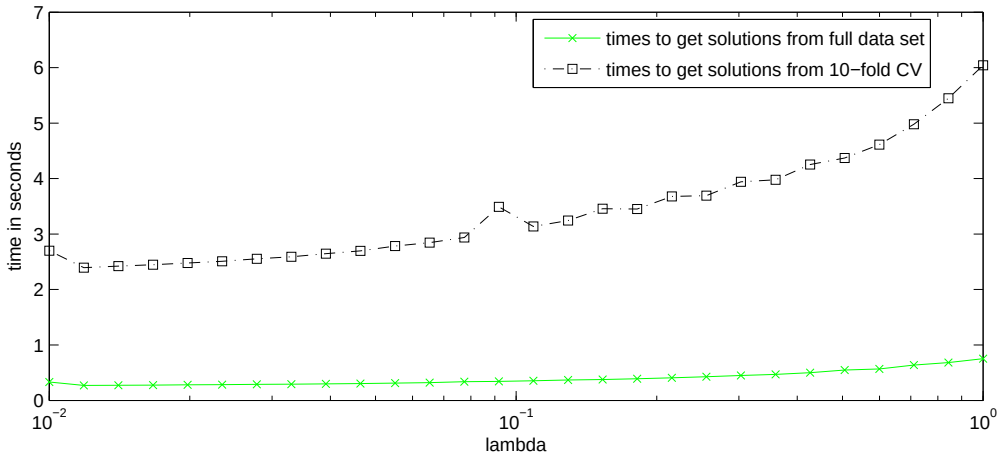

Figure 4: Times (in seconds) to get solutions for a sequence of λ in a typical run of the case: Gaussian copula with $\theta = 0.5$, sample size $n = 2000$, grid size $m = 64$.

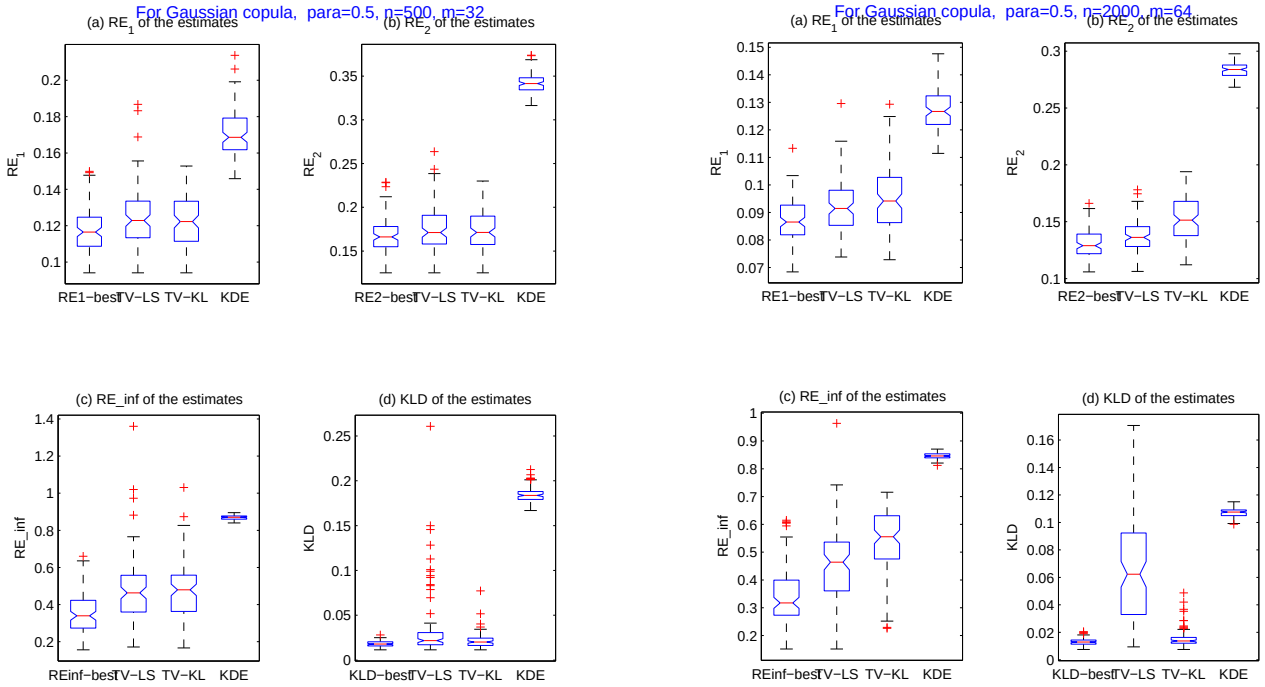

Figure 5: Boxplots of errors of different estimates for the Gaussian copula with $\theta = 0.5$.

The selected parametric copula is the one with the smallest $RE_q(\hat{\theta})$ among all parametric candidates.

A simulation study was to illustrate this. The true underlying copula density was Gaussian with $\theta = 0.5$. An i.i.d. standard uniform bivariate random sample $\{(U_i, V_i)\}_{i=1}^n$ was generated from it with $n = 500$. MPLE-TV estimate $\hat{c}_\lambda$ was constructed based on the data $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$ with grid size $m = 32$ and λ selected by 10-fold CV. The θ was estimated by the Canonical Maximum Likelihood (CML) method using MATLAB's *copulafit*() function based on the data $\{(\hat{U}_i, \hat{V}_i)\}_{i=1}^n$.

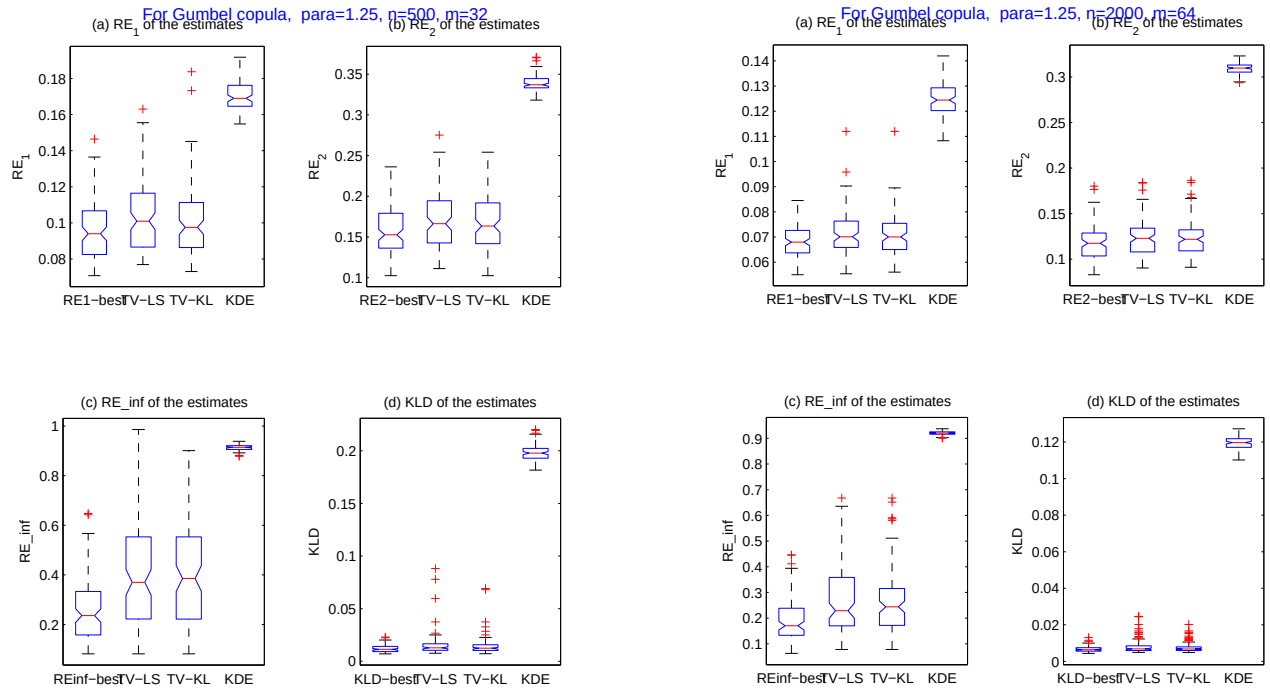

Figure 6: Boxplots of errors of different estimates for the Gumbel copula with $\theta = 1.25$.

Table 5 reports the $RE_q(\hat{\theta})$ for 4 different candidates. MPLE-TV estimate was closest to the Gaussian estimate in terms of relative errors. We correctly selected the Gaussian model among four parametric families considered.

Table 6 reports the number of successes of MPLE-TV estimate for selecting a parametric copula out of 4 different parametric candidates over 100 replications. For example, when $n = 500$ observations were generated from Frank copula with $\theta = 4.0$, TV-LS correctly selected Frank copula as the underlying density out of Clayton, Frank, Gaussian and Gumbel families 92 times in terms of the RE_1 criterion over 100 replications. The simulation results indicate that this parametric copula selection strategy is reliable with moderate and large sample sizes. There is no clear winner among RE_1, RE_2 and RE_∞ error measures, but RE_∞ tends to perform poorly because it uses only a single number (the maximum) of an estimate. RE_2 outperforms RE_1 in the Frank(4) case only. A close look at the true copula density reveals that the sharp peaks at the corners of the Frank(4) copula density are much lower than those of the other three. The high peak regions are harder to be estimated accurately than the low peak ones. Squares of the differences in RE_2 enlarge the error measure RE_2 in comparison to RE_1 when the estimated peak regions between the nonparametric estimate $\hat{c}_\lambda$ and parametric estimate $c_{\hat{\theta}}$ disagree greatly, which partly contribute to the under-performance of the RE_2 relative to RE_1 in the Clayton(0.8), Gaussian(0.5) and Gumbel(1.25) cases. In general, RE_1 seems to be a more robust error measure.

6. Application to Real Data

We apply our MPLE-TV method to a subset of the Framingham Heart study data (`http://www.framingham.com/heart/`). We focus on the dependence structure underlying the diastolic (DBP) and the systolic (SBP) blood pressures (in mmHg) measured on 663 male subjects at their

Table 2: Monte Carlo approximations to $E(\text{Loss}(\hat{c}_\lambda, c))$ over 100 replications for $n = 125$, $m = 16$

Copula	par.	Method	RE_1	RE_2	RE_∞	KLD
Gaussian	0.00	TV-best	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
Gaussian	0.00	TV-LS	0.013 (0.039)	0.022 (0.065)	0.109 (0.356)	0.003 (0.010)
Gaussian	0.00	TV-KL	0.010 (0.032)	0.018 (0.053)	0.090 (0.289)	0.002 (0.008)
Gaussian	0.00	KDE	0.230 (0.004)	0.281 (0.004)	0.613 (0.034)	0.280 (0.005)
Gaussian	0.50	TV-best	0.160 (0.023)	0.210 (0.034)	0.370 (0.125)	0.030 (0.008)
Gaussian	0.50	TV-LS	0.178 (0.036)	0.234 (0.048)	0.489 (0.163)	0.044 (0.066)
Gaussian	0.50	TV-KL	0.177 (0.035)	0.231 (0.045)	0.472 (0.149)	0.050 (0.072)
Gaussian	0.50	KDE	0.235 (0.027)	0.385 (0.030)	0.782 (0.027)	0.271 (0.028)
Gaussian	0.90	TV-best	0.299 (0.034)	0.321 (0.051)	0.332 (0.108)	0.149 (0.027)
Gaussian	0.90	TV-LS	0.311 (0.037)	0.334 (0.054)	0.409 (0.124)	0.346 (0.155)
Gaussian	0.90	TV-KL	0.329 (0.048)	0.353 (0.060)	0.466 (0.124)	0.186 (0.074)
Gaussian	0.90	KDE	0.463 (0.057)	0.595 (0.026)	0.838 (0.018)	0.355 (0.053)
Student	0.50	TV-best	0.324 (0.031)	0.448 (0.065)	0.488 (0.134)	0.142 (0.023)
Student	0.50	TV-LS	0.348 (0.044)	0.480 (0.069)	0.638 (0.117)	0.159 (0.031)
Student	0.50	TV-KL	0.345 (0.043)	0.475 (0.066)	0.632 (0.111)	0.156 (0.030)
Student	0.50	KDE	0.398 (0.031)	0.700 (0.018)	0.920 (0.014)	0.461 (0.045)
Clayton	0.80	TV-best	0.154 (0.026)	0.227 (0.051)	0.278 (0.132)	0.033 (0.009)
Clayton	0.80	TV-LS	0.174 (0.040)	0.254 (0.060)	0.411 (0.180)	0.044 (0.031)
Clayton	0.80	TV-KL	0.167 (0.034)	0.244 (0.058)	0.397 (0.181)	0.039 (0.016)
Clayton	0.80	KDE	0.243 (0.019)	0.444 (0.021)	0.868 (0.017)	0.289 (0.025)
Frank	4.00	TV-best	0.182 (0.027)	0.213 (0.033)	0.348 (0.096)	0.030 (0.009)
Frank	4.00	TV-LS	0.203 (0.037)	0.237 (0.045)	0.481 (0.195)	0.056 (0.075)
Frank	4.00	TV-KL	0.200 (0.035)	0.233 (0.040)	0.465 (0.173)	0.047 (0.047)
Frank	4.00	KDE	0.255 (0.034)	0.377 (0.031)	0.722 (0.030)	0.255 (0.029)
Gumbel	8.30	TV-best	0.534 (0.048)	0.697 (0.025)	0.785 (0.022)	0.729 (0.067)
Gumbel	8.30	TV-LS	0.564 (0.050)	0.711 (0.028)	0.794 (0.024)	0.785 (0.094)
Gumbel	8.30	TV-KL	0.579 (0.055)	0.716 (0.030)	0.797 (0.025)	0.777 (0.085)
Gumbel	8.30	KDE	0.843 (0.031)	0.878 (0.008)	0.941 (0.005)	1.320 (0.072)
Gumbel	1.25	TV-best	0.123 (0.020)	0.187 (0.037)	0.315 (0.145)	0.019 (0.006)
Gumbel	1.25	TV-LS	0.150 (0.039)	0.227 (0.055)	0.514 (0.186)	0.028 (0.013)
Gumbel	1.25	TV-KL	0.148 (0.039)	0.224 (0.055)	0.501 (0.193)	0.028 (0.013)
Gumbel	1.25	KDE	0.220 (0.011)	0.358 (0.019)	0.831 (0.021)	0.280 (0.017)

first visit. The scatterplot of the log-blood pressures and the scatterplot of the standardized ranks of the log-blood pressures can be found in Fig. 7. It is evident that there is a strong positive dependence between the two responses. Lambert (2007) analyzed this data set assuming the copula density of the log-blood pressures was a sub-family of copulas named Archimedean with unknown (strict) generator. Lambert (2007) proposed a ratio approximation of the Archimedean copula generator and of its first derivative using B-splines, estimated the associated parameters using Markov chains Monte Carlo methods, and found that Gumbel copula was appropriate for

Table 3: Monte Carlo approximations to $E(\text{Loss}(\hat{c}_\lambda, c))$ over 100 replications for $n = 500$, $m = 32$

Copula	par.	Method	RE_1	RE_2	RE_∞	KLD
Gaussian	0.00	TV-best	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
Gaussian	0.00	TV-LS	0.005 (0.017)	0.009 (0.025)	0.066 (0.226)	0.000 (0.001)
Gaussian	0.00	TV-KL	0.004 (0.015)	0.007 (0.022)	0.047 (0.140)	0.000 (0.001)
Gaussian	0.00	KDE	0.179 (0.008)	0.245 (0.006)	0.771 (0.033)	0.201 (0.002)
Gaussian	0.50	TV-best	0.118 (0.012)	0.167 (0.020)	0.351 (0.111)	0.018 (0.003)
Gaussian	0.50	TV-LS	0.125 (0.017)	0.177 (0.027)	0.473 (0.182)	0.034 (0.037)
Gaussian	0.50	TV-KL	0.122 (0.015)	0.175 (0.023)	0.469 (0.149)	0.022 (0.009)
Gaussian	0.50	KDE	0.171 (0.013)	0.343 (0.012)	0.869 (0.012)	0.185 (0.009)
Gaussian	0.90	TV-best	0.209 (0.018)	0.235 (0.035)	0.255 (0.105)	0.103 (0.016)
Gaussian	0.90	TV-LS	0.213 (0.018)	0.239 (0.036)	0.323 (0.126)	0.236 (0.075)
Gaussian	0.90	TV-KL	0.264 (0.027)	0.299 (0.040)	0.434 (0.128)	0.127 (0.042)
Gaussian	0.90	KDE	0.265 (0.022)	0.528 (0.010)	0.888 (0.008)	0.192 (0.013)
Student	0.50	TV-best	0.225 (0.018)	0.358 (0.053)	0.423 (0.111)	0.074 (0.008)
Student	0.50	TV-LS	0.230 (0.020)	0.372 (0.056)	0.520 (0.115)	0.081 (0.023)
Student	0.50	TV-KL	0.231 (0.020)	0.374 (0.056)	0.524 (0.114)	0.078 (0.010)
Student	0.50	KDE	0.266 (0.021)	0.681 (0.011)	0.948 (0.007)	0.279 (0.019)
Clayton	0.80	TV-best	0.112 (0.012)	0.187 (0.039)	0.228 (0.104)	0.019 (0.004)
Clayton	0.80	TV-LS	0.120 (0.017)	0.199 (0.041)	0.346 (0.164)	0.032 (0.038)
Clayton	0.80	TV-KL	0.116 (0.013)	0.199 (0.042)	0.359 (0.167)	0.022 (0.007)
Clayton	0.80	KDE	0.179 (0.012)	0.434 (0.012)	0.924 (0.009)	0.199 (0.011)
Frank	4.00	TV-best	0.129 (0.017)	0.152 (0.022)	0.319 (0.088)	0.016 (0.004)
Frank	4.00	TV-LS	0.136 (0.021)	0.162 (0.030)	0.426 (0.219)	0.040 (0.055)
Frank	4.00	TV-KL	0.134 (0.019)	0.159 (0.023)	0.398 (0.165)	0.018 (0.006)
Frank	4.00	KDE	0.181 (0.018)	0.307 (0.015)	0.805 (0.022)	0.174 (0.008)
Gumbel	8.30	TV-best	0.388 (0.021)	0.679 (0.016)	0.788 (0.014)	0.388 (0.024)
Gumbel	8.30	TV-LS	0.394 (0.022)	0.685 (0.017)	0.793 (0.014)	0.437 (0.045)
Gumbel	8.30	TV-KL	0.429 (0.036)	0.697 (0.019)	0.799 (0.015)	0.403 (0.033)
Gumbel	8.30	KDE	0.541 (0.022)	0.858 (0.004)	0.959 (0.002)	0.689 (0.024)
Gumbel	1.25	TV-best	0.095 (0.016)	0.157 (0.030)	0.259 (0.125)	0.012 (0.003)
Gumbel	1.25	TV-LS	0.104 (0.020)	0.170 (0.035)	0.391 (0.196)	0.016 (0.012)
Gumbel	1.25	TV-KL	0.102 (0.021)	0.167 (0.035)	0.393 (0.192)	0.015 (0.009)
Gumbel	1.25	KDE	0.171 (0.009)	0.339 (0.011)	0.913 (0.012)	0.198 (0.008)

this data without being fully satisfactory.

We applied our estimation procedure to this data set, and used 10-fold LS and KL CV to select the regularization parameter λ. The grid size m was set to 38. This (m, n) pair is roughly the linear interpolation between $(m, n) = (32, 500)$ and $(m, n) = (64, 2000)$. We estimated parametric copula densities by assuming Gumbel, Gaussian, Clayton and Frank copula respectively for the data. Table 7 lists relative errors $RE_q(\hat{\theta})$. We found that Gumbel copula was closet to our TV-LS estimate. This is in agreement with Lambert (2007)'s finding that Gumbel copula is appropriate

Table 4: Monte Carlo approximations to $E(\text{Loss}(\hat{c}_\lambda, c))$ over 100 replications for $n = 2000$, $m = 64$

Copula	par.	Method	RE_1	RE_2	RE_∞	KLD
Gaussian	0.00	TV-best	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	-0.000 (0.000)
Gaussian	0.00	TV-LS	0.005 (0.015)	0.009 (0.022)	0.088 (0.284)	0.000 (0.001)
Gaussian	0.00	TV-KL	0.004 (0.011)	0.007 (0.016)	0.052 (0.198)	0.000 (0.001)
Gaussian	0.00	KDE	0.131 (0.005)	0.189 (0.003)	0.678 (0.034)	0.126 (0.001)
Gaussian	0.50	TV-best	0.087 (0.008)	0.131 (0.013)	0.346 (0.100)	0.013 (0.003)
Gaussian	0.50	TV-LS	0.092 (0.010)	0.137 (0.014)	0.456 (0.132)	0.063 (0.036)
Gaussian	0.50	TV-KL	0.096 (0.012)	0.152 (0.019)	0.534 (0.116)	0.016 (0.007)
Gaussian	0.50	KDE	0.127 (0.007)	0.283 (0.007)	0.846 (0.011)	0.107 (0.003)
Gaussian	0.90	TV-best	0.153 (0.008)	0.181 (0.021)	0.195 (0.074)	0.098 (0.010)
Gaussian	0.90	TV-LS	0.156 (0.012)	0.184 (0.025)	0.246 (0.092)	0.162 (0.031)
Gaussian	0.90	TV-KL	0.264 (0.017)	0.317 (0.071)	0.480 (0.278)	0.111 (0.022)
Gaussian	0.90	KDE	0.183 (0.009)	0.459 (0.009)	0.858 (0.009)	0.096 (0.005)
Student	0.50	TV-best	0.159 (0.009)	0.315 (0.037)	0.417 (0.085)	0.042 (0.004)
Student	0.50	TV-LS	0.165 (0.011)	0.320 (0.038)	0.472 (0.087)	0.093 (0.047)
Student	0.50	TV-KL	0.162 (0.010)	0.335 (0.038)	0.495 (0.088)	0.045 (0.010)
Student	0.50	KDE	0.200 (0.008)	0.648 (0.007)	0.934 (0.006)	0.143 (0.007)
Clayton	0.80	TV-best	0.080 (0.007)	0.140 (0.026)	0.155 (0.071)	0.013 (0.003)
Clayton	0.80	TV-LS	0.085 (0.010)	0.145 (0.027)	0.223 (0.108)	0.035 (0.026)
Clayton	0.80	TV-KL	0.085 (0.011)	0.154 (0.031)	0.255 (0.125)	0.015 (0.006)
Clayton	0.80	KDE	0.135 (0.008)	0.408 (0.008)	0.921 (0.007)	0.111 (0.005)
Frank	4.00	TV-best	0.093 (0.009)	0.107 (0.010)	0.253 (0.054)	0.009 (0.002)
Frank	4.00	TV-LS	0.096 (0.011)	0.111 (0.012)	0.324 (0.138)	0.049 (0.048)
Frank	4.00	TV-KL	0.099 (0.012)	0.117 (0.015)	0.335 (0.065)	0.011 (0.005)
Frank	4.00	KDE	0.130 (0.009)	0.226 (0.008)	0.707 (0.022)	0.103 (0.003)
Gumbel	8.30	TV-best	0.270 (0.010)	0.661 (0.010)	0.790 (0.009)	0.227 (0.011)
Gumbel	8.30	TV-LS	0.272 (0.010)	0.664 (0.010)	0.795 (0.010)	0.250 (0.022)
Gumbel	8.30	TV-KL	0.322 (0.030)	0.679 (0.012)	0.801 (0.010)	0.235 (0.013)
Gumbel	8.30	KDE	0.348 (0.010)	0.823 (0.003)	0.945 (0.002)	0.324 (0.006)
Gumbel	1.25	TV-best	0.068 (0.006)	0.119 (0.019)	0.190 (0.078)	0.007 (0.002)
Gumbel	1.25	TV-LS	0.072 (0.009)	0.124 (0.021)	0.265 (0.127)	0.008 (0.004)
Gumbel	1.25	TV-KL	0.071 (0.008)	0.124 (0.020)	0.267 (0.129)	0.008 (0.003)
Gumbel	1.25	KDE	0.125 (0.007)	0.309 (0.006)	0.920 (0.008)	0.119 (0.004)

for this data. Fig. 8 plots the TV-LS on the left panel and the Gumbel copula density on the right panel. They look similar, with some difference in the front corner.

7. Concluding Remarks

We presented a TV penalized maximum likelihood copula density estimate subject to the constraints that the marginal distributions are standard uniforms. The linear equality constrained

Table 5: Relative error $RE_q(\hat{\theta})$ for the simulated data from Gaussian(0.5) with $n = 500$, $m = 32$

MPLE-TV Estimate	Parametric Estimate	$RE_1(\hat{\theta})$	$RE_2(\hat{\theta})$	$RE_\infty(\hat{\theta})$
TV-LS	Gaussian	0.1116	0.1807	0.7838
TV-KL	Gaussian	0.0938	0.1517	0.6361
TV-LS	Clayton	0.1373	0.1987	0.2440
TV-KL	Clayton	0.1164	0.1813	0.2343
TV-LS	Frank	0.1181	0.2202	1.9224
TV-KL	Frank	0.1069	0.1915	1.6805
TV-LS	Gumbel	0.5921	0.6974	0.9371
TV-KL	Gumbel	0.6135	0.7004	0.9255

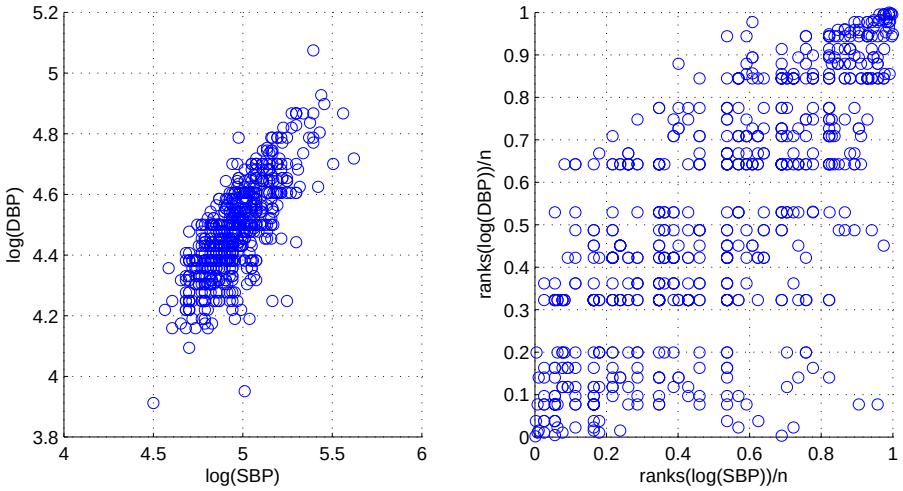

Figure 7: Left: log(DBP) vs. log(SBP); Right: standardized ranks of log(DBP) vs. those of standardized ranks of log(SBP)

TV regularized MPLE problem is solved by an augmented Lagrangian combined with operator-splitting algorithm. A fast ROF/TV-L2 denoising solver is at the core of the method. The K-fold CV regularization parameter selector based on integrated squared error or Kullback-Leibler distance works well.

The extension of our method to trivariate copula density estimation requires solving a 3D ROF/TV-L2 denoising problem in the step 1 of the operator-splitting iteration. The parametric maximum flow algorithms in Goldfarb and Yin (2009) and code from Yin (2010) used in this paper is capable of doing this. Hence our approach is readily to be extended to trivariate case. In higher dimensions, nonparametric copula density estimation is more challenging due to the "curse of dimensionality".

The theoretical questions such as the consistency and convergence rate of the estimator wait to be investigated. Few work exists regarding the asymptotic properties of TV regularized estimators, even in denoising problems. The consistency theorems in 1D TV-denoising problems have recently

Table 6: Number of successes of MPLE-TV estimate for selecting a parametric copula from 4 different parametric candidates over 100 replications

MPLE-TV Estimate	True Copula	n	m	RE_1	RE_2	RE_∞
TV-LS	Clayton(0.8)	125	16	53	38	35
TV-KL	Clayton(0.8)	125	16	52	36	30
TV-LS	Frank(4)	125	16	41	81	97
TV-KL	Frank(4)	125	16	36	83	95
TV-LS	Gaussian(0.5)	125	16	39	28	11
TV-KL	Gaussian(0.5)	125	16	42	28	11
TV-LS	Gumbel(1.25)	125	16	46	34	24
TV-KL	Gumbel(1.25)	125	16	47	39	30
TV-LS	Clayton(0.8)	500	32	96	80	65
TV-KL	Clayton(0.8)	500	32	94	81	65
TV-LS	Frank(4)	500	32	92	99	97
TV-KL	Frank(4)	500	32	79	99	98
TV-LS	Gaussian(0.5)	500	32	92	64	52
TV-KL	Gaussian(0.5)	500	32	92	55	42
TV-LS	Gumbel(1.25)	500	32	80	65	42
TV-KL	Gumbel(1.25)	500	32	79	61	40
TV-LS	Clayton(0.8)	2000	64	100	100	94
TV-KL	Clayton(0.8)	2000	64	100	100	85
TV-LS	Frank(4)	2000	64	100	100	99
TV-KL	Frank(4)	2000	64	94	100	100
TV-LS	Gaussian(0.5)	2000	64	99	91	64
TV-KL	Gaussian(0.5)	2000	64	100	46	27
TV-LS	Gumbel(1.25)	2000	64	97	96	84
TV-KL	Gumbel(1.25)	2000	64	97	95	83

been proved in Dumbgen and Kovac (2009). Much work is ahead to establish the consistency results for TV regularized density estimators.

The MATLAB code implementing the method is available on the authors website.

Acknowledgment

We thank Philippe Lambert for providing the Framingham Heart study data.

The second author's work was supported in part by NSF Grant DMS-07-48839, ONR Grant N00014-08-1-1101, and an Alfred P. Sloan Research Fellowship.

We thank the Editor, the anonymous associate editor and two referees for many constructive comments which led to a substantially improved paper.

Table 7: Relative error $RE_q(\hat{\theta})$ for the real data with $n = 663$, $m = 38$

MPLE-TV Estimate	Parametric Estimate	$RE_1(\hat{\theta})$	$RE_2(\hat{\theta})$	$RE_\infty(\hat{\theta})$
TV-LS	Gumbel	0.2299	0.3162	0.3085
TV-KL	Gumbel	0.2438	0.3261	0.3257
TV-LS	Gaussian	0.2326	0.3512	0.7155
TV-KL	Gaussian	0.2429	0.3519	0.7315
TV-LS	Clayton	0.4064	0.7008	0.8767
TV-KL	Clayton	0.3919	0.6944	0.8837
TV-LS	Frank	0.2598	0.3907	2.5778
TV-KL	Frank	0.2763	0.3880	2.4886

References

Autin, F., Pennecb, E.L., Tribouley, K., 2010. Thresholding methods to estimate the copula density. Journal of Multivariate Analysis 101, 200–222.

Balakrishnan, N., Lai, C.D., 2009. Continuous Bivariate Distributions. Springer.

Botev, Z., 2011. `http://www.mathworks.com/matlabcentral/fileexchange/17204-kernel-density-estimation`.

Botev, Z., Grotowski, J., Kroese, D.P., 2010. Kernel density estimation via diffusion. Annals of Statistics 38, 2916–2957.

Chambolle, A., 2005. Total variation minimization and a class of binary MRF models. Ecole Polytechnique.

Chen, S., Poon, S., 2007. Modelling international stock market contagion using copula and risk appetite. `http://ssrn.com/abstract=1024288`.

Cherubini, U., Luciano, E., Vecchiato, W., 2004. Copula Methods in Finance. John Wiley & Sons, New York.

Choros, B., Ibragimov, R., Permiakova, E., 2010. Copula estimation, in: Copula Theory and Its Applications, Lecture Notes in Statistics, V198, Springer. pp. 77–91.

Cox, D., Wermuth, N., 1996. Multivariate Dependencies: Models, Analysis and Interpretation. CRC.

Darbon, J., Sigelle, M., 2006. Image restoration with discrete constrained total variation, part i: fast and exact optimization. Journal of Mathematical Imaging and Vision 26, 261–276.

Dumbgen, L., Kovac, A., 2009. Extensions of smoothing via taut strings. Electronic Journal of Statistics 3, 41–75.

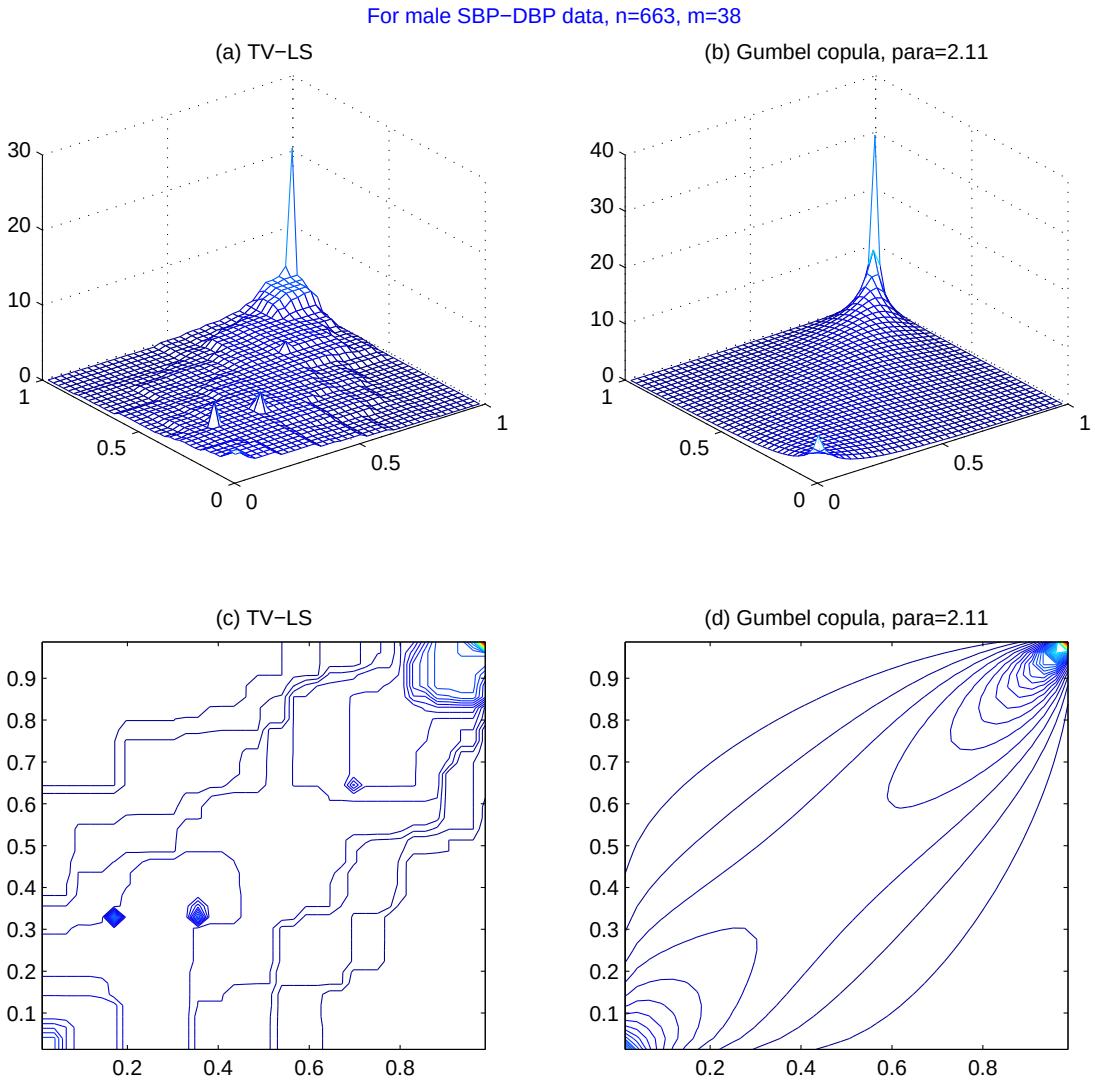

Figure 8: Left: TV based estimate, with λ chosen by 10-fold least squares (LS) cross-validation(CV); Right: parametric estimate assuming Gumbel copula

Embrechts, P., 2009. Copulas: a personal view. Journal of Risk and Insurance 76, 639–650.

Embrechts, P., Lindskog, F., McNeil, A.J., 2003. Modelling dependence with copulas and applications to risk management, in: Rachev, S.T. (Ed.), Handbook of Heavy Tailed Distributions in Finance. Elsevier, North-Holland, Amserdam. chapter 8, pp. 329–384.

Fermanian, J., Scaillet, O., 2003. Nonparametric estimation of copulas for time series. Journal of Risk 5, 25–54.

Genest, C., Gendron, M., Bourdeau-Brien, M., 2009. The advent of copulas in finance. The European Journal of Finance 15, 609–618.

Gijbels, I., Mielniczuk, J., 1990. Estimating the density of a copula function. Communications in Statistics - Theory and Methods 19, 445–464.

Glowinski, R., Tallec, P., 1989. Augmented Lagrangian and Operator-Splitting Methods. SIAM.

Goldfarb, D., Yin, W., 2009. Parametric maximum flow algorithms for fast total variation minimization. SIAM J. Scientific Computing 31, 3712–3743.

Goldstein, T., Osher, S., 2009. The split bregman method for l1 regularized problems. SIAM J. Imaging Sciences 2, 323–343.

Hall, P., 1987. On kullback-leibler loss and density estimation. Annals of Statistics 15, 1491–1519.

Hall, P., Neumeyer, N., 2006. Estimating a bivariate density when there are extra data on one or both components. Biometrika 93, 439–450.

Haug, S., Kluppelberg, C., Peng, L., 2011. Statistical models and methods for dependence in insurance data. Journal of the Korean Statistical Society In Press.

Junker, M., May, A., 2005. Measurement of aggregate risk with copulas. Econ J 8, 428–454.

Koenker, R., Mizera, I., 2007. Density estimation by total variation regularization, in: Nair, V. (Ed.), Aadvances in Statistical Modeling and Inference Essays in Honor of Kjell A Doksum. World Scientific. chapter 30, pp. 613–634.

Kolve, N., dos Anjos, U., Mendes, B., 2006. Copulas: a review and recent developments. Stochastic Models 22, 617–660.

Van der Laan, M.J., Dudoit, S., , Keles, S., 2004. Asymptotic optimality of likelihood based cross-validation. Statistical Applications in Genetics and Molecular Biology 3.

Lambert, P., 2007. Archimedean copula estimation using Bayesian splines smoothing techniques. Computational Statistics & Data Analysis 51, 6307–6320.

Li, C., Yin, W., Zhang, Y., 2009. TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms. `http://www.caam.rice.edu/~optimization/L1/TVAL3/`.

Li, D., 2000. On default correlation: a copula approach. J Fixed Income 9, 43–54.

Liu, L., Levine, M., Zhu, Y., 2009. A functional EM algorithm for mixing density estimation via nonparametric penalized likelihood maximization. Journal of Computational and Graphical Statistics 18, 481–504.

Malevergne, Y., Sornette, D., 2003. Testing the gaussian copula hypothesis for financial assets dependence. Quant Fin 3, 231250.

Malevergne, Y., Sornette, D., 2006. Extreme Financial Risks. Heidelberg: Springer.

Mikosch, T., 2006. Copulas: tales and facts. Extremes 9, 3–20.

Mohler, G.O., Bertozzi, A.L., Goldstein, T.A., Osher, S.J., 2010. Fast TV regularization for 2D maximum penalized likelihood estimation. Journal of Computational and Graphical Statistics .

Nelsen, R.B., 2006. An Introduction to Copulas. Lecture Notes in Statistics, Springer, New York. 2 edition.

Patton, A., 2009. Copula-based models for financial time series, in: Andersen, T., Davis, R., Kreiss, J.P., Mikosch, T. (Eds.), Handbook of Financial Time Series. Springer, Verlag, pp. 767–785.

Qu, L., Qian, Y., Xie, H., 2009. Copula density estimation by total variation penalized likelihood. Communications in Statistics - Simulation and Computation 38, 1891–1908.

Rodriguez, J., 2007. Measuring financial contagion: a copula approach. J Empir Fin 41, 401–423.

Sancetta, A., Satchell, S., 2004. The Bernstein copula and its applications to modeling and approximations of multivariate distributions. Econometric Theory 20, 535–562.

Sardy, S., Tseng, P., 2010. Density estimation by total variation penalized likelihood driven by the sparsity l1 information criterion. Scandinavian Journal of Statistics 37, 321–337.

Scott, D., 1992. Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley: New York.

Shen, X., Zhu, Y., Song, L., 2008. Linear b-spline copulas with applications to nonparametric estimation of copulas. Computational Statistics & Data Analysis 52, 3806–3819.

Shih, J.H., Louis, T.A., 1995. Inferences on the association parameter in copula models for bivariate survival data. Biometrics 51, 1384–1399.

Sklar, A., 1959. Fonctions de répartition à n dimensions et leurs marges. Publ. Inst. Stat. Univ. Paris.

Trivedi, P.K., Zimmer, D.M., 2007. Copula Modeling: An Introduction for Practitioners. Now Publishers, Hanover, Mass.

Yin, W., 2010. A parametric max-flow code for total variation and non-local total variation minimization. `http://www.caam.rice.edu/~wy1/ParaMaxFlow/`.