HAL open science

MTCopula: Synthetic Complex Data Generation Using Copula

Fodil Benali, Damien Bodénès, Nicolas Labroche, Cyril de Runz

To cite this version:

Fodil Benali, Damien Bodénès, Nicolas Labroche, Cyril de Runz. MTCopula: Synthetic Complex Data Generation Using Copula. 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP), 2021, Nicosia, Cyprus. pp.51-60. hal-03188317

HAL Id: hal-03188317

https://hal.science/hal-03188317v1

Submitted on 1 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MTCopula: Synthetic Complex Data Generation Using Copula

Fodil Benali, Damien Bodénès
Adwanted Group
Paris, France
{fbenali,dbodenes}@adwanted.com

Nicolas Labroche, Cyril de Runz
BDTLN - LIFAT, University of Tours
Blois, France
{nicolas.labroche,cyril.derunz}@univ-tours.fr

ABSTRACT

Nowadays, marketing strategies are data-driven, and their quality depends significantly on the quality and quantity of available data. As it is not always possible to access this data, there is a need for synthetic data generation. Most of the existing techniques work well for low-dimensional data and may fail to capture complex dependencies between data dimensions. Moreover, the tedious task of identifying the right combination of models and their respective parameters is still an open problem. In this paper, we present MTCopula, a novel approach for synthetic complex data generation based on Copula functions. MTCopula is a flexible and extendable solution that automatically chooses the best Copula model, between Gaussian Copula and T-Copula models, and the best-fitted marginals to catch the data complexity. It relies on Maximum Likelihood Estimation to fit the possible marginal distribution models and introduces Akaike Information Criterion to choose both the best marginals and Copula models, thus removing the need for a tedious manual exploration of their possible combinations. Comparisons with state-of-art synthetic data generators on a real use case private dataset, called AdWanted, and literature datasets show that our approach preserves better the variable behaviors and the dependencies between variables in the generated synthetic datasets.

1 INTRODUCTION

Nowadays, data are the new gold. Unfortunately, it is difficult to get this valuable data as sometimes companies do not have the means to collect large data sets relevant to their business. Others have difficulties sharing sensitive data due to the business contract confidentiality or record privacy [25], which is the case of ad planning, our industrial context. In this specific context, only very few high quality and complex data (multidimensional, multivariate, categorical/continuous, time series, *etc.*), supposedly representative of the whole dataset, are available for generating a large and realistic synthetic dataset. Therefore, there is a true need for a realistic complex data generator.

Our objective is to generate new data that maintains the same characteristics as the original data, such as the distribution of attributes and dependency between them. Moreover, it must be structurally and formally resembling the original data so that any work done on the original data can be done using the synthetic data [21]. This cannot be done using the usual one-dimensional synthetic data generation [17] method because, when applying it in a high dimensional context, it does not allow to model the dependency between variables. To tackle those issues, several recent works focused on deep learning approaches such as Generative Adversarial Network (GAN), but those approaches require a large amount of data for the learning step and thus can not be used for our problem.

© Copyright 2021 for this paper held by its author(s).
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Nevertheless, recently, there has been a growing interest in Copula-based models for estimating [1, 26] and sampling [10, 29] from a multivariate distribution function. Copula [15] are joint probability distributions in which any univariate continuous probability distribution can be plugged in as a marginal. The Copula captures the joint behavior of the variables and models the dependence structure, whereas each marginal models the individual behavior of its corresponding variable. Thus, our problem turns into **building a joint probability distribution** that best fits the marginal distribution of each variable and allows capturing different dependencies between these variables. This problem is often understood as a structure learning task that can be solved in a constructive way while attempting to maximize the likelihood or some information theory criterion [22].

Copula is a flexible mathematical tool that can support different configurations in terms of marginal fitting distribution and copula models. To choose the best configuration is not simple. For instance, the literature Copula-based data generators use Gaussian Copula model but this model has difficulties to capture tail dependencies, which may affect the quality of the data generation.

In this work, we present MTCopula, a flexible and extendable Copula-based approach to model and generate complex data (e.g., multivariate time series) with automatic optimization of Copula configurations. Our contributions are the following: (1) we formalize the problem of synthetic complex data generation, (2) we propose an approach MTCopula to learn Copulas and automatically choose the marginals and Copula models that best fit the data we want to generate, and (3) we describe experiments showing how well MTCopula preserves implicit relationships between variables in the synthetic datasets on a real use case and state-of-the-art datasets.

This paper is organized as follows: Section 2 presents the related works. Sections 3 and 4 introduce the main concepts related to dependency structures and Copulas. Section 5 provides the problem description while Section 6 describes MTCopula, our solution to model and generate data with their structure dependencies. Section 7 presents the experiments performed to show the properties and the efficiency of our approach. Finally, Section 8 presents the conclusion and opens future works.

2 RELATED WORK

The fundamental idea of the process of synthetic data generation involves sampling data from a pre-trained statistical model, then use the sample data in place of the original data. In this section, we study related works with regard to this preliminary notion and our problem, which is the generation of synthetic complex data. Complex data denotes a case where data can be a **mixture of continuous and categorical variables**, in a **high dimensional context**, and with **the possibility of having temporal relations** in the order of variables (time series) and dependencies in **variables' distributions tails**.

First, our problem is not about generating data from specifications: it is rather about generating synthetic data from real

CEUR-WS.org/Vol-2840/paper8.pdf

data samples, which, for different reasons, are generally available in small quantities but with good quality. Therefore approaches such as AutoUniv[1] cannot be applied.

Second, in the simplest case of one-dimensional synthetic data generation, sampling from a random variable X with a known probability distribution F is usually done using the classical approach Inverse Transform Sampling (ITS) [17], in which pseudo-random samples U_1, ...,U_N are generated from a uniform distribution U on $[0, 1]$ and then transformed by $F_X^{-1}(U_1)$, ..., $F_X^{-1}(U_N)$. The issue with applying such an approach in high dimensional synthetic data generation is that it will not allow modeling the dependency between variables. As a consequence, it generates an independent joint distribution. Therefore, this approach cannot capture the dependency structure, which is one of our problem's key elements.

Then, traditionally, a perturbation technique, called General Additive Data Perturbation (GADP) has been widely used for synthetic data generation [14]. The principle consists in fitting a multivariate Gaussian distribution on the input data, $X \sim \mathcal{N}(\mu, \Sigma)$. After that, the estimated multivariate Gaussian variable X is used to generate the synthetic data Y by adding a noise variable e, $Y = X + e$. where e is a Gaussian error. The problem with this method is that it does not allow us to best model the marginal behaviors of variables since it considers only Gaussian marginal distributions by construction, which can be limiting as observed in our experiments. Moreover, it does not model the tail dependence as is consider the correlation matrix Σ only. Another variant of GADP is the Dirichlet multivariate synthesizer based on MLE [24]. The problem with MLE for multivariate distribution fitting is that it has to be maximized over a potentially high-dimensional parameter space, which is computationally very expensive.

The rise of deep learning in the last years has brought forth new machine learning techniques such as generative adversarial networks (GANs)[18, 23]. These techniques perform better than state-of-the-art works in many fields but require large datasets for training, which can be a significant problem because collecting data is often expensive or time-consuming. Even when data is already collected, this type of method cannot be applied due to privacy or confidentiality issues. Moreover, GANs, like most of deep learning approaches, act as a black-box and does not allow a business expert to understand how the synthetic data are actually generated.

Recently, there has been a growing interest in Copula-based modeling and synthetic data generation. Despite the fact that Copula models can best model dependencies and the marginal behaviors of variables, most contributions suggested for synthetic data generation [10, 19] have focused on a single model: the Gaussian Copula. However, this model assumes a structure dependency that may only loosely capture the interaction between variables [11] as it does not allow to model the tail dependence. In addition, these contributions use the Pearson correlation factor to estimate the correlation matrix, which is not invariant under strictly monotone non linear transformation, and while this hypothesis is crucial in the Copula's context. As a consequence, this impacts structure dependency preservation during the copula learning fitting. Nevertheless, Copulas with both marginal fittings and its dependency structure allow for a transparent explanation of the generated data.

In conclusion, Copulas seems to be the best solution for generating datasets based on complex tiny real datasets, but there is a need for parameter calibration automation. Before introducing the Copula, we present the dependency structure notions in the next section.

[1] https://archive.ics.uci.edu/ml/datasets/AutoUniv

3 DEPENDENCY STRUCTURES

One of our goals is to capture the dependency structure relationship $\mathcal{D}$ between data/variables to finally be able to generate data respecting those dependencies. This section focuses on the main measures used to summarize dependency between components of a random vector.

3.1 Pearson Product–Moment Correlation

The Pearson product-moment correlation ρ is a measure of the linear relationship between two random variables X_1, X_2. A relationship is linear when a change in one variable is associated with a proportional change in the other variable. Pearson correlation takes values in the interval [-1, 1], and it is defined as:

$$\rho(X_1, X_2) = Cor(X_1, X_2) = \frac{Cov(X_1, X_2)}{\sqrt{Var(X_1)}\sqrt{Var(X_2)}}. \quad (1)$$

The problem with *Pearson correlation* ρ is that it is not invariant under non-linear strictly increasing transformations of the marginals [9].

3.2 Rank Correlation

In practice, we have a monotonic relationship between measurements in which variables tend to change together, but not necessarily at a constant rate. In this case, rank correlation statistics are well suited for determining whether there is a correspondence between random variables. We mention here the two important rank correlation measures, namely *Spearman* and *Kendall*.

Definition 3.1 (Spearman ρ_s correlation). Let (X_1,X_2) be a bivariate random vector with continuous marginal dfs F_1 and F_2. The Spearman's factor ρ_s is defined by:

$$\rho_s(X_1, X_2) = \rho(F_1(X_1), F_2(X_2)). \quad (2)$$

Definition 3.2 (Kendall's τ correlation). Kendall's τ is defined as the probability of concordance minus the probability of discordance of two random variables X_1 and X_2:

$$\begin{aligned}\tau(X_1, X_2) = P((X_{11}, X_{21})(X_{12}, X_{22}) > 0) - \\ P((X_{11}, X_{21})(X_{12}, X_{22}) < 0),\end{aligned} \quad (3)$$

where (X_{11}, X_{21}) and (X_{12}, X_{22}) are independent and identically distributed copies of (X_1, X_2).

Both Kendall's τ and Spearman's ρ_s are dependence invariant with respect to monotone transformations of the marginals. Their range of values is the interval [-1, 1] [3].

3.3 Tail Dependence

Understanding the dependence structure of rare events is fundamental in order to best model random variables behaviors. Measures of dependence like *Pearson linear correlation*, *Spearman* and *Kendall correlation* are not able to correctly capture and characterize the joint occurrence of large and small values of random variables [8]. The Pearson correlation describes how well two random variables are linearly correlated with respect to their entire distribution. However, this information is not useful to model the extreme behavior of two random variables [27].

To evaluate tail dependence, the tail dependence coefficient is calculated as follows:

Definition 3.3 (Upper and lower tail dependence coefficient). The upper tail dependence coefficient of a bivariate distribution is defined as:

$$\lambda^{upper} = \lim_{t \to 1^-} P(X_2 > F_2^{-1}(t) | X_1 > F_1^{-1}(t)). \quad (4)$$

The lower tail dependence coefficient is:

$$\lambda^{lower} = \lim_{t \to 0^+} P(X_2 \leq F_2^{-1}(t) | X_1 \leq F_1^{-1}(t)). \quad (5)$$

Using those definitions, we are now able to introduce the Copula on which our approach is based.

4 COPULA

This section is devoted to summarizing Copula principles as they are the key part for data generation that conserve dependencies. A deeper explanation about copula can be found in [15].

4.1 Copula Foundations

A *Copula* is a Latin term which means *link*. In recent years, due to its ability to catch the core of multivariate data distributions and their dependencies, copula was applied in a wide range of areas such as econometric modeling [20] and quantitative risk management [12].

This concept was first introduced in statistical modeling in 1959 by *Sklar* [28] to describe the function that "join together" one-dimensional distribution functions to form a multivariate distribution function. It is based on Sklar's Theorem 4.1.

Theorem 4.1 (Sklar's theorem). *Let $(X_1, ..., X_j, ..., X_d)$ be a d-dimensional random vector with joint distribution function H and marginal distribution functions F_i , i = 1,..., d, then there exists a d-copula $C : [0,1]^d \to [0,1]$, such that for all x in $\mathbb{R}^d$, the joint distribution function can be expressed as:*

$$H(x_1, .., x_j, ..x_d) = C(F_1(x_1), .., F_j(x_j), ..F_d(x_d)) \quad (6)$$

with associated density function h, expressed by the multiplication of the copula density function c and marginal densities:

$$h(x_1, ..., x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \times \prod_{k=1}^{d} f_k(x_k). \quad (7)$$

Conversely, Copula C corresponding to a multivariate distribution function G which marginal distribution functions F_i for i = 1,.., d, can be expressed as:

$$C(u_1.., u_d) = G(F_1^{-1}(u_1).., F_d^{-1}(u_d)), \forall (u_1, .., u_d) \in [0,1]^d \quad (8)$$

where $u_i = F_i(x_i)$ and F_i^{-1} is the inverse of the marginal distribution function of F_i.

The first equation of the Sklar's Theorem (Eq.6) describes the role of the Copula function which is connecting or coupling the marginal distribution functions F_1,..., F_d to form the multivariate distribution function H. This allows large flexibility in constructing statistical models by considering, separately, the univariate behavior of the components of a random vector and their dependence properties captured by some copulas. In particular, Copulas can serve for modeling situations where a different distribution is needed for each marginal, providing a valid substitute to several classical multivariate distribution functions such as Gaussian, Laplace, Gamma, Dirichlet, etc. This particularity represents one of the main advantages of the Copula's concept, as explained by Mikosch [13]: "[Copula] generate all multivariate distributions with flexible marginals".

Equation 8 describes the construction of the Copula that captures and estimates dependence between the standardized variables [3]. A typical example of this construction is the Gaussian Copula, which is obtained by taking G in (Eq.8) as the multivariate standard Gaussian d.f. This illustrates the founding principle of Copula that states that the dependence of data can be modeled independently from the marginals. It is thus possible to represent different original distributions just by changing the marginal distributions.

Real-world high dimensional data may have different marginals and joint distributions. Therefore, Copulas seem to be the right tools to overcome these difficulties.

4.2 The Invariance Principle Of Copula

Here, we would like to mention one of the principal properties of copulas inferred from *Sklar's* Theorem 4.1. This theorem is central for data generation using copula as it guarantees that the normalization applied on marginals by their respective cumulative distribution functions F, does not alter the measure of dependence between the variables that we want to capture with the copula.

Theorem 4.2 (Invariance Principle of Copula). *Let $X = (X_1, ..., X_j, ..., X_d)$ be a d-dimensional random vector with continuous joint distribution H, marginal distribution functions F_i , i = 1,..., d and a copula C. Let Tr_1,...,Tr_d be strictly increasing transformations on range X_1, .., X_d respectively. Then C is also the copula of the random variable $(Tr_1(X_1), ..., Tr_j(X_j), ..., Tr_d(X_d))$.*

Thus, Copulas, that describe the dependence of the components of a random vector, are invariant under increasing transformations of each variable. The power of this theorem manifests itself when moving from the multivariate distribution function (H) to the corresponding random vectors (X). In particular, when we want to sample from a multivariate distribution function. It gives us guarantees about dependency preservation when standardizing variables with their marginal distributions in order to capture dependency by taking Tr_i=F_i (cumulative distribution functions F_i are strictly increasing by construction). After that in order to return to the original data shape, we apply the inverse distribution F_i^{-1} (or the quasi-inverse) by taking $Tr_i = (F_i^{-1} o F_i)(x_i)$ is a strictly increasing transformation in the range of X_i.

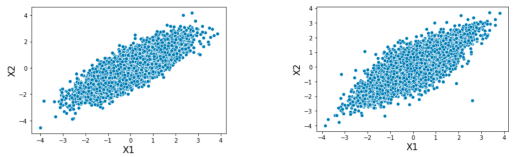
4.3 Families Of Copulas

In practice, there are many bivariate Copula families like the elliptical copulas, archimedean Copulas, and extreme-value Copulas [3], but only a few multivariate ones. This section focuses on the elliptical family because it contains two multivariate Copulas, the Gaussian Copula, and T-Copula.

4.3.1 Multivariate Gaussian Copula.

Definition 4.3 (Multivariate Gaussian Copula). The multivariate Gaussian Copula is the result of applying the inverse statement of Sklar's theorem (Eq.8) to the multivariate Gaussian distribution with zero mean vector and correlation matrix P.

The main drawback of Gaussian Copula is that it does not allow to capture tail dependence. The upper and the lower tail dependence coefficient between two variables (X_i, X_j) with correlation factor ρ, are the same and are given by [3]:



(a) Gaussian Copula. **(b) T-Student Copula .**

Figure 1: Comparison of Tail Dependency Capture by Gaussian Copula and T-Copula with Standard Gaussian Marginals and $\tau = 0.7$.

$$\lambda = \lim_{x \to \infty} 2\left(1 - \Phi\left(\frac{x\sqrt{1-\rho}}{\sqrt{1+\rho}}\right)\right) = 0. \tag{9}$$

4.3.2 Multivariate T-Copula.

Definition 4.4 (Multivariate T-Copula). The multivariate T-Copula yields from applying the inverse statement of Sklar's theorem (Eq.8) to the multivariate Student distribution.

In this case, considering (Eq.8), G corresponds to the multivariate T-Student d.f $T_d(\cdots ; P, \nu)$ with scale parameter matrix $P \in [-1, 1]^{d \times d}$ and ν>0 degree of freedom. Further T_ν^{-1} is the inverse of the univariate standard student c.d.f. T_ν. The main advantage of the T-Copula comparing to the Gaussian Copula is its ability to capture the tail dependence among extreme values [16]. The upper tail dependence coefficient λ_{ij}^{upper} between two variables (X_i, X_j) is equal to lower tail dependence coefficient λ_{ij}^{lower}, because T-Copula is symmetric and is given by:

$$\lambda_{ij} = 2T_{\nu+1}\left(-\sqrt{\nu+1}\frac{\sqrt{1-\rho_{ij}}}{\sqrt{1+\rho_{ij}}}\right). \tag{10}$$

4.3.3 Illustration. To compare T-Copula and Gaussian Copula's ability to capture tail dependence, Figure 1 shows two scatter plots that represent a bivariate distribution constructed using the two mentioned Copulas.

One important common characteristic in this comparison is that both Copulas use the Kendall's τ of two random variables (X_i, X_j) that has the same form for both T-Copula $C_{P,\nu}^t$ and Gaussian Copula C_P^Φ and it is defined by [4]:

$$\rho_{ij} = sin(\frac{\pi}{2}\tau(X_i, X_j)), \tag{11}$$

where ρ_{ij} is the Pearson correlation between the pair (X_i, X_j).

As we can notice from the lower left and upper right corners of the two scatter plots, the constructed bivariate distributions have significantly different behavior in their bivariate tails, although they have the same marginals and correlation factor. In fact, in the Gaussian Copula (left scatter), there seems to be no strong dependence in the lower left and upper right corners, while the T-Copula with three degrees of freedom (right scatter) emerges to have more mass and more structure in the lower and upper tail.

4.4 Copula Learning

Estimating Copula C as in (Eq.6) that belongs to a parametric family of Copulas C_θ such as the T and *Gaussian* Copula, consists in estimating the vector θ of unknown parameters. If the marginal distribution F_1, ...,F_d are known, the following sample would represent independent, identically distributed (*iid*) random samples of Copula.

$$U_i = (F_1(X_1), ..., F_d(X_d)), i \in \{1, ..., n\}. \tag{12}$$

Consequently, θ could be estimated using data distribution fitting techniques such as Maximum Likelihood Estimation (MLE). However, in reality, the marginals of H are unknown. For this reason, the marginals have to be estimated before that θ can be estimated. The Copula learning process, schematized in Figure 2, is structured in two steps – Marginal Distribution Fitting, and Copula Fitting – that are described in the following.

4.4.1 Marginal Distribution Fitting. Modeling marginal distribution F_1, ..., F_d can be achieved commonly in two ways [3, 4]: the first approach consists in fitting parametric distribution to each marginal, i.e., we assume $X_j \sim f_j(.;\gamma_j)$, the parameter γ_j is commonly estimated by maximum likelihood:

$$\hat{\gamma_j} := \underset{\gamma_j}{argmax} \prod_{i=1}^{n} f_j(x_{ij}; \gamma_j), j \in \{1, ..., d\}. \tag{13}$$

The associated marginal distribution function F_j is then estimated by $F_j(.; \hat{\gamma_j})$. The second approach consists of modeling the non-parametric marginals using the empirical distribution function $\hat{F_j}$ defined as:

$$\hat{F_j}(x) = \frac{1}{n+1}\sum_{i=1}^{n} 1_{\{x_i \leq x\}} \; for \; all \; x. \tag{14}$$

4.4.2 Copula Fitting. In both previous cases, we end up with data on the Copula scale, which will be used to estimate the Copula parameters θ of the chosen multivariate Copula family:

$$(u_{i1}, ..., u_{id}) = (\hat{F_1}(x_{i1}), .., \hat{F_d}(x_{id})), for \; i = 1, .., n. \tag{15}$$

Similar to marginal distribution parameters estimation, one method is Maximum likelihood estimation, which is commonly used to estimate the parameters vector θ of the Copula-based on pseudo-Copula data. If parametric marginal models (Eq.13) are used, then we talk about inference for marginals approach (IFM)[6] and if the empirical distribution of (Eq.14) is applied then we have a semi-parametric approach [5] also known as Canonical MLE (CMLE), and the likelihood function is given by:

$$\mathcal{L}(\theta | u_1, .., u_j, .., u_d) = \prod_{i=1}^{n} c(u_{i1}, .., u_{ij}, .., u_{id} | \theta). \tag{16}$$

The success of the first approach (IFM) depends on finding appropriate parametric models for the marginals. If the marginals are misidentified, the estimated parameter vector θ will be biased [7].

Finally, another simple method, called the method of moments, is based on the invariance property of Kendall's τ under strictly increasing transformations of the marginals. The method consists of calculating Kendall's τ for each bivariate marginal of the Copula and then using relationship in (Eq.11) to infer an estimate of the entire correlation matrix P of the considered elliptical Copula (Gaussian or T) [3].

In the case of T-Copula, to estimate the remaining parameter ν, MLE is generally used with correlation matrix held fixed [4].

5 PROBLEM FORMULATION

Our objective is, given a set of complex and representative observations (e.g. media channels with their user targets and respective daytime audiences) L_o, to generate a synthetic dataset L_s which is similar to the original dataset L_o under the following properties.

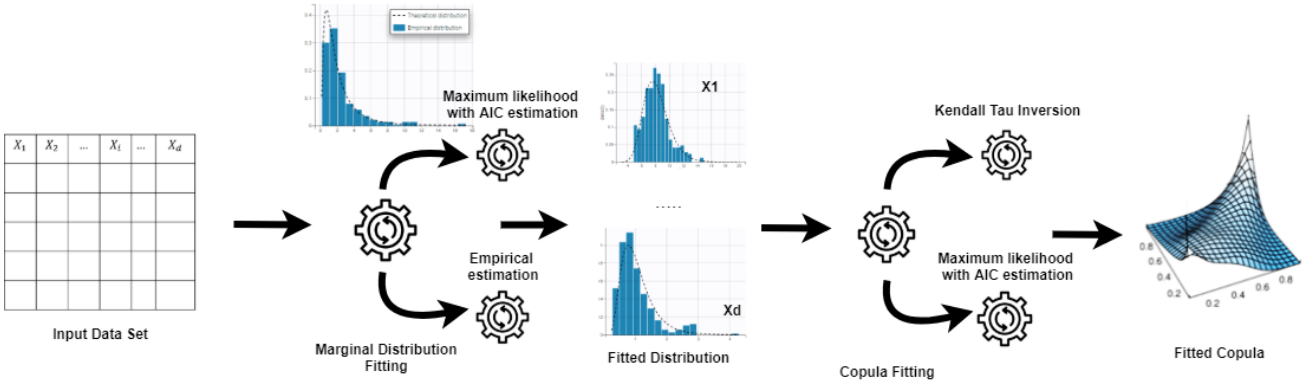


Figure 2: Copula Learning Process.

- For each attribute (variable) in the dataset, the generated values must be consistent with the distribution of the variable.
- Dependence between variables must remain the same in the new dataset.

This objective can be reformulated as: find automatically the statistical model that best fits the process of data generation. Therefore, using Copula and according to Section 4.4, this can be done by, first, estimating marginals parameters, and, second, estimating Copula distribution parameters. The fitting will almost never be exact, so the problem consists of determining the model parameters that minimize the relative amount of the lost information.

In the literature, the *Akaike Information Criterion* (AIC) [2] is often used to this extent, but not in the context of automatic determination of the best marginals or Copula models for data generation. Noticeably, AIC provides a trade-off between the goodness of fit and the model's simplicity by penalizing proportionally to the number of parameters. This, in turn, allows decreasing the risk of overfitting and underfitting at the same time. In what follows, we formulate our problem based on AIC without loss of generality as any other test could have been used, such as the *Kolmogorov-Smirnov* test, which does not penalize models with more parameters. Based on AIC, our synthetic data generation problem becomes the following two-steps optimization problem:

(1) Sampling values consistent with each variable behavior consists in finding the corresponding marginal distribution density function (f_j, γ_j) such that:

$$minimize\ AIC = 2k - 2\ln(\hat{\mathcal{L}}(\hat{\gamma_j}|x_j)),\ j = 1..d \tag{17}$$

where $\hat{\mathcal{L}}(\hat{\gamma_j}|x_j) = \prod_{i=1}^{n} f_j(x_{ij}|\hat{\gamma_j})$ represents the maximized likelihood function of a candidate marginal density f_j with k-dimensional vector of parameters $\hat{\gamma_j}$ given by:

$$\hat{\gamma_j} = \underset{\gamma_j}{argmax} \prod_{i=1}^{n} f_j(x_{ij}; \gamma_j). \tag{18}$$

(2) Characterizing the inter-dependency behavior of variables together consists in finding the joint distribution density (copula parameters) (h, θ) that:

$$minimize\ AIC = 2k - 2\ln(\hat{\mathcal{L}}(\hat{\theta}|x_1, .., x_j, .., x_d)) \tag{19}$$

where $\hat{\mathcal{L}}(\theta|x_1, .., x_j, .., x_d) = \prod_{i=1}^{n} h(x_1, .., x_j, .., x_d|\hat{\theta})$ is the ML estimation of the model h with parameters θ, and k is the number of parameters. $\hat{\theta}$ is given by:

$$\hat{\theta} = \underset{\theta}{argmax} \prod_{i=1}^{n} h(x_{i1}, ..., x_{ij}, ..., x_{id}; \theta). \tag{20}$$

6 SOLUTION DESCRIPTION

This section illustrates the general problem and describes its solution in the specific context of complex data generation with multivariate time series paired with categorical variables as found in our problem of media channel data generation. Our system, which is called MTCopula, is broken down into three steps: (1) data preparation, (2) copula model learning, and (3) synthetic data generation. Noticeably, only step (1) is specific to our problem, while steps (2) and (3) are entirely generic to any complex synthetic data generation scenario.

6.1 Data Preparation

6.1.1 General Pipeline. Copula, as a multivariate distribution function, requires a continuous representation of independent and identically distributed d-dimensional random variables. Due to this requirement, the multiple multivariate time series in the input must be preprocessed before learning the Copula model that, in a next step, generates synthetic data. Figure 3 illustrates the different steps of our data preparation process.

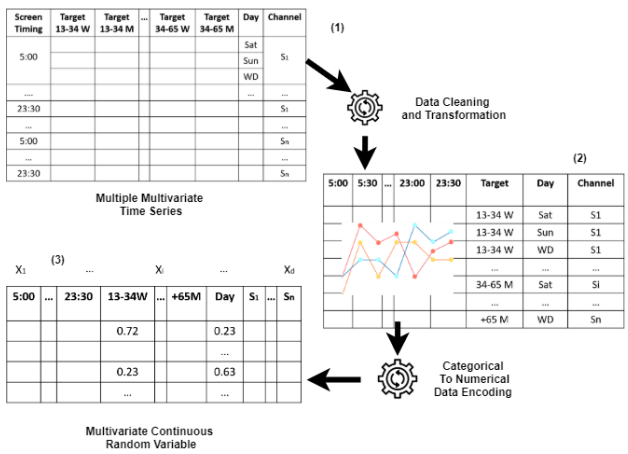


Figure 3: Data Preparation Workflow.

Our preprocessing process includes data cleaning, which consists of first removing missing values and normalizing data representation (ex. lower casing). Then, each column representation of the multivariate time series data is converted into a row representation of multiple time series. This allows to change the observation structure and, as a consequence, allows removing the dependence due to the time series nature where an observation at time t depends on previous time slots. In our case, the multivariate time series is defined by 6 time-dependent variables – $\{\text{Women}, \text{Men}\} \times \{13-34 \text{ years}, 34-65 \text{ years}, 65+ \text{ years}\}$ – and two categorical features – the media channel and the day of the week – as visible in the first table in Figure 3. Each one will produce a six-time series paired with a vector of three categorical variables (Target, Channel, and day). As a result of the preprocessing step, we have a set of independent and identically distributed observation defined by a vector of continuous and categorical (discrete) variables as shown in Figure 3.

6.1.2 Categorical variables encoding and Copulas. Categorical data cannot be modeled directly by the Copula, so we propose to replace them with continuous data. To this end, we consider two options. The first option consists of only considering distribution based encoding but fails to model the dependence between values of a categorical variable.

The second option consists of performing first a one-hot encoding to capture dependence between values of the same categorical variable. Applying this to the Target variable allows to model the multivariate dependence between the different values of this variable (women 13-34, men 13-34, women 34-65, men 34-65, women +65, men +65), and, as a consequence, models the multivariate time series behavior. The distribution-based encoding technique is used in order to transfer the discrete representation of the categorical variable to the continuous representation in the range $[0, 1]$. Figure 4 illustrates distribution based encoding technique using the Truncated Gaussian. This process gives dense areas at the center of each interval and ensures that the numbers are well differentiated. This facilitates the inverse process (decoding), given a value $v \in [0, 1]$, we can identify the corresponding category based on the value interval. Once the categorical variables are transformed, we have a set of observations of d-dimensional continuous random variables (Table 3 Figure 3). This dataset will be the input of the next step in order to estimate the copula parameters.

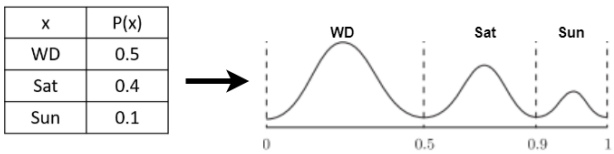


Figure 4: Categorical (Working Day, Saturday, Sunday) to Continuous Data Encoding using a Truncated Gaussian.

6.2 Copula Model Learning

As we explain in Section 4.4, the Copula learning process is done in two steps: the marginal distribution fitting and the Copula fitting.

6.2.1 Marginal distribution fitting. Our system proposes two methods to estimate the marginal distributions. The first one is non-parametric, via empirical distribution, as described in (Eq.14), and the second one is parametric and uses MLE (Eq.13). Algorithm 1 presents the steps of MLE to fit the marginals and, most importantly, AIC to automate the choice of the best marginal distribution among a set of preselected distributions. Currently, we choose, without loss of generality, among the following bounded distributions: Truncated Gaussian, GaussianKDE (Kernel Density Estimator), Beta, Truncated Exponential, and Uniform.

Algorithm 1: Marginal Distribution Fitting and Selection Using Maximum Likelihood and AIC.

Input: L_j dataset of n observation from the random variable X_j,
Output: the best fitted distribution F_j with estimated parameters $\hat{\gamma}_j$.

```
distributions = { Truncated Gaussian, GaussianKDE, Beta,
  Exponential, Uniform, or any bounded distribution } ;
best_aic = +∞;
for dist in distributions do
    fitted_params = Fit(dist, L_j, method ='maximum
      likelihood');
    aic = AIC(dist, fitted_params);
    if aic ≤ best_aic then
        best_aic = aic;
        γ̂_j = fitted_params;
        F_j = CDF(dist);
    end
end
```

The estimated marginal distributions are used to construct pseudo-Copula observations via the probability integral transformation as described in (Eq.15). A model selection criterion, such as AIC, is used to select the copula C that best fits pseudo-Copula data and characterizes dependence between marginals. Algorithm 2 presents the steps of Copulas fitting using AIC.

6.2.2 Copula fitting. Most of the works, done in synthetic data generation based on Copula, use a Gaussian copula with MLE approach to estimate marginals. Our system gives flexibility in terms of Copula model choice based on AIC, which, in turn, allows learning different Copula models and choose the model which best fits the input data. For the moment, we fit two models, Gaussian and T-Student Copula, as they are able to capture different dependence structures: linear like the correlation using Gaussian Copula, and non-linear behavior like the tail dependency using T-Copula.

Interestingly, our work addresses a recurrent problem observed when using Copulas: most contributions use Gaussian copula paired with a Pearson Correlation [10, 19] in order to estimate the correlation factor of the Gaussian Copula. However, the Pearson correlation factor is not invariant under strictly monotone non linear transformation, which may impact the process of estimation when standardizing with marginal distribution functions. Our contribution MTCopula uses the Kendall's τ inversion, which is based on the relationship between the Elliptical Copula (T-Copula or Gaussian Copula) correlation parameter and the Kendall's τ of two random variables (see Eq.11). For the T-Copula, another step is required to estimate the degrees of freedom, which is based on MLE with the correlation matrix held fixed.

6.3 Data Generation And Reconstruction

For synthetic data generation, copula samples are generated by sampling from the Copula density function c that corresponds

Algorithm 2: Copula Fitting with AIC.

Input: Dataset L of n observations from a d-dimensional vector X, a method m (e.g.: Kendall τ inversion) for parameters estimation and marginal distributions $F_1, \ldots, F_d$.
Output: the best fitted copula C with estimated parameters θ.

```
copulas = { Gaussian Copula, T-Copula } ;
best_aic = +∞;
copula_data = standardize(L, F_1,..., F_d);
for copula in copulas do
    fitted_params = Fit(copula, L, method=m);
    aic = AIC(copula, fitted_params);
    if aic ≤ best_aic then
        best_aic = aic;
        θ = fitted_params;
        C = copula;
    end
end
```

to the estimated Copula joint distribution function C. Then, the inverse probability transformation (F_j^{-1}) is applied to transform the Copula samples back to the natural distribution of the data (see Eq.8). Algorithm 3 presents the steps to sample based on Copula C and fitted marginal distributions $(F_1, F_2, \ldots, F_d)$.

Algorithm 3: Sampling Based On Copula

Input: Best Fitted Copula C with parameters vector θ, Fitted marginal distributions $(F_1, \hat{\gamma_1})$, $(F_2, \hat{\gamma_2})$, ..., $(F_d, \hat{\gamma_d})$.
Output: synthetic d-dimensional observation $\widetilde{X}$.

Sampling d-dimensional copula data U, $U \sim (\mathrm{c}, \theta)$;
Return $\widetilde{X} = (F_1^{-1}(U_1, \hat{\gamma_1}), F_2^{-1}(U_2, \hat{\gamma_2}), ..., F_d^{-1}(U_d, \hat{\gamma_d}))$;

For the moment, our system MTCopula supports two Copula models: Gaussian and T-Copula. For generating correlated random variables, our method uses the Cholesky factorization, which is commonly used in Monte Carlo simulation to produce efficient estimates of simulated values [30].

Once the synthetic data generation process is finished, a reconstruction operation is performed in order to re-convert the categorical variable to its original representation by replacing interval values with their corresponding, most likely, categories. Finally, the row representation of the time series is re-transformed into a column representation.

7 EXPERIMENTS

In this section, we report the experiments that were conducted to validate MTCopula ability to generate synthetic data[2]. In order to evaluate our approach, we answer the following research questions:

(1) MTCopula relies on the central hypothesis that Copulas are pertinent to generate synthetic data. To confirm it, we propose experiments where state-of-the-art generators (ITS, GADP, MLE, and CMLE) are compared with different Gaussian Copulas and T-Copula. As a Gaussian Copula is defined by its correlation matrix to model dependency, our test incorporates several ways to estimate this correlation matrix: Kendall's τ, Pearson and Spearman coefficients. In conclusion, this experiment validates the choices of both Copula and the Kendall's τ.
(2) The main bottleneck of methods based on Copula is (i) to be able to choose among the marginal models, and (ii) to choose among the Copula models that may have different properties to capture the dependency. MTCopula automatises the process by using the AIC criterion as a measure to automatically determine the best model either for marginals or Copula. We show to which extent this choice is efficient in our context.
(3) Finally, to answer the first question raised in this paper, we show the efficiency of MTCopula to generate multiple/multivariate time series based on our initial real industrial use case on media planning and synthetic media channels data generation.

For our experiments, we use the 4 datasets presented in Table 1. The XYZ dataset was generated using a mixture of Beta and Gaussian distributions with a correlation between Y and Z only, in order to simulate complex marginal distributions. The Abalone and Breast Cancer Wisconsin datasets come from the UCI dataset platform [3]. The AdWanted dataset[4] comes from Adwanted Group company and provides a rich and real use case for our approach based on media channels. For this specific dataset, the input data, which is 27000 instances in 10 dimensions, is first preprocessed following the methodology presented in Section 6.1 for Copula model learning. This produces a multivariate continuous data set with 1440 instances of 60 dimensions that we use in our tests.

Dataset	Type	Number Attributes	Attribute Characteristics	Number Instances
XYZ	Multivariate	3	Continuous	1000
Abalone	Multivariate	8	Continuous, Discret, Categorical	4177
Breast Cancer Wisconsin (Diagnostic)	Multivarate	32	Continuous, Categorical	569
AdWanted	Multivarate Timeseries	60	Continuous, Categorical	1440

Table 1: Datasets Used For Experiments.

7.1 Copula For Synthetic Data Generation

This section evaluates to which extent Copula models answer our need to generate synthetic datasets that fit with our two objectives presented in Section 5.

7.1.1 Copula versus other state-of-art generators. We first evaluate the ability of the Copula framework to generate synthetic data that better preserve dependency structure when compared to the following state-of-the-art approaches: ITS [17], GADP [14], MLE and CMLE [5]. In order to show the Copula framework efficiency, we couple different marginals by changing the copula itself: either T-Copula or Gaussian copula. For the Gaussian Copula, we use different methods to estimate the correlation matrix P: Gaussian Copula with Kendall's τ (GCK), Gaussian Copula with Spearman (GCS), and Gaussian Copula with Pearson (GCP).

We evaluate, on our four datasets, the dependence structure preservation based on the Root Mean Square Error (RMSE) between the correlation matrix of the original dataset and the generated dataset. The lower the RMSE, the better the dependency structure is captured. The final reported errors, presented in Table 2, are averaged over **50 runs**, except for MLE and CMLE due to their time computation costs on the three most complex

[2]The source codes are available at https://github.com/cderunz/MTCopula.

[3]https://archive.ics.uci.edu/ml/datasets.php

[4]The AdWanted dataset is not shareable due to privacy issues

	XYZ dataset		Breast Cancer WD		Abalone dataset		AdWanted dataset	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
ITS[17]	0.4465	0.0155	0.4725	0.0018	0.7237	0.0027	0.3447	0.037
GADP[14]	0.1659	0.0137	0.2392	0.0397	0.2855	0.0169	0.2482	0.0224
MLE	0.4456	0.0147	0.4734	-	0.7266	-	0.8953	-
CMLE[5]	0.1735	0.0132	0.4698	-	0.7120	-	0.8671	-
GCP	0.1639	0.0159	0.0794	0.0059	0.0571	0.0217	0.1057	0.0028
GCS	0.1579	0.0114	0.0785	0.0052	0.0554	0.0225	0.0982	0.0017
GCK	0.1451	0.0105	0.0658	0.0053	0.0547	0.0161	0.0931	0.002
TC	0.1596	0.0111	0.0993	0.0095	0.0315	0.0091	0.0881	0.0005

Table 2: RMSE Evaluation of Dependency Structure Preservation using Different Methods: GCx denotes Gaussian Copula and x indicates the correlation (p: Pearson, s: Spearman, k: Kendall). TC denotes the T-Student Copula.

dataset. We observe clearly that the dependency structure is better respected with Copulas than with state-of-the-art approaches. For instance, on the Breast Cancer Wisconsin Dataset, the mean RMSE of ITS, GADP, MLE, and CMLE are higher than 0.2 when it is lower than 0.1 for any type of Copulas.

7.1.2 Choice of Dependency Structure Estimation Method. In order to validate our choice that Kendall's τ is relevant and accurate to estimate and preserve dependency structure, we compare several methods to estimate the correlation matrix P of the Gaussian Copula: Kendall, Spearman, and Pearson. Noticeably, we limit our study to Copula whose dependency structure $\mathcal{D}$ is expressed as a correlation matrix.

From Table 2, we can observe that Kendall, Spearman, and Pearson methods, for which the *RMSE* median is between 0.01 to 0.2 depending on the dataset, are significantly more accurate than RMSE scores for ITS, GADP, MLE, CMLE methods for which the means are respectively between 0.34 and 0.72, 0.16 and 0.28, 0.44 and 0.89, and between 0.17 and 0.86. We can also observe that the Gaussian Copula with Kendall performs slightly better than the Gaussian Copulas with both Pearson and Spearman.

These results illustrate the robustness and the effectiveness of Kendall's method against the others method for correlation matrix estimation in the specific case of Gaussian copula. Therefore, our choice of Kendall's τ to capture the dependency structure is validated both experimentally and theoretically, as illustrated before in Section 3. The dependency structure estimation method choice is thus confirmed.

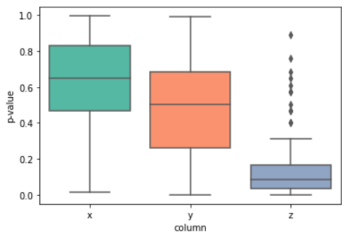
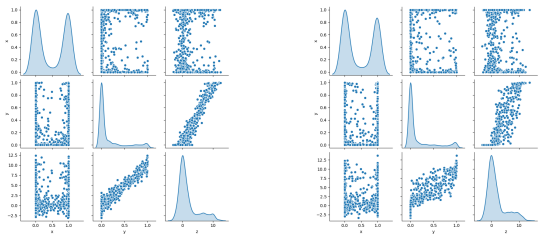


Figure 5: Marginals Fitting Evaluation Using Two Sided Kolmogorov-Smirnov Test with $\alpha = 0.05$ on XYZ dataset.

7.1.3 Impact of the marginal fitting on the quality of data generation. Figure 5 illustrates a box plot for the variation of the P-Value of the two 2-Samples Kolmogorov-Smirnov Test, which determines whether the synthetic attributes values and the real attributes values are derived from the same distribution. We notice that for the first 2 variables X and Y, the median P-value (resp. ≈ 0.65 and ≈ 0.50), are above the threshold $\alpha = 0.05$, so we cannot reject the null hypothesis, that the synthetic and the real marginals are derived from the same distribution. Although the median P-value of Z (≈ 0.09) is also slightly larger than α, it is significantly less accurate than the others. This is due to problems with the marginal fitting of this distribution. As a consequence, correlation is impacted between Y and Z as visible in Figure 6. The Figure 6 shows that globally data generation using Copula with structure dependency capture is able to answer our problem, but the better we fit both marginals and Copula, the more realistic the generated data are. As a consequence, our problem boils down to selecting the most effective marginals and Copula models to generate the most realistic data. That is the goal of our approach MTCopula, that relies on AIC as described in the next section.



(a) Real Data **(b) Synthetic Data.**

Figure 6: Pair Plot of XYZ Dataset and Synthetic Data Generated Using Kendall Method.

7.2 Interest Of AIC For Models Selection

This section presents the benefits of using AIC to determine the best model for both marginal fitting and copula choices.

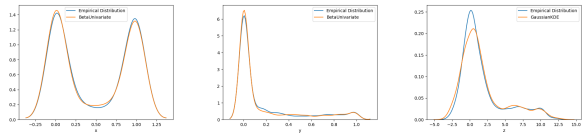
7.2.1 Choice of the marginals. To evaluate the importance of AIC in selecting the most appropriate marginal distribution that best fits the behavior of marginal variables, we fit a list of bounded distributions: Beta distribution, Uniform distribution, Truncated Exponential, Truncated Gaussian, and Kernel density estimation, using the MLE method for each variable. For each of these distributions, we evaluate the AIC using the fitted parameters. The distribution with the minimum value of AIC is selected to model the behavior of the variable. Note that we use a list of bounded distribution in order to avoid generating outliers. In addition, we incorporate a Kernel density estimation algorithm to fit more complex distribution shapes. Table 3 illustrates the evaluation of AIC of the marginal distributions fitting of XYZ dataset variables.

From Table 3 we can observe that for both X and Y variables. Beta distribution has a very small value of AIC (-11718.86 and -11001.61 respectively). As a consequence, we notice that the real data distribution (blue color in Figure 7) and the fitted distribution (orange color Figure 7) are almost identical (see Figures 7a and 7b). While, for the variable Z, the value of the minimum AIC is not as small (4435.44) compared to the other variables. As a result, we observe a significant difference between the fitted and the real data distribution in Figure 7c. This is because AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model.

7.2.2 Choice of the copula models. In this experiment, we investigate the impact of the copula model choice on the quality

Variable	Beta	KDE	Uniform	Truncated Exponential	Truncated Gaussian
X	-11718.86	-281.47	4.0	98.99	133.62
Y	-11001.61	-1116.15	3.96	-1497.21	-690.05
Z	240273.73	4435.44	5480.97	5040.59	4896.43

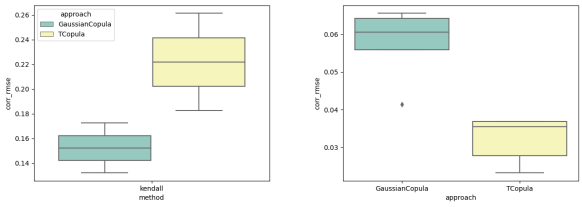
Table 3: AIC Evaluation on XYZ Dataset Marginals.



(a) X Fitting using Beta distribution **(b) Y Fitting using Beta distribution** **(c) Z Fitting using KDE distribution**

Figure 7: Marginal Distribution Obtained After Fitting Using Algorithm 1.

of data generation, and we demonstrate the importance of AIC to choose the best copula model. To this end, we fit two copulas models, the Gaussian and the T-Copula, on two different datasets XYZ and Abalone. For both models, we use the Kendall method to estimate the correlation matrix P. The degree of freedom ν of T-Copula is estimated by the CMLE method with correlation matrix P held fixed. Results are averaged after **10 runs**. Figure 8 illustrates the RMSE evaluation of the dependency preservation using the two copulas.



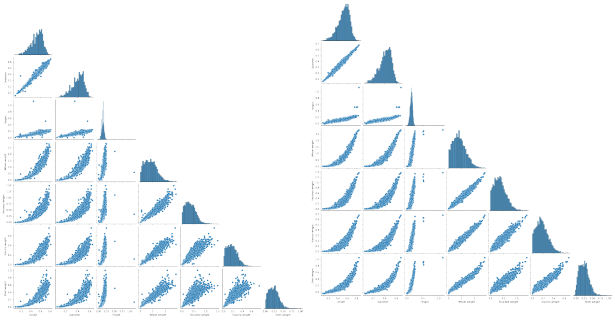
(a) RMSE Variation XYZ **(b) RMSE Variation Abalone**

Figure 8: Dependency Structure Preservation Evaluation using different Copula Models.

From Figure 8a), we can observe that, for XYZ dataset, the Gaussian Copula performs better than the T-Copula. On the other side, as shown in Figure 8b, T-Copula outperforms the Gaussian Copula on Abalone dataset. This is because XYZ dataset does not expose a tail dependence structure (see Figure 6a). Consequently, the use of T-Copula will impact the correlation matrix (see eq. 10) by considering dependencies in the tails that do not appear in original data. Conversely, Abalone dataset shows a lower tail dependence structure as illustrated in Figure 9a. As a result, using a T-Copula for data generation will correct the dependencies in tails, while it is not the case with the Gaussian Copula. For the moment, we use the T-Copula only for tail dependence modeling, which has a symmetric tail structure, the reason for which, we do not control the upper tail structure in the generated synthetic data as shown in Figure 9b. Results in Table 4 confirm those conclusions. For XYZ dataset, the *min AIC* that best fits the data corresponds to the Gaussian Copula (3993.73). On the other hand, the T-Copula has the minimal value of *AIC* that best fits Abalone dataset (9507.26). This confirms the AIC interest in choosing the best copula model that best fits the data generation process.

Database	Copula Model	AIC Value
XYZ	Gaussian	3993.73
	T-Student	3998.18
Abalone	Gaussian	12388.88
	T-Student	9507.26
AdWanted	Gaussian	202532.88
	T-Student	127444.74

Table 4: AIC Evaluation of Gaussian and T-Copula Models.



(a) Real Data. **(b) Synthetic Data using T-Copula.**

Figure 9: Pair Plot illustration of Abalone Dataset.

Through this section, we have demonstrated the effectiveness of MTCopula to select among different combinations of marginal fittings and Copula models, the most appropriate models that best represent the process of data generation, and we showed the importance and the relevance of the AIC criterion in this process.

7.3 MTCopula Applied To Media Channels

The objective of this experimentation is to measure the effectiveness of MTCopula on real media dataset as provided by *Adwanted* company. According to Table 4, as AIC for T-Copula ($127k$) is lower than AIC for the Gaussian Copula ($202k$), MTCopula is capable to automatically select the T-Copula for this dataset to sample synthetic multivariate time series. These data will be used in the following experiments to evaluate the business-related qualities of the generated data. The results, in terms of RMSE, presented in Table 1, confirm this choice, as T-Copula obtain a slightly better performance: ≈ 0.088 with standard deviation ≈ 0.0005 for T-Copula and ≈ 0.093 with standard deviation ≈ 0.002 for Gaussian Copula with Kendall's τ.

To study the utility of the generated time series, we compare each time series in the generated dataset with its counterpart from the same target user category, the same day in the week, and the same channel in the real data set. For each pair, we measure the MAE variation of the statistical properties of time series, respectively the Min, Max, Mean, Median, Standard deviation, and 95 Percentile. Figure 10 shows the MAE of those measures. From this Figure, we can observe an overall variation smaller than 0.2, which is a very good result as it is significantly smaller than the observed standard deviation of those statistics in the original dataset (respectively ≈ 1.66, ≈ 0.54, ≈ 0.46, ≈ 0.44, and

1.44). Noticeably, for the min statistic, because we have a standard deviation ≈ 0.2, this result reflects the ability of MTCopula to preserve the time series's characteristics when generating synthetic data. This overall good business-related performance gives guarantees on the utility of the synthetic time series in several situations when access to the real data is not possible.

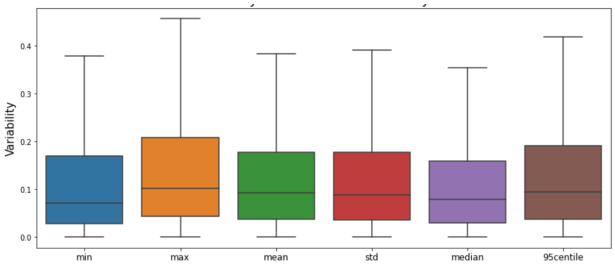


Figure 10: MAE Variation of Synthetic Time Series statistics.

8 CONCLUSION

This paper proposed MTCopula a flexible, extendable, and generic solution for synthetic complex data generation. It incorporates different Copula models (for the moment Gaussian and T-Copula) in order to capture different dependency structures including tail dependence. To bypass the non invariance problem of Pearson-Correlation based Copula methods, MTCopula involves Kendall τ, which is robust to outliers and invariant under strictly monotone transformations. This ensures dependency preservation during the process of copula learning. Unlike the GADP approach that uses only the Gaussian distribution to model the marginals, our solution incorporates a variety of bounded distribution in order to best fit the behavior of variables and do not generate outliers. In addition, MTCopula is less restrictive in terms of the quantity of the input data and is more explainable than GANs. MTCopula is able to automatically select both the univariate marginal distributions and the copula model that best fit the input data. For that, it uses MLE to fit the possible marginal distribution model, and then AIC to choose both the best distribution and the best Copula Model between the T-Copula and the Gaussian one. MTCopula handles multiple data types including complex tabular datasets and multiple/multivariate time series. The proposed experiments show MTCopula's interest and efficiency compared to existing methods.

In our future works, first, further experiments will be conducted to evaluate (*i*) the sensitivity of MTCopula to the number of parameters it has to fit to correctly estimate the marginals or Copula models, by varying the number and the nature of the variables, (*ii*) how it deals with asymmetric tail dependency behaviors as this problem is still open in MTCopula. Second, we will work on making our approach robust to missing values in the original datasets. Third, we plan to study the use of synthetic data for machine learning model fitting, in order to see how qualitative is the new data for different tasks. Fourth, an important way to see how much using MTCopula could be interesting for machine learning tasks is also to analyze its scalability according to the number of original and generated data. Fifth, we want to tackle a new research problem: how can MTCopula efficiently consider conditional dependencies between variables. Using Vine Copula seems to be a promising solution that we need to study.

9 ACKNOWLEDGMENTS

This work is funded by the ANRT CIFRE Program (2019/0877).

REFERENCES

[1] Ruzanna Ab Razak and Noriszura Ismail. 2019. Dependence Modeling and Portfolio Risk Estimation using GARCH-Copula Approach. *Sains Malaysiana* 48, 7 (2019), 1547–1555.

[2] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE transactions on automatic control* 19, 6 (1974), 716–723.

[3] Claudia Czado. 2019. Analyzing Dependent Data with Vine Copulas. *Lecture Notes in Statistics, Springer* (2019).

[4] Stefano Demarta and Alexander J McNeil. 2005. The t copula and related copulas. *International statistical review* 73, 1 (2005), 111–129.

[5] Christian Genest, Kilani Ghoudi, and L-P Rivest. 1995. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika* 82, 3 (1995), 543–552.

[6] Harry Joe. 2005. Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of multivariate Analysis* 94, 2 (2005), 401–419.

[7] Harry Joe. 2014. *Dependence modeling with copulas.* CRC press.

[8] Samuel Kotz and Saralees Nadarajah. 2000. *Extreme value distributions: theory and applications.* World Scientific.

[9] Dorota Kurowicka and Roger M Cooke. 2006. *Uncertainty analysis with high dimensional dependence modelling.* John Wiley & Sons.

[10] Zheng Li, Yue Zhao, and Jialin Fu. 2020. SYNC: A Copula based Framework for Generating Synthetic Data from Aggregated Sources. *arXiv preprint arXiv:2009.09471* (2020).

[11] Donald MacKenzie and Taylor Spears. 2014. 'The formula that killed Wall Street': The Gaussian copula and modelling practices in investment banking. *Social Studies of Science* 44, 3 (2014), 393–417.

[12] Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. 2015. *Quantitative risk management: concepts, techniques and tools-revised edition.* Princeton university press.

[13] Thomas Mikosch. 2006. Copulas: Tales and facts. *Extremes* 9, 1 (2006), 3–20.

[14] Krishnamurty Muralidhar, Rahul Parsa, and Rathindra Sarathy. 1999. A general additive data perturbation method for database security. *management science* 45, 10 (1999), 1399–1415.

[15] Roger B Nelsen. 2007. *An introduction to copulas.* Springer Science & Business Media.

[16] Aristidis K Nikoloulopoulos, Harry Joe, and Haijun Li. 2009. Extreme value properties of multivariate t copulas. *Extremes* 12, 2 (2009), 129–148.

[17] Sheehan Olver and Alex Townsend. 2013. Fast inverse transform sampling in one and two dimensions. *arXiv preprint arXiv:1307.1223* (2013).

[18] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data Synthesis Based on Generative Adversarial Networks. *Proc. VLDB Endow.* 11, 10 (2018), 1071–1083.

[19] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA).* IEEE, 399–410.

[20] Andrew Patton. 2013. Copula methods for forecasting multivariate time series. In *Handbook of economic forecasting.* Vol. 2. Elsevier, 899–960.

[21] L. Petricioli, L. Humski, M. Vranić, and D. Pintar. 2020. Data Set Synthesis Based on Known Correlations and Distributions for Expanded Social Graph Generation. *IEEE Access* 8 (2020), 33013–33022.

[22] Stéphanie Portet. 2020. A primer on model selection using the Akaike information criterion. *Infectious Disease Modelling* 5 (2020), 111–128.

[23] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. 2017. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems.* 3236–3246.

[24] Jerome P Reiter, Quanli Wang, and Biyuan Zhang. 2014. Bayesian estimation of disclosure risks for multiply imputed, synthetic data. *Journal of Privacy and Confidentiality* 6, 1 (2014).

[25] Marko Robnik-Šikonja. 2015. Data generators for learning systems based on RBF networks. *IEEE transactions on neural networks and learning systems* 27, 5 (2015), 926–938.

[26] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. 2019. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems.* 6827–6837.

[27] Francesco Serinaldi. 2008. Analysis of inter-gauge dependence by Kendall's τ K, upper tail dependence coefficient, and 2-copulas with application to rainfall fields. *Stochastic Environmental Research and Risk Assessment* 22, 6 (2008), 671–688.

[28] M Sklar. 1959. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris* 8 (1959), 229–231.

[29] Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. 2019. Copulas as High-Dimensional Generative Models: Vine Copula Autoencoders. In *Advances in Neural Information Processing Systems.* 6528–6540.

[30] Honggang Zhu, LM Zhang, Te Xiao, and XY Li. 2017. Generation of multivariate cross-correlated geotechnical random fields. *Computers and Geotechnics* 86 (2017), 95–107.