Summer 2015

Supervised Classification Using Copula and Mixture Copula

Sumen Sen
Old Dominion University

SUPERVISED CLASSIFICATION USING COPULA AND MIXTURE COPULA

by

Sumen Sen
B.Sc. 2006, Calcutta University, India
M.Sc. 2008, IIT Kharagpur, India
M.S. 2011, University of Central Florida, FL

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MATHEMATICS AND STATISTICS

OLD DOMINION UNIVERSITY
August 2015

Approved by:

Norou Diawara (Director)

N. Rao Chaganty (Member)

Michael J. Doviak (Member)

Khan M. Iftekharuddin (Member)

ABSTRACT

SUPERVISED CLASSIFICATION USING COPULA AND MIXTURE COPULA

Sumen Sen
Old Dominion University, 2015
Director: Dr. Norou Diawara

Statistical classification is a field of study that has developed significantly after 1960's. This research has a vast area of applications. For example, pattern recognition has been proposed for automatic character recognition, medical diagnostic and most recently in data mining. Classical discrimination rule assumes normality. However in many situations, this assumption is often questionable. In fact for some data, the pattern vector is a mixture of discrete and continuous random variables. In this dissertation, we use copula densities to model class conditional distributions. Such types of densities are useful when the marginal densities of a pattern vector are not normally distributed. This type of models are also useful for a mixed discrete and continuous feature types. Finite mixture density models are very flexible in building classifier and clustering, and for uncovering hidden structures in the data. We use finite mixture Gaussian copula and copula of the Archimedean family based mixture densities to build classifier. The complexities of the estimation are presented. Under such mixture models, maximum likelihood estimation methods are not suitable and regular expectation maximization algorithm may not converge, and if it does, not efficiently. We propose a new estimation method to evaluate such densities and build the classifier based on finite mixture of copula densities. We develop simulations scenarios to compare the performance of the copula based classifier with classical normal distribution based models, the logistic regression based model and the Independent model. We also apply the techniques to real data, and present the misclassification errors.

I dedicate this dissertation to my parents, Sabir and Anita Sen, my brother, Amit, my wife Sandhani and my son Soham.

ACKNOWLEDGEMENTS

I would like to express my heartfelt and deepest gratitude to my mentor Dr. Norou Diawara for his patient guidance, encouragement and support through out my thesis. This dissertation would not be possible without his constant support. I have been very grateful to get the opportunity to work under his supervision. He cared so much about my work, and who responded to my questions and queries promptly. I do not have enough words to express my deep and sincere appreciation.

I would also like to extent my gratitude to Dr. N. Rao Chaganty, Dr. Michael J. Doviak and Dr. Khan M. Iftekharuddin for serving on my dissertation committee. I would specially like to thank Dr. N. Rao Chaganty for all his advising throughout my graduate study and arranging the graduate teaching assistantship. Teaching undergraduate student was a enriching experience for me, it has definitely improved my teaching skills. I would also like to thank the department of mathematics and statistics for providing me the platform.

I also would like to thank Dr. Nak-Kyeong Kim, Dr. Michael J. Doviak, Dr. N. Rao Chaganty and Dr. Norou Diawara for everything I learnt from their class. I would like to take this opportunity to thank Late Dr. Dayanand Naik for his insightful teaching. I learnt a lot from his Linear models class. I personally thank Dr. Edgard Maboudou from UCF for teaching Statistical computing with R.

I am also grateful to Dr.Hideaki Kaneko, Chair of the Department of Mathematics and Statistics, and Dr. Raymond Cheng, the Graduate Program Director, for their kind support. Thanks are due to fiscal manager Ms. Sheila Hegwood, department office manager, Ms. Miriam Venable.

Finally, I would also like to thank my friends Arjun, Amanda, Corinne, Shreya, Sameera and Pooja for their help and support throughout this journey. Last but not the least I would like to thank my parents, my brother and my wife for their immense support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 STATISTICAL CLASSIFICATION

Statistical classification is a field of study that has developed significantly after 1960. Significant findings have been obtained in this area. This method has a vast area of applications in many different regions, for example automatic character recognition, medical diagnostic and most recently in data mining. Using this technology one can build a machine with "brain-like" performance. Where one can train an algorithm using a training data set and then the algorithm will eventually learn and be able to classify the unknown patterns. This type of classification is called Supervised classification. Where as in unsupervised classification you don't have a labeled traning data set, and the algorithm tries to cluster the data into different groups. In recent year, there have been many extensions both in the methodology and application point of views. These developments include kernel-based methods and, Bayesian methods. Suppose there are two or more distinct populations or characteristics associated with data. Discriminant analysis is a search tool used to regulate which population or characteristics an observation come from, underline mechanism is called discriminant rule. Standard classification methods are quadratic discriminant analysis (QDA), linear discriminant analysis (LDA) and regularised discriminant analysis (RDA) (Friedman (1989). These methods are well investigated in the literature such as associated probabilities, adaptability, use of priors. Robustness of the discrimination rule to outliers is discussed by Todorov et al. (1994). Aeberhard et al. (1994) showed that RDA performs better compare to LDA only when the class covariance matrices are identical and if a large training set is available. Alternative approaches to the problem of discriminant analysis with singular covariance matrices are described by Krzanowski et al. (1995). When the mechanism rule is well chosen the misclassification error will be minimised. Extensions of linear and quadratic discriminant analysis to data sets where the patterns are curves or functions are developed by James and Hastie (2001). All above methods assume marginal normally.

However, most of the time the features and characteristics associated are such that data violates normality assumption. There are data sets where all the features are discrete or some of them are continuous and some of them are discrete, that is data is mixed. Analysis must find a way to capture the dependency. Copulas are very useful tools in statistics for modelling dependence and derive multivariate distribution with specific margins (Joe (2014)). More recently copula based models have been used in different areas such as climate (Scholzel (2008)), oceanography (De-Waal et al. (2005)), insurance claims (Claudia et al. (2012)), engineering (Grigoriu (2007)). Using copula, one can construct multivariate density for a mixed data, where the marginal distributions are discrete and continuous or both. Salinas-Gutierrez (2011) used copula model for discriminant analysis assuming all the marginal distributions are continuous. Leon & Wu (2011) used copula based models, under latent variable, for pattern recognition. In this dissertation, we used copula and finite mixture copula models for classification. We also introduced two stage estimation process for mixture copula models and used those models for classification in simulated and real data sets.

1.1.1 STATISTICAL CLASSIFICATION PROBLEM

In this section we describe the basic model structure for the statistical classification. We use the term "pattern" to denote the p-dimensional data vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^t$ of measurements. The components of the pattern vector are measurements of the features of an object. Features are variables specified by the investigator and thought to be important for classification. We also assume there are G groups or classes, denoted by $\{\omega_1, \omega_2, \ldots, \omega_G\}$. There are two types of classification: *Supervised classification* and *Unsupervised classification.* In this dissertation, we concentrate only on supervised classification. Under supervised classification method we assume we have a set of patterns of known class $\{(\boldsymbol{x}_i, z_{ig}), i = 1, 2, \ldots n_g; g = 1, 2 \ldots G\}$, where $\boldsymbol{x}_i \in R^p$ and $z_{ig} = 1$ if the pattern belongs to ω_g and $z_{ig} = 0$ otherwise, for $g = 1, 2, \ldots, G$. This set of data is called design set and will be used to build a classifier.

1.1.2 BAYES' DECISION RULE

Consider G classes, $\omega_1, \omega_2, \ldots, \omega_G$ with known prior probabilities $p(\omega_1), p(\omega_2), \ldots, p(\omega_G)$. If we had no information regarding an pattern $\boldsymbol{x}$ other than the class probability distribution then, in order to minimize the probability of making an error, we would assign $\boldsymbol{x}$ to class ω_g if:

$$p(\omega_g) > p(\omega_k) \text{ for all } g = 1, 2, \ldots, G \text{ and } g \neq k.$$

For classes with equal prior probabilities, patterns are assigned arbitrarily between those classes. However, we do have an observation vector $\boldsymbol{x}$, and a decision rule based on probabilities is to assign $\boldsymbol{x}$ to class ω_g if the probability of class ω_g given the observation $\boldsymbol{x}$, that is $p(\omega_g|\boldsymbol{x})$, is greatest over all G classes. That is, assign $\boldsymbol{x}$ to class ω_g if:

$$p(\omega_g|\boldsymbol{x}) > p(\omega_k|\boldsymbol{x}) \text{ for all } g = 1, 2, \ldots, G \text{ and } g \neq k. \tag{1}$$

Now, using Bayes' theorem one can write:

$$p(\omega_g|\boldsymbol{x}) = \frac{p(\omega_g|\boldsymbol{x})p(\omega_g)}{p(\boldsymbol{x})}. \tag{2}$$

Then using Equation (1) and (1) Bayes' minimum error rule can be written as: assign the unknown pattern vector $\boldsymbol{x}$ to class ω_g if

$$p(\omega_g)p(\boldsymbol{x}|\omega_g) > p(\omega_k)p(\boldsymbol{x}|\omega_k) \text{ for all } g = 1, 2, \ldots, G \text{ and } g \neq k. \tag{3}$$

The probability of making an error, $p(error)$, can be expressed as:

$$p(error) = \sum_{g=1}^{G} p(error|\omega_g)p(\omega_g).$$

In the above expression $p(error|\omega_g)$ $g = 1, 2, \ldots, G$ is the probability of the misclassifying pattern from the class ω_g. The decision rule in Equation (3) minimizes the error, as shown in Webb and Copsey (2011) . Assuming the prior probabilities $p(\omega_g)$ are known, then in order to make a decision we need to estimate the class conditional densities $p(\boldsymbol{x}|\omega_g)$. Estimation of the density is based on a sample of observation $\{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \ldots, \boldsymbol{x}_{n_g}^g\}$ $(\boldsymbol{x}_i^g \in R^p)$ from class ω_g.

1.2 QUADRATIC DISCRIMINANT ANALYSIS:

Quadratic Discriminant analysis (QDA) is one of the most widely used classifier, and it's based on multivariate normal distribution. In this method, for g^{th} class the class conditional density is assumed to be multivariate normal distribution, with mean μ_g and covariance matrix Σ_g. Then the quadratic discriminant rule allocates observations $\boldsymbol{x}$ to class ω_g after minimizing the square Mahalanobis distance between $\boldsymbol{x}$ and the class ω_g. Class conditional density $p(\boldsymbol{x}|\omega_g)$ defined as:

$$p(\boldsymbol{x}|\omega_g) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_g|^{\frac{1}{2}}} exp\left[\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_g)^T\Sigma_g^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_g)\right]. \tag{4}$$

Classification is done by assigning a pattern vector $\boldsymbol{x}$ to class for which the posterior probability $p(\omega_g|\boldsymbol{x})$ or equivalently $log(p(\omega_g|\boldsymbol{x}))$ is the greatest. If the prior is available from each population, we can use that information under Bayes' rule. Using Bayes' rule and the normality assumption in Equation (4) for the conditional density, we can write:

$$\begin{aligned} p(\omega_g|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\omega_g)p(\omega_g)}{p(\boldsymbol{x})} \\ \Rightarrow log(p(\omega_g|\boldsymbol{x})) &= log(p(\boldsymbol{x}|\omega_g)) + log(p(\omega_g)) - log(p(\boldsymbol{x})) \\ &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_g)^T\Sigma_g^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_g) - \frac{1}{2}log(|\Sigma_g|) \\ &\quad -\frac{p}{2}log(2\pi) + log(p(\omega_g)) - log(p(x)). \end{aligned} \tag{5}$$

As $p(\boldsymbol{x})$ does not depend on class ω_g, the discriminant rule is: assign $\boldsymbol{x}$ to ω_g if $\delta_g > \delta_k$ for all $g, k \in \{1, 2, \ldots, G | g \neq k\}$ where δ_g is given by:

$$\delta_g(\boldsymbol{x}) = log(p(\omega_g)) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_g)^T\Sigma_g^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_g) - \frac{1}{2}log(|\Sigma_g|), \tag{6}$$

for $g \in \{1, 2, \ldots, G\}$. Based on a training sample, the quantities μ_g and Σ_g are replaced by their estimates. Classifying a pattern vector based on the values of $\delta_g(\boldsymbol{x})$, $g = 1, 2, \ldots, G$ is called normal-based quadratic discriminant function (McLachlan 1992).

1.3 LINEAR DISCRIMINANT ANALYSIS:

Linear Discriminant Analysis (LDA), also called Fisher's linear discriminant analysis assumes the pattern vector $\boldsymbol{x}$ is normally distributed in each class but unlike

QDA this approach assumes class covariance matrices, that is $\Sigma_1 = \Sigma_2 = \ldots, \Sigma_G = \Sigma$ are same. Then, ignoring the terms that are not changing with class, the discriminant rule in Equation (6) can be written as:

$$\delta_g(\boldsymbol{x}) = log(p(\omega_g)) + \boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}_g - \frac{1}{2}\boldsymbol{\mu}_g^T\Sigma^{-1}\boldsymbol{\mu}_g. \tag{7}$$

The parameters can be estimated based on a sample. To get the estimated discriminant rule, we replace in Equation (7) $\boldsymbol{\mu}_g$ by $\bar{\boldsymbol{x}}^g$, the arithmetic mean of g^{th} class. Correlation matrix Σ can be estimated by

$$\hat{\Sigma} = \frac{1}{N-G}\sum_{g=1}^{G}\sum_{i=1}^{n_g}(\boldsymbol{x}_i^g - \bar{\boldsymbol{x}}^g)(\boldsymbol{x}_i^g - \bar{\boldsymbol{x}}^g)^T, \tag{8}$$

where $\boldsymbol{x}_i^g$, $i = 1, 2, \ldots, n_g$ are observations from g^{th} class, and $N = \sum_{g=1}^{G} n_g$, is the total number of observation in all class. Fahrmeir and Hamerle (1984) discussed briefly about this method.

1.4 REGULARIZED DISCRIMINANT ANALYSIS

Regularized discriminant analysis method was proposed by Friedman (1989) It is applicable when sample size is small and dimension is high. Two parameters are involved in this method: a complexity parameter, denoted by $\lambda \in [0, 1]$, providing an intermediate between a linear and a quadratic discriminant rule; and a shrinkage parameter for covariance matrix updates, denoted by $\gamma \in [0, 1]$. In this method the covariance matrix $\widehat{\Sigma}_g$ for g^{th} class is replaced by a linear combination, Σ_g^λ, of the sample covariance matrix $\widehat{\Sigma}_g$ and the pooled covariance matrix $\widehat{S}_p$ as:

$$\Sigma_g^\lambda = \frac{(1-\lambda)n_g\widehat{\Sigma}_g + \lambda N\widehat{S}_p}{(1-\lambda)n_g + N\lambda}, \tag{9}$$

where $\widehat{S}_p = \sum_{g=1}^{G} \frac{n_g}{N}\widehat{\Sigma}_g$. Second parameter, γ is used to regularized the sample class covariance further beyond that provided by Equation (9).

$$\Sigma_g^{\lambda,\gamma} = (1-\gamma)\Sigma_g^\lambda + \gamma c_g(\lambda)I_p, \tag{10}$$

where I_p is a $p \times p$ identity matrix, and $c_g(\lambda) = trace(\Sigma_g^\lambda)/p$, the average of eigenvalues of Σ_g^λ. Finally the matrix $\Sigma_g^{\lambda,\gamma}$ is used as a plug-in estimate of the covariance matrix in the normal based discriminant rules. This method can improve classification performance when the covariance matrices are not equal or the sample size is too small for quadratic discriminant analysis to be viable.

1.5 PERFORMANCE OF A CLASSIFIER

In the literature, several methods have been proposed to estimate the classification error (Gupta and Dordrect (1987) and Mclachlan (1976)). Three major methods are discussed below:

- **The re-substitution method**: In this method all observations are used to design the classifier and used again to estimate its performance.
- **Hold-out method:** Let the total sample size be $N = \sum_{g=1}^{G} n_g$, then in this method one portion of the set of observations is used to design the classifier, and the remaining $(N - k)$ observations, known as test set, are used to estimate the error rate.
- **Random Sub-sampling**: In this method, k_g samples are chosen randomly from each group ω_g $g = 1, 2, \ldots, G$. Classifier is designed by those k_g samples. Then misclassification error E_i is estimated using remaining sample from each group. This process is repeated k times and true error E is obtained as the average of E_i's.

The re-substitution method can be used only when the sample size is significantly large, shown analytically in Rudys (1978). Hold-out error counting has basic drawbacks. Misclassification error rates highly depend on the split. If we change the split estimated error rate can significantly change. To overcome these problems we used random sub-sampling method to estimate error rates. For small sample size bootstrap methods can be used to estimate error rate.

1.6 OVERVIEW OF THE DISSERTATION

From the previous discussion, we can see that the method of constructing a classifier is not just a density estimation problem but a computational algorithm. A decision rule can be constructed through explicit estimation of the class conditional densities: $p(\boldsymbol{x}|\omega_g)$. In this dissertation, we use copula based model to parameterize the class conditional densities. In Chapter 2 we provide an introduction of copulas and their estimation process. We discussed MLE estimation and Inference Function Method (IFM) estimation process and provide simulation results. In Chapter 3 we introduce finite Gaussian and Archimedean copula mixture distributions and developed an estimation method for such mixture densities. Implementation of the proposed estimation method for several finite copula mixture distributions are discussed using simulated data. In Chapter 4 Gaussian copula based models and finite mixture Gaussian copula models are applied for classification. We provide decision rules and decision boundaries for those copula and mixed copula based classifiers. In this chapter we also discussed other classical methods, IM and LR, for classification. In Chapter 5 we implemented these methods and compared performance (misclassification rate) with the classical normal based methods, independence model and logistic regression. We implement these models in simulated data and real life data and compare misclassification errors. Finally we conclude our study in Chapter 6 and in the Appendix we provide expressions of score functions and **R** code we used to obtain the estimates.

CHAPTER 2

COPULA AND ESTIMATION

2.1 DEFINITION AND EXAMPLES

One of the modern approach to derive a multivariate distribution with specified marginal is through copula. Joe (2014) and Song (2007) has briefly described these methods in their books. Copula is a multivariate distribution with univariate margins that are uniform on the interval [0,1]. The basic idea behind the construction of a multivariate distribution using copula is *probability integral transform.* For a given continuous random variable variable X with a CDF $F(.)$, the transformation $F(X)$ follows a uniform distribution on [0,1], and the definition of copula is give below:

Definition 1. *A p-dimension copula is a function* $C : [0,1]^p \to [0,1]$ *with the following properties.*

1. $C(1,\ldots,u_i,\ldots,1) = u_i \; \forall i = 1,2,\ldots,p$ *and* $u_i \in [0,1]$.

2. $C(u_1,u_2,\ldots,u_p) = 0$ *if at least one* $u_i = 0$ *for* $i = 1,2,\ldots,p$.

3. *For any* $u_{i1}, u_{i2} \in [0,1]$ *with* $u_{i1} \leq u_{i2}$, *for* $i = 1,2,\ldots,p$,

$$\sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\ldots+j_p}C(u_{1j_1},u_{2j_2}\ldots,u_{pj_p}) \geq 0.$$

2.1.1 EXAMPLES OF COPULAS

In this sub-section, we provide some examples of well known copulas.

Example 1. *The* independence copula *is a function given by*

$$C(u_1,u_2,\ldots u_p) = \prod_{i=1}^{p} u_i, \quad u_i \in [0,1]. \tag{11}$$

Example 2. . *The* Comonotonicity Copula *is a function is given by*

$$C(u_1,u_2,\ldots u_p) = min\{u_1,u_2,\ldots,u_p\}, \quad u_i \in [0,1]. \tag{12}$$

Example 3. *The* Clayton Copula *is a function is given by*

$$C(u_1, u_2, \ldots, u_p) = \left(max\{u_1^{-\theta} + u_2^{-\theta} \cdots + u_p^{-\theta} - p + 1, 0\}\right)^{-1/\theta}, \quad \theta \in [-1, \infty) - \{0\}, \quad u_i \in [0,1]. \tag{13}$$

Example 4. *The bivariate* Frank Copula *is a function given by*

$$C(u_1, u_2|\theta) = -\frac{1}{\theta} log\left(1 + \frac{(e^{-u_1\theta} - 1)(e^{-u_2\theta} - 1)}{e^{-\theta} - 1}\right), \quad \theta \in \mathbb{R} - \{0\}. \tag{14}$$

In this dissertation the theory is developed focusing on Gaussian copula. This type of copula is very popular in the literature and the association structure is similar to multivariate normal distribution. When dimension is large, then estimation of the parameter is relatively simple, on the other hand for other type of copula complexity increases as dimension increases.

Example 5. . *The copula associated with standard multivariate gaussian distribution is called Gaussian copula. The* Gaussian Copula *is a function given by*

$$C_\Phi(u_1, u_2, \ldots u_p|R(\boldsymbol{r})) = \Phi_{R(\boldsymbol{r})}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_p)), \tag{15}$$

where $u_1, u_2, \ldots u_p \in \mathbb{R}$ and Φ^{-1} is the inverse cumulative distribution function of a standard normal and $\Phi_{R(\boldsymbol{r})}$ is the joint cumulative distribution function of a standard multivariate normal distribution covariance matrix equal to the correlation matrix $R(\boldsymbol{r})$. Gaussian copula density, c_Φ ,in Equation (16), density is defined as:

$$\begin{aligned} c_\Phi(u_1, u_2, \ldots, u_p|R(\boldsymbol{r})) &= \frac{\partial^p}{\partial u_1 \partial u_2 \ldots \partial u_p} C_\Phi(u_1, u_2, \ldots u_p|R(\boldsymbol{r})) \\ &= \frac{1}{\sqrt{|R(\boldsymbol{r})|}} exp\left(-\frac{1}{2}\boldsymbol{U}^T(R(\boldsymbol{r})^{-1} - I_p)\boldsymbol{U}\right), \end{aligned} \tag{16}$$

where $\boldsymbol{U} = (\Phi^{-1}(u_1)\ \Phi^{-1}(u_2) \ldots \Phi^{-1}(u_p))^T$. Figure 1 shows bivariate Gaussian copula density under different correlation values.

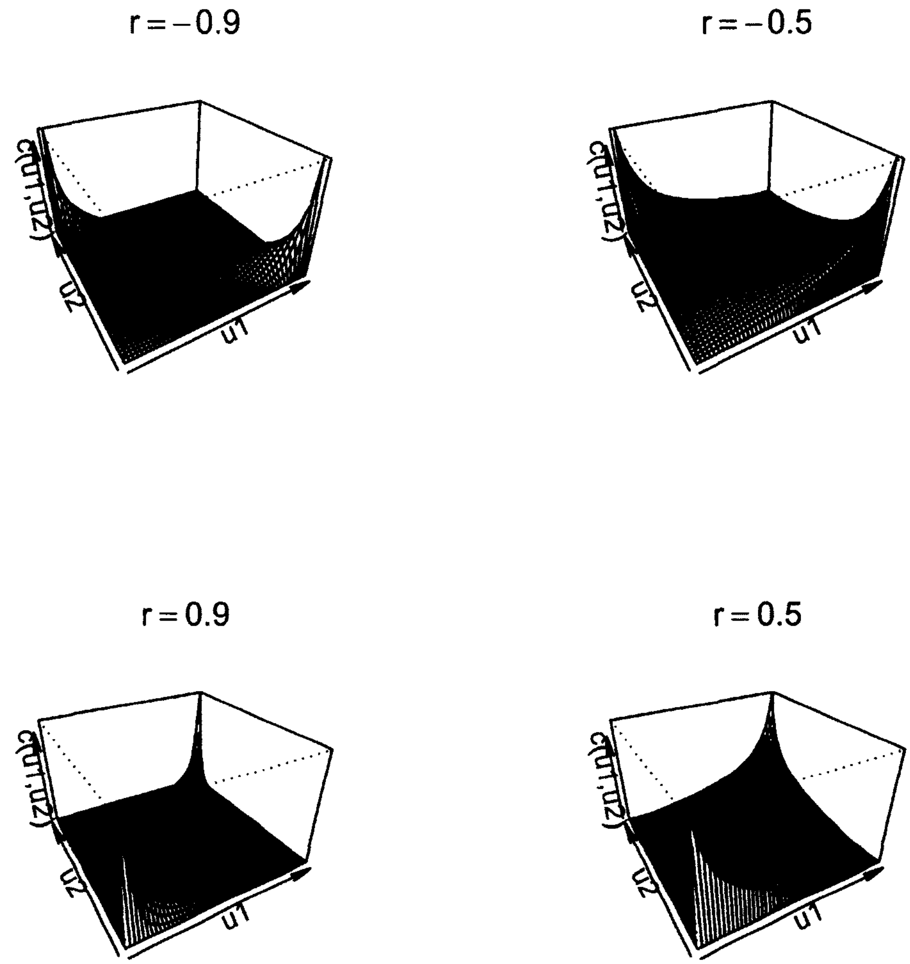


Figure 1. Bivariate Gaussian copula density for different values of r.

.

The most fundamental theorem related to copula is Sklar's theorem, which allows us to glue the known marginal densities through a copula.

Theorem 2.1.1. (*Sklar's Theorem*) *Let $X_1, X_2, \ldots, X_p$ be random variables with marginal distribution functions $F_1, F_2, \ldots, F_p$ and joint cumulative distribution function F. Then the followings hold.*

1. *There exist a p dimensional copula C such that for all $x_1, x_2, \ldots, x_p \in (-\infty, \infty)$*

$$F(x_1, x_2, \ldots, x_p) = C(F_1(x_1), F_2(x_2), \ldots, F_p(x_p)).$$

2. *If $X_1, X_2 \ldots, X_p$ are continuous, the copula C in unique. Otherwise C can be uniquely determined on a n dimensional rectangle $Range(F_1) \times Range(F_2) \times \cdots \times Range(F_p)$.*

More discussion in copulas and their properties is presented in Joe (2014), Nelsen (2006) and Jaworski et al. (2010).

2.2 MAXIMUM LIKELIHOOD AND IFM ESTIMATION OF COPULA

Our goal is to use copula based densities for classification. In order to build classifiers we need to estimate parameters in the copula density, using most efficient and fast method of estimation available in the literature. This section describes briefly two estimation process of copula densities. There are two methods one can estimate the parameters in a copula density. One method maximizes the likelihood function (MLE) to obtain estimates. Another method is Inference Function for Margin (IFM) method; this second method was introduced by Joe (2005), and he also showed this method is as efficient as MLE. For a p dimensional joint density, this method consist of doing p separate optimization of the univariate likelihoods, followed by an optimization of the multivariate likelihood as a function of dependence parameter vector. This method is simple to implement and its converges quickly compared to MLE. We used one additional iteration step to IFM method of estimating parameters. Steps of this estimation method is given bellow:

Step1: Ignoring the dependence parameter R, estimate each marginal parameters $\boldsymbol{\theta}$ by maximizing each univariate likelihoods.

Step2: In the second step use those estimated values of $\boldsymbol{\theta}$ from the previous step and maximize the complete data likelihood to get an estimate of the dependence parameter R.

Step3: Finally using the estimated value of R obtained from the previous step, maximize the complete data likelihood to obtain final estimates of $\boldsymbol{\theta}$.

We will show few examples of MLE and IFM estimation methods for discrete, continuous and mixed joint distribution using Gaussian copula distributions and compare their standard errors.

2.3 JOINT CONTINUOUS DISTRIBUTION USING GAUSSIAN COPULA AND ESTIMATION

According to Theorem 2.1.1, any joint distribution function F with continuous marginal distributions $F_1, F_2, \ldots, F_p$ can be associated to a copula function C_Φ. If we know the marginal distributions, we can derive the joint probability density function using Sklar's Theorem. Let $X_1, X_2 \ldots X_p$ be continuous random variables with probability density functions $f_1, f_2, \ldots, f_p$. Then the joint density f of $(X_1, X_2, \ldots, X_p)$ can be written as

$$f(x_1, x_2 \ldots, x_p) = c_\Phi\left(F_1(x_1), F_2(x_2), \ldots, F_p(x_p)|R(\boldsymbol{r})\right) \prod_{i=1}^{p} f_i(x_i), \tag{17}$$

where $c_\Phi(\boldsymbol{u}) = \frac{\partial C_\Phi(\boldsymbol{u})}{\partial \boldsymbol{u}}$ is the Gaussian copula density function. One advantage of using Gaussian copula is flexible correlation structure. Different correlation structures and their estimation is provided in the next section.

2.3.1 STRUCTURE OF R

If the dimension of the random variable $\boldsymbol{X}$ is p we have $\binom{p}{2}$ elements in the association matrix R. As p increases, the number of parameter increases dramatically. If in a model, there are too many parameters to estimate, estimation can be computationally challenging and also efficiency can be lost. To avoid these, in this dissertation we assume structured R matrix.

1) Equi-correlation Structure:

Under this structure we assume $R(r) = r\mathbf{1}\mathbf{1}^t - (1-r)I_p$, where I_p is a p dimensional identity matrix, $r \in (-\frac{1}{p-1}, 1)$, and $\mathbf{1}$ is p dimensional column vector of ones. It follows from Olkin and Pratt (1958) that:

$$R^{-1}(r) = \frac{1}{1-r} I_p - \frac{r}{(1-r)\{1+(p-1)r\}} \mathbf{1}\mathbf{1}^t. \tag{18}$$

2) AR-1 structure:

Under this structure, the $(i,j)^{th}$ element of $R(r)$ is given by $r^{|i-j|}$, with $r \in (-1, 1)$. The inverse of this matrix is given below (Chaganty 1997)

$$R^{-1}(r) = \frac{1}{1-r^2}(I_p - r^2 M_2 - r M_1), \tag{19}$$

where $M_2 = diag(0, 1, \ldots, 1, 0)$ and M_1 is a tridiagonal matrix with 0 on the main diagonal and 1 on the upper and lower diagonals.

2.3.2 ESTIMATION

Consider a random variable $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ with the marginal pdf and cdfs are $f_j(x_j|\boldsymbol{\theta}_j)$, $F_j(x_j|\boldsymbol{\theta}_j)$ respectively, for $j = 1, 2, \ldots, p$. Then the joint distribution can be written as:

$$\begin{aligned} f(x_1, x_2 \ldots, x_p|\Theta, R(\boldsymbol{r})) &= c_\Phi\left(F_1(x_1|\boldsymbol{\theta}_1), F_2(x_2|\boldsymbol{\theta}_2), \ldots, F_p(x_p|\boldsymbol{\theta}_p)|R(\mathbf{r})\right) \\ &\times \prod_{j=1}^{p} f_i(x_j|\boldsymbol{\theta}_j), \end{aligned} \tag{20}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, ..., .\boldsymbol{\theta}_p)$ is the parameter vector. Then, based on a random sample- of size n, $\boldsymbol{X} = (\boldsymbol{X_1}, \boldsymbol{X_2}, ..., \boldsymbol{X_n})$, the log likelihood function is given by:

$$\begin{aligned} l(\boldsymbol{\Theta}|\boldsymbol{X}) &= \sum_{i=1}^{n} c_\Phi\left(F_1(x_{1i}|\boldsymbol{\theta}_1), F_2(x_{2i}|\boldsymbol{\theta}_2), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_p)|R(\boldsymbol{r})\right) \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{p} log(f_j(x_{ji}|\boldsymbol{\theta_j})) \\ &= \frac{n}{2}log|R(\boldsymbol{r})| - \sum_{i=1}^{n}\frac{1}{2}\boldsymbol{q}_i^t(R(\boldsymbol{r})^{-1} - I_p)\boldsymbol{q}_i + \sum_{i=1}^{n}\sum_{j=1}^{p} log(f_j(x_{ji}|\boldsymbol{\theta_j})), \end{aligned} \tag{21}$$

where $\boldsymbol{q}_i = (q_{1i}, q_{2i}, \ldots, q_{pi})$ with components $q_{ji} = \Phi^{-1}(F_j(x_{ji}))$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$. The score functions for the parameters can be obtained by taking :

$$\frac{\partial l}{\partial \boldsymbol{\theta}_j} = \sum_{i=1}^{n}\sum_{j=1}^{p} \frac{1}{f(x_{ij}|\boldsymbol{\theta}_j)} \frac{\partial f(x_{ij}|\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} - \sum_{i=1}^{n} \frac{\partial \boldsymbol{q}_i^t}{\partial \boldsymbol{\theta}_j}(R(\boldsymbol{r})^{-1} - I_p)\boldsymbol{q}_i, \tag{22}$$

$$\begin{aligned} \frac{\partial l}{\partial \boldsymbol{r}} &= \frac{\partial}{\partial \boldsymbol{r}}\left(-\frac{n}{2}log|R(\boldsymbol{r})| - \frac{1}{2}\sum_{i=1}^{n} trace((R(\boldsymbol{r})^{-1} - I_p)\boldsymbol{q}_i\boldsymbol{q}_i^t)\right) \\ &= \frac{\partial}{\partial \boldsymbol{r}}\left(-\frac{n}{2}log|R(\boldsymbol{r})| - \frac{1}{2}\sum_{i=1}^{n} trace(R(\boldsymbol{r})^{-1}\boldsymbol{q}_i\boldsymbol{q}_i^t - \boldsymbol{q}_i\boldsymbol{q}_i^t)\right) \\ &= \frac{\partial}{\partial \boldsymbol{r}}\left(-\frac{n}{2}log|R(\boldsymbol{r})| - \frac{n}{2}trace(R(\boldsymbol{r})^{-1}\sum_{i=1}^{n}\boldsymbol{q}_i\boldsymbol{q}_i^t)\right). \end{aligned} \tag{23}$$

As the score functions are highly nonlinear, closed form expression of MLE estimates are not available. We use numerical methods to solve those score equations and obtain MLE estimates of the parameter. A popular choice numerical algorithm would be Quasi-Newton method given in Nash (1979). The algorithm can be describe as follows:

Step1. Start with an initial estimate $\hat{\Theta}_i$ of Θ

Step2. $(k+1)^{th}$ step iteration proceeds as:

$$\hat{\Theta}^{k+1} = \hat{\Theta}^{k} - \delta B(\hat{\Theta}^{k})\dot{l}(\hat{\Theta}^{k})$$

where $B(\Theta)$ is an approximation to the inverse of the Hessian matrix, $\dot{l}(\Theta) = \frac{\partial l(\Theta)}{\partial \Theta}$ and δ is a constant.

Step3. Repeat Step 2 until $\hat{\Theta}^{k+1} \cong \hat{\Theta}^{k}$ and take $\hat{\Theta} = \hat{\Theta}^{k+1}$ as the MLE.

2.3.3 EXAMPLE

We use Sklar's theorem to construct bivariate gamma density using a Gaussian copula. We consider two random variables X_1 and X_2 with the density:

$$f_i(x_i|\alpha_i, \beta_i) = \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i - 1} e^{-\beta_i x_i} \quad \text{for } i = 1, 2, \tag{24}$$

with $x_i \in (0, \infty)$, $\alpha_i > 0$, $\beta_i > 0$. Using Equation (20) we can construct the bivariate gamma distribution. The joint density of X_1 and X_2 is given below:

$$\begin{aligned} f(x_1, x_2|\alpha_1, \beta_1, \alpha_2, \beta_2, R(r)) &= c_\Phi(F_1(x_1|\alpha_1, \beta_1), F_2(x_2|\alpha_2, \beta_2)|R(r)) \\ &\times f_1(x_1|\alpha_1, \beta_1) f_2(x_2|\alpha_2, \beta_2), \end{aligned} \tag{25}$$

where $F_i(.)$ is the CDF of gamma density, and c_Φ, is the bivariate Gaussian copula density, defined as:

$$c_\Phi(u_1, u_2) = \frac{1}{\sqrt{|R|}} exp\left(-\frac{1}{2} \begin{pmatrix} \Phi^{-1}(u_1) \\ \Phi^{-1}(u_2) \end{pmatrix}^T (R^{-1} - I_2) \begin{pmatrix} \Phi^{-1}(u_1) \\ \Phi^{-1}(u_2) \end{pmatrix} \right), \tag{26}$$

where $u_1, u_2 \in [0, 1]$, Φ is the standard normal distribution function, I_2 is a 2×2 identity matrix and $R(r) = \left(\begin{smallmatrix} 1 & r \\ r & 1 \end{smallmatrix}\right)$ is a 2×2 symmetric matrix, called *association matrix* and for continuous marginal distribution r is the Pearson correlation between two normal scores. That is r can be expressed as:

$$r = cor[\Phi^{-1}\{F_1(X_1)\}, \Phi^{-1}\{F_2(X_2)\}].$$

Using Equations (26) and (25) the joint density function become:

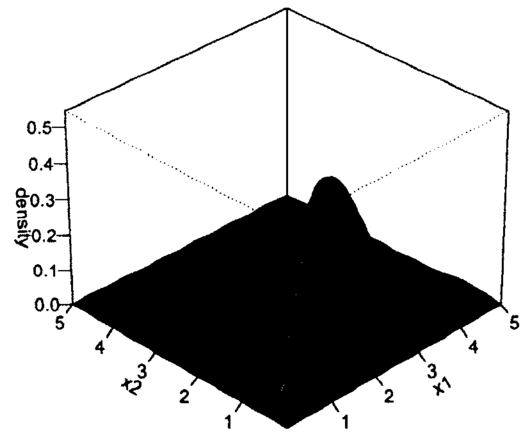
$$\begin{aligned} f(x_1, x_2|\alpha_1, \beta_1, \alpha_2, \beta_2, r) &= \frac{1}{\sqrt{1-r^2}} exp\left(\frac{-1}{2\sqrt{1-r^2}}(q_1^2+q_2^2-2q_1q_2r)\right. \\ &+ \left.\frac{1}{2}(q_1^2+q_2^2)\right) \times f_1(x_1|\alpha_1,\beta_1) f_2(x_2|\alpha_2,\beta_2), \end{aligned} \tag{27}$$

where $q_1 = \Phi^{-1}\{F_1(x_1|\alpha_1, \beta_1)\}$ and $q_2 = \Phi^{-1}\{F_2(x_2|\alpha_2, \beta_2)\}$. Figure 2 shows the plot of the joint distribution of (X_1, X_2) defined in Equation (25).

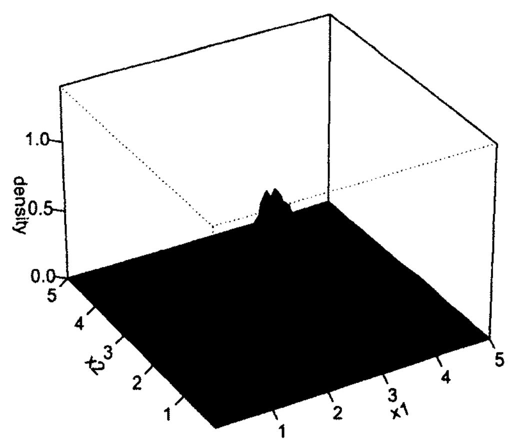
Using the method described in Nelsen 2006 we can simulate random number from the joint pdf in Equation (25). Based on a random sample $\boldsymbol{X}_i = (x_{1i}, x_{2i})$ of size n the log-likelihood function is given by:

$$\begin{aligned} l(\Theta|\boldsymbol{X}) &= -\frac{n}{2}log(1-r^2) - \frac{r}{2(1-r^2)}\sum_{i=1}^{n}\{r({q_{1i}}^2+{q_{2i}}^2) - 2q_{1i}q_{1i}\} \\ &+ \frac{1}{2}\sum_{i=1}^{n}(q_{1i}^2+q_{2i}^2) + n(\alpha_1 log(\beta_1) + \alpha_2 log(\beta_2)) \\ &+ (\alpha_1 - 1)\sum_{i=1}^{n} log(x_{1i}) + (\alpha_2 - 1) + \sum_{i=1}^{n} log(x_{2i}) \\ &- n\{log(\Gamma(\alpha_1)) + log(\Gamma(\alpha_2))\} - \sum_{i=1}^{n}\beta_1 x_{1i}^j + \beta_2 x_{2i}, \end{aligned} \tag{28}$$

where $\Theta = (\alpha_1, \beta_1, \alpha_2, \beta_2, \rho)$, $q_{1i} = \Phi^{-1}(F_1(x_{1i}))$, $q_{2i} = \Phi^{-1}(F_2(x_{2i}))$ and ρ is the association parameter. From Equation (28) we can see that direct maximization of the likelihood function is tedious and numerically unstable, because the marginal parameters $\alpha_1, \beta_1, \alpha_2, \beta_2$ appear in the likelihood through a complex normal score function $\Phi^{-1}(.)$. We apply the numerical algorithm describe in previous section to obtain MLEs of the parameters. Simulation results for different sample sizes are given in Table 1. We also implemented IFM method and compare standard errors of the estimates.



(a) $r = .10$



(b) $r = .90$

Figure 2. Bivariate gamma density using *Gaussian copula.*

Table 1
Parameter estimates using MLE and IFM method for bivariate gamma density based on Gaussian copula

	MLE			
	Sample Size=100		*Sample size=500*	
Parameters	Estimates	SE	Estimates	SE
α_1=5.1	5.4026	0.8028	5.1132	0.3376
β_1=3.2	3.3963	0.5439	3.2046	0.2254
α_2=2.1	2.1833	0.3221	2.1043	0.1218
β_2=1.2	1.2577	0.2179	1.2037	0.0830
r=0.25	0.2410	0.0858	0.2487	0.0425
	IFM			
α_1=5.1	5.4025	0.8029	5.1118	0.3379
β_1=3.2	3.3960	0.5440	3.2036	0.2258
α_2=2.1	2.1834	0.3219	2.1044	0.1217
β_2=1.2	1.2578	0.2178	1.2037	0.0831
r=0.25	0.2411	0.0858	0.2487	0.0424

.

2.3.4 EXAMPLE

Consider $\boldsymbol{X} = (X_1, X_2, X_3)$ be a random variable. Marginal densities are given below.

$$f_1(x_1|\lambda_1) = \lambda_1 e^{-\lambda_1 x_1}, \quad x_1 \in [0, \infty), \tag{29}$$

$$f_2(x_2|\lambda_2) = \lambda_2 e^{-\lambda_2 x_2}, \quad x_2 \in [0, \infty), \tag{30}$$

$$f_3(x_3|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} {x_3}^{\alpha-1} e^{-\beta x_3}, \quad x_3 \in (0, \infty), \tag{31}$$

with $(\lambda_1, \lambda_2, \lambda_3, \alpha, \beta) \in (0, \infty)$. Then using Theorem [2.1.1] we can derive the joint density as:

$$\begin{aligned} f(x_1, x_2, x_3|\lambda_1, \lambda_2, \alpha, \beta, R(r)) &= c_\Phi(F_1(x_1|\lambda_1), F_2(x_2|\lambda_2), F_3(x_3|\alpha, \beta)|R(r)) \\ &\times f_1(x_1|\lambda_1) f_1(x_2|\lambda_2) f_3(x_3|\alpha, \beta), \end{aligned} \tag{32}$$

where $F_i(x_i)$ are the corresponding distribution function for $i = 1, 2, 3$. We simulate random sample from the above density and used numerical methods to obtain estimates of the parameters. We used equi-correlation structure for the simulation. Results are given Table 2.

Table 2
Parameter estimates and standard errors(SE) for trivariate joint distribution with exponential and gamma margins.

	MLE			
	Sample Size=100		*Sample size=500*	
Parameters	Estimates	SE	Estimates	SE
λ_1=2	1.9274	0.1429	1.9324	0.0660
λ_2=5	4.8738	0.2189	4.8555	0.1095
α=3.5	3.3816	0.4668	3.3271	0.2120
β=1.3	1.3452	0.1863	1.3123	0.0825
ρ=0.15	0.1964	0.0651	0.1968	0.0302
	IFM			
λ_1=2	1.9917	0.1456	1.9967	0.06638
λ_2=5	5.0417	0.2196	5.0196	0.1086
α=3.5	3.5889	0.4584	3.5290	0.2116
β=1.3	1.3451	0.1872	1.3103	0.0819
ρ=0.15	0.1643	0.0558	0.1654	0.0264

2.4 JOINT DISCRETE DISTRIBUTION USING GAUSSIAN COPULA AND ESTIMATION

Using copula we can derive joint discrete distribution also. When the marginal distributions, $F_1(x_1|\boldsymbol{\theta_1})$,$F_2(x_2|\boldsymbol{\theta_2})$, ..., $F_p(x_p|\boldsymbol{\theta_p})$ are discrete then we can obtain joint probability mass function using copula as follows :

$$\begin{aligned} f(\boldsymbol{x}|\Theta, R(\boldsymbol{r})) &= P(X_1 = x_1, X_2 = x_2, ..., X_p = x_p|\Theta, R(r)) \\ &= \sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\ldots+j_p}C_\Phi(u_{1j_1}, u_{2j_2}\ldots, u_{pj_p}|R(\boldsymbol{r})), \end{aligned} \tag{33}$$

where $\Theta = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_p)$, $C_\Phi(.|R(\boldsymbol{r}))$ is the Gaussian copula with association matrix $R(\boldsymbol{r})$, $u_{j1} = F_j(x_j)$ and $u_{j2} = F_j(x_j-)$. Here $F_j(x_j-)$ is the left-hand limit of F_j at x_j, which is equal to $F_j(x_j - 1)$.

2.4.1 ESTIMATION

Based on random sample $\boldsymbol{X} = (\boldsymbol{X}_1, ..., \boldsymbol{X}_n)$ the log-likelihood function can be written as :

$$l(\Theta|\boldsymbol{X}) = \sum_{i=1}^{n} log(f(\Theta|\boldsymbol{X}_i). \tag{34}$$

Then score functions is given by :

$$\frac{\partial l(\Theta|\boldsymbol{X})}{\partial \boldsymbol{\theta}_j} = \sum_{i=1}^{n} \frac{1}{f(\Theta|\boldsymbol{X}_i)} \frac{\partial}{\partial \boldsymbol{\theta}_j} f(\Theta|\boldsymbol{X}_i), \tag{35}$$

where,

$$\begin{aligned} \frac{\partial f(\Theta|Y_i)}{\partial \boldsymbol{\theta}_j} &= \sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\ldots+j_p} \\ &\times \sum_{k=1}^{p} \frac{\partial C_\Phi(u_{i1j_1}, u_{i2j_2}\ldots, u_{ipj_p}|R(\boldsymbol{r}))}{\partial u_{ikj_k}} \frac{\partial u_{ikj_k}}{\partial \boldsymbol{\theta}_j}. \end{aligned} \tag{36}$$

Now as we are using Gaussian copula, the expression above can further simplified as:

$$\begin{aligned} \frac{\partial C_\Phi(u_{i1j_1}, u_{i2j_2}\ldots, u_{ipj_p}|R(\boldsymbol{r}))}{\partial u_{ikj_k}} &= \frac{\partial \Phi_p(u_{i1j_1}, u_{i2j_2}\ldots, u_{ipj_p}|\boldsymbol{0}, R(\boldsymbol{r}))}{\partial u_{ikj_k}} \\ &= \Phi_{p-1}(\boldsymbol{u}_{-k}|\mu_{-k}, R_{-k}(\boldsymbol{r})) \\ &\times \phi(u_{i(k)j_k}|0,1), \end{aligned} \tag{37}$$

where $\boldsymbol{u} = (u_{i1j_1}, ..., u_{i(k-1)j_{k-1}}, u_{i(k+1)j_{k+1}}, \ldots, u_{ipj_p})$, $\boldsymbol{\mu}_{-k}$ and $R_{-k}(\boldsymbol{r})$ is the corresponding mean and association matrix of the vector $\boldsymbol{u}$. Finally

$$\frac{\partial f(\Theta|Y_i)}{\partial \boldsymbol{r}} = \sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\ldots+j_p}\frac{\partial C_\Phi(u_{i1j_1}, u_{i2j_2}\ldots, u_{ipj_p}|R(\boldsymbol{r}))}{\partial \boldsymbol{r}}, \tag{38}$$

with,

$$\begin{aligned} \frac{\partial C_\Phi(u_{i1j_1}, u_{i2j_2}\ldots, u_{ipj_p}|R(\boldsymbol{r}))}{\partial \boldsymbol{r}} &= \int_{-\infty}^{\Phi^{-1}(u_{i1j_1})}\cdots\int_{-\infty}^{\Phi^{-1}(u_{ipj_p})} \frac{\partial log(\phi_p(\boldsymbol{x}|R(\boldsymbol{r})))}{\partial \boldsymbol{r}} \\ &\times \phi_p(\boldsymbol{x}|R(\boldsymbol{r})) \\ &= \int_{-\infty}^{\Phi^{-1}(u_{i1j_1})}\cdots\int_{-\infty}^{\Phi^{-1}(u_{ipj_p})} -\frac{\partial}{\partial \boldsymbol{r}}\left(\frac{1}{2}log(|R(\boldsymbol{r})|)\right. \\ &\left.+\frac{1}{2}trace(R(\boldsymbol{r})^{-1}\boldsymbol{x}^t\boldsymbol{x})\right)\phi_p(\boldsymbol{x}|R(\boldsymbol{r})). \end{aligned} \tag{39}$$

Solving the score function numerically, we obtain MLE of the parameters.

2.4.2 EXAMPLE

Lets consider a random vector $\boldsymbol{X} = (X_1, X_2)$ with marginal probability mass functions are given bellow:

$$f(x_1) = P(X_1 = x_1) = \frac{\lambda_1^{x_1} e^{-\lambda_1}}{x_1!}, \quad x_1 \in 0, 1, 2, ... \tag{40}$$

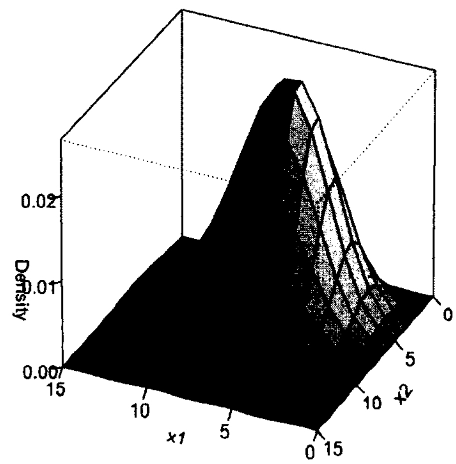
$$f(x_2) = P(X_2 = x_2) = \frac{\lambda_2^{x_2} e^{-\lambda_2}}{x_2!}, \quad x_2 \in 0, 1, 2, ... \tag{41}$$

with $\lambda_1 > 0$ and $\lambda_2 > 0$. Then the joint probability mass function $f(X = x_1, X_2 = x_2)$ can be obtain using Equation (33). Plot of such density is given in Figure 3. R is used to simulate data and numerical methods were used to obtain estimates of the parameters. Simulation results are given in Table 3.

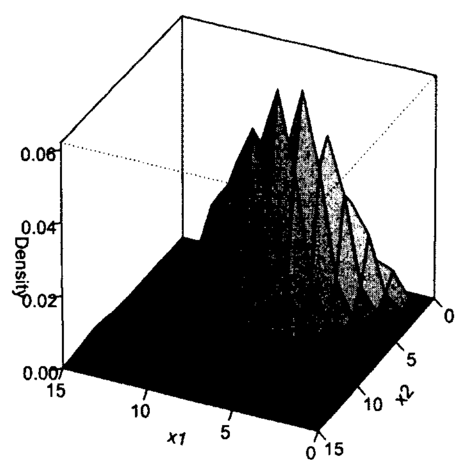
Table 3

Parameter estimates and standard errors (SE) for bivariate Poisson.

	MLE			
	Sample Size=100		*Sample size=500*	
Parameters	Estimates	SE	Estimates	SE
λ_1=6	5.6158	0.2525	5.6204	0.1266
λ_2=4	3.7378	0.1922	3.7458	0.0995
r=0.60	0.6201	0.0518	0.6139	0.0243
	IFM			
λ_1=6	6.0050	0.2523	6.0049	0.1271
λ_2=4	3.9971	0.1968	4.0015	0.1006
r=0.60	0.6155	0.0539	0.6092	0.0252



(a) $r = .10$



(b) $r = .90$

Figure 3. Bivariate Poisson density using *Gaussian copula.*

2.5 JOINT MIXED DISTRIBUTION USING GAUSSIAN COPULA AND ESTIMATION:

When the margins appear to be mixed, say the first p_1 margins are continuous and rest $p_2 = p - p_1$ are discrete, then the joint pdf can be written as (Song 2005):

$$\begin{aligned} f(\boldsymbol{x}) &= \prod_{j=1}^{j=p_1} f_j(x_j) \sum_{j_{p_1+1}=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_{p_1+1}+\cdots+j_p} \\ &\times \; C_{\Phi}^{p1}(F_1(x_1), \ldots, F_{p1}(x_{p_1}), u_{p_1+1, j_{p_1+1}}, \ldots, u_{p,j_p}), \end{aligned} \tag{42}$$

where, $u_{j1} = F_j(x_j)$ and $u_{j2} = F_j(x_j-)$ and,

$$C_{\Phi}^{p_1}(\boldsymbol{u}) = \frac{\partial^{p_1}}{\partial u_1 \ldots \partial u_{p_1}} C_{\Phi}(u_1, u_2 \ldots u_{p_1} \ldots u_p).$$

2.5.1 EXAMPLE

We use above expression of density function to obtain a bivariate pdf with mixed margins. Consider two random variable (X_1, X_2) with distributions $f_1(x_1)$ and $f_2(x_2)$, where

$$f_1(x_1|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} {x_1}^{\alpha-1} e^{-\beta x_1}, \quad x_1 \in (0, \infty), \tag{43}$$

$$f(x_2) = P(X_2 = x_2) = \frac{\lambda_1^{x_2} e^{-\lambda_1}}{x_2!}, \quad x_2 \in 0, 1, 2, \ldots \tag{44}$$

Joint density of (X_1, X_2) can be obtain using Equation (42). As the density itself is complicated and the score functions are not is a closed form, closed form solution for maximum likelihood estimates are not available. Numerical optimization method is used to obtain MLE's. Data were simulated from this density and, both estimation methods, IFM and MLE, were applied to obtain estimated. Simulation results are given in Table 4. In all the above cases we can see the IFM method is as efficient as MLE, and this true for not only continuous distributions but also for discrete and mixed types of models. This estimation method is efficient in terms of computing time. As the goal of this research is to use copula based densities to build classifier, a fast estimation method is needed to estimate the classifier. In Chapter 4 we use this IFM method to build copula based classifiers. In the next chapter we introduced finite copula mixture model and their estimation along with simulation studies.

Table 4
Parameter estimates and standard errors(SE) for Poisson and gamma mixed density.

	MLE			
	Sample Size=30		*Sample size=130*	
Parameters	Estimates	SE	Estimates	SE
α_1=3.5	3.6138	0.5423	3.4234	0.2293
β_1=6.3	6.7202	1.0957	6.3498	0.4533
λ_1=3	3.0023	0.1692	3.0012	0.0858
r=0.20	0.1974	0.0951	0.2051	0.0459
	IFM			
α_1=3.5	3.7036	0.5436	3.5086	0.2292
β_1=6.3	6.6879	1.0893	6.3067	0.4512
λ_1=3	3.0020	0.1691	3.0012	0.0858
r=0.20	0.1889	0.0864	0.1968	0.0421

CHAPTER 3

GAUSSIAN COPULA MIXTURE MODEL AND ESTIMATION

Finite mixture model is a probabilistic model represented as a weighted sum of a few parametric densities. Lindsay (1995) and McLachlan and Peel (2000) have shown that this type of mixture model is very useful for uncovering hidden structures in the data. Combining copulas as a finite mixture model helps us to not only fully understand the different dependence patterns between observed random variables, but also add more flexibility into the model. Vrac et al. (2005) used a mixture model of Frank copulas. Cuvelier and Noirhomme-Fraiture (2005) proposed a mixture model of Clayton copulas for clustering in data mining. Hu (2006) used a mixture of three copulas (Gaussian, Gumbel and Gumbel-Survival) to model dependence of monthly returns between a pair of stock indexes. In all the above cases, authors constructed a mixture copula by mixing different copulas as components of the finite mixture, and from there used the new mixture copula function to construct multivariate distribution functions using Sklar's Theorem. But here we construct the distribution first and then use them as components of finite mixture model. Due to complexity of these types of mixture models, most of the cases, MLE method of estimation is not convergent, and EM algorithm (Dempster et al. (1977)) may converge but the convergence process is too slow. In this section we propose a new estimation algorithm to estimate parameters for such mixture models.

3.1 GAUSSIAN MIXTURE COPULA MODEL

A mixture model is a powerful tool to investigate hidden structure in the data and represent complex probability density functions. The mixture model is semiparametric in that it does not put much structure to the data, unlike a fully parametric density, and does not produce model estimates highly dependent on the observed data, opposed to a fully non-parametric model (Lindsay 1995; McLachlan and Peel (2000)). In this section, we introduced mixture copula model and there estimation methods.

3.1.1 FINITE MIXTURE

A p-dimensional random vector $\boldsymbol{X} = (\boldsymbol{X}_1, ..., \boldsymbol{X}_p)$ is said to be generated from a mixture of M-component densities if its density function can be written as:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{45}$$

where, $\boldsymbol{\Theta} = (\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^M)$, $\boldsymbol{\theta}^j = (\boldsymbol{\theta}_{1j}, \boldsymbol{\theta}_{2j}, ..., \boldsymbol{\theta}_{pj})$; $R^j(\boldsymbol{r})$ is the correlation matrix of j^{th} mixture component, π_j is the mixing proportion of the j^{th} component satisfying $0 < \pi_j < 1$ and $\sum_{j=1}^{M} \pi_j = 1$. Consider the finite continuous copula mixture model where all the margins are continuous and assume that each $f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r}))$ defined as:

$$f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) = c_\Phi\left(F_1(x_1|\boldsymbol{\theta}_{1j}), F_2(x_2|\boldsymbol{\theta}_{2j}), \dots, F_p(x_p|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r})\right) \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}), \tag{46}$$

where, $c(\boldsymbol{u}) = \frac{\partial C(\boldsymbol{u})}{\partial \boldsymbol{u}}$ is the copula density function, as defined in Equation (26). To simplify the notations we will write $f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r}))$ as $f_j(\boldsymbol{x}|\boldsymbol{\theta}^j)$. Our goal is to build a likelihood function so that estimation of parameters can be performed for the mixture of M classes.

3.1.2 LIKELIHOOD AND ESTIMATION

In this section, we develop the likelihood function using a latent variable and also provide brief discussion about proposed estimation process. One way to build a likelihood function is to introduce a latent variable z_{ij} defined as:

$$z_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in j^{th}\text{class } j = 1, 2, ..., M, \; i = 1, 2, \dots, n, \\ 0 & \text{otherwise, with } \boldsymbol{x}_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \in \mathbb{R}^p. \end{cases} \tag{47}$$

Then, the random variable $\boldsymbol{Z_i} = (z_{i1}, \dots, z_{iM})$ is a multinomial random variable with parameter $\boldsymbol{\pi} = (\pi_1, \dots, \pi_M)$. Based on a random sample of size n the mixture model can be formulated by generating latent unobserved variable z_{ij} defined as above. Using such discrete latent variables, the log-likelihood function of the complete data can be written as:

$$\begin{aligned}
l(\Theta|\boldsymbol{x}) &= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_j(\boldsymbol{x}_i|\boldsymbol{\theta}^j)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + \sum_{k=1}^{p} logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\} \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}log\{c_\Phi\left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r})\right)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_1(x_{1i}|\boldsymbol{\theta}_{1j})\} + \ldots \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_p(x_{pi}|\boldsymbol{\theta}_{pj})\} + \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{(1-p)log\pi_j\} \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}log\{c_\Phi\left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r})\right)\} \\
&= l_1 + l_2 + \ldots + l_p + L_c + \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{(1-p)log\pi_j\}, \qquad (48)
\end{aligned}$$

where,

$$\begin{aligned}
l_k &= l_k(\boldsymbol{\theta}_k^j) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\}, \quad k = 1, 2, \ldots, p, \text{ and} \qquad (49)
\end{aligned}$$

$$\begin{aligned}
L_c &= L_c(\boldsymbol{\theta}^j, R(\boldsymbol{r}^j)) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}log\{c_\Phi\left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r})\right)\},
\end{aligned} \qquad (50)$$

with the parameter set $\Theta = \{\boldsymbol{\theta}^j, R^j(\boldsymbol{r})|1 \leq j \leq M\}$, $\boldsymbol{\theta}^j = \{\boldsymbol{\theta_{1j}}, \boldsymbol{\theta_{2j}}, ..., \boldsymbol{\theta_{pj}}\}$, and $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_M) \in [0,1]^M$. Now without the help of the latent indicator variable z_{ij} the log likelihood is derived as:

$$l(\Theta, |\boldsymbol{x}) = \sum_{i=1}^{n} log\left[\sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x_j}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r}))\right], \qquad (51)$$

where $f_j(.)$'s are given by Equation (46). Because of the complexity of the density and likelihood function, obtaining estimates by maximizing the likelihood function is computationally challenging. Quasi Newton method described in Chapter 2 does not converge for these types of complex functions. One can use EM algorithm to estimate the parameters, but EM algorithm is too slow, as the optimization To estimate the parameters $\Theta = \{\boldsymbol{\theta}^j, \boldsymbol{\pi}, R^j(\boldsymbol{r})\}$, we propose a two stage estimation process, using the EM algorithm. Algorithm for this two stage method is given below.

3.1.3 TWO STAGE ALGORITHM

We propose a two stage estimation procedure to obtain near optimal solution. Under the Equation (48), we can break up the likelihood in to two parts: the marginal likelihood functions l_k given in Equation (49) and the likelihood functions of copulas given in Equation (50). Our approach is to:

1. Maximize each likelihood l_k given in Equation (49), to obtain $\hat{\boldsymbol{\theta}}^j$ an estimate of the set of parameter $\boldsymbol{\theta}^j$. Use EM algorithm to obtain $\hat{\boldsymbol{\theta}}_{kj}$ for $k = 1, 2, ..., p$ and $j = 1, 2, ..., M$. For each $k = 1, 2, ..., p$, at l^{th} iteration step start with a initial value $\boldsymbol{\theta}_{kj}{}^{(l)}, \pi_k^{(l)}$. At E step, using Bayes' rule, calculate:

$$\begin{aligned} E(z_{ij}|x_{ki}) &= T_{ijk}^{(l)}(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)}) \\ &= \frac{\pi_j^{(l)} f_k(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})}{\sum_{j=1}^{M} \pi_j^{(l)} f_k(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})}, \text{ at each } i = 1, 2, \ldots, n. \end{aligned} \tag{52}$$

   Now, in M step find the parameters that maximize the function:

$$\sum_{i=1}^{n} \sum_{j=1}^{M} T_{ijk}^{(l)}(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})\{log\pi_j + logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\}, \tag{53}$$

   and set

$$\hat{\boldsymbol{\theta}}_{kj}^{(l+1)} = argmax\left(\sum_{i=1}^{n} \sum_{j=1}^{M} T_{ijk}^{(l)}(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})\{log\pi_j + logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\}\right). \tag{54}$$

   Repeat the process until convergence, to obtain $\hat{\boldsymbol{\theta}}^j = (\hat{\boldsymbol{\theta}}_{1j}, \hat{\boldsymbol{\theta}}_{2j}, ..., \hat{\boldsymbol{\theta}}_{pj})$.

2. Now use $\hat{\boldsymbol{\theta}}^j$ and maximize the likelihood function given below:

$$(\widehat{R^j(\boldsymbol{r})}, \hat{\boldsymbol{\pi}}) = argmax\left(\sum_{i=1}^{n} log \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}_j|\hat{\boldsymbol{\theta}}^j, R^j(\boldsymbol{r}))\right). \tag{55}$$

After estimating the first set of parameters and obtaining $\hat{\boldsymbol{\theta}}^j$, we choose to use the likelihood given by Equation (51) instead of using the likelihood function given by Equation (48). As the copula function is complicated and EM algorithm is an iterative optimization process, this will lead to a very slow algorithm. For complicated objective function EM algorithm is not that helpful. That is the reason behind using the likelihood function in Equation (51) to obtain estimates for second set of parameters.

3.2 EXAMPLES

In this section few examples of copula mixture models and their estimation using two stage algorithm as discussed in previous section.

Mixture exponential distribution:

Consider a mixture copula distribution with exponential margins. The theory is developed for a p variate mixture with M mixing components. Simulation for bivariate and multivariate mixtures models were performed. A p-variate exponential mixture density is given as below:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\lambda}^j, R^j(\boldsymbol{r})), \tag{56}$$

where, $\Theta = (\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \boldsymbol{r}^j, \boldsymbol{\pi})$, $\boldsymbol{\lambda}^j = (\lambda_{1j}, \lambda_{2j}, \ldots, \lambda_{pj})$, $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_M)$ are the mixing proportions with $\sum_{j=1}^{M} \pi_j = 1$, $R(\boldsymbol{r}^j)$, is the $p \times p$ association matrix for j^{th} mixture component and

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\lambda}^j, \boldsymbol{r}^j) &= \prod_{k=1}^{p} \{\lambda_{kj}\} e^{\{-\sum_{k=1}^{p} \lambda_{kj} x_k\}} \\ &\times \; c_{\Phi}(F(x_1|\lambda_{1j}), F(x_2|\lambda_{2j}), \ldots, F(x_1|\lambda_{pj})|R^j(r)), \; j = 1, 2, \ldots, M. \end{aligned} \tag{57}$$

In Equation (57), $F(.)$ is CDF of exponential distribution and $c_{\Phi}(.)$ is Gaussian copula density function. For bivariate case plot of this type of mixture density is given in the Figure 4.

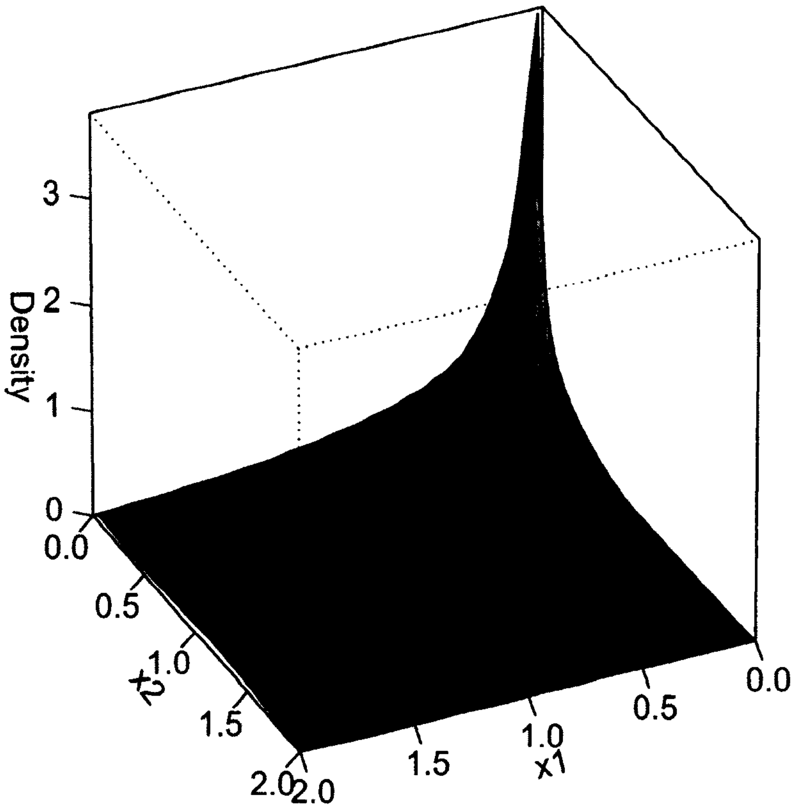


Figure 4. Bivariate exponential mixture density using *Gaussian copula.*

To estimate the parameters, based on a sample of size n, we used two stage estimation process, described in previous section.

Step 1: For each $j = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, p$, start with an initial estimate $(\widehat{\lambda}_{kj}^{(l)}, \widehat{\pi}_k^{(l)})$ and calculate $T_{ijk}^{(l)}$ as:

$$T_{ijk}^{(l)} = \frac{\widehat{\pi}_k^{(l)}\widehat{\lambda}_{kj}^{(l)}e^{-\widehat{\lambda}_{kj}^{(l)}x_{ki}}}{\sum_{j=1}^{M}\widehat{\pi}_k^{(l)}\widehat{\lambda}_{kj}^{(l)}e^{-\widehat{\lambda}_{kj}^{(l)}x_{ki}}}, \quad j = 1, 2, \ldots, M, \; k = 1, 2, \ldots, p. \tag{58}$$

Now choose

$$\widehat{\lambda}_{kj}^{(l+1)} = argmax\left(\sum_{i=1}^{n}\sum_{j=1}^{M}T_{ijk}^{(l)}\{log\pi_j + log\lambda_{kj} - \lambda_{kj}x_{ki}\}\right). \tag{59}$$

Maximizing the above function given in Equation (59) we get the set of estimates at $(l+1)^{th}$ iteration step as:

$$\widehat{\lambda}_{kj}^{(l+1)} = \frac{\sum_{i=1}^{n}T_{ijk}^{(l)}}{\sum_{i=1}^{n}x_{ki}T_{ijk}^{(l)}}, \quad j = 1, 2, \ldots, M \text{ and } k = 1, 2, \ldots, p. \tag{60}$$

Then repeat the process until convergence to obtain final set of estimates $\widehat{\lambda}_{kj}$.

Step 2: In this step use the set of estimates obtained from Step 1 to maximize the likelihood given below:

$$\begin{aligned} l(R^j(\boldsymbol{r}), \boldsymbol{\pi}|\boldsymbol{x}) &= \sum_{i=1}^{n} log \sum_{j=1}^{M} \pi_j \prod_{k=1}^{p} \{\widehat{\lambda}_{kj}\} e^{\{-\sum_{k=1}^{p}\widehat{\lambda}_{kj}x_{ki}\}} c_\Phi \Big(F(x_{1i}|\widehat{\lambda}_{1j}), \ldots \\ &\quad \ldots, F(x_{pi}|\widehat{\lambda}_{pj})|R^j(\boldsymbol{r})\Big). \end{aligned} \tag{61}$$

As the likelihood function given by Equation (61) is complex and highly nonlinear, we used numerical optimization method to obtain the estimates $\widehat{R^j(\boldsymbol{r})}$, and $\widehat{\pi}_j$, for $j = 1, 2, \ldots, M$. Simulations for three scenarios are carried out results are given in the Tables 5, 6 and 7 below. The results show that MSE is lower as we increase the sample size and MSE is a function of the value of the parameter magnitude.

We performed simulation for tri-variate mixtures, for an unstructured association matrix. As we increase the dimension p or number of component M, number of parameters will increase rapidly. To avoid such situation one can impose simple structure on the association matrix $R^j(\boldsymbol{r})$, as defined in Section 2.3.2.

Table 5

Bivariate exponential mixture density.

	Simulation-1(p=2,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=3.5	3.5326	0.0823	3.5279	0.0522
λ_{12}=0.8	0.7945	0.0077	0.7981	0.0061
λ_{21}=5.6	5.6523	0.2702	5.6420	0.0932
λ_{22}=1.6	1.5810	0.0271	1.6103	0.0170
r_1=0.65	0.6319	0.0019	0.6435	0.0007
r_2=0.15	0.1355	0.0128	0.1398	0.0045
π_1=0.70	0.6997	0.0021	0.6954	0.0016

Table 6

Tri-variate exponential mixture density with unstructured correlation.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=11.5	11.7562	1.6921	11.4484	0.8162
λ_{21}=12.1	12.2357	1.1821	12.0342	0.5887
λ_{31}=10.6	10.9668	1.9678	10.7698	0.9593
λ_{12}=3.1	3.2250	0.3511	3.0983	0.1155
λ_{22}=1.6	1.6278	0.0349	1.5892	0.0149
λ_{32}=2.1	2.1369	0.0975	2.1241	0.0446
r_{12}^1=0.55	0.5342	0.0019	0.5451	0.0009
r_{13}^1=0.65	0.6321	0.0015	0.6401	0.0009
r_{23}^1=0.45	0.4351	0.0027	0.4454	0.0001
r_{12}^2=0.35	0.3292	0.0073	0.3381	0.0033
r_{13}^2=0.15	0.1354	0.0102	0.1298	0.0053
r_{23}^2=0.22	0.2221	0.0924	0.2058	0.0042
π_1=0.68	0.6730	0.0018	0.6793	0.0003

Table 7
Tri-variate exponential mixture density with equi-correlation structure.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=14.5	14.6357	2.6232	14.4363	1.2205
λ_{21}=12.6	12.6628	1.1530	12.6359	0.4866
λ_{31}=10.6	11.0340	2.1073	10.7667	0.8656
λ_{12}=3.1	3.1129	0.1576	3.0776	0.0873
λ_{22}=1.6	1.6103	0.0373	1.5982	0.1997
λ_{32}=2.1	2.1602	0.1309	2.1473	0.0478
r^1=0.55	0.5439	0.0009	0.5479	0.0006
r^2=0.22	0.2241	0.0032	0.2227	0.0015
π_1=0.70	0.6948	0.0005	0.6967	0.0002

.

Mixture gamma model

Gaussian copula can capture dependence in many non Gaussian densities. In this example, we consider marginal distribution to be of gamma type. The mixture density is given below:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R^j(r)), \tag{62}$$

where $\Theta = (\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R^j(\boldsymbol{r}), \boldsymbol{\pi})$, $\boldsymbol{\alpha}^j = \{\alpha_{kj}|j = 1, 2, \ldots, M \text{ and } k = 1, 2, \ldots, p\}$ are the shape, $\boldsymbol{\beta}^j = \{\beta_{kj}|j = 1, 2, \ldots, M \text{ and } k = 1, 2, \ldots, p\}$ are the scale, $R(\boldsymbol{r}^j)$ is the $p \times p$ association matrix, and

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R(\boldsymbol{r}^j)) &= \prod_{k=1}^{p} \left\{ \frac{\beta_{kj}{}^{-\alpha_{kj}}}{\Gamma(\alpha_{kj})} x_k^{\alpha_{kj}-1} \right\} \left\{ e^{\sum_{k=1}^{p} -\frac{x_k}{\beta_{kj}}} \right\} c_\Phi \left(F(x_1|\alpha_{1j}, \beta_{1j}), \ldots \right. \\ & \left. \ldots, F(x_p|\alpha_{pj}, \beta_{pj}) | R^j(\boldsymbol{r}) \right), \end{aligned} \tag{63}$$

where, $c_\Phi(.)$ denotes the p-variate Gaussian copula density and F(.) is gamma distribution function. Plot of the density is given by Figure 5. As the density is not easily tractable obtaining MLE's turns out to be challenging. Two step estimation method, described in previous section, was implemented to obtain the estimates.

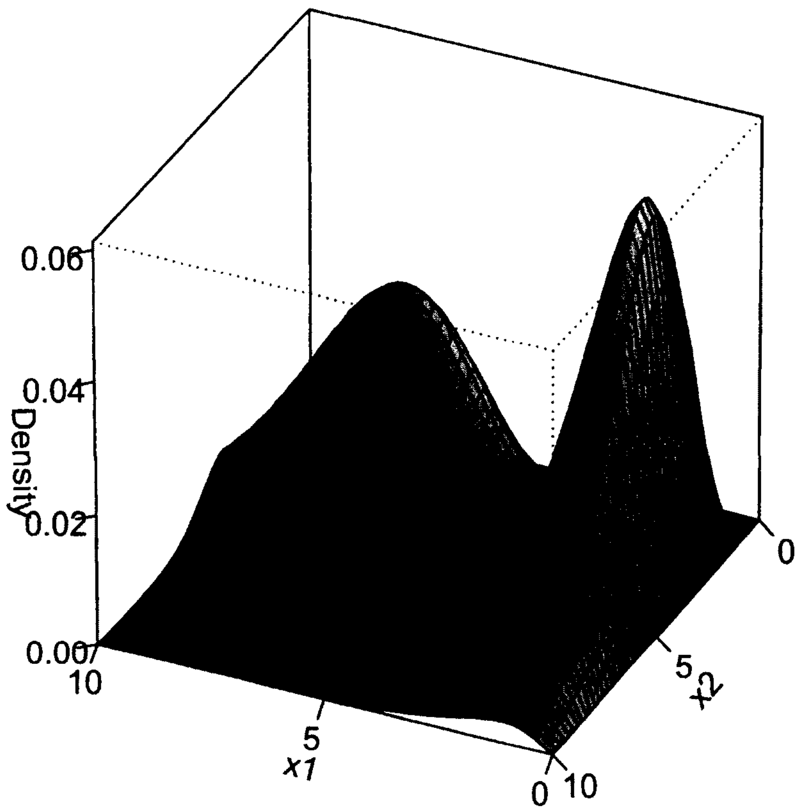


Figure 5. Bivariate gamma mixture density using *Gaussian copula*

Two step algorithm for multivariate mixture gamma model using Gaussian copuls is given below:

Step 1: In this step start with a initial set of estimates $\alpha_{kj}^{(l)}$, $\beta_{kj}^{(l)}$ and $\pi_j^{(l)}$. Using this initial set of estimates calculate:

$$T_{ijk}^{(l)} = \frac{\pi_j^{(l)} \frac{\beta_{kj}^{(l)^{-\alpha_{kj}^{(l)}}}}{\Gamma(\alpha_{kj}^{(l)})} x_{ki}^{\alpha_{kj}^{(l)}-1} e^{\frac{x_{ki}}{\beta_{kj}^{(l)}}}}{\sum_{j=1}^{M} \pi_j^{(l)} \frac{\beta_{kj}^{(l)^{-\alpha_{kj}^{(l)}}}}{\Gamma(\alpha_{kj}^{(l)})} x_{ki}^{\alpha_{kj}^{(l)}-1} e^{\frac{x_{ki}}{\beta_{kj}^{(l)}}}}. \tag{64}$$

Then use numerical optimization methods to get:

$$\begin{aligned}(\alpha_{kj}^{(l+1)}, \beta_{kj}^{(l+1)}) &= argmax\left(\sum_{i=1}^{n}\sum_{j=1}^{M} T_{ijk}^{(l)}(log\pi_j - \alpha_{kj}log\beta_{kj} - log\Gamma(\alpha_{kj})\right. \\ &+ \left.(\alpha_{kj}-1)logx_{ki} - \frac{x_{kj}}{\beta_{kj}})\right). \end{aligned} \tag{65}$$

To get final estimates $(\widehat{\alpha}_{kj}, \widehat{\beta}_{kj})$, $j = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, p$ repeat the process until convergence.

Step 2: In this step, use the estimates obtained from step 1 and maximize the likelihood given below to obtain $\widehat{\pi}_j$ and $\widehat{R^j(\boldsymbol{r})}$ as:

$$\begin{aligned}(\widehat{\pi}_j, \widehat{R^j(\boldsymbol{r})}) &= argmax\left(\sum_{i=1}^{n} log\left(\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p}\{\frac{\widehat{\beta}_{kj}^{-\widehat{\alpha}_{kj}}}{\Gamma(\widehat{\alpha}_{kj})} x_{ki}^{\widehat{\alpha}_{kj}-1}\}\{e^{\sum_{k=1}^{p} - \frac{x_{ki}}{\widehat{\beta}_{kj}}}\}\right.\right. \\ & c_\Phi\left.\left.\left(F(x_{1i}|\widehat{\alpha}_{1j}, \widehat{\beta}_{1j}), \ldots, F(x_{pi}|\widehat{\alpha}_{pj}, \widehat{\beta}_{pj})|R^j(\boldsymbol{r})\right)\right)\right). \end{aligned} \tag{66}$$

Data from bivariate and tri-variate gamma are simulated, and different correlation structures are imposed on them. Simulation results are given in Table 8, 9 and 10.

Table 8
Bivariate gamma mixture density.

	Simulation (p=2,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
α_{11}=2.3	2.1992	0.1036	2.2996	0.0234
β_{11}=0.2	0.1870	0.0115	0.1893	0.0015
α_{12}=10.2	10.2918	1.9478	10.4239	0.9135
β_{12}=3.3	2.9817	0.6623	3.0506	0.2133
α_{21}=1.9	1.8971	0.0176	1.9115	0.0024
β_{21}=3.2	3.1880	0.0702	3.2040	0.0053
α_{22}=12.5	12.8645	1.9448	12.5107	0.9881
β_{22}=9.3	9.1872	1.4355	9.3341	0.5173
r_1=0.65	0.6484	0.0487	0.6564	0.0225
r_2=0.25	0.2270	0.0061	0.2346	0.0024
π_1=0.68	0.7012	0.0004	0.7001	0.0001

.

Table 9
Tri-variate gamma mixture density, with equi-correlation structure.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
α_{11}=2.3	2.3120	0.0239	2.3347	0.0156
β_{11}=3.2	3.1857	0.0629	3.1848	0.0396
α_{12}=12.2	12.4811	1.5713	12.3412	0.6885
β_{12}=13.3	13.0438	1.1543	13.1937	0.7740
α_{21}=5.9	5.9629	0.2332	5.9846	0.1745
β_{21}=1.2	1.1892	0.0808	1.1917	0.0063
α_{22}=10.5	10.6835	1.4592	10.6078	0.6452
β_{22}=11.3	11.2237	1.3791	11.2138	0.6823
α_{31}=8.9	8.2628	0.7701	9.0093	0.6284
β_{31}=4.2	4.0433	0.1541	4.1834	0.1167
α_{32}=16.5	16.6993	1.3089	16.5341	0.7102
β_{32}=7.2	7.1805	0.5237	7.2318	0.4481
r_1=0.60	0.5958	0.0081	0.5974	0.0004
r_2=0.20	0.1929	0.0031	0.1987	0.0009
π_1=0.60	0.5985	0.0025	0.6001	0.0014

Table 10
Tri-variate gamma mixture density, with unstructured correlation.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
α_{11}=2.3	2.3091	0.0230	2.3224	0.0147
β_{11}=3.2	3.2054	0.0503	3.1727	0.0318
α_{12}=12.2	12.3913	1.6012	12.2499	1.1529
β_{12}=13.3	13.2185	1.7219	13.3517	1.5085
α_{21}=5.9	5.9434	0.1816	5.9675	0.1015
β_{21}=1.2	1.1954	0.1147	1.1869	0.0411
α_{22}=10.5	10.7406	1.3784	10.6559	0.5861
β_{22}=11.3	11.2384	1.0139	11.2019	0.5983
α_{31}=8.9	9.0390	0.6536	9.0162	0.2796
β_{31}=4.2	4.1618	0.1743	4.1542	0.0674
α_{32}=16.5	16.9413	1.7396	16.6165	1.5676
β_{32}=7.2	7.1975	1.3007	7.1712	0.6177
r_{12}^1=0.60	0.5976	0.0369	0.5961	0.0035
r_{13}^1=0.40	0.3930	0.0218	0.3978	0.0063
r_{23}^1=0.50	0.4972	0.0166	0.4987	0.0083
r_{12}^2=0.20	0.1952	0.0059	0.2005	0.0002
r_{13}^2=0.15	0.1557	0.0062	0.1456	0.0031
r_{23}^2=0.33	0.3421	0.0554	0.3237	0.0028
π_1=0.66	0.6599	0.0036	0.6594	0.0009

3.3 FINITE MIXTURE OF MULTIVARIATE DISCRETE DISTRIBUTIONS

The idea behind the two stage algorithm is to apply EM algorithm to each marginal and estimate the parameters and using those estimates maximize the full likelihood function to obtain estimates of mixing proportion and association matrix. We applied this estimation method for discrete mixture distributions. For discrete random variable $\boldsymbol{X} = (X_1, X_2, \dots, X_p)$ the M component mixture density is given by:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{67}$$

where,

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) &= P(X_1 = x_1, X_2 = x_2, \dots, X_p = x_p|\boldsymbol{\Theta_j}, R^j(\boldsymbol{r})) \\ &= \sum_{t_1=1}^{2}\sum_{t_2=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+t_2+\cdots+t_p} C_\Phi\left(u^j_{1t_1}, u^j_{2t_2}, \dots,\right. \\ &\quad \left.\dots, u^j_{pt_p}|R^j(\boldsymbol{r})\right), \end{aligned} \tag{68}$$

with, $C_\Phi(.)$ is the Gaussian copula with association matrix $R(\boldsymbol{r})$ given in Equation (15), and $u^j_{k1} = F^j_k(x_k|\boldsymbol{\theta}^{kj})$, $u^j_{k2} = F^j_k(x_k - |\boldsymbol{\theta}_{kj})$, for $k = 1, 2, \dots, p$ and $j = 1, 2, \dots, M$. In the above expression, $F_j(x_k - |\boldsymbol{\theta}^{kj})$ is the left-hand limit the distribution function F_j at x_k. Implementation of the estimation process is given below for different mixture distributions. When the data is discrete, copula function offers an approach to capture the correlation. We consider mixture of multivariate Poisson distribution using Gaussian copula.

Mixture Poisson distribution:

We apply the proposed two stage estimation method to mixed Poisson distribution, and performed simulation. For a p-variate random variable $\boldsymbol{X} = (X_1, X_2, \dots, X_3)$, with $X_k \in \{0, 1, 2, 3 \dots, \}$, $k = 1, 2, \dots, p$; and $F^j(x_k|\lambda_{kj})$ is the Poisson distribution function, mixed Poisson density is given by Equation (67) and (68). Density plot is given by Figure 6.

For a p-variate mixture Poisson model, based on a sample of size n, the two stage estimation process is given below:

Step 1: First use EM algorithm, with initial estimates $\lambda_{kj}^{(l)}$ and $\pi_j^{(l)}$ and, calculate:

$$T_{ijk}^{(l)} = \frac{\pi_j^{(l)} \frac{\lambda_{kj}^{(l)} e^{-\lambda_{kj}^{(l)}}}{x_k!}}{\sum_{j=1}^{M} \pi_j^{(l)} \frac{\lambda_{kj}^{(l)} e^{-\lambda_{kj}^{(l)}}}{x_k!}}, \quad j = 1, 2 \ldots M \text{ and } k = 1, 2, \ldots, p. \tag{69}$$

Using the above value update the set of estimates by:

$$\widehat{\lambda}_{kj}^{(l+1)} = \frac{\sum_{i=1}^{n} x_{ki} T_{ijk}^{(l)}}{\sum_{i=1}^{n} T_{ijk}^{(l)}}, \quad j = 1, 2, \ldots, M \text{ and } k = 1, 2, \ldots, p. \tag{70}$$

Continue the process until convergence to obtain final set of estimates $\widehat{\lambda}_{kj}$.

Srep 2: Obtain $\widehat{R^j(\boldsymbol{r})}$ and $\widehat{\pi}_j$ by:

$$\begin{aligned}(\widehat{R(\boldsymbol{r}^j)}, \widehat{\pi}_j) \ &= argmax \ \sum_{i=1}^{n} log \left(\sum_{j=1}^{M} \pi_j \sum_{t_1=1}^{2} \sum_{t_2=1}^{2} \cdots \sum_{t_p=1}^{2} (-1)^{t_1+t_2+\cdots+t_p} C_\Phi \left(\widehat{u}_{i1t_1}^j, \ldots \right.\right. \\ &\left.\left. \ldots, \widehat{u}_{ipt_p}^j | R^j(\boldsymbol{r}) \right) \right), \end{aligned} \tag{71}$$

where, $\widehat{u}_{ik1}^j = F_k^j(x_{ki}|\widehat{\lambda}_{kj})$, $u_{ik2}^j = F_k^j(x_{ki} - 1|\widehat{\lambda}_{kj})$, for $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, M$. Applications of this algorithm for mixture Poisson are given in Tables 11, 12, and 13 below:

Table 11

Bivariate Poisson mixture density.

	Simulation-1(p=2,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=5	4.7299	0.3615	4.4516	0.3161
λ_{12}=8	8.0585	0.1081	8.0201	0.0413
λ_{21}=3	2.6725	0.1670	2.6325	0.1509
λ_{22}=7	6.5768	0.2705	6.7704	0.2141
r_1=0.60	0.6033	0.0024	0.6225	0.0012
r_2=0.30	0.3062	0.0041	0.3070	0.0021
π_1=0.68	0.6417	0.0038	0.6573	0.0026

Table 12

Tri-variate Poisson mixture density with unstructured correlation.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=5	4.8017	0.0585	4.8385	0.0307
λ_{21}=3	2.8985	0.0202	2.9242	0.0065
λ_{31}=2	1.8990	0.0147	1.9404	0.0062
λ_{12}=8	8.2462	0.1177	8.2942	0.1019
λ_{22}=9	8.9578	0.0663	9.0721	0.0409
λ_{32}=7	6.9544	0.0913	7.0728	0.0398
r_{12}^1=0.55	0.5603	0.0043	0.5621	0.0022
r_{13}^1=0.30	0.3708	0.0468	0.3512	0.0014
r_{23}^1=0.36	0.4025	0.0278	0.3898	0.0118
r_{12}^2=0.20	0.1716	0.0072	0.1921	0.0051
r_{13}^2=0.40	0.3724	0.0764	0.4109	0.0071
r_{23}^2=0.65	0.6274	0.0556	0.6417	0.0061
π_1=0.70	0.7149	0.0017	0.7055	0.0008

Table 13

Trivariate Poisson mixture density with equi-correlation structure.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=5	4.6474	0.1542	4.6787	0.1123
λ_{21}=3	2.7842	0.2585	2.7901	0.0508
λ_{31}=2	1.8212	0.0481	1.8181	0.0376
λ_{12}=8	8.1772	0.0712	8.1507	0.0368
λ_{22}=9	8.7528	0.1735	8.7638	0.1371
λ_{32}=7	6.7123	0.1685	6.8067	0.1119
r^1=0.55	0.5821	0.0150	0.5734	0.0013
r^2=0.25	0.2644	0.0022	0.2565	0.0008
π_1=0.64	0.6539	0.0014	0.6527	0.0002

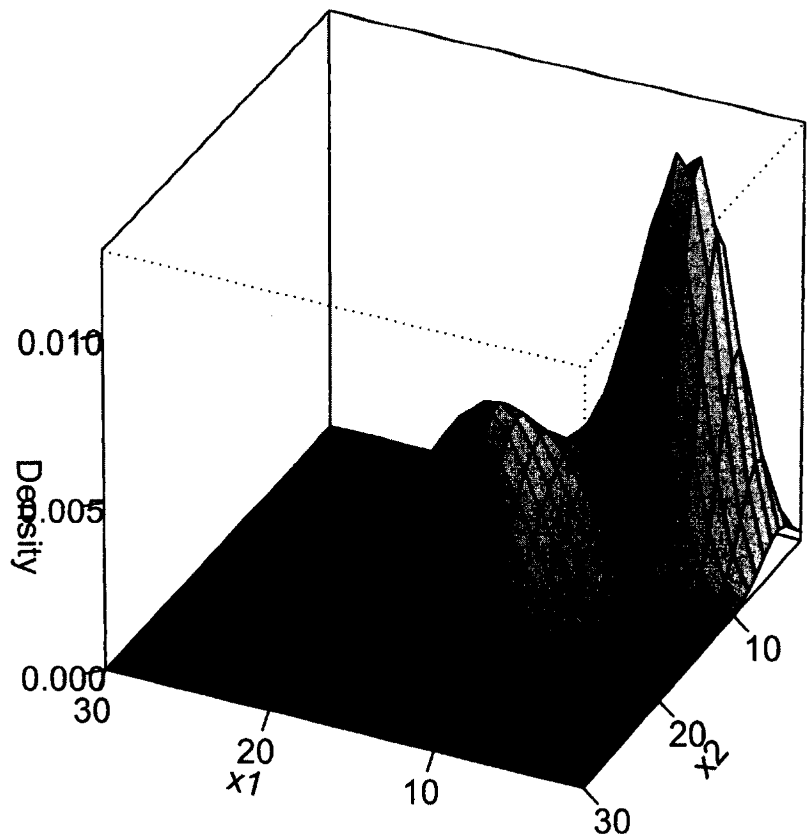


Figure 6. Bivariate Poisson mixture density using *Gaussian copula*

3.4 FINITE MIXTURE OF MULTIVARIATE MIXED DISTRIBUTIONS

Consider a p dimensional random variable $\boldsymbol{X} = (X_1, X_2, \dots, X_p)$, where first p_1 random variables are continuous and rest $p_2 = p - p_1$ of them are discrete. Then the joint density is given by Equation (42). We applied the proposed estimation method to estimate parameters for mixture of those densities given in Equation (42). Using the same notation as we used in Equation (45), consider the mixture density as:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(x|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{72}$$

where,

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) &= \prod_{k=1}^{p_1} f_k(x_k|\boldsymbol{\theta}_{kj}) \sum_{l_{p_1+1}=1}^{2} \cdots \sum_{l_p=1}^{2} (-1)^{l_{p_1+1}+\cdots+l_p} \\ &\times \; C_{\Phi}^{p_1}(F_1(x_1|\boldsymbol{\theta}_{1j}), \dots, F_{p_1}(x_{p_1}|\boldsymbol{\theta}_{p_1 j}), u^j_{p_1+1, l_{p_1+1}}, \dots, u^j_{p,l_p}), \end{aligned} \tag{73}$$

with, $u^j_{k1} = F_k(x_k|\boldsymbol{\theta}_{kj})$ and $u^j_{k2} = F_k(x_k - |\boldsymbol{\theta}_{kj})$ and,

$$C_{\Phi}^{p_1}(\boldsymbol{u}) = \frac{\partial^{p_1}}{\partial u_1, \dots, \partial u_{p_1}} C_{\Phi}(u_1, u_2, \dots, u_{p_1}, \dots, u_p).$$

3.4.1 EXAMPLE:

As an example, we mixed data from gamma and Poisson distribution. Consider a bivariate random variable $\boldsymbol{X} = (X_1, X_2)$, with joint mixture bivariate copula density as:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{2} \pi_j f_j(x|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{74}$$

where

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) &= \frac{{\beta_{1j}}^{-\alpha_{1j}}}{\Gamma(\alpha_{1j})} x_1^{\alpha_{1j}-1} e^{-\frac{x_1}{\beta_{1j}}} \left(C_{\Phi}^1(F_1(x_1|\alpha_{1j}, \beta_{1j}), F_2(x_2|\lambda_{2j})|R^j(\boldsymbol{r})) \right. \\ &\left. - \, C_{\Phi}^1(F_1(x_1|\alpha_{1j}, \beta_{1j}), F_2(x_2 - 1|\lambda_{2j})|R^j(\boldsymbol{r})) \right), \end{aligned} \tag{75}$$

with marginal distributions are mixtures of gamma and Poisson respectively. $F_1(.|\alpha_{1j}, \beta_{1j})$ and $F_2(.|\lambda_{2j})$ are CDF of gamma and Poisson distribution and the function $C^1(.)$ is given by:

$$\begin{aligned} C^1_\Phi(u_1,u_2|r) &= \frac{\partial}{\partial u_1} C_\Phi(u_1,u_2|R(r)) \\ &= \frac{\partial}{\partial u_1}\frac{1}{2\pi\sqrt{(1-r^2)}}\int_{-\infty}^{\Phi^{-1}(u_1)}\int_{-\infty}^{\Phi^{-1}(u_2)} exp\{-\frac{1}{2}\boldsymbol{x}R(r)^{-1}\boldsymbol{x}\}d\boldsymbol{x} \\ &= \frac{1}{2\pi\sqrt{(1-r^2)}}\int_{-\infty}^{\Phi^{-1}(u_2)} exp\{-\frac{(r\Phi^{-1}(u_1)-x_2)^2}{2(1-r^2)}\}dx_2 \\ &= \Phi\left(\frac{\Phi^{-1}(u_2)-r\Phi^{-1}(u_1)}{\sqrt{1-r^2}}\right). \end{aligned} \tag{76}$$

Plot of this type of density is given in Figure 7. To estimate the parameters we used two stage algorithm and results are given in the Table 14.

Table 14

Bivariate gamma and Poisson mixture density.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
α_{11}=3.3	3.3464	0.2901	3.3135	0.2019
β_{11}=1.2	1.1916	0.1152	1.1993	0.0790
α_{12}=11.3	11.2245	1.1271	11.3085	0.8268
β_{12}=4.3	4.3342	0.4264	4.2885	0.3239
λ_{21}=2	1.9978	0.1223	2.0079	0.0894
λ_{22}=8	7.9481	0.2378	8.0130	0.1851
r^1=0.60	0.5813	0.0358	0.5958	0.0258
r^2=0.35	0.3424	0.0567	0.3461	0.0396
π_1=0.54	0.5524	0.0011	0.5404	0.0008

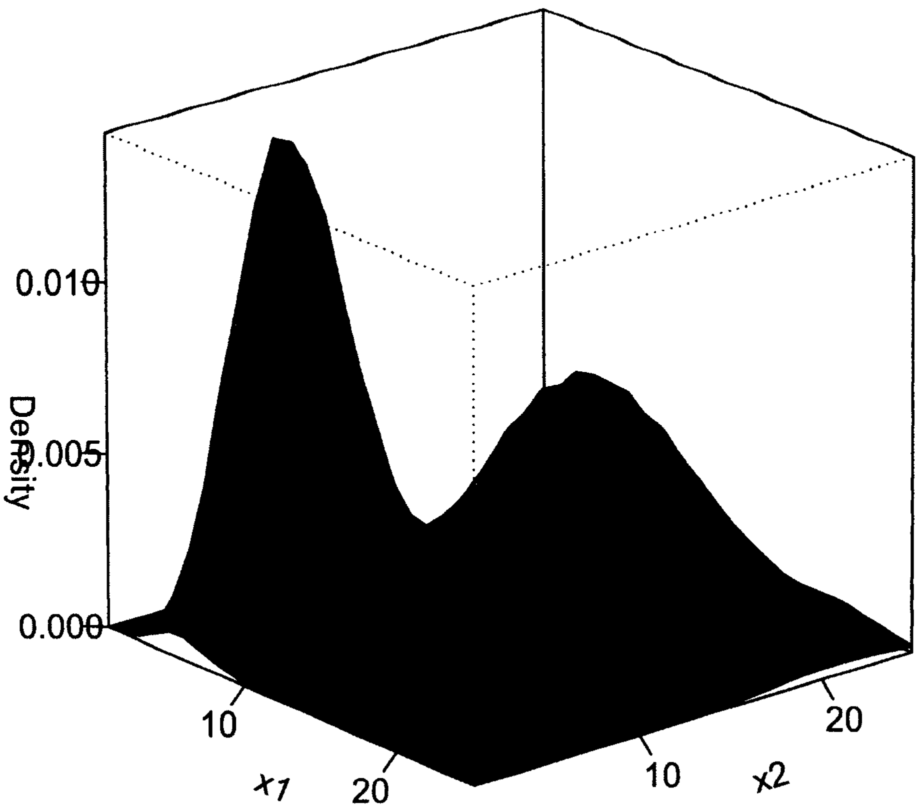


Figure 7. Bivariate gamma and Poisson mixture density using *Gaussian copula*

3.5 FINITE MIXTURE MODEL USING OTHER COPULAS

We can also construct finite mixture copula model using other Archimedean copula such as Clayton and Frank. A copula is called Archimedean if it can be written as:

$$C(u_1, u_2, \dots, u_p|\theta) = \psi^{-1}(\psi(u_1, \theta) \dots, \psi(u_2, \theta)). \tag{77}$$

In the above equation $\psi(.)$ is a continuous function, called generator, satisfies the conditions below:

- $\psi : [0, \infty) \to [0, 1]$, with $\psi(0) = 1$ and $\lim_{x\to\infty} \psi(x) = 0$.
- ψ^{-1} is given by $\psi^{-1} = inf\{v : \psi(v) \leq x\}$.
- ψ is strictly decreasing on $[0, \psi^{-1}(0)]$.

Clayton and Frank copulas are the special cases of Archimedean copula. One of the advantage of using Archimedean copula is to reduce number of parameter in the model. In Archimedean copulas there is only one dependence parameter. In this section, we simulate data from multivariate exponential mixture using Clayton and Frank copula, and used two stage algorithm to estimate parameters. Results are given Tables 15 and 16.

Table 15

Tri-variate exponential mixture density using Clayton copula.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=14.5	14.5802	1.2281	14.5080	1.0529
λ_{21}=12.6	12.7528	1.2099	12.4603	0.8378
λ_{31}=10.6	10.4453	0.5664	10.5193	0.7040
λ_{12}=3.1	3.0678	0.1489	3.1686	0.1033
λ_{22}=1.6	1.6672	0.0388	1.6107	0.0164
λ_{32}=2.1	2.0444	0.0936	2.1227	0.0391
θ^1=7	6.5926	0.2837	6.7956	0.1792
θ^2=3	2.8943	0.1933	2.9145	0.0682
π_1=0.70	0.7043	0.0017	0.7064	0.0010

Table 16
Tri-variate exponential mixture density using Frank copula.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	MSE	Estimates	MSE
λ_{11}=14.5	14.3768	1.4601	14.5102	0.9058
λ_{21}=12.6	12.4761	0.8795	12.5928	0.5274
λ_{31}=10.6	10.5177	1.0625	10.5814	0.4829
λ_{12}=3.1	3.0808	0.1897	3.0861	0.0763
λ_{22}=1.6	1.6227	0.0470	1.6158	0.0194
λ_{32}=2.1	2.1238	0.0957	2.1229	0.0369
θ^1=7	6.8281	0.1842	6.9782	0.0752
θ^2=3	2.9358	0.2339	3.1283	0.1119
π_1=0.70	0.7106	0.0052	0.7047	0.0002

Discussion:

In this chapter we proposed a two stage algorithm to estimate parameters for a finite mixture copula models. We performed simulation studies and showed this method is applicable for continuous, discrete and mixed types distributions. We also performed simulation using Archimedean copula(Clayton and Frank). This method is very fast, as we split the estimation process in two stages. For this types of finite mixture densities where direct maximization of the likelihood function by Quasi Newton Raphson method do not converge, this two stage method is very useful.

CHAPTER 4

APPLICATION TO CLASSIFICATION

In Chapter 2 we showed the estimation process of multivariate copula densities and in Chapter 3 we proposed an estimation process for finite mixture of those densities. In this chapter, we applied these estimation techniques to pattern recognition problem. Rogelio Salinas-Gutierrez et al. (2011) used copula based classifiers for image data, where all the feature vectors were continuous. We first extend this classification methods for discrete and mixed types of features. Finite mixture models are useful for classification, Hastie and Tibshirani (1996) used finite mixture under normal model with common correlation for classification. Finite copula mixture models, described in previous chapter, will be used to build the classifier for discrete or mixed type of data, and under other copula. We compared the classification power of these copula based models with logistic regression method and models with assumption of independence. For finite mixture models two stage estimation, describes in Chapter 3, is used to estimate parameters.

4.1 PROBABILISTIC CLASSIFIER USING GAUSSIAN COPULA

In this section, we used copula based densities to model the class conditional density when the features are discrete and mixed type.

4.1.1 DISCRETE FEATURES

Assuming we have G classes $\{\omega_1, \omega_2, \ldots, \omega_G\}$ and a discrete feature vector $\boldsymbol{x}^g = (x_1^g, x_2^g, \ldots, x_p^g)$ from ω_g, with marginal pmf and cdf as f_i^g and F_i^g, respectively. When all the features are discrete, then using Gaussian copula we can model the class conditional distribution as:

$$p(\boldsymbol{x}|\omega_g) = \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_1+j_2+\cdots+j_p} C_\Phi(u_{1j_1}^g, u_{2j_2}^g \ldots, u_{pj_p}^g), \tag{78}$$

where $u_{j1}^g = F_j^g(x_j)$ and $u_{j2}^g = F_j(x_j - 1)$, $g = 1, 2, \ldots, G$ and the support of F_j^g is formed by integers. Associated parameters can be estimated based on a sample of size n_g observations $\{\boldsymbol{x}_1^g, \boldsymbol{x}_2^g, \ldots, \boldsymbol{x}_{n_g}^g\}$ ($\boldsymbol{x}_i^g \in \mathbb{R}^p$) from class ω_g, $1 \leq g \leq G$. Using the class conditional density in Equation (78) one can build the classifier as:

$$\begin{aligned} \delta_g(\boldsymbol{x}) &= log(p(\omega_g)) + log(p(\boldsymbol{x}|\omega_g)) \\ &= log(p(\omega_g)) + log\left(\sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\cdots+j_p}\right. \\ &\times \left. C_\Phi(u_{1j_1}^g, u_{2j_2}^g \ldots, u_{pj_p}^g)\right), \end{aligned} \tag{79}$$

where prior probabilities are estimated as $\widehat{p(\omega_g)} = \frac{n_g}{\sum_{g=1}^{G} n_g}$, $g = 1, 2, \ldots, G$. Assign the vector $\boldsymbol{x}$ to the class ω_g, if

$$\delta_g(\boldsymbol{x}) = \underset{k}{\operatorname{argmax}}\ \delta_k(\boldsymbol{x}); \quad k = 1, 2, \ldots, G. \tag{80}$$

Parameters in Equation (79) are replaced by their estimates obtained by using IFM method of maximizing the likelihood function. The decision boundary between two classes g and l given by:

$$\{\boldsymbol{x} : \delta_g(x) = \delta_l(x)\}, \tag{81}$$

or the following holds:

$$\frac{\sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\cdots+j_p}C_\Phi(u_{1j_1}^g, u_{2j_2}^g \ldots, u_{pj_p}^g)}{\sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\cdots+j_p}C_\Phi(u_{1j_1}^l, u_{2j_2}^l \ldots, u_{pj_p}^l)} = 1. \tag{82}$$

4.1.2 MIXED TYPE FEATURES

Now consider the case of mixed feature vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_{p_1}, \ldots, x_p)$, where the first p_1 features are continuous and the rest of the $p_2 = p - p_1$ features are discrete. Then using Gaussian copula, model the class conditional density for g^{th} class as:

$$\begin{aligned} p(\boldsymbol{x}|\omega_g) &= \prod_{k=1}^{p_1} f_k^g(x_k) \sum_{j_{p_1+1}=1}^{2}\cdots\sum_{j_p=1}^{2}(-1)^{j_{p_1+1}+\cdots+j_p} \\ &\times C_\Phi^{p_1}(F_1^g(x_1), \ldots, F_{p1}^g(x_{p_1}), u_{p_1+1,j_{p_1+1}}^g, \ldots, u_{p,j_p}^g), \end{aligned} \tag{83}$$

where

$$\begin{aligned} C_\Phi^{p_1}(\boldsymbol{u}) &= \frac{\partial^{p_1}}{\partial u_1 \ldots \partial u_{p_1}} C_\Phi(u_1, u_2, \ldots, u_{p_1} \ldots u_p) \\ &= (2\pi)^{-\frac{p_2}{2}}|R|^{-\frac{1}{2}} \\ &\times \int_{-\infty}^{\phi^{-1}(u_{p_1+1})} \cdots \int_{-\infty}^{\phi^{-1}(u_p)} exp\left\{-\frac{1}{2}(\mathbf{q}_1^t, \boldsymbol{x}_2^t)^t R^{-1}(\boldsymbol{q}_1^t, \boldsymbol{x}_2^t)^t\right. \\ &+ \left.\frac{1}{2}\boldsymbol{q}_1^t\boldsymbol{q}_1\right\} d\boldsymbol{x}_2, \end{aligned} \tag{84}$$

with $\boldsymbol{x}_1 = (x_1, x_2, ...x_{p_1})$, $\boldsymbol{x}_2 = (x_{p_1+1}, ...x_p)$, $\boldsymbol{q}_1 = (\phi^{-1}(u_1), ..., \phi^{-1}(u_{p_1}))$, $\boldsymbol{q}_2 = (\phi^{-1}(u_{p_1+1}), ..., \phi^{-1}(u_p))$ and u_{t,j_t}'s are as defined in Equation (78). Assuming the prior probabilities are known in this case, one can build the classifier as:

$$\begin{aligned} \delta_g(\boldsymbol{x}) &= log(p(\omega_g)) + log(p(\boldsymbol{x}|\omega_g)) \\ &= log(p(\omega_g)) + \sum_{j=1}^{p_1} log(f_j^g(x_j)) + log(\sum_{j_{p_1+1}=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_{p_1+1}+\cdots+j_p} \\ &\times \; C_\Phi^{p_1}(F_1^g(x_1), \ldots, F_{p1}^g(x_{p_1}), u_{p_1+1, j_{p_1+1}}^g, \ldots, u_{p,j_p}^g)), \end{aligned} \tag{85}$$

and classify the feature $\boldsymbol{x}$ in to the class ω_g if:

$$\delta_g(\boldsymbol{x}) = \underset{k}{\operatorname{argmax}} \; \delta_k(\boldsymbol{x}); \quad k = 1, 2, \ldots, G. \tag{86}$$

Parameters in Equation (86) are replaced by their estimates obtained based on a training sample of size n_g from the class ω_g. We use IFM method to obtain the estimates. In this case the decision boundary between class g and class l is given by:

$$\{\boldsymbol{x} : \delta_g(x) = \delta_l(x)\}, \tag{87}$$

or the following holds:

$$\frac{\sum_{j=1}^{p_1} log(f_j^g(x_j)) + log(C_\Phi^{p_1}(x|g))}{\sum_{j=1}^{p_1} log(f_j^l(x_j)) + log(C_\Phi^{p_1}(x|l))} = 1, \tag{88}$$

where $C_\Phi^{p_1}(x|k) = C_\Phi^{p_1}(F_1^k(x_1), \ldots, F_{p1}^k(x_{p_1}), u_{p_1+1, j_{p_1+1}}^k, \ldots, u_{p,j_p}^k)$ for $k = g, l$.

4.1.3 FINITE MIXTURE COPULA MODELS

Finite copula mixture models, as described in Chapter 3, can also be use to model the class conditional densities. This section shows how to implement such type of models for continuous and mixed types of features. We model the class conditional density as:

$$p(\boldsymbol{x}|\omega_g) = \sum_{j=1}^{M} \pi_j^g f_j^g(\boldsymbol{x}|\boldsymbol{\theta}_j^g, R_j^g(\boldsymbol{r})). \tag{89}$$

with $\pi_j^g \in [0, 1]$ and $\sum_{j=1}^{M} \pi_j^g = 1$ $g = 1, 2, \ldots, G$. Based on the features: whether they are continuous, discrete and mixed type. In each cases, the form of $f_j^g(\boldsymbol{x}|\boldsymbol{\theta}_j^g, R_j^g(\boldsymbol{r}))$ is given by Equations (17), (68) and (73), respectively. To estimate the parameters we use two stage estimation as proposed in Chapter 3.

For continuous features:

Consider the feature vector $\boldsymbol{x}=(x_1,\ldots,x_p)$ is continuous. Then model the class conditional density, $p(\boldsymbol{x}|\omega_g)$ as a mixture of M multivariate densities using Gaussian copula:

$$\begin{aligned} p(\boldsymbol{x}|\omega_g) &= \sum_{j=1}^{M} \pi_j^g c_\Phi \left(F_1^g(x_1|\boldsymbol{\theta^g}_{1j}), F_2^g(x_2|\boldsymbol{\theta^g}_{2j}), \ldots, F_p^g(x_p|\boldsymbol{\theta^g}_{pj})|R_j^g(\boldsymbol{r})\right) \\ & \prod_{k=1}^{p} f_k^g(x_k|\boldsymbol{\theta^g}_{kj}). \end{aligned} \tag{90}$$

Based on the equation above, classifier can be written as:

$$\begin{aligned} \delta_g(\boldsymbol{x}) &= log(p(\omega_g)) + log\left(\sum_{j=1}^{M} \pi_j^g c_\Phi \left(F_1^g(x_1|\boldsymbol{\theta^g}_{1j}), \ldots, F_p^g(x_p|\boldsymbol{\theta^g}_{pj})|R_j^g(\boldsymbol{r})\right)\right. \\ & \left.\prod_{k=1}^{p} f_k^g(x_k|\boldsymbol{\theta^g}_{kj})\right). \end{aligned} \tag{91}$$

Using the above function, $\delta_g(\boldsymbol{x})$, in Equation (91) one can classify the feature vector $\boldsymbol{x}$ in to the class ω_g if:

$$\delta_g(\boldsymbol{x}) = \underset{k}{\text{argmax}} \ \delta_k(\boldsymbol{x}); \quad k = 1, 2, \ldots, G. \tag{92}$$

The parameters in Equation (91) can be estimated based on a labeled sample and using the algorithm proposed in Chapter 3.

Discrete case:

If the feature $\boldsymbol{x}=(x_1,\ldots,x_p)$ is discrete then one can model the class conditional density, $p(\boldsymbol{x}|\omega_g)$, as:

$$p(\boldsymbol{x}|\omega_g) = \sum_{j=1}^{M} \pi^g \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_1+j_2+\cdots+j_p} C_\Phi(u_{1j_1}^g, u_{2j_2}^g \ldots, u_{pj_p}^g). \tag{93}$$

In this case, the classifier can be built as:

$$\begin{aligned} \delta_g(\boldsymbol{x}) = log(p(\omega_g)) + log &\left(\sum_{j=1}^{M} \pi^g \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_1+j_2+\cdots+j_p}\right. \\ & \left. C_\Phi(u_{1j_1}^g, u_{2j_2}^g \ldots, u_{pj_p}^g)\right). \end{aligned} \tag{94}$$

Using Equation 94 assign the feature vector $\boldsymbol{x}$ to the class g if:

$$\delta_g(\boldsymbol{x}) = \underset{k}{\text{argmax}}\ \delta_k(\boldsymbol{x}); \quad k = 1, 2, \ldots, G. \tag{95}$$

Estimates of the parameters can be obtained from a test data set and using the algorithm described in Chapter 3.

Mixture case:

If the feature vector, $\boldsymbol{x}$=$(x_1, \ldots, x_{p_1}, \ldots, x_p)$, is mixed; that is first p_1 random variables are continuous and rest of them are discrete then model the class conditional density as:

$$\begin{aligned} p(\boldsymbol{x}|\omega_g) &= \sum_{j=1}^{M} \pi^g \prod_{k=1}^{p_1} f_k^g(x_k) \sum_{j_{p_1+1}=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_{p_1+1}+\cdots+j_p} \\ &\times \; C_{\Phi}^{p_1}(F_1^g(x_1), \ldots, F_{p1}^g(x_{p_1}), u_{p_1+1, j_{p_1+1}}^g, \ldots, u_{p,j_p}^g). \end{aligned} \tag{96}$$

Define $\delta_g(\boldsymbol{x})$ as:

$$\begin{aligned} \delta_g(\boldsymbol{x}) &= log(p(\omega_g)) + \sum_{j=1}^{M} \pi^g \prod_{k=1}^{p_1} f_k^g(x_k) \sum_{j_{p_1+1}=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_{p_1+1}+\cdots+j_p} \\ &\times \; C_{\Phi}^{p_1}(F_1^g(x_1), \ldots, F_{p1}^g(x_{p_1}), u_{p_1+1, j_{p_1+1}}^g, \ldots, u_{p,j_p}^g). \end{aligned} \tag{97}$$

The the decision rule can be written as:
Assign the feature $\boldsymbol{x}$ to the class g if:

$$\delta_g(\boldsymbol{x}) = \underset{k}{\text{argmax}}\ \delta_k(\boldsymbol{x}); \quad k = 1, 2, \ldots, G. \tag{98}$$

To estimate the parameters we used the algorithm describe in Chapter 3.

4.2 OTHER CLASSIFIER

When the features are of discrete or mixed types, then one should not model the class conditional density using multivariate normal. For such cases, two methods are very popular the Naive Bayes or independent model and the logisticregression.

4.2.1 INDEPENDENT MODEL

This model assumes a conditional independence among the features. This model can be considered as a special case of the copula models when the correlation matrix is the identity matrix. Because of its structure, this model is easy to implement. This model is very useful even the features are highly correlated (Zhang 2004). In this model, we modelthe class conditional density as:

$$p(\boldsymbol{x}^g|\omega_g) = \prod_{i=1}^{p} f_i^g(x_i), \quad g = 1, 2, \ldots, G \quad \boldsymbol{x} = (x_1, \ldots, x_p). \tag{99}$$

4.2.2 LOGISTIC REGRESSION

Logistic regression is considered as a classification tool when the features are discrete or mixed. In the literature, many authors have compared the classification power of logistic regression (LR) and LDA, QDA (Efron 1975, Maja Pohar et al. 2004). LDA is a more appropriate method when the explanatory variables are normally distributed. But whenever the assumptions of LDA are not met, LR is another popular alternative and it gives good results regardless of the distribution. As the estimates for LR are obtained by the maximum likelihood method, they have a number of asymptotic properties as well. In the next chapter, we propose simulation and real data example for each of the scenarios, and present misclassification error rates.

CHAPTER 5

SIMULATION AND APPLICATIONS

Having discussed all the techniques of building copula classifier and estimation process, simulations are great tools to evaluate the performance of the proposed method. We will compare the performance of our proposed copula model with the independent and logistic regression models when the features are of discrete and mixed types. For simplicity, we present our simulations for $G = 2$ classes.

5.1 COPULA CLASSIFIER

In this simulation setup, we consider trivariate feature vector $\boldsymbol{x} = (x_1, x_2, x_3)$, and we also assume we have only two classes, that is $G = 2$. Setup for simulations are given in Table 17.

Table 17
Simulation setup.

Class 1	Class 2
Simulation-1	
$X_1 \sim Gamma(\alpha_1 = 3.2, \beta_1 = 1.3)$	$X_1 \sim Gamma(\alpha_2 = 2.3, \beta_2 = 4.3)$
$X_2 \sim Poisson(\lambda_1 = 5.4)$	$X_2 \sim Poisson(\lambda_2 = 4.4)$
$X_3 \sim Geometric(p_1 = 0.28)$	$X_3 \sim Geometric(p_2 = 0.32)$
Simulation-2	
$X_1 \sim Gamma(\alpha_1 = 5.2, \beta_1 = 3.3)$	$X_1 \sim Gamma(\alpha_2 = 6.3, \beta_2 = 2.1)$
$X_2 \sim Poisson(\lambda_1 = 3.4)$	$X_2 \sim Poisson(\lambda_2 = 5.3)$
$X_3 \sim Geometric(p_1 = 0.18)$	$X_3 \sim Geometric(p_2 = 0.12)$
Simulation-3	
$X_1 \sim Bernoulli(p_{11} = 0.30)$	$X_1 \sim Bernoulli(p_{21} = 0.25)$
$X_2 \sim Bernoulli(p_{12} = 0.35)$	$X_2 \sim Bernoulli(p_{22} = 0.55)$
$r_1 = 0.65$	$r_1 = 0.20$

The statistical software R is used to simulate data with above marginal distributions for different values of association parameters r_1 and r_2, where r_1 and r_1 are the correlations between features in group 1 and group 2, respectively . Sample size of each group was 200. A selection of 80% observations were randomly chosen, without replacement, to build the classifier. Then, the remaining 20 observations were used to estimate misclassification error. This process was repeated 20% times and the true error was obtained as the average of those estimated errors obtained in each steps.

In the simulation apart from using Gaussian copula, we also used couple of Archimedean copula family, the Clayton and Frank copula. Misclassification rates of Gaussian copula (GC), Clayton copula (CC), Frank copula (FC), logistic regression (LR) and independent model (IM) models are given in Table 18 and 19, for a fixed value of r_j and various values of r_j $i \neq j$, $i, j = 1, 2$. Surface plot the misclassification errors for Gaussian copula, logistic regression, independent model and are given in Figures 8, 9 and 10. For simulation 3, we simulated correlated binary data for two class, and modeled the class conditional density as multivariate binary distribution using Gaussian copula. Results are given in the Table 20.

Table 18

Misclassification error of Copula LR and IM model for simulation 1.

$r_1 = 0.1$

r_2	GC	CC	FC	LR	IM
0.1	0.4120	0.4158	0.4249	0.4235	0.4188
0.2	0.4116	0.4155	0.4211	0.4190	0.4179
0.3	0.4051	0.4125	0.4381	0.4156	0.4087
0.4	0.3787	0.3995	0.4325	0.4121	0.4112
0.5	0.3556	0.3868	0.4201	0.4108	0.4045
0.6	0.3154	0.3611	0.3590	0.4035	0.4041
0.7	0.2875	0.3310	0.3648	0.3650	0.4227
0.8	0.2425	0.2913	0.3260	0.3829	0.3905
0.9	0.1675	0.2315	0.2587	0.3724	0.4031

$r_2 = 0.1$

r_1	GC	CC	FC	LR	IM
0.1	0.3895	0.4021	0.4110	0.3859	0.3952
0.2	0.4258	0.4301	0.4212	0.4095	0.4128
0.3	0.3925	0.3975	0.3960	0.3910	0.4005
0.4	0.3770	0.3890	0.4210	0.3790	0.3845
0.5	0.3597	0.3782	0.3845	0.4125	0.4230
0.6	0.3315	0.3675	0.3715	0.4150	0.4292
0.7	0.3082	0.3315	0.3522	0.3975	0.4150
0.8	0.2462	0.2890	0.3020	0.3960	0.4320
0.9	0.1875	0.2210	0.2580	0.3475	0.4125

Table 19
Misclassification error of Copula LR and IM model for simulation 2.

$r_1 = 0.5$

r_2	GC	CC	FC	LR	IM
0.1	0.2971	0.3196	0.3507	0.3046	0.3257
0.2	0.3000	0.3214	0.3201	0.3571	0.3714
0.3	0.2914	0.3017	0.3339	0.2928	0.3182
0.4	0.3103	0.3082	0.3725	0.3118	0.3303
0.5	0.2760	0.2853	0.3207	0.2817	0.3028
0.6	0.2803	0.2832	0.3400	0.2810	0.3025
0.7	0.3107	0.3360	0.3639	0.3103	0.3332
0.8	0.2261	0.2325	0.2610	0.2482	0.3085
0.9	0.1767	0.2078	0.2221	0.2196	0.3471

$r_2 = 0.5$

r_1	GC	CC	FC	LR	IM
0.1	0.3114	0.3177	0.3358	0.3127	0.3351
0.2	0.2911	0.2860	0.3242	0.3096	0.3214
0.3	0.3125	0.3118	0.3425	0.3142	0.3196
0.4	0.2757	0.3085	0.3496	0.2828	0.3167
0.5	0.2750	0.2792	0.3060	0.2778	0.3267
0.6	0.2735	0.2989	0.3439	0.2825	0.3235
0.7	0.2767	0.3053	0.3250	0.3093	0.3850
0.8	0.2328	0.2764	0.3253	0.2868	0.3225
0.9	0.1711	0.2203	0.2968	0.2378	0.3307

Table 20
Misclassification errors of Copula, LR, IM model for simulation 3.

Gaussian Copula	LR	IM
0.3237	0.4337	0.4025

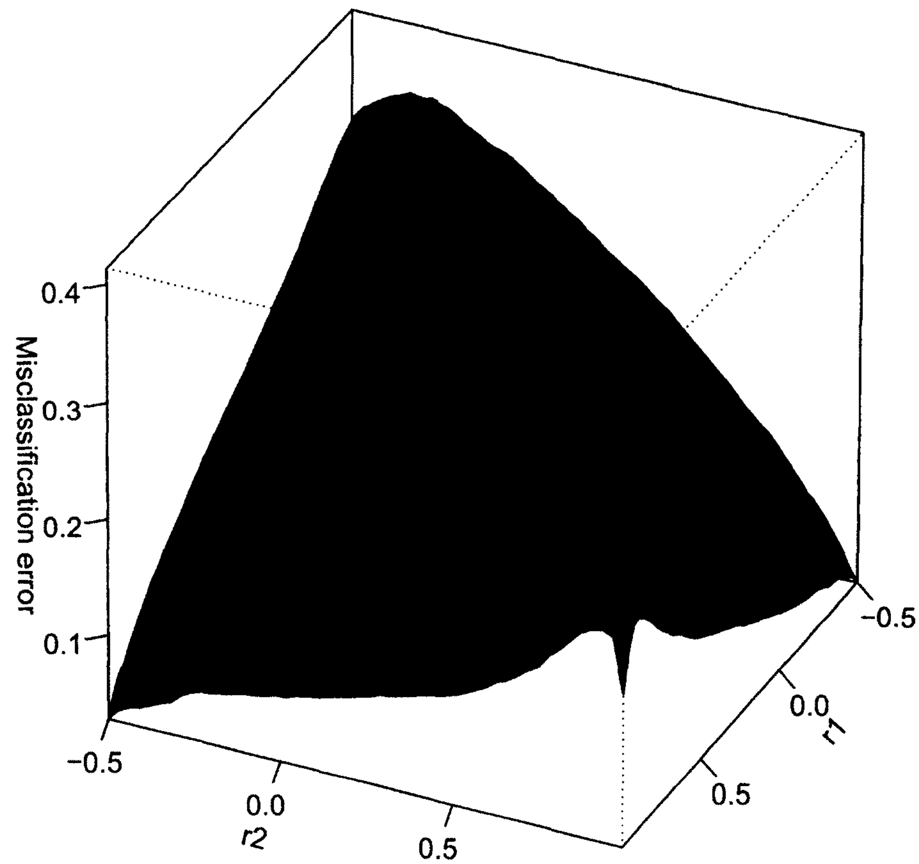


Figure 8. Surface plot of misclassification error using Gaussian copula

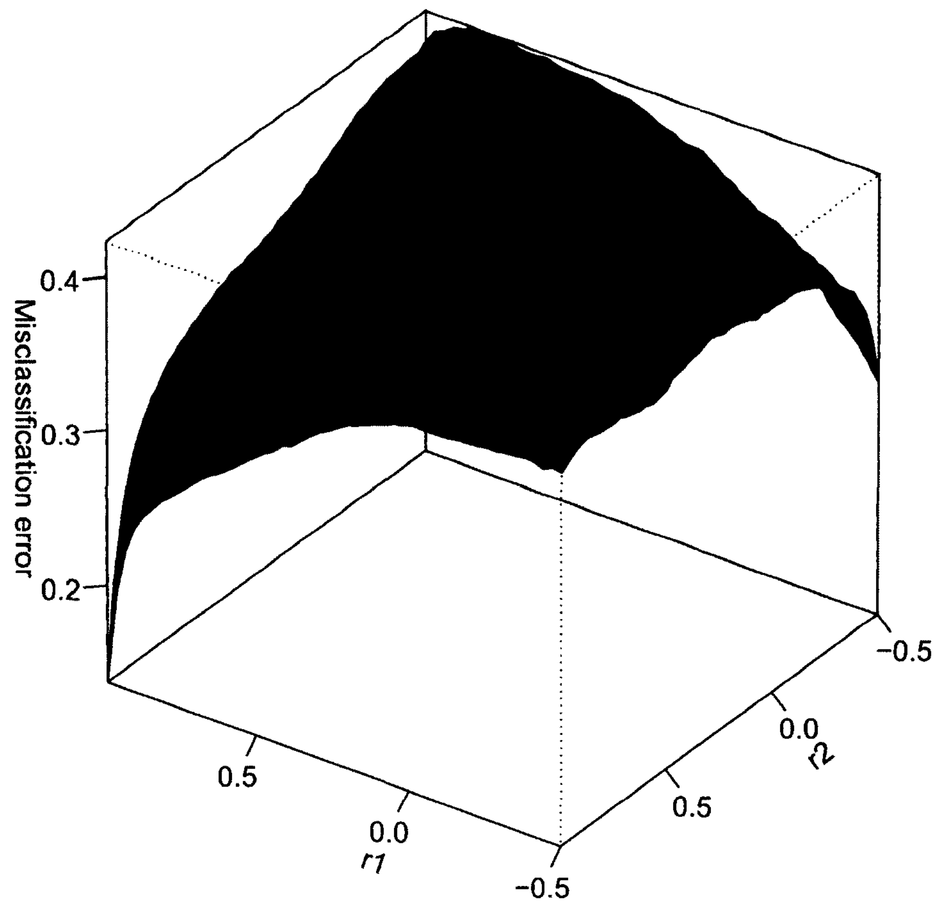


Figure 9. Surface plot of misclassification error using logistic regression

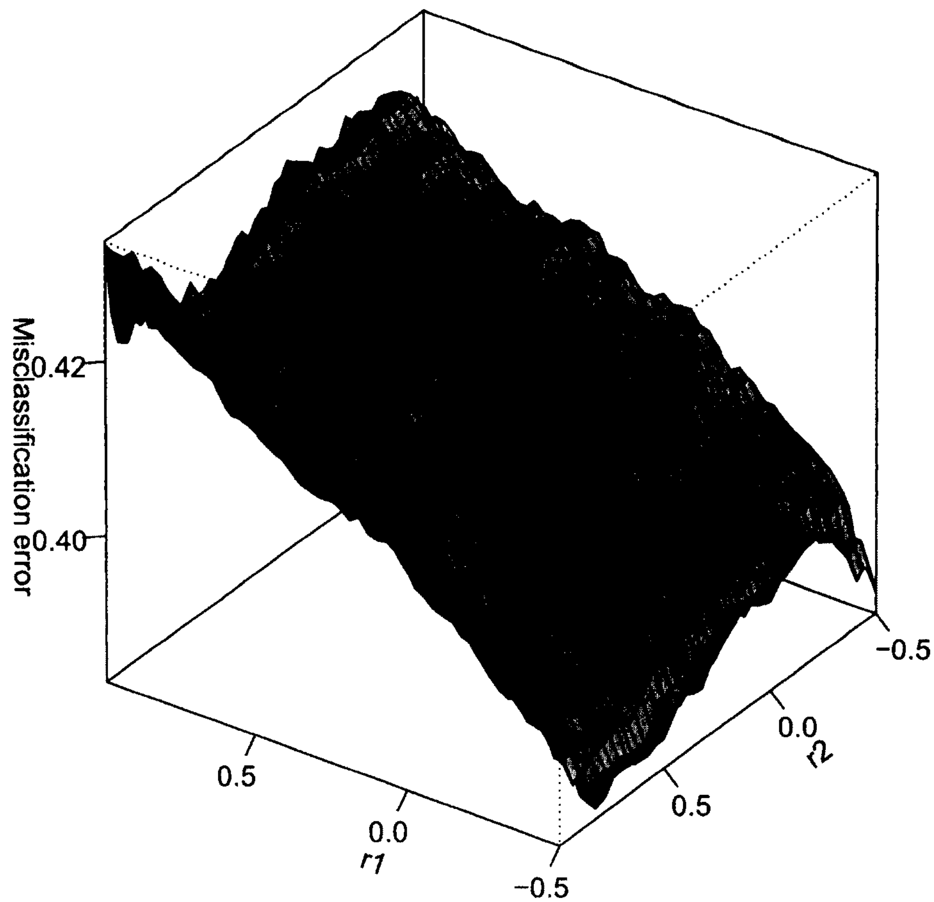


Figure 10. Surface plot of misclassification error using IM

From the simulation study, we can see the copula, LR and IM model produces similar misclassification rate when the association between the features are same among the groups. When association between the features are not equal, then copula models outperform other two models.

To extend the results of single copula classification, we propose simulations to a mixture of copula models.

5.2 USING MIXTURE COPULA MODEL

In this simulation study, we use finite mixture copula models for continuous features and mixed features. For simplicity we assume we have two groups ($G = 2$), two mixture components ($M = 2$), and $p = 2$, that is dimension is also two.

5.2.1 CONTINUOUS FEATURES

We assume all the features are continuous. Data were simulated from the mixture density given below for two sets of parameters.

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{2} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R^j(r)), \tag{100}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R^j(\boldsymbol{r}), \boldsymbol{\pi})$, $\boldsymbol{\alpha}^j = \{\alpha_{kj}|j = 1,2 \text{ and } k = 1,2\}$, $\boldsymbol{\beta}^j = \{\beta_{kj}|j = 1,2 \text{ and } k = 1,2\}$, $R^j(\boldsymbol{r})$ is the 2×2 association matrix, and

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\alpha}^j, \boldsymbol{\beta}^j, R^j(\boldsymbol{r})) \quad &= \quad \prod_{k=1}^{2}\{\frac{\beta_{kj}{}^{-\alpha_{kj}}}{\Gamma(\alpha_{kj})} x_k^{\alpha_{kj}-1}\}\{e^{\sum_{k=1}^{2} -\frac{x_k}{\beta_{kj}}}\} c_\Phi\left(F(x_1|\alpha_{1j}, \beta_{1j}),\right. \\ &\left. F(x_2|\alpha_{2j}, \beta_{2j})|R^j(\boldsymbol{r})\right). \end{aligned} \tag{101}$$

Chosen set of parameters for each classes are given in Table 21. For each class, model the class conditional density, $p(\boldsymbol{x}^g|\omega_g)$, as a mixture copula density as given in Equation (100). We simulated 1000 samples from each group, and randomly choose 800 sample for training and the rest 200 samples to estimate misclassification error. We repeat this process 20 times to obtain average misclassification error rate. In this case we also compared the misclassification error with multivariate normal mixture model (MVN mixture). Results are given in Table 22.

Table 21
Parameter sets for simulation.

Sample size=1000 (p=2,M=2)	
Class-1	Class-2
α_{11}=2.3	α_{11}=5.1
β_{11}=3.4	β_{11}=1.2
α_{12}=12.2	α_{12}=17.3
β_{12}=1.3	β_{12}=4.3
α_{21}=5.9	α_{21}=3.9
β_{21}=1.2	β_{21}=2.2
α_{22}=10.5	α_{22}=13.5
β_{22}=4.3	β_{22}=7.3
r_1=0.65	r_1=0.25
r_2=0.55	r_2=0.35
π_1=0.57	π_1=0.67

Table 22
Misclassification errors of mixture copula, QDA, LDA, IM model and MVN nuxture.

Mixture Copula	QDA	LDA	IM	MVN mixture
0.2142	0.3517	0.3942	0.5592	0.3102

In the above table we can see mixture copula model out performed the classical methods.

5.2.2 DISCRETE FEATURES

In this simulation study, we assume all the marginal distributions are discrete. We simulated data from the mixture copula density given in Equation (67) assuming all the margins are Poisson. For the g^{th} class fit the class conditional density, $p(\boldsymbol{x}|\omega_g)$, as:

$$p(\boldsymbol{x}|\omega_g) = f^g_{mix}(\boldsymbol{x}|\Theta_g) = \sum_{j=1}^{M} \pi_j f^g_j(\boldsymbol{x}|\boldsymbol{\theta}^j_g, R^j_g(\boldsymbol{r})), \quad g = 1, 2, \tag{102}$$

where $f^g_j(\boldsymbol{x}|\boldsymbol{\theta}^j_g, R^j_g(\boldsymbol{r}))$ is given by Equation (68). Simulation setup for two classes are given in Table 23. As mentioned above from a sample of size 1000, we randomly choose a sample of size 800 sample for training and 200 samples to estimate misclassification error, and repeat this process 20 times average misclassification errors are given in Table 24.

Table 23
Parameter sets for simulation.

Sample size=1000 (p=2, M=2)	
Class-1	Class-2
λ_{11}=2	λ_{11}=15
λ_{12}=10	λ_{12}=3
λ_{21}=3	λ_{21}=12
λ_{22}=12	λ_{22}=4
r_1=0.62	r_1=0.55
r_2=0.33	r_2=0.25
π_1=0.70	π_1=0.60

Table 24
Misclassification errors of mixture copula, LR and IM model.

Mixture Copula	LR	IM
0.2812	0.3315	0.4259

5.2.3 MIXED FEATURES

Finite mixture copula models can be applied on mixed type features. In this simulation, we assume one discrete and one continuous feature, and simulated data from the density given by Equation (73), assuming the margins are Poisson and gamma. For classification, model g^{th} class conditional density as in Equation (102), where each mixture density is given by Equation (73). Simulation setup is given by the Table 25.

Table 25
Parameter sets for simulation.

Sample size=1000 (p=2, M=2)	
Class-1	Class-2
α_{11}=2.3	α_{11}=12.3
β_{11}=0.2	β_{11}=0.3
α_{12}=10.2	α_{12}=5.1
β_{12}=3.5	β_{12}=2.2
λ_{12}=2	λ_{12}=3
λ_{22}=7	λ_{22}=9
r_1=0.60	r_1=0.65
r_2=0.45	r_2=0.15
π_1=0.65	π_1=0.72

From a sample of size 1000, we randomly choose 800 samples for training and the rest 200 samples for testing, we repeated this process for 20 times to estimate misclassification error rates for mixture copula, LR and IM models are given in Table 26.

Table 26
Misclassification errors of mixture copula, LR and IM model.

Mixture Copula	LR	IM
0.052	0.320	0.091

From the above simulation we can see that finite copula mixture models outperformed classical models. Unlike the classical models copula models do not assume any normality of independence, and these models can be applied for discrete and mixed type features.

5.3 APPLICATION TO REAL LIFE DATA

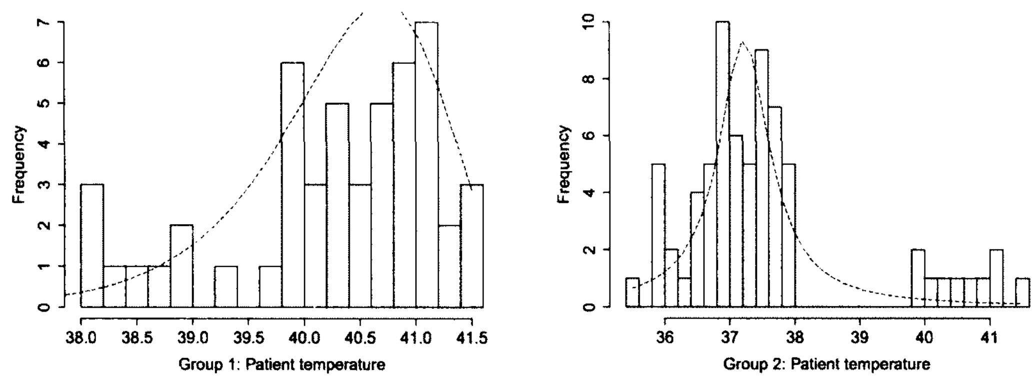
In this section we apply copula models and mixture copula models to real life data, and compare the misclassification rates with classical methods.

5.3.1 APPLICATION OF COPULA MODELS IN ACUTE INFLAMMATIONS DATA

Copula classification model is implemented to *Acute inflammations data*, used by a medical expert as a data set to test the expert system, which will perform the presumptive diagnosis of two diseases of urinary system. This data set is available online in UCI machine learning archive (http://archive.ics.uci.edu/ml/). It has five binary variables and one continues variable. At first, two binary variables, (1) *Continuous need for urination* and (2) *Micturition pain* was chosen and copula, logistic regression and independent methods were implemented. *Nephritis of renal pelvis origin* was chosen as a class variable. $N_1 = 50$ of the patients have nephritis and $N_2 = 70$ of them does not have nephritis. Cross-validation method was used to obtain misclassification error. For each group 80% of the sample were used for estimation and 20% sample for testing.

Table 27 shows that copula model outperforms LR and IM models. Further another continuous variable, *Temperature of patient* is included. Histograms for the two continuous variables, displayed in Figure 11, suggest that assuming Weibull and Cauchy distributions for the variables is reasonable. We then build misclassification error rates under these input arguments, and show comparisons with IM and LR. All three models were implemented, Table 27 summarizes the findings.

Results show that with two binary variable copula model has the lowest error rate. But at the same time when the continuous variable was added copula with unstructured (UN) association matrix R outperforms LR and IM model.
One should only impose a structure R if the dimension is high. That will reduce number of parameters we need to estimate and increase the efficiency of the computation.



(a) Weibull variable (b) Cauchy variable

Figure 11. Histograms of continuous variables for Acute inflammations data.

Table 27
Misclassification error of Copula LR and IM model.

	Copula(UN)	Copula(Equi)	LR	IM
Two binary variables:		0.2250	0.5012	0.5208
Continuous variable included:	0.0588	0.1332	0.1188	0.1375

5.3.2 APPLICATION OF FINITE MIXTURE COPULA MODELS IN WILT DATA SET

In this section, we applied finite mixture copula model in to *Wilt data set.* The pine sawyer beetle is primary causes of Japanese Pine Wilt (JPW) disease, and the oak platypodid beetle is primary cause for Japanese Oak Wilt (JOW) disease. This data set contains training and testing data from the study done by Johnson et al. This data involved detecting diseased trees in Quickbird imagery. Rapid detection of newly infected trees are very important, as without any treatment this can spread rapidly in to the forest. This data set consists of image segments, generated by segmenting the pansharpened image. Discription of the data set can be found in Johnson et al. (2013). Training data set contains 4339 observations and test data set has 500 samples. There are few training samples for the "diseased trees" class (74) and many for "other land cover" class (4265). Attribute information for this data set is given below:

1. Class: "w" (diseased trees), "n" (all other land cover).
2. GLCM_Pan: GLCM mean texture (Pan band).
3. Mean_G: Mean green value.
4. Mean_R: Mean red value.
5. Mean_NIR: Mean NIR value.
6. SD_Pan: Standard deviation (Pan band).

From above information, we can see there are five feature variables and one binary class variable. We choose three feature variables Mean_G, Mean_R, and Mean_NIR and model the class conditional density, $p(\boldsymbol{x}|\omega_j)$, as in Equation (62) and (63) with two mixture component ($M = 2$) and as the training sample size for the "diseased trees" class is large we assume equi-correlation structure for association martix R. Training data set was use for estimation and testing data set was use to estimate misclassification error rate. Estimates of the parameters for mixture copula model and Misclassification error rates for mixture copula model, LDA and QDA are given by the tables below:

Table 28
Misclassification error of mixture Copula, Copula, LDA, QDA and MVN mixture methods.

Mixture Gaussian Copula(Equi)	Gaussian Copula(Equi)	LDA	QDA	MVN mixture
0.19	0.37	0.38	0.23	0.22

Table 29
Estimates and SE for Wilt data.

Wilt data			
Class 1(other land cover)		*Class 2(diseased trees)*	
Estimates	SE	Estimates	MSE
$\hat{\alpha}_{11}$=99.2486	0.4559	33.7613	1.0598
$\hat{\beta}_{11}$= 2.2301	0.0103	0.6041	0.0341
$\hat{\alpha}_{12}$= 69.6198	0.9625	87.9913	2.0041
$\hat{\beta}_{12}$= 0.5366	0.0029	1.0793	0.2404
$\hat{\alpha}_{21}$= 10.3254	0.3497	19.7540	1.4679
$\hat{\beta}_{21}$= 44.5061	0.8249	3.4988	0.5144
$\hat{\alpha}_{22}$= 9.2675	0.2045	47.8965	1.5628
$\hat{\beta}_{22}$= 38.4753	0.0673	0.4860	0.0385
$\hat{\alpha}_{31}$= 3.3903	0.0695	100.4856	2.9416
$\hat{\beta}_{31}$= 42.3946	0.9014	1.2637	0.1195
$\hat{\alpha}_{32}$= 25.4736	0.3837	25.2849	1.4810
$\hat{\beta}_{32}$= 24.4573	0.7974	18.3809	1.1329
$\hat{r}_1$= 0.4654	0.0043	0.27297	0.2306
$\hat{r}_2$= 0.2780	0.0078	0.50302	0.2292
$\hat{\pi}_1$= 0.7043	0.0018	0.16516	0.0887

From the above table we can see that mixture copula model performs better than other models (LDA, QDA, Mixture Multivariate normal and Copula). Mixture copula models can perform better than copula models without mixture components.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this dissertation we have proposed supervised classification using copula. Gaussian copula has nice properties. It is relatively simple and inherits good properties to uncover the hidden structure of the correlation in the multivariate normal. The components of the vector are independent if the correlation matrix is diagonal. According to Song (2007), the theoretical and numerical complexity of the Gaussian copula remains the same regardless of the dimension of the vector. By using the proposed copula, one can use different correlation structures, such as the Auto regressive or equi- correlation or unstructured. We have used Gaussian and Archimedean copula based distribution for statistical classification. We have used these types of density for mixed type of features. We have also proposed an estimation process for the mixture copula model and used these models for statistical classification. This estimation process is also applicable for heterogeneous mixture distribution. Along with simulations, two real life data set were used to implement this types of model and compared their performance with classical methods. We have introduced copula and aspects of copula theory but extension to Bayesian analysis formulation methods will have substantial impact on applied multivariate data. In future, we would like to work on building more efficient algorithm, in terms of computing time, for estimating mixture copula models. Non parametric estimation process of copulas are popular in literature, as and they are very efficient (Chen and Huang (2007). We would like to extent those methods for finite copula mixture case. Finally we would like to work on building nonparametric copula classifiers and compare their error rate with existing non parametric methods like "k" nearest neighbor, decision tree and random forest. We would also like to extent those methods for finite copula mixture case.

REFERENCES

Aeberhard, S., D. Coomans, and O. de Vel. 1994. Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition.* 27(8), 10651077.

Chaganty, N. Rao. 1997. An alternative approach to the analysis of longitudinal data via generalized estimating equations. *Journal of Statistical Planning and Inference.* 63, 39-54.

Cortez, P., A. Cerdeira, F. Almeida, T. Matos and J. Reis. 2009. Modeling wine preferences by data mining from physicochemical properties in Decision Support Systems. *Elsevier.* 47(4), 547-553, ISSN: 0167-9236.

Cuvelier, E. and M. Noirhomme-Fraiture. 2005. Clayton copula and mixture decomposition, In Janssen, J., Lenca, P., (Eds.), *Applied Stochastic Models and Data Analysis (ASMDA 2005).* Brest, France, May, 699-708.

Czado, C. , R. Kastenmeier, E. C. Brechmann1, A. Min. 2012. A mixed copula model for insurance claims and claim Sizes. *Scandinavian Actuarial Journal.* (4), 278305.

De Leon, A. R. & B. Wu. 2011. Copula-based regression models for a bivariate mixed discrete and continuous outcome. *Statistics in Medicine.* 30(2), 175-185.

Dempster A.P., N.M. Laird , D.B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological).* 39(1), 1-38.

De-Waal, D.J., P.H.A.J.M. Van-Gelder. 2005. Modelling of extreme wave heights and periods through copulas. *Extremes.* 8(4), 345356.

Efron, B. 1975. The Efficiency of Logistic Regression Compared to Normal Discriminant Analys. *Journal of the American Statistical Association.* 70, 892-898.

Fahrmeir, L., Hamerle A. and Tutz G. 1984. Multivariate statistische Verfahren (1st ed.). *de Gruyter, Berlin,* 1984.

Friedman, J.H. 1989. Regularized discriminant analysis. *Journal of the American Statistical Association.* 84, 165-175.

Grigoriu, M. 2007. Multivariate distributions with specified marginals: Applications to Wind Engineering. *Journal of Engineering Mechanics.* 133(2), 174184.

Gupta, A.K. , Ed Dordrect. 1987. Error rate estimation in discriminant analysis:Recent advances. *Advances in Multivariate Statistical Analysis*, 233-252.

Hastie, T.J. and R.J. Tibshirani. 1996. Discriminant analysis by Gaussian mixturee. *Journal of the Royal Statistical Society Series B.* 58(1), 155176.

Hu, L. 2006. Dependence patterns across financial markets: a mixed copula approach. *Applied Financial Economics.* 16(10), 717-729.

James, G.M. and T.J. Hastie. 2001. Functional linear discriminant analysis for irregularly samples curves.

Jaworski, P., F. Durante, W. K. Hrdle, T. Rychlik. 2010. *Copula Theory and Its Applications Lecture Notes in Statistics.* Springer. *Journal of the Royal Statistical Society Series B.* 3(3), 533-550.

Joe, H. 2005. Asymptotic efficiency of the two-stage estimation method for copula-based models", *Journal of Multivariate Analysis.* 94, 401-419.

Joe, H. 2014. *Dependence Modeling with Copulas*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability.

Johnson, B., R. Tateishi, N. Hoan. 2013. A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. *International Journal of Remote Sensing.* 34(20), 6969-6982.

Krzanowski, W. J., P. Jonathan, W.V. McCarthy, and M.R. Thomas, Discriminant analysis with singular covariance matrices, methods and applications to spectroscopic data. *Applied Statistics.* 44(1), 101-115.

Lindsay, B.G. 1995. Mixture Models: Theory, Geometry and Applications, *In: NSF-CBMS Regional Conference Series in Probability and Statistics, Vol.5.*

Institute of Mathematical Statistics and the American Statistical Association, Alexandria, Virginia.

McLachlan, G.J. 1992. Discriminant Analysis and Statistical Pattern Recognition. *John Wiley and Sons, Ltd.*

McLachlan, G. J. 1976, The bias of the apprent error rate in discriminant analysis. *Biometrica*, 63, 239-272.

McLachlan, G.J., D. Peel. 2000. *Finite Mixture Models*, Wiley.

Nash, J. C. 1978. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation.* Second Edition. Hilger, New York.

Nelsen, R. B. 2006. *An introduction to copulas.* Spinger series in statistics.

Olkin, I., J. W. Pratt. 1958. Unbiased Estimation of Certain Correlation Coefficients. *The Annals of Mathematical Statistics.* 29(1), 201-211.

Peter, X.K. Song,Y. FAN, and J. D. Kalbfleisch. 2005. Maximization by parts in likelihood inferance, *Journal of the American Statistical Association.* 100(472), 1145-1167.

Peter, X.K. Song. 2007. *Correlated Data Analysis: Modeling, Analytics, and Applications (1st ed.).* Springer Series in Statistics vol. 365. New York, NY: Springer.

Pohar, M., M. Blas, and S. Turk. 2004. Comparison of Logistic Regression and Linear Discriminant Analysis. *Metodoloki zvezki.* 1(1), 143-161.

Raudys, S. J. and Anil K. Jain. 1991. Small samle size effects in statistical pattern recognization: Recommendation for practitioners. *IEEE transactions on pattern analysis and mechine intelligence*, 13(3), 252-264.

Salinas-Gutierrez, R., A. Hernandez-Aguirre, M. J. J. Rivera-Meraz, and E. R. Villa-Diharce. Using Gaussian Copulas in Supervised Probabilistic Classification". *Soft Computing for Intelligent Control and Mobile Robotics Studies in Computational Intelligence*, 318, 355-372.

Scholzel, C. and P. Friederichs. 2008. Multivariate non-normally distributed random variables in climate research and introduction to the copula approach. *Nonlin. Processes Geophys.* 15, 761-772.

Song, Xi C. and T. M. Huang. 2007. Nonparametric estimation of copula functions for dependence modelling, *The Canadian Journal of Statistics / La Revue Canadienne de Statistique.* 35(2), 265-282.

Todorov, V., N. Neyko, P. Neytchev. "Stability of High Breakdown Point Robust PCA, in Short Communications", *COMPSTAT'94; Physica Verlag, Heidelberg.*

Webb, Andrew R. and Keith D. Copsey. 2011. Statistical pattern recognization-3rd ed. *John Wiley & Sons, Ltd.*

W.J. Krzanowski, W.J. 1993. The location model for mixtures of categorical and continuous variables. *Journal of Classification*, 10(1), 25-49.

Vrac, M., A. Chdin, E. Diday. 2005. Clustering a Global Field of Atmospheric Profiles by Mixture Decomposition of Copulas, *Journal of Atmospheri and Oceanic Technology.* 22(10), 1445-1459.

Zhang, H. 2004. The Optimality of Naive Bayes. *FLAIRS 2004 conference. (available online: PDF (http:/ / www. cs. unb. ca/ profs/ hzhang/publications/ FLAIRS04ZhangH. pdf)).*

APPENDIX A

EXPRESSIONS OF THE SCORE FUNCTION OF FINITE MIXTURE COPULA DENSITY

A.1 CONTINUOUS COPULA MIXTURE DENSITY

Here we provide the expression of the derivatives of the copula mixture densities described in chapter 3. When all the marginal distributions are continuous mixture copula density is given by:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{103}$$

where, $f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r}))$ defined as:

$$f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) = c_\Phi\left(F_1(x_1|\boldsymbol{\theta}_{1j}), F_2(x_2|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_p|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r})\right) \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) \tag{104}$$

To estimate the parameters we used the proposed two stage method described in Chapter 3. In the first stage use EM algorithm to estimate $(\theta_{1j}, \ldots, \theta_{pj})$. In the second stage we used those estimated values to estimate $R^j(r)$ and π_j by maximizing the likelihood function below:

$$l(\boldsymbol{\pi}, R^j(\boldsymbol{r})|\boldsymbol{x}) = \sum_{i=1}^{n} log \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}^j}, R^j(\boldsymbol{r})) \tag{105}$$

Expression of the derivatives of this function with respect to π_j and $R^j((r))$ is given below:

$$\frac{\partial l}{\partial \pi_j} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi\left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r})\right)}{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi\left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r})\right)} \tag{106}$$

$$\frac{\partial l}{\partial R^j(\boldsymbol{r})} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) \frac{\partial}{\partial R^j(r)} c_\Phi\left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r})\right)}{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi\left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r})\right)} \tag{107}$$

Where,

$$\begin{aligned}
\frac{\partial}{\partial R} c_{\Phi}(q_1,\ldots,q_p|R) &= \frac{\partial}{\partial R} log(c_{\Phi}(q_1,\ldots,q_j|R))c_{\Phi}(q_1,\ldots,q_p|R) \\
&= -\frac{1}{2}\frac{\partial}{\partial R}\{log|R| + \boldsymbol{q}^{\boldsymbol{T}}(R^{-1} - I)\boldsymbol{q}\}c_{\Phi}(q_1,\ldots,q_p|R) \\
&= -\frac{1}{2}\frac{\partial}{\partial R}\{log|R| + \boldsymbol{q}^{\boldsymbol{T}}R^{-1}\boldsymbol{q} - \boldsymbol{q}^{\boldsymbol{T}}I\boldsymbol{q}\}c_{\Phi}(q_1,\ldots,q_p|R) \\
&= -\frac{1}{2}\{R^{-1} + R^{-1}\boldsymbol{q}\boldsymbol{q}^{\boldsymbol{T}}R^{-1}\}c_{\Phi}(q_1,\ldots,q_p|R), \qquad (108)
\end{aligned}$$

with $q = (q_1, q_2, \ldots, q_p)$. Equating the equations given by Equations (106) and (107) to zero, and solving them numerically MLE estimates can be obtained.

A.2 DISCRETE COPULA MIXTURE DENSITY

Here we provide the expressions of the derivatives of the discrete copula density, described in chapter 3. The joint mixture density is given by Equation (67) and (68). After obtaining the estimates $\boldsymbol{\theta}^{j}$ in the first stage we used these estimates to obtain estimates of $R^j(\boldsymbol{r})$ and π_j, by maximizing the likelihood function below:

$$\begin{aligned}
l(R^j(\boldsymbol{r}), \boldsymbol{\pi}|\boldsymbol{x}) &= \sum_{i=1}^{n} log\left(\sum_{j=1}^{M}\pi_j\sum_{t_1=1}^{2}\sum_{t_2=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+t_2+\cdots+t_p}C_{\Phi}\left(\widehat{u}_{i1t_1}^{j},\ldots,\right.\right. \\
&\left.\left.\ldots,\widehat{u}_{ipt_p}^{j}|R^j(\boldsymbol{r})\right)\right) \qquad (109)
\end{aligned}$$

Where $\widehat{u}_{ik1}^{j} = F_k^j(x_{ki}|\widehat{\theta}_{kj})$, $u_{ik2}^{j} = F_k^j(x_{ki} - 1|\widehat{\theta}_{kj})$, for $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, M$.

Score functions are given below:

$$\frac{\partial l}{\partial \pi_j} = \sum_{i=1}^{n} \frac{\sum_{t_1=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+\cdots+t_p}C_{\Phi}\left(\widehat{u}_{i1t_1}^{j},\ldots,\widehat{u}_{ipt_p}^{j}|R^j(\boldsymbol{r})\right)}{\sum_{j=1}^{M}\pi_j\sum_{t_1=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+\cdots+t_p}C_{\Phi}\left(\widehat{u}_{i1t_1}^{j},\ldots,\widehat{u}_{ipt_p}^{j}|R^j(\boldsymbol{r})\right)}, \qquad (110)$$

and

$$\frac{\partial l}{\partial R^j(\boldsymbol{r})} = \sum_{i=1}^{n} \frac{\pi_j\sum_{t_1=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+\cdots+t_p}\frac{\partial}{\partial R^j(\boldsymbol{r})}C_{\Phi}\left(\widehat{u}_{i1t_1}^{j},\ldots,\widehat{u}_{ipt_p}^{j}|R^j(\boldsymbol{r})\right)}{\sum_{j=1}^{M}\pi_j\sum_{t_1=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+\cdots+t_p}C_{\Phi}\left(\widehat{u}_{i1t_1}^{j},\ldots,\widehat{u}_{ipt_p}^{j}|R^j(\boldsymbol{r})\right)}, \qquad (111)$$

where

$$
\begin{aligned}
\frac{\partial}{\partial R} C_\Phi\left(q_1, \ldots, q_p \mid R\right) &= \int_{-\infty}^{\Phi^{-1}(q_1)} \cdots \int_{-\infty}^{\Phi^{-1}(q_p)} \frac{\partial log(\phi_p(\boldsymbol{x}|R))}{\partial R} \\
&\times \ \phi_p(\boldsymbol{x}|R) d\boldsymbol{x} \\
&= \int_{-\infty}^{\Phi^{-1}(q_1)} \cdots \int_{-\infty}^{\Phi^{-1}(q_p)} -\frac{\partial}{\partial R}\left(\frac{1}{2} log(|R|) d\boldsymbol{x}\right. \\
&\left. +\frac{1}{2} trace(R^{-1}\boldsymbol{x}^t\boldsymbol{x})\right) \phi_p(\boldsymbol{x}|R) d\boldsymbol{x} \\
&= \int_{-\infty}^{\Phi^{-1}(q_1)} \cdots \int_{-\infty}^{\Phi^{-1}(q_p)} -\frac{1}{2}(R^{-1} - R^{-1}\boldsymbol{x}\boldsymbol{x}^t R^{-1})\phi_p(\boldsymbol{x}|R) d\boldsymbol{x} \\
&= \int_{-\infty}^{\Phi^{-1}(q_1)} \cdots \int_{-\infty}^{\Phi^{-1}(q_p)} -\frac{1}{2} R^{-1}(R - \boldsymbol{x}\boldsymbol{x}^t) R^{-1}\phi_p(\boldsymbol{x}|R) d\boldsymbol{x}.
\end{aligned}
\tag{112}
$$

Solving the Equations (110) and (111) one can obtain MLE.

A.3 MIXED COPULA MIXTURE DENSITY

Expression of the score functions for the mixed type of copula, density defined in Chapter 3, is given in this section. The likelihood function using the estimates obtained in the first stage of estimation process can be written as:

$$
\begin{aligned}
l(R^j(\boldsymbol{r}), \boldsymbol{\pi}|\boldsymbol{x}) &= \sum_{i=1}^{n} log\{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p_1} f_k(x_{ki}|\widehat{\boldsymbol{\theta}}_{kj}) \sum_{l_{p_1+1}=1}^{2} \cdots \sum_{l_p=1}^{2} (-1)^{l_{p_1+1}+\cdots+l_p} \\
&\times \ C_\Phi^{p_1}(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_{p_1}(x_{p_1 i}|\widehat{\boldsymbol{\theta}}_{p_1 j}), \widehat{u}^j_{ip_1+1, l_{p_1+1}}, \ldots, \widehat{u}^j_{ip, l_p}|R^j(\boldsymbol{r}))\},
\end{aligned}
\tag{113}
$$

with, $\widehat{u}^j_{ik1} = F_k(x_{ki}|\widehat{\boldsymbol{\theta}}_{kj})$ and $\widehat{u}^j_{ik2} = F_k(x_{ki} - |\widehat{\boldsymbol{\theta}}_{kj})$, $k = p_1, \ldots, p$ and,

$$
\begin{aligned}
C_\Phi^{p_1}(\boldsymbol{u}|R) &= \frac{\partial^{p_1}}{\partial u_1, \ldots, \partial u_{p_1}} C_\Phi(u_1, u_2, \ldots, u_{p_1}, \ldots, u_p|R) \\
&= (2\pi)^{-\frac{p_2}{2}} |R|^{-\frac{1}{2}} \int_{-\infty}^{\Phi^{-1}(u_{p_1+1})} \cdots \int_{-\infty}^{\Phi^{-1}(u_p)} exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t) R^{-1}\right. \\
&\left. (\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t + \frac{1}{2}\boldsymbol{q_1}^t\boldsymbol{q_1}\right\} d\boldsymbol{x_2},
\end{aligned}
\tag{114}
$$

with $\boldsymbol{q_1} = (\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_{p_1}))$. Score functions are given below:

$$
\frac{\partial l}{\partial \pi_j} = \sum_{i=1}^{n} \frac{f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))}{\sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))} \tag{115}
$$

$$\frac{\partial l}{\partial R^j(\boldsymbol{r})} = \sum_{i=1}^{n} \frac{\pi_j \frac{\partial}{\partial R^j(\boldsymbol{r})} f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))}{\sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))} \tag{116}$$

where $f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))$ is given by Equation (73). Now to obtain $\frac{\partial}{\partial R^j(\boldsymbol{r}} f_j(\boldsymbol{x}_i|\boldsymbol{\theta}_j, R^j(\boldsymbol{r}))$ we need to derive the expression for $\frac{\partial}{\partial R} C_\phi^{p_1}(.|R)$.

$$\begin{aligned}
\frac{\partial}{\partial R} C_\phi^{p_1}(.|R) &= \frac{\partial}{\partial R}(2\pi)^{-\frac{p_2}{2}}|R|^{-\frac{1}{2}} \int_{-\infty}^{\Phi^{-1}(u_{p_1+1})} \cdots \int_{-\infty}^{\Phi^{-1}(u_p)} exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1}\right. \\
&\quad \left.(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t + \frac{1}{2}\boldsymbol{q_1}^t\boldsymbol{q_1}\right\} d\boldsymbol{x_2} \\
&= Const \times \int_{-\infty}^{\Phi^{-1}(u_{p_1+1})} \cdots \int_{-\infty}^{\Phi^{-1}(u_p)} \frac{\partial}{\partial R}\{|R|^{-\frac{1}{2}}\} exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1}\right. \\
&\quad \left.(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t\right\} + |R|^{-\frac{1}{2}} \frac{\partial}{\partial R} exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t\right\} d\boldsymbol{x_2} \\
&= Const \times \int_{-\infty}^{\Phi^{-1}(u_{p_1+1})} \cdots \int_{-\infty}^{\Phi^{-1}(u_p)} \frac{-1}{2}|R|^{\frac{1}{2}} R^{-1} exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1}\right. \\
&\quad \left.(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t\right\} - \frac{1}{2}|R|^{-\frac{1}{2}} R^{-1}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1} \\
&\quad exp\left\{-\frac{1}{2}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)R^{-1}(\boldsymbol{q_1}^t, \boldsymbol{x_2}^t)^t\right\} d\boldsymbol{x_2},
\end{aligned} \tag{117}$$

where $Const = (2\pi)^{-\frac{p_2}{2}} exp\{\frac{1}{2}\boldsymbol{q_1}^t\boldsymbol{q_1}\}$. All in the above cases score functions are highly nonlinear, we used numerical methods to solve then and obtain MLE's.

APPENDIX B

SELECTED R CODE

B.1 MLE ESTIMATES FOR TRIVARIATE GAMMA USING GAUSSIAN COPULA

```
#EM algorithm
library(optimx)
library(mixtools)
library(copula)
library(mvtnorm)
ssize=500
prob1=0.57
s1=ssize*prob1;s1
s2=ssize-s1;s2
d1=rep(1,times=285,each = 1)
d2=rep(0,times=215,each = 1)
d=c(d1,d2)
z=sample(d)
length(z)
a11=a21=b11=b21=a12=a22=b12=b22=a13=c()
a23=b13=b23=r1=r2=r3=r4=r5=r6=pi=c()
for(v in 1:20)
{
data1=matrix(0,ssize,3)
for(kk in 1:ssize)
{if(z[kk]==1)
{copg1 = normalCopula(c(0.6,0.4,0.5),dim = 3,dispstr = "un")
 mvdcg1 <- mvdc(copg1,c("gamma","gamma","gamma"),
 list(list(shape=2.3,scale=3.2),
 list(shape=5.9,scale=1.2),list(shape=8.9,scale=4.2)))
 set.seed(v+88595+100*kk)
 data1[kk,]=rMvdc(1,mvdcg1)
}
else
{ copg2 = normalCopula(c(0.2,0.15,0.33),dim = 3,dispstr = "un")
  mvdcg2 <- mvdc(copg2, c("gamma","gamma","gamma"),
```

```
    list(list(shape=12.2,scale=13.3),
    list(shape=10.5,scale=11.3),list(shape=16.5,scale=7.2)))
    set.seed(v+78865+20*kk)
    data1[kk,]=rMvdc(1,mvdcg2)
}
}
out1=gammamixEM(data1[,1],lambda=c(1/2,1/2)
,maxrestarts=20,epsilon=1e-8,maxit=5000)
out2=gammamixEM(data1[,2],lambda=c(1/2,1/2)
,maxrestarts=20,epsilon=1e-8,maxit=5000)
out3=gammamixEM(data1[,3],lambda=c(1/2,1/2)
,maxrestarts=20,epsilon=1e-8,maxit=5000)
a11[v]=out1$gamma.pars[1]
a21[v]=out1$gamma.pars[3]
b11[v]=out1$gamma.pars[2]
b21[v]=out1$gamma.pars[4]
a12[v]=out2$gamma.pars[1]
a22[v]=out2$gamma.pars[3]
b12[v]=out2$gamma.pars[2]
b22[v]=out2$gamma.pars[4]
a13[v]=out3$gamma.pars[1]
a23[v]=out3$gamma.pars[3]
b13[v]=out3$gamma.pars[2]
b23[v]=out3$gamma.pars[4]
n=length(data1[,1])
gcopula<-function(u1,u2,u3,r1,r2,r3)
{   if((u1==1)|(u2==1)|(u3==1))
    { u1=0.999999
      u2=0.999999
      u3=0.999999
    }
    U=c(qnorm(u1),qnorm(u2),qnorm(u3))
    R=matrix(c(1,r1,r2,r1,1,r3,r2,r3,1),3,3)
    I=matrix(c(1,0,0,0,1,0,0,0,1),3,3)
    as.numeric(1/(sqrt(det(R)))*exp(-0.5*(t(U)%*%(solve(R)-I)%*%U)))
}
log.lik1<-function(p,data)
{   sm=c()
    for(ii in 1:n)
    { sm[ii]=log(p[7]*dgamma(data[ii,1],shape=a11[v],scale=b11[v])
     *dgamma(data[ii,2],shape=a12[v],scale=b12[v])*
```

```
    dgamma(data[ii ,3],shape=a13[v],scale=b13[v])*
    gcopula(pgamma(data[ii ,1],shape=a11[v],scale=b11[v])
    ,pgamma(data[ii ,2],shape=a12[v],scale=b12[v]),pgamma(data[ii ,3],
    shape=a13[v],scale=b13[v]),p[1],p[2],p[3]) + (1-p[7])
    *dgamma(data[ii ,1],shape=a21[v],scale=b21[v])
    *dgamma(data[ii ,2],shape=a22[v],scale=b22[v])
    *dgamma(data[ii ,3],shape=a23[v],scale=b23[v])
    *gcopula(pgamma(data[ii ,1],shape=a21[v],scale=b21[v]),
    pgamma(data[ii ,2],shape=a22[v],scale=b22[v])
    ,pgamma(data[ii ,3],shape=a23[v],scale=b23[v]),p[4],p[5],p[6]))

  }
  -sum(sm)
  }
  p=c(0.5,0.33,0.4,0.15,0.10,0.22,0.5)
  log.lik1(p,data1)
  op1=optimx(p,log.lik1,data=data1,method="Nelder-Mead")
  r1[v]=op1$p1
  r2[v]=op1$p2
  r3[v]=op1$p3
  r4[v]=op1$p4
  r5[v]=op1$p5
  r6[v]=op1$p6
  pi[v]=op1$p7
}
#Estimates and SE

mean(a11);sd(a11);mean(a21);sd(a21);mean(b11);sd(b11);mean(b21)
sd(b21);mean(a12);sd(a12);mean(a22);sd(a22);mean(b12);sd(b12)
mean(b22);sd(b22)mean(a13);sd(a13);mean(a23);sd(a23);mean(b13)
sd(b13);mean(b23);sd(b23);mean(r1);sd(r1);mean(r2);sd(r2)
mean(r3);sd(r3);mean(r4);sd(r4);mean(r5);sd(r5)
mean(r6);sd(r6);mean(pi);sd(pi)
```

VITA

Sumen Sen
Department of Mathematics and Statistics
Old Dominion University
Norfolk, VA 23529

Education

Ph.D.	Computational and Applied Mathematics (Statistics) Old Dominion University, Norfolk, VA. (Aug 2015)
M.S.	Statistics University of Central Florida, Orlando, FL. (May 2011)
B.Sc.	Mathematics University of Calcutta, India. (September 2008)

Experience Publications:

08/2011 - 07/2015	Graduate Teaching Assistant Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA
08/2009 - 07/2011	Graduate Teaching Assistant Department of Mathematics and Statistics, University of Central Florida, Orlando, FL.

S.Sen, N. Diawara and R. Lamichhane (2014) "A bivariate distribution with conditional gamma and its multivariate form". Journal of Modern Applied Statistical Methods: Vol. 13: Iss. 2, Article 9.
S.Sen, N. Diawara and Khan M. Iftekharuddin (2015) "Statistical pattern recognition using Gaussian copula". Journal of Statistical Theory and Practice, DOI: 10.1080/15598608.2015.1008607.
S.Sen, N. Diawara (2015) "Statistical classification using finite mixture Gaussian copula models". In preparation.

Typeset using LaTeX.