Using copulas to model time dependence in stochastic frontier models

Christine Amsler — Michigan State University
Artem Prokhorov — Concordia University
Peter Schmidt — Michigan State University

November 2008

Abstract

Current ways of modeling time dependence in stochastic frontier panel data models are unduly restrictive or computationally intractable. They are based on a restrictive assumption on the nature of dependence such as the "scaling" property or involve T-dimensional integration, where T is the number of cross-sections in the panel. This paper demonstrates the use of copulas as a simple means of accounting for time dependence in panel frontier models. The range of dependence it allows to model is unrestricted and the computational task it involves is easy. We also point out to improved asymptotic efficiency of copula-based estimators and consider some of their finite sample properties.

JEL Classification: C13

Keywords: Copula, Frontier Models, MLE, Productivity Analysis

1 Introduction

In this paper we consider the traditional stochastic frontier model proposed by Aigner et al. (1977) and Meeusen and van Den Broeck (1977):

$$y_{it} = x'_{it}\beta + v_{it} - u_{it}, \tag{1}$$

where $u_{it} \geq 0, t = 1, ..., T$. Here i denotes individual firms, countries, production units, etc, and t denotes time. We assume that v_{it}'s are independent over i and t and distributed as $N(0, \sigma_v^2)$. We also assume that the distribution of u_{it} is $N(\mu_{it}, \sigma_{it}^2)^+$, i.e. it is obtained by a truncation to the left of zero of the normal distribution with mean μ_{it} and variance σ_{it}^2. The distribution parameters $(\mu_{it}, \sigma_{it}^2)$ may be constant or they may depend on explanatory variables z_{it}. The truncated error term u_{it} represents inefficiency. Typically we assume it is independent over i but independence over t is not an attractive assumption. One would expect

that inefficiency correlates positively over time – firms that are relatively inefficient in one time period will probably also be inefficient in other time periods.

In this setting, let f_{it} denote the marginal (single t) density for the composed error $\varepsilon_{it} = v_{it} - u_{it}$. We can write it as follows:

$$f_{it} = f_{it}(y_{it} - x'_{it}\beta). \tag{2}$$

Forming the likelihood for all observations requires the joint density of $(\varepsilon_{i1}, ..., \varepsilon_{iT})$. There are four major approaches to this in current literature.

First, one may assume independence (see, e.g., Battese and Coelli, 1995). Then the log-likelihood can be written as follows:

$$\ln L = \sum_i \sum_t \ln f_{it} \tag{3}$$

The quasi maximum likelihood estimator (QMLE) maximizes this function. A variation of the same approach is to assume that u_{it}'s are independent conditional on z_{it} (see, e.g., Wang, 2002).

Second, one may assume that u_{it} is time invariant (see, e.g., Pitt and Lee, 1981). Then,

$$u_{it} = u_i \text{ for all } t$$

Clearly, this is a special case of the time independence assumption above.

Third, one may assume a multivariate truncated distribution for $(u_{i1}, \ldots, u_{iT})$, e.g. multivariate truncated Normal (see, e.g., Pitt and Lee, 1981). Then, the MLE involves a hardly tractable optimization of a likelihood function with T-dimensional integrals.

Finally, one may assume "scaling". The scaling property assumes that u_{it} has the following multiplicative representation:

$$u_{it} = h(z_{it}; \delta)u^*_{it},$$

where u^*_{it} is a random variable, $h(z_{it}; \delta)$ is a known function, δ is a vector of unknown parameters and z_{it} contains variables that affect inefficiency.

There are many versions of the stochastic frontier model that satisfy the scaling property (see Alvarez et al., 2006, for a survey of such models). In some versions, u^*_{it} is assumed to be time invariant and z_{it} includes a time trend (see, e.g., Battese and Coelli, 1992; Kumbhakar, 1990). Other models assume u_{it} is distributed as $N(0, \sigma^2_{it})^+$, where σ^2_{it} depends on a set of relevant variables z_{it} (Caudill et al., 1995, e.g.,). But the key feature of the assumption is that u^*_{it} does not depend on z_{it}, and so changes in z_{it} only affect the scale of the distribution of u_{it} (through the nonrandom function $h(z_{it}, \delta)$) but not its shape (determined by the distribution of u^*_{it}).

The variables z_{it} may include functions of inputs x_{it} or various characteristics of the environment in which the firm operates. The assumption that firms with different z_{it} do not differ in the shape of the distribution

of inefficiency (differ only in the scale) or that there is no correlation in inefficiencies over time are hardly realistic. In this paper we want to develop a tractable method of allowing for dependence without scaling.

We propose using copulas to do that. A distinctive feature of copulas is that they allow to model marginal distributions separately from their dependence structure. As a result we have a flexible joint distribution function, whose marginals are specified by the researcher. The joint distribution can accommodate any degree of dependence. No simplifying assumptions such as scaling or time-invariance are required. Meanwhile, the task of maximizing the joint likelihood remains simple because it involves no T-dimensional integration. Some computational burden remains – one needs to evaluate the composed error marginal CDFs – but this is an easy task in one-dimensional numeral integration or a straightforward simulation.

Advantages of the proposed approach are not only computational. By assuming a copula we assume a joint distribution. This is important for estimation precision. If the joint distribution is correctly specified we have the efficient score function. It is well known that the maximum likelihood estimator that solves the efficient score equation is asymptotically efficient in a class of regular estimators. This leads to an asymptotic efficiency bound – the maximum precision level at which the parameters of the frontier model can be estimated as the size of the cross section grows. But this also suggests that one may do better than QMLE in terms of precision. By specifying an arbitrary, possibly incorrect, copula function, we make use of some dependence information and so we can approach (though not reach) the efficiency bound.

We study the efficiency issues in detail in a different paper (see Prokhorov and Schmidt, 2008). Here we only discuss how the main results of that work apply to stochastic frontier models. We use the setting of the Generalized Method of Moments (GMM) to point out that copulas contain information that is not in general redundant. So besides the computational advantages, copulas offer improvement in precision of estimation.

The plan of the paper is as follows. Section 2 sets the stage for the efficiency discussion by writing the traditional QMLE as a GMM estimator and reviewing its properties. In Section 3, we describe how to use copulas to allow for dependence over time. We use the GMM to motivate the two estimators that arise in this setting. Section 4 addresses the computational issues associated with maximum likelihood estimation of the stochastic frontier model, in which time dependence is not restricted. Section 5 presents simulation results. Section 6 concludes.

2 Quasi-MLE and score-based GMM

Given the densities of v and u, one can derive the density of the composed error $\varepsilon = v - u$ by convolution as follows:

$$f(\epsilon) = \int_0^\infty f_v(\epsilon + u) f_u(u) du, \tag{4}$$

where f_v and f_u are the densities of v and u, respectively, and where we have omitted the it subscript for simplicity.

This density is used in constructing the likelihood for the QMLE. For example, when $v \sim N(0, \sigma_v^2)$ and $u \sim N(0, \sigma_u^2)^+$, this integral has a known form:

$$f(\epsilon) = \frac{2}{\sigma}\phi\left(\frac{\epsilon}{\sigma}\right)\left[1 - \Phi\left(\frac{\epsilon\lambda}{\sigma}\right)\right], \tag{5}$$

where $\sigma^2 = \sigma_v^2 + \sigma_u^2, \lambda = \frac{\sigma_u}{\sigma_v}$, and ϕ and Φ are standard normal density and distribution functions, respectively (see, e.g., Aigner et al., 1977). To obtain the QMLE, one would form the log-likelihood by summing the log-likelihood contributions over all over i and t as in (3).

It is well known that, under regularity conditions, the QMLE is consistent so long as we have a correctly specified marginal density f_{it} even if there is no independence over t. In case there is in fact no independence the QMLE standard errors need to be adjusted to account for that (see, e.g., Hayashi, 2000, Section 8.7). However, the QMLE is generally inefficient. There is a more efficient estimator, which uses the same information as the QMLE.

To motivate this improved estimator, we rewrite the QMLE as a GMM estimator based on the score function. This well known representation uses the asymptotic equivalence of the first order condition that the QMLE solves and the corresponding moment condition (see, e.g., Godambe, 1960). Let s_{it} denote the score of the density function f_{it}, i.e.

$$s_{it} = \nabla_\theta \ln f_{it} \tag{6}$$

Note that s_{it} is a function of the model parameters. We will denote the vector of these parameters by θ, so, when $v \sim N(0, \sigma_v^2)$ and $u \sim N(0, \sigma_u^2)^+$, θ contains β, σ_u^2, and σ_v^2. Then, the QMLE first order condition is

$$\sum_i \sum_t s_{it} = 0 \tag{7}$$

and we can view the QMLE as a GMM estimator based on the moment condition:

$$\mathbf{E}s_i = 0, \text{ where } s_i = \sum_t s_{it}. \tag{8}$$

The key to improving efficiency of the QMLE is to notice that the equivalent GMM estimator uses a suboptimal weighting matrix. Indeed, the moment condition in (8) involves a summation of score functions over t. The theory of GMM suggests that we can do better. To be more precise, define the vector of T score functions for each individual i:

$$s_i^* = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{iT} \end{bmatrix}.$$

Note that each element of this vector has zero mean, because we are using a correctly specified density function f_{it}. Thus,

$$\mathbf{E}s_i^*(\theta) = 0 \tag{9}$$

and the GMM can be based on the stacked score functions.

We call the GMM estimator based on (9) the improved QMLE (IQMLE). The IQMLE is consistent as long as the QMLE is. For both estimators, consistency involves a correct specification of the density function f_{it}. However, the IQMLE is in general more efficient than QMLE. Stacking the score functions, rather than summing them, permits to account for correlation between them and results in a more precise estimator. In a different paper, we prove this statement formally and consider several special cases when QMLE and IQMLE are equally efficient (see Prokhorov and Schmidt, 2008). It is not surprising that when there is in fact independence over t both QMLE and IQMLE are efficient. However, there are other correlation patterns when it is impossible to improve on the traditional QMLE.

3 Copulas and pseudo MLE

Using the improved quasi likelihood is a way to get a more efficient estimator of stochastic frontier models. However, this is not a way to model time dependence. The GMM estimation based on the stacked score functions allows for correlation over time but does not allow to model it explicitly. Explicit modeling of time dependence requires a joint density of $(\varepsilon_{i1}, \ldots, \varepsilon_{iT})$. Specifically, we want that $(\varepsilon_{i1}, ..., \varepsilon_{iT})$ have marginal cdf's $F_{i1}, \ldots, F_{iT}$, respectively, and a joint distribution that allows arbitrary dependence. Copulas can do that.

Briefly defined, a copula is a multivariate distribution function with uniform marginals (a rigorous treatment of copulas can be found in Nelsen, 2006). Specifically, if we let $C(w_1, \ldots, w_T)$ denote the copula cdf and $c(w_1, \ldots, w_T)$ denote the copula density, each $w_t, t = 1, \ldots, T$, is distributed as uniform on $[0, 1]$. Copulas usually contain at least one parameter that models dependence between w's. We give several examples of copulas in the Appendix.

An important feature of copula functions is that they differ in the range of dependence they can cover. Some copulas can cover the entire range of dependence – they are called comprehensive copulas – while others can only accommodate a certain range of dependence. Suppose $T = 2$ and we measure dependence by Pearson's correlation coefficient. Then the Frank and Plackett copulas are comprehensive, the FGM can model correlations only between about -0.3 and +0.3. This will be important in empirical applications because the copula-based likelihood we construct should be able to capture the degree of dependence contained in the data. Clearly, we would like to use comprehensive copulas since we are looking to model arbitrary degree of dependence.

By a celebrated theorem due to Sklar (1959), if we are given the marginals and a continuous joint distribution $H(\varepsilon_1, \ldots, \varepsilon_T)$, there exists a unique copula such that

$$H(\varepsilon_1, \ldots, \varepsilon_T) = C(F_1(\varepsilon_1), \ldots, F_T(\varepsilon_T)) \text{ or}$$

$$h(\varepsilon_1, \ldots, \varepsilon_T) = c(F_1(\varepsilon_1), \ldots, F_T(\varepsilon_T)) \ \cdot f_1(\varepsilon_1) \cdot \ldots \cdot f_T(\varepsilon_T).$$

We now have a flexible joint distribution function, whose marginals we can specify in advance. We can use it to form a joint log-likelihood. The log-likelihood will have the following form:

$$\sum_i (\ln c(F_{i1}, \ldots, F_{iT}) + \ln f_{i1} + \ldots + \ln f_{iT}). \tag{10}$$

Here the first term in the summation is the copula contribution to the likelihood. It reflects time dependence between the cross sections and allows to model it separately from the cross sectional distributions. As before, the marginal densities f_{it} are functions of the parameter vector θ, but the copula term contains both θ and the copula dependence parameter – we denote it by ρ.

We call the estimator of θ and ρ that maximizes (10) the Pseudo MLE. The prefix "pseudo" reflects the fact that we do not know what the true joint distribution $(\varepsilon_{i1}, ..., \varepsilon_{iT})$ is and use a possibly incorrect copula to form it. Of course if the copula produces the true joint distribution, the PMLE is just the full MLE, and the usual results on consistency and asymptotic efficiency of the MLE applies. However, this is not the case in our setting and the question of relative efficiency of copula-based ML estimators is legitimate. Specifically, we want to know if there is another estimator that is more efficient that the PMLE.

The GMM representation of MLE, which we used in previous section, will be useful here. The PMLE solves the score equation that corresponds to the likelihood function in (10). This can be viewed as the GMM estimator based on the moment conditions

$$\mathbf{E}\left[\begin{array}{c} \nabla_\theta \ln c_i + \nabla_\theta \ln f_{i1} + \ldots + \nabla_\theta \ln f_{iT} \\ \nabla_\rho \ln c_i \end{array}\right] = 0, \tag{11}$$

where $c_i = c(F_{i1}, \ldots, F_{iT})$. Again, the key to developing an improved estimator is the observation that the moment conditions in (11) are obtained by applying a specific (not necessarily optimal) weighting scheme to an extended set of valid moment conditions.

The alternative estimator is based on the moment conditions

$$\mathbf{E}\left[\begin{array}{c} \nabla_\theta \ln f_{i1} \\ \cdots \\ \nabla_\theta \ln f_{iT} \\ \nabla_\theta \ln c_i \\ \nabla_\rho \ln c_i \end{array}\right] = 0. \tag{12}$$

This estimator will be consistent so long as we have the correctly speficied marginal densities f_{it} and the PMLE is consistent. However, it is generally more efficient (see Prokhorov and Schmidt, 2008, for the proof). We call it the improved PMLE (IPMLE).

Both PMLE and IPMLE allow to model arbitrary time dependence in panel stochastic frontier models while offering important computational advantages. We discuss these advantages in the next section.

4 Evaluation of integrals

Theoretically, there are two ways of using copula in stochastic frontier models. First, one may define a joint distribution for $(u_1, \ldots, u_T)$ using a copula and then obtain the joint distribution of $(\varepsilon_1, \ldots, \varepsilon_T)$ from it by integration. Second, one may use a copula to define a joint distribution for $(\varepsilon_1, \ldots, \varepsilon_T)$ directly using the marginal distribution of $\varepsilon_t = v_t - u_t$, for all t. In the discussion above we have followed possibility two. The reason for doing so is mainly computational: possibility one involves evaluation of a T-dimensional integral over $(u_1, \ldots, u_T)$ while possibility two involves only one-dimensional integration over u_t. Moreover, for commonly used one-sided distributions of u_t such as half-normal or exponential the density of ε_t is already known (see, e.g., Aigner et al., 1977). So possibility two is more practical.

Another method of modeling time dependence in stochastic frontier models is to adopt a multivariate truncated distribution. This method allows for arbitrary dependence just like our copula approach but imposes a critical computational task. Consider the multivariate truncated normal distribution used by Pitt and Lee (1981). Let u_i denote the vector of one-sided errors $(u_{i1}, \ldots, u_{iT})$ and let $h(u_i)$ denote its density function. The one-sided error vector is distributed as multivariate truncated normal with the parameter matrix Σ if

$$h(u_i) = (2\pi)^{-T/2}|\Sigma|^{-1/2} \exp\left\{\frac{1}{2}u_i'\Sigma^{-1}u_i\right\}/P_0, \tag{13}$$

where P_0 is the probability that $u_i \geq 0$. The denominator term P_0 is the following T-dimensional integral, which is a function of Σ:

$$P_0 = \int_0^\infty \cdots \int_0^\infty (2\pi)^{-T/2}|\Sigma|^{-1/2} \exp\{\frac{1}{2}u_i'\Sigma^{-1}u_i\}du_i, \tag{14}$$

In order to form the likelihood we need the joint density of $\varepsilon_i = (\varepsilon_{i1}, \ldots, \varepsilon_{iT})$. As before, assume that $v_{it} \sim N(o, \sigma_v^2)$ and that u_{it} and v_{it} are independent of each other. Then, the joint density can be obtained as another T-dimensional integral over u_i:

$$h(\varepsilon_i) = \int_0^\infty \cdots \int_0^\infty \Pi_{t=1}^T (2\pi\sigma_v^2)^{-1/2} \exp\left\{\frac{1}{2\sigma_v^2}(\varepsilon_{it} + u_{it})^2\right\} h(u_i)\, du_{i1} \ldots du_{iT}. \tag{15}$$

Note that inside this integral there is the joint density $h(u_i)$, which is itself calculated using a T-dimensional integral. The resulting likelihood is very difficult though not impossible to evaluate.

The copula approach avoids T-dimensional integration altogether. However, it is not entirely free of computational issues. Specifically, in order to obtain the PMLE using (10) or IPMLE using (12) we need to evaluate the integral

$$F(\varepsilon) = \int_{-\infty}^{\varepsilon} f(\epsilon) d\epsilon$$

for a given value of the parameter vector θ – these probabilities are used as the arguments of the copula density function. However, this computational task in one-dimensional integration is much easier.

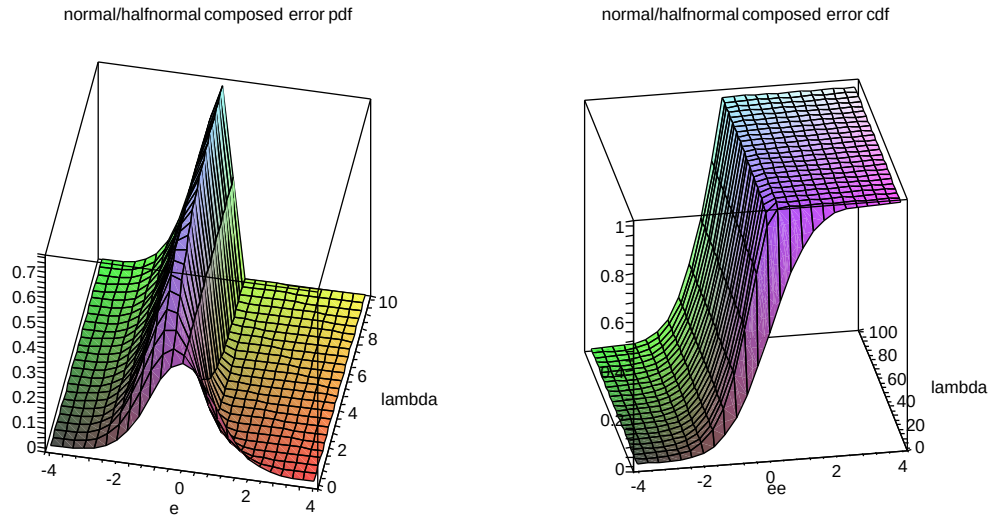

Figure 1: Density and cdf of the composed error ϵ with $\sigma = 1$

In the simulations that follow we use the setting in which v is normal and u is truncated normal – the relevant composed error density is given in (5). It is worth discussing the details of the integral computation in this setting. First note that $F_\sigma(\varepsilon) = F_1(\varepsilon/\sigma)$, where the subscript denotes the value of σ used in F. So we can express quantiles of $F(\varepsilon)$ for any σ as σ times the corresponding quantile for $\sigma = 1$. We therefore only need to evaluate the integral for $\sigma = 1$. Wang et al. (2008) tabulate some quantiles of this distribution using an alternative parametrization, in which σ is replaced with $\sigma_v\sqrt{1+\lambda^2}$. They obtain their quantiles by simulation. We chose the parametrization (σ, λ) rather than (σ_v, λ) because it bounds the variance of each error component by the value of σ^2. This is convenient because given σ the quantiles do not become large negative numbers as λ increases (the cdf does not flatten out). As can be seen from Figure 1, our cdf is effectively in the $[0, 1]$ range on the domain $[-4, 4]$ for all λ. We also use numerical integration rather than simulation. Our experience suggests that numeral integration is substantially faster than simulation for the same level of precision. We give our quantiles in Table 1 (a GAUSS code, which can be used to obtain any number of quantiles at desired precision, is available from the authors).

5 Monte Carlo

In this section we study the finite-sample behavior of the score-based estimators discussed in previous sections. So we conduct a simple Monte Carlo experiment.

Realizations are drawn from a bivariate distribution obtained using normal/half-normal composed error marginals and the Farlie-Gumber-Morgenstern copula. The true parameter values in both marginals are $\sigma_u = \sigma_v = 1$, $\sigma = \sqrt{2}$, $\lambda = 1$; in copula, ρ is set equal to 0.9. Simulation from $N(0, \sigma_u^2)^+$, is done by repeatedly sampling from $N(0, \sigma_u^2)$ until there are enough nonnegative observations. Simulation from the FGM copula is done using the conditional distribution method (see, e.g. Nelsen, 2006, p. 37):

- start by simulating one cross section of half normal composed errors ε_1
- obtain $u_1 \equiv F(\varepsilon_1)$ numerically as described in previous section
- generate $t \sim U(0, 1)$
- set $u_2 = C_2(u_2|u_1 = t) \equiv \frac{\partial C(u_1,u_2)}{\partial u_1}|_{u_1=t}$
- recover $\varepsilon_2 = F^{-1}(u_2)$

Then, the pairs $(\varepsilon_1, \varepsilon_2)$ are normal/half-normal composed errors with copula $C(u_1, u_2)$.

We then obtain the QMLE and PMLE using various copulas. The estimates are given in Table 2.

Several features of Table 2 are worth discussing. First, there is an upward bias in estimates that seems to vanish only at $N = 10,000$. Second, there is a striking similarity between the QMLE, the FGM-based PMLE and the PMLE's based on the incorrect copulas. All the estimates of the marginal distribution are close to one another and are insignificantly different from their true values even for the smallest sample size. We also note that standard errors of the PMLE are usually smaller (though not much smaller) than those of the QMLE even for the incorrect copulas (particularly for the two larger samples).

6 Concluding Remarks

In this paper we propose a simple way of accounting for dependence between cross sections in panel stochastic frontier models. Compared to available alternatives, this method is simple and flexible. It allows modeling arbitrary dependence and does not involve much computational costs. An interesting extension of this paper would be to compare robustness of such copula-based estimators with that of estimators based on the scaling property. After all both approaches may involve an incorrectly specified joint likelihood function.

References

Aigner, D., C. A. K. Lovell, and P. Schmidt (1977): "Formulation and estimation of stochastic frontier production function models," *Journal of Econometrics*, 6, 21–37.

Alvarez, A., C. Amsler, L. Orea, and P. Schmidt (2006): "Interpreting and Testing the Scaling Property in Models where Inefficiency Depends on Firm Characteristics," *Journal of Productivity Analysis*, 25, 201–212.

Battese, G. and T. Coelli (1995): "A model for technical inefficiency effects in a stochastic frontier production function for panel data," *Empirical Economics*, 20, 325–32.

Battese, G. E. and T. J. Coelli (1992): "Frontier production functions, technical efficiency and panel data: With application to paddy farmers in India," *Journal of Productivity Analysis*, 3.

Caudill, S. B., J. M. Ford, and D. M. Gropper (1995): "Frontier Estimation and Firm-Specific Inefficiency Measures in the Presence of Heteroscedasticity," *Journal of Business and Economic Statistics*, 13, 105–111.

Godambe, V. P. (1960): "An optimum property of regular maximum likelihood estimation," *The Annals of Mathematical Statistics*, 31, 1208–1211.

Hayashi, F. (2000): *Econometrics*, Princeton University Press.

Kumbhakar, S. C. (1990): "Production frontiers, panel data, and time-varying technical inefficiency," *Journal of Econometrics*, 46, 201–211.

Meeusen, W. and J. van Den Broeck (1977): "Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error," *International Economic Review*, 18, 435–444.

Nelsen, R. B. (2006): *An Introduction to Copulas*, vol. 139 of *Springer Series in Statistics*, Springer, 2 ed.

Pitt, M. M. and L.-F. Lee (1981): "The measurement and sources of technical inefficiency in the Indonesian weaving industry," *Journal of Development Economics*, 9, 43–64.

Prokhorov, A. and P. Schmidt (2008): "Robustness, redundancy and validity of copulas in likelihood models," *mimeo, Concordia University.*

Sklar, A. (1959): "Fonctions de répartition à n dimensions et leurs marges," *Publications de l'Institut de Statistique de l'Université de Paris*, 8, 229–231.

Wang, H.-J. (2002): "Heteroscedasticity and Non-Monotonic Efficiency Effects of a Stochastic Frontier Model," *Journal of Productivity Analysis*, 18, 241–253.

Wang, W. S., C. Amsler, and P. Schmidt (2008): "Goodness of fit tests in stochastic frontier models," *mimeo, Michigan State University.*

A Some copula families

- Independence or product copula:

$$C(u_1, \ldots, u_T) = u_1 \times \ldots \times u_T$$

- Normal copula:

$$C(u_1, \ldots, u_T; R) = \Phi_T(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_T); R)$$

  where Φ denotes the normal cdf and R denotes the covariance matrix.

- Gumbel copula:

$$C(u_1, \ldots, u_T; \rho) = \exp\left[-((-\ln u_1)^\rho + \ldots + (-\ln u_T)^\rho)^{1/\rho}\right], \quad \rho \in [1, \infty)$$

- Clayton copula:

$$C(u_1, \ldots, u_T; \rho) = \max\left[(u_1^{-\rho} + \ldots + u_T^{-\rho})^{-1/\rho}, 0\right], \quad \rho \in [-1, \infty) \ \{0\}$$

- Farlie-Gumbel-Morganstern (FGM) copula:

$$C(u_1, \ldots, u_T; \rho) = \prod_{t=1}^{T} u_t (1 + \sum_{t=2}^{T} \sum_{1 \le j_1 < \ldots < j_k \le T} \rho_{j_1 \ldots j_k} (1 - u_{j_1}) \ldots (1 - u_{j_k}))$$

  where $\rho_j \in [-1, 1]$

- General copula by inversion

  · start with cdf's $K(x_1, \ldots, x_T)$, $u_1 = F_1(x_1)$,...,$u_T = F_T(x_T)$

  · obtain $x_1 = F_1^{-1}(u_1)$,...,$x_T = F_T^{-1}(u_T)$ and

$$C(u_1, \ldots, u_T) \quad = \quad K(F_1^{-1}(u_1), \ldots, F_T^{-1}(u_T))$$

- Archimedean copulas

  · start with a generator function $\varphi : (0, 1) \to [0, \infty], \varphi' < 0$ and $\varphi'' > 0$

  · obtain

$$C(u_1, \ldots, u_T) = \varphi^{-1}(\varphi(u_1) + \ldots + \varphi(u_T))$$

- multivariate copulas can often be obtained recursively from a bivariate copula (not generally true for other families)
- e.g., Gumbel copula is Archimedean with $\varphi(t) = (-\log t)^\rho$, Frank copula – with $\varphi(t) = \log \frac{1-e^{-\rho}}{1-e^{-\rho t}}$

Table 1: Quantiles of normal/half-normal error distribution: $\sigma = 1$

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	-1.2816	-0.8416	-0.524	-0.2536	0	0.2536	0.5248	0.8416	1.2816
0.1	-1.3568	-0.9184	-0.6016	-0.332	-0.0792	0.1736	0.4432	0.76	1.1984
0.2	-1.4224	-0.9872	-0.6744	-0.4064	-0.156	0.0944	0.3616	0.6752	1.1096
0.3	-1.4768	-1.048	-0.7392	-0.4752	-0.228	0.0184	0.2816	0.5904	1.0176
0.4	-1.5208	-1.0992	-0.7952	-0.536	-0.2944	-0.052	0.2064	0.508	0.9264
0.5	-1.556	-1.1416	-0.8432	-0.5896	-0.3528	-0.1168	0.136	0.4304	0.8376
0.6	-1.5816	-1.1752	-0.884	-0.6352	-0.404	-0.1744	0.0712	0.3584	0.7536
0.7	-1.6016	-1.2016	-0.9168	-0.6744	-0.4488	-0.2248	0.0144	0.292	0.6752
0.8	-1.6152	-1.2224	-0.9432	-0.7064	-0.4864	-0.268	-0.0368	0.2328	0.6032
0.9	-1.6248	-1.2384	-0.964	-0.7328	-0.5184	-0.3056	-0.0808	0.18	0.5376
1.0	-1.632	-1.2504	-0.9808	-0.7536	-0.5448	-0.3384	-0.12	0.1328	0.4784
1.1	-1.6368	-1.2592	-0.9936	-0.7712	-0.5672	-0.3656	-0.1536	0.0912	0.4248
1.2	-1.6392	-1.2656	-1.004	-0.7856	-0.5856	-0.3888	-0.1824	0.0544	0.3776
1.3	-1.6416	-1.2704	-1.012	-0.7968	-0.6008	-0.4088	-0.208	0.0224	0.3344
1.4	-1.6424	-1.2736	-1.0176	-0.8064	-0.6136	-0.4256	-0.2304	-0.0064	0.296
1.5	-1.6432	-1.276	-1.0224	-0.8136	-0.624	-0.44	-0.2488	-0.0312	0.2616
1.6	-1.644	-1.2776	-1.0264	-0.8192	-0.6328	-0.4528	-0.2656	-0.0536	0.2304
1.7	-1.644	-1.2792	-1.0288	-0.824	-0.64	-0.4632	-0.28	-0.0736	0.2024
1.8	-1.644	-1.28	-1.0304	-0.828	-0.6464	-0.472	-0.2928	-0.0904	0.1776
1.9	-1.6448	-1.28	-1.032	-0.8304	-0.6512	-0.4792	-0.3032	-0.1064	0.1544
2.0	-1.6448	-1.2808	-1.0336	-0.8328	-0.6552	-0.4856	-0.3128	-0.12	0.1344
2.1	-1.6448	-1.2808	-1.0344	-0.8352	-0.6584	-0.4912	-0.3216	-0.1328	0.1152
2.2	-1.6448	-1.2808	-1.0344	-0.8368	-0.6616	-0.496	-0.3288	-0.1432	0.0984
2.3	-1.6448	-1.2808	-1.0352	-0.8376	-0.664	-0.5	-0.3352	-0.1536	0.0832
2.4	-1.6448	-1.2816	-1.0352	-0.8384	-0.6656	-0.504	-0.3408	-0.1624	0.0688
2.5	-1.6448	-1.2816	-1.036	-0.8392	-0.6672	-0.5064	-0.3456	-0.1704	0.056
2.6	-1.6448	-1.2816	-1.036	-0.84	-0.6688	-0.5088	-0.3504	-0.1776	0.0448
2.7	-1.6448	-1.2816	-1.036	-0.84	-0.6696	-0.5112	-0.3544	-0.184	0.0336
2.8	-1.6448	-1.2816	-1.036	-0.8408	-0.6704	-0.5136	-0.3576	-0.1896	0.024
2.9	-1.6448	-1.2816	-1.036	-0.8408	-0.6712	-0.5152	-0.3608	-0.1952	0.0144
3.0	-1.6448	-1.2816	-1.036	-0.8408	-0.672	-0.516	-0.3632	-0.2	0.0064
3.1	-1.6448	-1.2816	-1.036	-0.8408	-0.672	-0.5176	-0.3656	-0.204	-0.0016
3.2	-1.6448	-1.2816	-1.036	-0.8408	-0.6728	-0.5184	-0.368	-0.208	-0.0088
3.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6728	-0.5192	-0.3696	-0.212	-0.0152
3.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6736	-0.52	-0.3712	-0.2152	-0.0216
3.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6736	-0.5208	-0.3728	-0.2184	-0.0272
3.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6736	-0.5216	-0.3744	-0.2208	-0.0328
3.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6736	-0.5216	-0.3752	-0.2232	-0.0376
3.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6736	-0.5224	-0.3768	-0.2256	-0.0424
3.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5224	-0.3776	-0.228	-0.0472

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
4.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5224	-0.3784	-0.2296	-0.0512
4.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5232	-0.3792	-0.232	-0.0544
4.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5232	-0.38	-0.2336	-0.0584
4.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5232	-0.3808	-0.2352	-0.0616
4.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.5232	-0.3808	-0.236	-0.0648
4.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3816	-0.2376	-0.068
4.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3816	-0.2384	-0.0704
4.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3824	-0.24	-0.0736
4.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3824	-0.2408	-0.076
4.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3832	-0.2416	-0.0784
5.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3832	-0.2424	-0.08
5.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3832	-0.2432	-0.0824
5.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.244	-0.0848
5.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.2448	-0.0864
5.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.2456	-0.088
5.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.2464	-0.0896
5.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.2464	-0.0912
5.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.384	-0.2472	-0.0928
5.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2472	-0.0944
5.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.248	-0.096
6.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.248	-0.0968
6.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2488	-0.0984
6.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2488	-0.0992
6.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2496	-0.1008
6.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2496	-0.1016
6.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2496	-0.1024
6.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2504	-0.1032
6.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2504	-0.1048
6.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2504	-0.1056
6.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2512	-0.1064
7.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2512	-0.1072
7.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2512	-0.108
7.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2512	-0.1088
7.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.2512	-0.1088
7.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.1096
7.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.1104
7.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.1112
7.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.112
7.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.112
7.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.1128
8.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3848	-0.252	-0.1136

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
8.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.252	-0.1136
8.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1144
8.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1144
8.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1152
8.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1152
8.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.116
8.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.116
8.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1168
8.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1168
9.0	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1176
9.1	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1176
9.2	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1176
9.3	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1184
9.4	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1184
9.5	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1192
9.6	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1192
9.7	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1192
9.8	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.1192
9.9	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.12
10	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2528	-0.12
20	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
30	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
40	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
50	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
60	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
70	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
80	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
90	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256
100	-1.6448	-1.2816	-1.036	-0.8416	-0.6744	-0.524	-0.3856	-0.2536	-0.1256

Table 2: QMLE and PMLE with standard errors for following true parameter values: $\mu_1 = \mu_2 = 0$, $\sigma_1 = \sigma_2 = \sqrt{2} = 1.4142$, $\lambda_1 = \lambda_2 = 1$, $\rho_{FGM} = 0.9$

QMLE	$N = 100$	$N = 1,000$	$N = 10,000$
μ_1	0.2487(0.3981)	0.0901(0.1336)	-0.0607(0.0628)
σ_1	1.5946(0.2558)	1.4668(0.0852)	1.3640(0.0347)
λ_1	1.2972(0.7479)	1.1717(0.2435)	0.9064(0.0989)
μ_2	0.0588(0.6429)	0.1014(0.1368)	0.0263(0.0493)
σ_2	1.3835(0.3488)	1.4561(0.0912)	1.4393(0.0303)
λ_2	0.9780(1.0058)	1.1481(0.2428)	1.0540(0.0832)
FGM	$N = 100$	$N = 1,000$	$N = 10,000$
μ_1	0.2878(0.3409)	0.1089(0.1241)	-0.0429(0.0551)
σ_1	1.6021(0.2306)	1.4775(0.0806)	1.3724(0.0311)
λ_1	1.3520(0.6732)	1.2011(0.2312)	0.9321(0.0883)
μ_2	0.1732(0.6577)	0.1081(0.1343)	0.0197(0.0541)
σ_2	1.4448(0.3822)	1.4578(0.0904)	1.4340(0.0328)
λ_2	1.1452(1.1484)	1.1538(0.2393)	1.0398(0.0901)
ρ	0.8043(0.2898)	0.8458(0.0771)	0.9300(0.0223)
Plackett	$N = 100$	$N = 1000$	$N = 10000$
μ_1	0.2717(0.3411)	0.1097(0.1286)	-0.0381(0.0580)
σ_1	1.5977(0.2318)	1.4783(0.0832)	1.3748(0.0329)
λ_1	1.3538(0.6772)	1.2031(0.2395)	0.9407(0.0933)
μ_2	0.1304(0.7168)	0.1050(0.1355)	0.0221(0.0477)
σ_2	1.4218(0.4101)	1.4557(0.0908)	1.4362(0.0294)
λ_2	1.0866(1.2197)	1.1508(0.2415)	1.0480(0.0804)
ρ	2.2164(0.6642)	2.3598(0.2037)	2.4458(0.0646)
Frank	$N = 100$	$N = 1,000$	$N = 10,000$
μ_1	0.2775(0.3399)	0.1068(0.1306)	-0.0339(0.0547)
σ_1	1.6003(0.2314)	1.4766(0.0842)	1.3776(0.0312)
λ_1	1.3569(0.6770)	1.1965(0.2424)	0.9471(0.0884)
μ_2	0.1402(0.6335)	0.1053 (0.1367)	0.0248(0.0464)
σ_2	1.4264(0.3617)	1.4566(0.0915)	1.4379(0.0287)
λ_2	1.0997(1.0752)	1.1490(0.2430)	1.0510(0.0783)
ρ	1.6228(0.6262)	1.8069(0.1941)	1.9112(0.0597)