Mohamed Khider University, Biskra Faculty of Exact Science, Science of Nature and Life
Departement of Mathematics

PhD Thesis

Presented to obtain
The Doctorate Degree In Mathematics

OPTION : Statistics

By
KARIMA FEMMAM

Contribution On the Estimation of the Copulas Parameters

Publicly defended, In 26/10/2023, front of the jury members:

Pr. SAYAH Abdallah	**University of Mohamed Khider, Biskra**	**President**
Pr. Brahim Brahimi	**University of Mohamed Khider, Biskra**	**Supervisor**
Dr. Jihane Abdelli	**University of Mohamed Khider, Biskra**	**Co-supervisor**
Pr. Benatia Fatah	**University of Mohamed Khider, Biskra**	**Examiner**
Pr. Terissa Labib Sadok	**University of Mohamed Khider, Biskra**	**Examiner**
Pr. Aissaoui Adel	**University of Echahid Hamma Lakhdar, El Oued**	**Examiner**

AKNOWLEDGEMNT

I would like to express my sincere gratitude to my primary supervisor, Pr. Brahim Brahimi and Pr. Smain Femmam, for their guidance, support, and mentorship throughout my PhD program.

I would like to express my sincere appreciation to the members of the thesis comittee for their time, expertise, and critical evaluation of my work.

I would also like to extend my thanks to my family for their love, support, and encouragement during this long and challenging journey.

Lastly, I am deeply grateful to my colleagues, friends, and mentors who have provided me with valuable feedback, advice, and inspiration.

Dedication

To my beloved parents, their sacrifices and unwavering support have made this accomplishment possible.

To the soul of my aunt,.

To my dear sisters, their love, patience, and understanding have been my greatest comfort.

To my friends and colleagues who cheered me on during the long hours of research and writing.

Thank you, everyone, for your role in this journey.

Publications based on this thesis

A K. Femmam, B. Brahimi, and S. Femmam, Springer "**An optimized feature selection technique based on bivariate copulas "GBCFS"**", Springer, publication in Journal of Combinatorial Optimization. (2023) 45:74 https://doi.org/10.1007/s10878-023-01006-9, https://www.springer.com/journal/10878

B K. Femmam and S. Femmam, "**Improving the dimensionality reduction of PCA using bivariate copulas**", Advances and Applications in Statistics, Volume 86, Issue 1, Pages 47 - 64 (March 2023) http://dx.doi.org/10.17654/0972361723015, http://www.pphmj.com/abstract/14752.htm

Conference papers and awards based on this thesis

- K. Femmam and S. Femmam, "**Fast and Efficient Feature Selection Method Using Bivariate Copulas**", *The 14th International Conference on Computer Science and Information Technology, ICCSIT 2021* in Paris, France. October 15-17, 2021, iccsit.org/index.html. In proceeding of ICCSIT 2021.
- K. Femmam, **Reducing the Dimension of Big Data Using Multivariate Copulas**. *The 1st International Conference on Innovative Academic Studies, ICIAS'2022*. Konya-Turkey. September 10-13, 2022.
- K. Femmam, **Efficient dimensionality reduction based on bivariate copulas**. *The 2nd National Conference on Pure and Applied Mathematics, NCPAM'2022*. Laghouat, Algeria. December 18-19, 2022.
- K. Femmam, **Reducing the Dimensions of Big Data using Bivariate Copulas**. *Study Days in Mathematics*. Biskra, Algeria. December 12–14, 2022.

TABLE OF CONTENTS

List of Tables

LIST OF FIGURES

ACRONYMS

AB AdaBoost. 44

ALDE Angle Linear Discriminant Embedding. 27

ANN Artificial Neural Network. 31

BAKPCA Block Adaptive Kernel Principal Component Analysis. 29

BCFS Bivariate copulas based Feature Selection. 3

BCFS-PCA Bivariate copulas based Feature Selection based Principal Component Analysis. 3

CDF Cumulative Distribution Function. 16

CSVD Constrained Singular Value Decomposition. 26

FDA Fisher Discriminant Analysis. 27

FFT Fast Fourier Transformation. 24

FS Forward Selection. 29

GBCFS Grouped Bivariate copulas based Feature Selection. 3

GPCA Generalized Principal Component Analysis. 27

ICA Independent Component Analysis. 28

IsoMap Isometric Mapping. 29

k-NN k-Nearest Neighbors. 31

KPCA Kernel Principle Component Analysis. 3

LARS Least Angle Regression. 30

LASSO Least Absolute Shrinkage and Selection Operator. 30

LDA Linear discriminant analysis. 27

LL-RDA Regularized Linear Regression Discriminant Analysis. 28

LLE Locally Linear Embedding. 29

LPCA Local Principal Component Analysis. 27

LPP Local Preserving Projections. 28

LS Least Square. 30

LU Lower Upper. 31

LU-C LU based Copulas. 31

MC-PCA Multivariate Copulas based Principal Component Analysis. 31

NB Naive Baysian. 31

NMMP Neighborhood Min-Max Projection. 27

PCA Principle Component Analysis. 3

PCs Principal Components. 26

PDF Probability Density Function. 14

RF Random Forest. 31

S2R Sparse Subspace Representation-based Dimensionality Reduction. 30

SB Sequential Backward. 29

SKPCA Subset Kernel Principal Component Analysis. 29

SPCA Sparse Principal Component Analysis. 27

SSSP Small Sample Size Problem. 27

SVD Singular Value Decomposition. 3

SW StepWise. 30

NOTION

$C^t_{\nu,\theta}$ Student Copula. 17

C^C_θ Clayton Copula. 19

C^F_θ Frank Copula. 19

C^{Ga}_θ Gaussian Copula. 15

C^G_θ Gumbel Copula. 19

C Bivariate Copula. 6

I Unit segment. 5

M Maximu Copula. 8

$T_{\nu,\theta}$ Bivariate cumulative student distribution. 17

W Minimum Copula. 8

Γ Gamma function. 14

Φ_θ Bivariate cumulative normal distribution. 15

Φ Multivariate normal density distribution. 14

Π Product Copula. 8

$\hat{C}$ Empirical Copula. 22

λ_L Lower tail dependence. 12

λ_U Upper tail dependence. 12

$\overline{C}$ Survival Copula. 7

ρ_s Spearman's rho. 9

ρ Pearson's correlation coefficient. 8

τ Kendall's tau. 9

φ Archimedean copula's generator. 19

t_ν Multivariate student density distribution. 14

General Introduction

This thesis is dedicated to the development and application of a Copula-based approach for modeling multi-dimensional dependence in data. The theory of Copula has generated enormous interest among researchers in various scientific fields. It is widely used in the study of multidimensional data since it allows for easier estimation and description of the distribution of random variables by estimating margins and Copulas separately. Our goal is to examine the benefits of using Copulas in the field of data mining, specifically when it comes to handling large data.

Problematic

Every Field has its hands wet with data, in the search of optimization and efficiency, but most of the time useless and redundant observations are added to datasets, which can increase the time to treat them, while also add inaccuracy to them, thus storage capacity growth almost exponentially. This growth does not necessary indicate a gain of more information, since the accumulated data is frequently poor quality, crude and contains irrelevant attributes. The presence of redundancy in the datasets is often inconvenient in algorithm when dealing with data and modeling methods, since it results in noise and misspecification for models parameters, which results in low accuracy and bad performance of the model. The collection of these numerous data is increasing in quantity and being expressed in more and larger dimensions. Besides, redundant information is present in many attributes and may have an impact on other variables that are potentially relevant and less present. In this case, important information risk of being drowned among many attributes that all express the same idea of no interest to the user and its extraction is only possible if the original data is cleaned and prepared.

Recent years have shown that authors are interested in using stochastic models in problem solving for data pre-processing issues. Theoretically, most of these models call for certain assumptions about the dependencies between the data attributes. For that, a knowledge of the joint probability distribution is required in order to model the dependency structure between these variables. In this light, a number of methods have been put out in the scientific literature, some of which have complex mathematical complications and less precised results.

Motivations

As we have seen in the previous section, data pre-processing became a must, since most of data contain noise and irrelevant attribute. For that, we are interested in reducing the dimensions of large data using stochastic process and mathematical operations in this thesis. Dimensionality Reduction methods are divided into two major types, feature selection and feature extraction. Feature selection is a technique that clears and reduces redundancy by selecting only relevant attributes from the data. While feature extraction is a technique that extracts important information from the data by projecting the high dimensional data into a lower dimensional subspace called the principal subspace. In this thesis, we developed two feature selection methods and one feature extraction method, our goal is to suggest and investigate the possibility of producing high-quality, multidimensional sample that capture only important information and relevant attributes that hold the same information and statistical behavior as the original data. Therefore, the main objective of this thesis are giving as follow:

- Detect inter-correlation and eliminate redundancy in multi-dimensional data.
- estimate the joint distribution function without the need to impose hypothesis about the marginal distributions of the attributes.
- Develop new efficient methods to reduce the dimension of data using Copulas in the field of Dimensionality Reduction.
- Utilise Copulas as a tool to detect irrelevant variables and eliminate it.
- Establish fast calculation methods for Dimensionality Reduction techniques.

Numerous experimental studies have demonstrated the shortcomings of several algorithms, since it cannot handle treating extremely large volumes of complex and diverse data, such as real-world data sets. As a result, statistical techniques are unable to overcome the numerical challenges posed by the presence of irrelevant or redundant variables. In other words, the more data are large, the hardest is to extract important information from it. Therefore, such issue is managed by reducing the dimensions of these data. As a consequence, only relevant attributes are selected for experimental studies. For that, Dimensionality Reduction intend to:

- Clean the data and prepare it for machine learning and modeling.
- Improve models performance by increasing the accuracy and the efficiency of the models.
- Reduce the dimensions of large data and reduce the memory complexity in the machine.

Hence, Dimensionality Reduction is an important step in data pre-processing and a must for preparing the data to be modeled.

Contributions

The contribution of this thesis is to develop new feature selection and extraction techniques as a step of pre-processing, cleaning the data and preparing it to be modeled. The three methods are based on Copulas distribution function, since Copulas help us to estimate the joint bivariate probability distribution function without the need to estimate the marginals distributions, and the fact that Copulas enable us to detect the inter-correlation and eliminate it. All the proposed techniques are compared against well known methods, in term of reduction and computational time, and also in term of efficiency by fitting the obtained results of reduction of each method to several models and compare the accuracy of these models. A short description of each method is as follow:

- **BCFS**: the first developed method named bivariate Copulas based Feature Selection (**BCFS**) consists on eliminating the redundancy using correlation, we say that two attributes are redundant if they hold the same information. In other words, if two attributes are correlated we say that one of them is redundant and we eliminate one of these attributes. This correlation is detected using bivariate Copulas and the elimination between the two attributes is random. The proposed algorithm on the other hand is characterised with its fast computational time, where the time complexity is given by O($m \times nlogn$). The method is simulated using real-world datasets, and showed a good performance against well known methods in term of reduction and accuracy of the models.

- **GBCFS**: the second developed method is an improvement of the first proposed technique by treating the issue of the random elimination in **BCFS**. For that, a grouping technique of the correlated attributes is proposed. This technique help us when it comes to choosing which attributes to eliminate and to select for more reduction and reducing redundancy. The technique showed good results against **BCFS** and other feature selection techniques in term of reduction, computational time and accuracy of the models for each obtained reduced data.

- **BCFS-PCA**: the last developed technique in this thesis is based on the **BCFS** and the most used feature extraction technique the Principal Component Analysis (**PCA**). The method is built under two stages. In the first stage, **BCFS** technique is applied to reduce redundancy. While in the second stage, the **PCA** method is performed on the obtained subset of the first stage. The intention of this method is to improve the performance of **PCA** reduction and information extraction. This technique is compared against the baseline method **PCA**, **SVD**, **KPCA** and another method that combines multivariate Copulas and **PCA** using real-world datasets, and the classification accuracy of the new reduced data.

Thesis Organisation

This thesis is constructed of two main parts, where the theoretical framework used in our research is giving in two chapters in the first part. While the contribution and the practical work is giving in format of two chapters in the second part of the thesis. Figure 0.1 shows how the manuscript is organised and the link between the chapters.

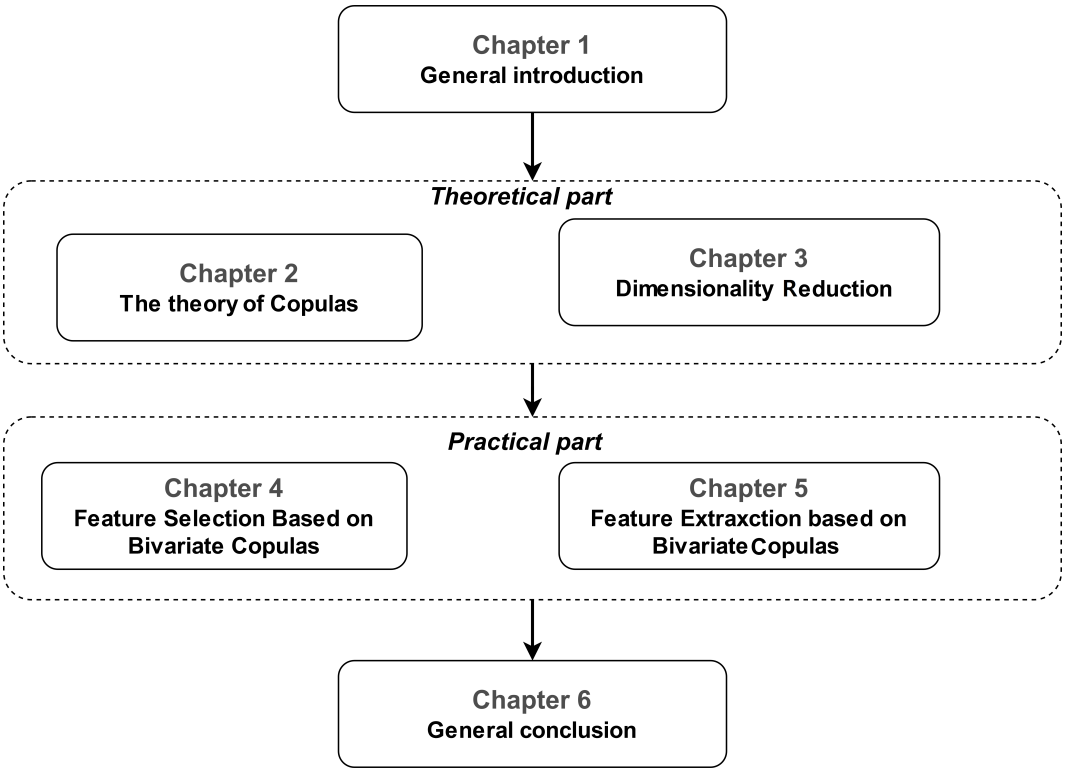


Figure 0.1: Thesis organisation.

- Chapter 1 introduces the idea of our work, how the idea inspired us and how we developed it.
- The second chapter gives an overview about the theory of Copulas, dependency and the mathematical formula needed for our research.
- Chapter 3 explains the notion of Dimensionality Reduction, the state of the art of feature extraction and feature selection, and what challenges authors are currently facing. We also explain how researchers used Copulas to reduce the dimensions of large data, discuss their methods, give a critical discussion and how our method deals with these issues.
- Our contribution is discussed in chapter 4 ans 5 in details.
- The last chapter presents a general conclusion of this thesis, where we outline certain points that could inspire readers for future work.

CHAPTER 1

THE THEORY OF COPULAS

1.1 Introduction

The Copula notion was firstly introduced by Sklar in 1959 [82], its name comes from the latin word "*copũlae*", as an indication of coupling the marginal distribution functions. They have been widely used in statistics for modeling dependencies, especially when it comes to describe the nonlinear dependencies in computer science applications, since they are used to cover the inconvenient of the distribution functions when it comes to modeling dependency and separating the impact of dependence from the impact of marginal distributions in a joint distribution.

This chapter focus on summarising the basic definition of the bivariate Copulas (2-dimensional Copulas) needed for our research. For that, we define the following notions: $I = [0,1]$ is the unit segment, $I^2 = [0,1] \times [0,1]$ is the unit square and for any $0 \leq u_1 \leq u_2 \leq 1, 0 \leq v_1 \leq v_2 \leq 1$, $D = [u_1, u_2] \times [v_1, v_2]$ is a rectangular region in the unit square. The bivariate Copula is a special case of the multivariate Copula, it focus on modeling the dependence between two random variables instead of all variables at once as the case of multivariate Copula. For more details about the multivariate Copulas, see [65].

Definition 1.1.1. *Let $G(u,v)$ be a function from I^2 to I and D be a rectangular region of the unit square.*

- *The G-**volume** of the region D is given as:*

$$V_G(D) = G(u_2,v_2) - G(u_1,v_2) - G(u_2,v_1) + G(u_1,v_1). \tag{1.1}$$

- *We say that $G(u,v)$ is **quasi-monotone**,for any rectangular area D in the unit square, if its G-volume is nonnegative.*

- $G(u,v)$ *is* ***grounded*** *on* I^2 *if* $G(0,v) = G(u,0) = 0$ *for any* $u,v \in I$.

- $G(u,v)$ *is* ***2-increasing*** *if* $V_G(D) \geq 0$*, for any rectangular area* D *in the unit square.*

Definition 1.1.2. *We say that* C *is a 2-dimensional Copula (bivariate Copula) in* I^2 *if it satisfies the following properties:*

- $C(0,v) = C(u,0) = 0$*, for any* u,v *in* I.

- $C(1,v) = v$*,* $C(u,1) = u$*, for any* u,v *in* I.

- $V_C(D) \geq 0$*, for any rectangular region* $D \subseteq I^2$.

For a Copula $C(u,v)$, the partial derivatives $\frac{\partial C}{\partial u}$ and $\frac{\partial C}{\partial v}$ exist for almost all u,v in I. Let's say that $\frac{\partial^2 C}{\partial u \partial v}$ and $\frac{\partial^2 C}{\partial v \partial u}$ are continuous and exist in I^2. Then, the Copula density is defined as:

$$c(u,v) = \frac{\partial^2 C}{\partial u \partial v} = \frac{\partial^2 C}{\partial v \partial u}. \tag{1.2}$$

The bivariate Copula is a cumulative distribution function with Uniform [0,1] margin, it is used to describe the inter-correlation (dependency structure) between two random variables by combining the bivariate distribution function with their one-dimension marginal distribution function. Therefore, let $u = F_X(x)$ and $v = F_Y(y)$ be distribution functions, then any Copula of the form $C(u,v) = C(F_X(x), F_Y(y))$ is a valid bivariate distribution function, and the inverse is also true [82].

1.2 Sklar's theorem

Following Sklar's theorem [65], any bivariate joint distribution can also be written as univariate marginal distribution functions (a unique Copula C in $[0,1]$), and standard uniform marginal distributions which display the dependencies between the variables. this relationship is giving in the following theorem.

Theorem 1.2.1. *Let* H *be a joint distribution function with the* F_X *and* F_Y *margins. Then a Copula* C *exists such that for any x, y.*

$$H(x,y) = C(F_X(x), F_Y(y)). \tag{1.3}$$

If F_X *and* F_Y *are continuous,* C *is unique. Otherwise,* C *is uniquely defined on* $Ran(F_X) \times Ran(F_Y)$*, where* $Ran(F_X)$ *and* $Ran(F_Y)$ *are respectively the ranges of* F_X *and* F_Y*. In contrast, the function* H *defined in (1.3) is a joint distribution function with margins* F_X *and* F_Y *if* C *is a Copula and* F_X *and* F_Y *are distribution functions.*

For the proof of Theorem 1.2.1, see [65].
According to Sklar's theorem, any Copula with marginal distributions as arguments is a valid bivariate distribution, and every valid bivariate distribution may be represented as a Copula of its marginals. It is always possible to separate the dependency structure from the univariate marginals for the continuous case. Howerver, for the discrete one-dimensional marginal distributions, we cannot assume that (1.3) is unique.

Corollary 1.2.1.1. *Let H be a joint distribution function with margins F_X and F_Y, C a Copula defined as in (1.3), $F_X^{(-1)}(u) = inf\{x \in \mathbb{R} | F_X(x) \geq u\}$ and $F_Y^{(-1)}(v) = inf\{y \in \mathbb{R} | F_Y(v) \geq v\}$. Then, for any u, v in [0,1]*

$$C(u,v) = H(F_X^{(-1)}(u), F_Y^{(-1)}(v)). \tag{1.4}$$

The formula giving in (1.4) presents the inversion method for constructing Copulas for joint distribution functions in the case of continuous margins. For the case when F_X and F_Y are discrete marginal distributions, see [65].
If F_X and F_Y are continuous marginal distributions, Copulas are either invariant or do not change often for strictly monotone transformations of random variables [65].

Theorem 1.2.2. *Let X and Y be continuous random variables with Copula C_{XY} and marginals of u, v in I^2. Let a and b be strictly monotone on $Ran(X)$ and $Ran(Y)$, respectively.*

- *If a and b are strictly increasing, then*

$$C_{a(X)b(Y)}(u,v) = C_{XY}(u,v). \tag{1.5}$$

- *If a is strictly increasing and b is strictly decreasing, then*

$$C_{a(X)b(Y)}(u,v) = u - C_{XY}(u, 1-v). \tag{1.6}$$

- *If a is strictly decreasing and b is strictly increasing, then*

$$C_{a(X)b(Y)}(u,v) = v - C_{XY}(1-u, v). \tag{1.7}$$

- *If a and b are strictly decreasing, then*

$$C_{a(X)b(Y)}(u,v) = u + v - 1 + C_{XY}(1-u, 1-v). \tag{1.8}$$

  This Copula is named the survival Copula and will be defined by $\overline{C}$ and satisfies all Copula properties. For proof, see [65]

Let X and Y be random variables with distribution functions $F_X(x)$ and $F_Y(y)$, respectively. The survival function is the probability that X occurs after time x, it is defined by $S_X(x) = P[X > x] = 1 - F_X(x)$. Similarly, the survival function for Y is $S_Y(y) = P[Y > y] = 1 - F_Y(y)$ and

the joint distribution is $S(x,y) = P[X > x, Y > y]$. According to Sklar's theorem, in [65], authors demonstrated that univariate and joint survival functions are connected as univariate and joint distribution functions. For that, this relationship is expressed as follow:

$$S(x,y) = \overline{C}(S_X(x), S_Y(y)). \tag{1.9}$$

Equation (1.10) is the product Copula Π. We say that X and Y are independent if and only if its Copula is giving in (1.10).

$$\Pi(u,v) = uv. \tag{1.10}$$

According to [33], the Copula C is the maximum Copula M, $C(u,v) = M(u,v) = min(u,v)$, if Y is a monotone increasing function of X. While we say C is the minimum Copula W, $C(u,v) = W(u,v) = max(u+v1,0)$, if Y is a decreasing function of X.

Theorem 1.2.3. *For any Copula C and any u,v in I. The following inequalities maintain.*

$$W(u,v) \leq C(u,v) \leq M(u,v). \tag{1.11}$$

The functions $W(u,v)$ and $M(u,v)$ are named as the lower and upper Fréchet-Hoeffding Bounds, respectively. For more details, see [40].

1.3 Measures of dependence

Classical measures of dependence face the challenge of including the joint distribution functions in their structure. Fortunately, Sklar's theorem (Theorem 1.2.1) allows us to overcome this challenge and replace the joint distribution functions with Copulas. By doing so, it is possible to fully remove the information from the random variable. This will reinstate the measure as non-parametric [80], with the property of the invariance under monotonic transformations. As a result, the measurement becomes a rank statistic, which can be taken as a more robust measure of dependence [13]. In this section, we define the comment measures that are associated with Copula. And the main measure used for our contribution.
The Pearson's linear correlation coefficient is the most commonly used measure of dependence. However, it is limited and fails to capture the dependency for most heavy tailed distributions. Luckily, there exist other measures of dependence such as Kendall's concordance and Spearman's rank correlation, they are concordance Copula-based measures. Other graphical methods to describe the dependency between two variables are defined in [33] such as the χ-plot and K-plot.

1.3.1 Linear correlation

Definition 1.3.1. *Let $(X,Y)^t$ be a vector of random variables with nonzero finite variances. The linear correlation coefficient ρ for $(X,Y)^t$ is:*

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}, \tag{1.12}$$

where $Cov(X,Y) = E[XY] - E[X]E[Y]$ *is the covariance of* $(X,Y)^t$, $Var(X) = E[(X - E[X])^2]$ *and* $Var(Y) = E[(Y - E[Y])^2]$ *are the variances of* X *and* Y, *respectively.*

The estimation of the linear correlation coefficient (1.12) is giving as:

$$\widehat{\rho}(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} = \frac{\sum_{i=1}^{n} x_i y_i - n\overline{xy}}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}, \tag{1.13}$$

where $\overline{x} = \sum_{i=1}^{n} x_i$ and $\overline{y} = \sum_{i=1}^{n} y_i$ are the sample means of X and Y, respectively and n is the sample size of a bivariate sample (x_i, y_i), $i = 1, ..., n$. The equation (1.12) of ρ, known as the linear correlation coefficient has the following properties:

- $-1 < \rho(X,Y) < 1$.
- If $|\rho(X,Y)| = 1$, then there is a perfect linear correlation between X and Y.
- $\rho(X,Y) = 0$ if X and Y are independent. The inverse is not true.
- $\rho(\alpha x + \beta, \gamma Y + \theta) = sgn(\alpha\gamma)\rho(X,Y)$, where α, $\gamma \in \mathbb{R}\backslash\{0\}$, $beta$ and $\theta \in \mathbb{R}$. This means that ρ is invariant under strictly increasing linear transformations, where

$$sgn(\alpha\gamma) = \begin{cases} 1 & \text{if } \alpha\gamma > 0, \\ -1 & \text{if } \alpha\gamma < 0\ . \end{cases} \tag{1.14}$$

- X and Y are positively correlated if $\rho(X,Y) > 0$.
- X and Y are negatively correlated if $\rho(X,Y) < 0$.
- X and Y are linearly independent if $\rho(X,Y) = 0$.

Pearson's correlation coefficient is frequently used since it is calculated using basic distributions parameters (means and variances). However, linear correlation is invariant under only linear increasing transformations but not under general increasing transformations.

1.3.2 Measures of concordance

The measures of concordance are considered as non-parametric measures of dependence. The most popular measures of concordance are Kendall's tau τ and Spearman's rho ρ_s, since they overcome the drawbacks of linear correlation ρ and can capture the association for non-linear correlation.

Definition 1.3.2. *Let $(x,y)^t$ and $(\tilde{x},\tilde{y})^t$ be two observations of the vector $(X,Y)^t$ of continuous random variables. We say that $(x,y)^t$ and $(\tilde{x},\tilde{y})^t$ are concordant if $(x,y)(\tilde{x},\tilde{y})>0$ and discordant if $(x,y)(\tilde{x},\tilde{y})<0$.*

Theorem 1.3.3. *Let $(X,Y)^t$ and $(\tilde{X},\tilde{Y})^t$ be independent vectors of continuous random variables with distribution functions H and $\tilde{H}$ respectively, with the margins F_X of X and $\tilde{X}$ and F_Y of Y and $\tilde{Y}$. Let C and $\tilde{C}$ be the Copulas of $(X,Y)^t$ and $(\tilde{X},\tilde{Y})^t$ respectively. Therefore $H(x,y)=C(F_X(x),F_Y(y))$ and $\tilde{H}(x,y)=\tilde{C}(F_X(x),F_Y(y))$. Let Q be the difference between the probability of concordance and discordance of $(X,Y)^t$ and $(\tilde{X},\tilde{Y})^t$, i.e, let*

$$Q=P[(X-\tilde{X})(Y-\tilde{Y})>0]-P[(X-\tilde{X})(Y-\tilde{Y})<0], \tag{1.15}$$

then

$$Q=Q(C,\tilde{C})=4\iint_{I^2}\tilde{C}(u,v)dC(u,v)-1. \tag{1.16}$$

For the proof, see [21].

Corollary 1.3.3.1. *Let $C,\tilde{C}$, and Q be as in the previous Theorem 1.3.3. We state that:*

- *$Q(C,\tilde{C})=Q(\tilde{C},C)$, i.e. Q is symmetric in its arguments.*
- *If $C\prec C'$, $Q(C,\tilde{C})\leq Q(C',\tilde{C})$, i.e. Q is nondecreasing in each argument.*
- *$Q(C,\tilde{C})=Q(\overline{C},\overline{\tilde{C}})$, i.e. We can replace Copulas by survival Copulas in Q.*

Definition 1.3.4. *Let X and Y be two continuous random variables with the joint Copula C and δ be a measure of dependence between X and Y. δ is a measure of concordance if it satisfies the following properties:*

$\triangleright$ *δ exists for every pair X, Y of continuous random variables.*

$\triangleright$ $-1\leq\delta_{X,Y}\leq 1, \delta_{X,X}=1$ *and* $\delta_{X,-X}=-1$.

$\triangleright$ $\delta_{X,Y}=\delta_{Y,X}$.

$\triangleright$ *If X and Y are independent, then* $\delta_{Y,X}=\delta_{\Pi}=0$.

$\triangleright$ $\delta_{-X,Y}=\delta_{X,-Y}=-\delta_{X,Y}$.

$\triangleright$ *If C and $\tilde{C}$ are Copulas such that $C\prec\tilde{C}$, then $\delta_C\leq\delta_{\tilde{C}}$.*

$\triangleright$ *If (X_n,Y_n) is a sequence of continuous random variables with Copulas C_n, and if C_n converges pointwise to C, then* $\lim_{x\to\infty}\delta_{C_n}=\delta_C$.

Definition 1.3.4 is taking from [76].

1.3.2.1 Kendall's tau τ

Kendall's tau τ is the probability of concordance minus the probability of discordance [21]. Its definition is giving as follow.

Definition 1.3.5. *Let $(X,Y)^t$ be a random vector. Kendall's tau τ is defined by:*

$$\tau(X,Y) = P[(X-\tilde{X})(Y-\tilde{Y})>0] - P[(X-\tilde{X})(Y-\tilde{Y})<0], \tag{1.17}$$

where $(X,Y)^t$ and $(\tilde{X},\tilde{Y})^t$ are independent copies.

Kendall's tau τ can be estimated easily. Let A be the number concordant pairs and B the number of discordant pairs, we also state that in the bivariate sample of size n, there are $\binom{2}{n}$ distinct pairs (x,y) and $(\tilde{x},\tilde{y})$. Therefore, Kendall's tau τ can be estimated as follow:

$$\hat{\tau} = \frac{A-B}{A+B} = \binom{2}{n}^{-1}(A-B). \tag{1.18}$$

Kendall's tau τ is known as Copula-based measure of dependence. Which means that there is a relationship between τ and Copulas, this relationship will allow us to estimate the Copula's parameter directly, as we will discuss it later. The following definition defines the relation between Copula and Kendall's tau τ.

Definition 1.3.6. *Kendall's tau for a random vector $(X,Y)^t$ is defined as:*

$$\tau(X,Y) = Q(C,C) = 4\iint_{I^2} C(u,v)dC(u,v) - 1. \tag{1.19}$$

The integral above is the expected value of the random variable $C(U,V)$, where U and V are uniform margins with the joint distribution function C. This means

$$\tau(X,Y) = 4\mathbf{E}(C(U,V)) - 1. \tag{1.20}$$

1.3.2.2 Spearman's rho ρ_s

Spearman's rho ρ_s is another Copula-based measure of concordance. Its definition is giving as follow.

Definition 1.3.7. *Let $(X,Y)^t$ be random variables. Spearman's pho ρ_s formula is defined by:*

$$\rho_s = 3(P[(X-X')(Y-Y'')>0] - P[(X-X')(Y-Y'')<0], \tag{1.21}$$

where $(X,Y)^t$, $(X',Y')^t$ and $(X'',Y'')^t$ are independent copies.

From theorem 1.3.3 and definition 1.3.4, we get the following results.

Theorem 1.3.8. *For a random vector $(X,Y)^t$ of continuous variables with Copula C. Then Spearman's rho ρ_s is defined by:*

$$\rho_s = 3Q(C,\Pi) = 12\iint_{I^2} uvdC(u,v) - 3 = 12\iint_{I^2} C(u,v)dudv - 3, \tag{1.22}$$

thus, if $X \sim F_1$, $Y \sim F_2$, $U = F_1(X)$ and $V = F_2(Y)$, then

$$\rho_s = 12\iint_{I^2} C(u,v)dudv - 3 = \frac{\mathbf{E}(UV) - 1/4}{1/12} = \frac{COV(U,V)}{\sqrt{Var(U)Var(V)}} = \rho(F_1(X),F_2(Y)). \tag{1.23}$$

Theorem 1.3.9. *Let X and Y be continuous random variables whose Copula is C, then Kendall's tau τ and Spearman's rho ρ_s satisfy the properties in Definition 1.3.4 for a measure of concordance.*

The proof of theorem 1.3.9 is available at [66]. This theorem demonstrates that both Kendall's tau τ and Spearman's rho ρ_s are measures of concordance.

1.3.3 Tail dependence

Another common measure of dependence is tail dependence, it is a statistical concept that is used to measure the degree of dependence between the extreme values of two random variables. It is specifically designed to measure the dependence between the upper or lower tails of two random variables. In other word, tail dependence are used to measure the probability that both variables have extreme values at the same time. The two types of tail dependence are upper tail dependence and lower tail dependence. Upper tail dependence measure the probability that both variables have large values simultaneously. while lower tail dependence measure the probability that both variables have small values simultaneously. In this section we give the basic formula of the two coefficients motioned above. More details can be found in [65] [46]. They include many examples and exercises to help readers better understand and apply these concepts.
The upper tail dependence coefficient λ_U is defined as:

$$\lambda_U = \lim_{p\to 1} P(Y > F_Y^{-1}(p) | X > F_X^{-1}(p)), \tag{1.24}$$

where $F_X^{-1}(p)$ and $F_Y^{-1}(p)$ are the inverse cumulative distribution functions of X and Y, respectively, and p is a probability level that approaches 1. While the lower tail dependence coefficient λ_L is defined as:

$$\lambda_L = \lim_{p\to 0} P(Y \leq F_Y^{-1}(p) | X \leq F_X^{-1}(p)), \tag{1.25}$$

where p is a probability level that approaches 0.

Intuitively, the upper tail dependence coefficient measures the probability that X is larger than its expected value given that Y is larger than its expected value, as the probability level p approaches 1. The lower tail dependence coefficient measures the probability that X is smaller than its expected value given that Y is smaller than its expected value, as the probability level p approaches 0.

If both λ_U and λ_L are close to 1, it indicates strong tail dependence, meaning that the two variables tend to have extreme values together. If both coefficients are close to 0, it indicates weak tail dependence, meaning that the two variables are relatively independent in their extreme values. If λ_U is close to 1 and λ_L is close to 0, it indicates that the two variables have only upper tail dependence, while if λ_L is close to 1 and λ_U is close to 0, it indicates that the two variables have only lower tail dependence.

Definition 1.3.10. *Let X and Y be two random variable with margins F_X and F_Y and C be the joint Copula. The upper tail dependence coefficient can be expressed in term of Copula as follow:*

$$\lambda_U = \lim_{u \to 1} \frac{1 - 2u + C(u,u)}{1-u} = \lim_{u \to 1} \frac{\overline{C}(u,u)}{1-u}. \tag{1.26}$$

Similary, The lower tail dependence coefficient can be expressed in term of Copula as follow:

$$\lambda_L = \lim_{u \to 0} \frac{C(u,u)}{u}. \tag{1.27}$$

Tail dependence are particularly useful in modeling extreme events, which can have a significant impact on financial markets, insurance portfolios, and other applications where risk management is important. By capturing the dependence structure in the tails of the distributions, tail dependence can help to more accurately model the joint behavior of random variables and better estimate the likelihood of extreme events. This can lead to more effective risk management strategies and better decision-making in a variety of fields.

1.4 Families of Copulas

There are many families of Copulas that have been developed over the years, each with its own unique characteristics and applications. We distinguish two type of Copulas.

1.4.1 Elliptical Copulas

Elliptical Copulas are a family of Copulas that have been extensively studied in the literature of Copula theory. They are named after their characteristic elliptical shape and are popular because of their simplicity and flexibility. These Copulas have many desirable properties, including the ability to capture various types of dependence structures, such as positive or negative dependence, tail dependence, and asymmetry. They are also flexible enough to be used in a wide range of applications, including finance, insurance, engineering, and environmental studies.
There are many studies on the properties of elliptical Copulas, their applications, and their estimation methods. The book [46] provides a comprehensive introduction to Copulas and their applications. It includes a detailed discussion of elliptical Copulas and their properties. While the paper [33] provides a detailed overview of Copulas, with a focus on their use in hydrology.

Elliptical Copulas are characterized by their generator function, which is a radial function that satisfies certain conditions. The most common generator functions for elliptical Copulas are the Gaussian and t-generator functions. The Gaussian generator function gives rise to the Gaussian Copula, while the t-generator function gives rise to the t-Copula. In the next subsections, we will introduce the two most used elliptical bivariate Copulas, we refer to [65] and [46] for details about these two Copulas and other elliptical Copulas.

1.4.1.1 Elliptical Distributions

Elliptical distributions are a family of probability distributions that exhibit a certain type of symmetry and can be expressed as a transformation of a standard random vector. The name "elliptical" comes from the fact that the contours of the distribution in the multivariate case form ellipsoids. We will only introduce the needed background for this thesis. For more details see [51].

Definition 1.4.1. *Let X be a n-dimensional random vector. The Probability Density Function (PDF) of an elliptical distribution can be written as:*

$$\psi(X) = |\Sigma|^{-1/2} g((X-\mu)\Sigma^{-1}(X-\mu)^t), \tag{1.28}$$

where X is a n-dimensional random vector, μ is the mean vector, Σ is the covariance matrix, and $g(.)$ is a scalar function known as the generator function of the elliptical distribution.

The generator function determines the shape of the distribution and can take different forms depending on the specific type of elliptical distribution. Some examples of elliptical distributions include the Gaussian distribution (when the generator function is the exponential function), the Student's t-distribution (when the generator function is a power function), and the Laplace distribution (when the generator function is the absolute value function). The multivariate normal distribution Φ is a specific type of elliptical distribution, with the following PDF:

$$\phi(X) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left\{-\frac{1}{2}\left(X-\mu\right)\Sigma^{-1}\left(X-\mu\right)^t\right\}. \tag{1.29}$$

Other examples of elliptical distributions include the multivariate t-distribution, which has a PDF that is similar to the multivariate normal distribution but includes a degrees of freedom parameter. The PDF of the multivariate t-distribution t_ν is given by:

$$t_\nu(x) = |\Sigma|^{-\frac{1}{2}} \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\pi\nu)^{\frac{d}{2}}} \left(1 + \frac{X^t\Sigma^{-1}X}{\nu}\right), \tag{1.30}$$

where ν is the degrees of freedom parameter. and Γ is the Gamma function defined as:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \tag{1.31}$$

where $x > 0$.

1.4.1.2 The Bivariate Gaussian Copula

The bivariate Gaussian Copula C_θ^{Ga} is a Copula function that is mostly commonly used to model the dependence between two random variables. Using (1.4), it can be defined as:

$$C_\theta^{Ga}(u,v) = \Phi_\theta(\Phi^{-1}(u), \Phi^{-1}(v)), \tag{1.32}$$

where u and v are the marginal distribution functions of the two random variables, Φ_θ is the cumulative distribution function of the bivariate standard normal distribution with correlation coefficient θ, and Φ^{-1} is the inverse of the standard normal cumulative distribution function. The range of θ is $[-1,1]$, where $\theta = -1$ indicates perfect negative dependence, $\theta = 0$ indicates no dependence, and $\theta = 1$ indicates perfect positive dependence. On the other hand, The Gaussian Copula density is expressed as follow:

$$c_\theta^{Ga}(u,v) = \frac{\phi_\theta(s,r)}{\phi(s)\phi(r)}, \tag{1.33}$$

where $\phi_\theta(s,r)$ is the density function of the standardised bivariate normal distribution with correlation θ and ϕ is the density of the standard normal distribution, $s = \Phi^{-1}(u)$ and $r = \Phi^{-1}(v)$. Therefore, the equation (1.33) can be written explicity as follow:

$$c_\theta^{Ga}(u,v) = \frac{1}{\sqrt{1-\theta^2}} \exp\left[-\frac{\theta^2(s^2+r^2) - 2\theta sr}{2(1-\theta^2)}\right]. \tag{1.34}$$

Since Gaussian Copula belongs to the elliptical family, we state that it has radial symmetry, that is $C(u,v) = \overline{C}(u,u)$. As a consequence, the upper and lower tail dependence coefficients are equal. As we've said before, the Gaussian Copula have no tail dependence, ie:

$$\lambda_U = \lambda_L = 0. \tag{1.35}$$

The relationship between Kendall's tau and Copula's parameter is given as follow:

$$\tau = \frac{2}{\pi}\arcsin(\theta). \tag{1.36}$$

While the Spearman's rho ρ_s is given as follow:

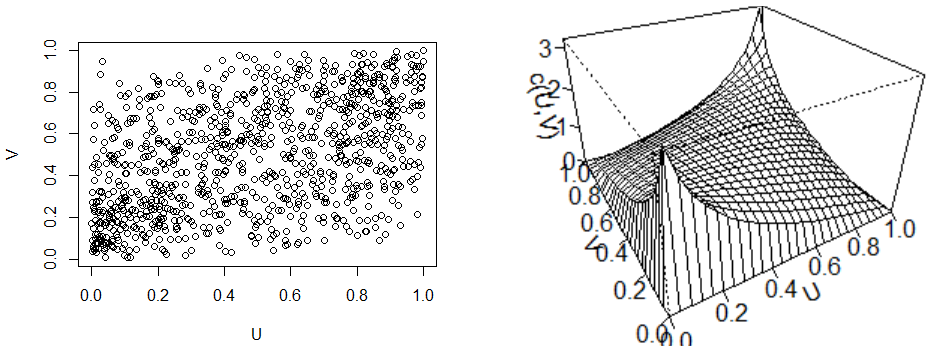
$$\rho_s = \frac{6}{\pi}\arcsin(\theta). \tag{1.37}$$

The formulas given in (1.36) and (1.37) can be used to model the inter-correlation between the variables.

Simulation

- Choose a value for the correlation coefficient, θ, where $-1 \leq \theta \leq 1$.

- Compute the Cholesky decomposition of the correlation matrix Σ, which is a 2×2 matrix with 1's on the diagonal and θ as the off-diagonal elements. This yields to a lower triangular matrix A such that $\Sigma = AA^t$.
- Generate two independent standard normal random variables Z_1 and Z_2.
- Transform the random variables Z_1 and Z_2 using the Cholesky decomposition matrix A to obtain the transformed variables X and Y as $(X,Y)^t = AZ$.
- Compute the Cumulative Distribution Function (CDF) of the standard normal distribution for each transformed variable X and Y, i.e., $U = \Phi(X)$ and $V = \Phi(Y)$, where $\Phi(x)$ is the CDF of the standard normal distribution.
- The resulting random vector (U,V) has a Gaussian Copula with correlation coefficient θ, where U and V are uniformly distributed random variables on $[0,1]$ and follow the Copula distribution C_θ^{Ga}.



(a) Copula (b) Density Copula

Figure 1.1: The Gaussian Copula plot with the paramater $\theta = 0.5$ and normal margins, where $n = 1000$.

The corresponding R code of the simulation above of the Gaussian bivariate Copulas and the plot shown in Figure 1.1 are giving in details in Appendix 4.4.
The bivariate Gaussian Copula is commonly used in various applications to model complex dependence structures between two random variables. Unlike traditional linear models, such as the Pearson correlation coefficient, it can capture dependencies that are not well represented by linear relationships.

One advantage of the bivariate Gaussian Copula is that it is easy to implement and interpret, and it allows for the use of standard statistical techniques for estimation and inference. However, it may not be suitable for modeling non-Gaussian or heavy-tailed distributions, as it assumes that the marginal distributions of the two variables are normal. In such cases, other types of Copula functions, such as the t-Copula or the Clayton Copula, may be more appropriate.

1.4.1.3 The Bivariate Student t-Copula

The bivariate Student Copula $C^t_{\nu,\theta}$ is another elliptical Copula function that is used to model the dependence between two random variables that follow a multivariate Student t-distribution. Using the formula (1.4), it can be defined as:

$$C^t_{\nu,\theta}(u,v) = T_{\nu,\theta}(T_\nu^{-1}(u), T_\nu^{-1}(v)), \tag{1.38}$$

where u and v are the marginal distribution functions of the two random variables, $T_{\nu,\theta}$ is the bivariate t-distribution with ν degrees of freedom parameter and the correlation parameter θ between the two variables and T_ν^{-1} is its inverse. The density of the Student t-Copula can be expressed as:

$$c^t_{\nu,\theta} = \frac{t_{\nu,\theta}(s,r)}{t_\nu(s)t_\nu(r)}, \tag{1.39}$$

where $s = T_\nu^{-1}(u)$ and $r = T_\nu^{-1}(v)$. For that, using (1.30), it can be written explicitly as follow:

$$c^t_{\nu,\theta}(u,v) = \frac{\Gamma\left(\frac{\nu+2}{2}\right)\Gamma\left(\frac{\nu}{2}\right)}{\sqrt{1-\theta^2}\,\Gamma^2\left(\frac{\nu+1}{2}\right)} \frac{\left[\left(1+\frac{s^2}{\nu}\right)\left(1+\frac{r^2}{\nu}\right)\right]^{\frac{\nu+1}{2}}}{\left[1+\frac{s^2+r^2-2\theta sr}{\nu(1-\theta^2)}\right]^{\frac{\nu+2}{2}}}. \tag{1.40}$$

Similarly to the Gaussian Copula, the Student Copula is also has radial symmetry and the tail dependence coefficients are equal. However, they exist. They are expressed as follow

$$\lambda_U = \lambda_L = 2T_{\nu+1}(t), \tag{1.41}$$

where $T_{\nu+1}$ is the univariate Student distribution function with $\nu+1$ degrees of freedom and

$$t = -\sqrt{\nu+1}\sqrt{\frac{1-\theta}{1+\theta}}. \tag{1.42}$$

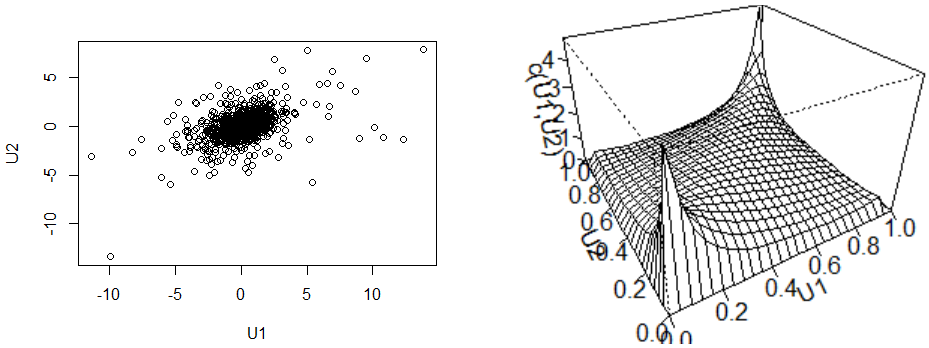
Simulation

- Choose a value for the correlation coefficient, θ, where $-1 \le \theta \le 1$ and the degree of freedom ν.
- Set

$$\Sigma = \begin{pmatrix} 1 & \theta \\ \theta & 1 \end{pmatrix}. \tag{1.43}$$

- Compute the Cholesky decomposition L of the correlation matrix Σ.
- Generate two independent standard normal random variables Z_1 and Z_2.
- Generate a χ^2 random variable s with ν degrees of freedom.

- Compute the correlated standard normal random variables Y_1 and Y_2 as $Y = LZ$.
- Compute the correlated t-distributed random variables X_1 and X_2 as $X = \sqrt{\frac{\nu}{s}}Y$.
- Transform X_1 and X_2 to t-distributed random variables U and V with ν degrees of freedom. ie, set $U = T_\nu(X_1)$ and $V = T_\nu(X_2)$.
- The resulting random vector (U,V) has a Student Copula with correlation coefficient θ, where U and V are uniformly distributed random variables on $[0,1]$ and follow the Copula distribution C_θ^t.



(a) Copula (b) Density Copula

Figure 1.2: The Student Copula plot with the paramater $\theta = 0.5$ and normal margins, where $n = 1000$.

The R code of the simulation above of the Student bivariate Copula and the corresponding plot as shown in Figure 1.2 are given in details in Appendix 4.4.
The bivariate Student Copula allows for the modeling of heavy-tailed distributions and is more robust to outliers compared to the bivariate Gaussian Copula. It also provides a more flexible modeling approach since the degrees of freedom parameter ν can be estimated from the data. However, the estimation of the bivariate Student Copula can be computationally intensive, and the interpretation of the degrees of freedom parameter ν may not be straightforward. Additionally, the bivariate Student Copula may not be suitable for modeling extreme values, as it assumes that the tails of the distributions are symmetric.

Extensions of the bivariate Student Copula, such as the asymmetric Student Copula and the skewed t-Copula, have been proposed to overcome some of these limitations and to provide a more flexible modeling framework. See [46] and [33] for more details.

1.4.2 Archimedean Copulas

Bivariate Archimedean Copulas are useful when it comes to modeling the dependence between two random variables with non-Gaussian distributions, where other types of Copulas may not

be appropriate. The class of Archimedean Copulas has been named by Ling in [58], but it was recognized by Schweizer and Sklar in [79]. It's an important class of Copula because of the easy way of constructing it. We will not include a lot of details about the Archimedean Copulas since they were not used in the application chapters of this thesis. For more information about it, see [65], [33] and [46]. The general form of a bivariate Archimedean Copula is given by:

$$C(u,v) = \varphi^{-1}(\varphi(u) + \varphi(v)), \tag{1.44}$$

where φ is called generator that it's a continuous strictly decreasing convex function defined from I to $[0,\infty]$ such that $\varphi(1) = 0$ and φ^{-1} be the inverse of φ and C be a Copula. If the second derivative $\varphi''(t)$ exists, then the density of Archimedean Copula can be defined as:

$$c(u,v) = -\frac{\varphi''(C(u,v))\varphi'(u)\varphi'(v)}{(\varphi'(C(u,v)))^3} \tag{1.45}$$

There are several type of Archimedean Copulas, among them the Gambel C_θ^G, Clayton C_θ^C and Frank C_θ^F Copulas. All of these Archimedean Copulas have useful properties that make them popular choices for modeling dependence structures in multivariate and bivariate distributions. In particular, they are computationally efficient and allow for easy estimation of parameters. Their generators are given in Table 1.1, and the Copula formula is given in Table 1.2, where θ is the Copula's parameter for each Copula. While 1.3 present the density and Copula plots for Gumbel, Clayton and Frank Copula with the parameters 1.3, 1 and -1 respectively.

Table 1.1: Generator of Archimedean Copulas.

Copula	$\varphi(t)$	$\varphi^{-1}(s)$	$\theta \in$
Gumbel	$(-\log t)^\theta$	$\exp\{-((-\log(u))^\theta + (-\log(v))^\theta)^{\frac{1}{\theta}}\}$	$[1,\infty)$
Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$\max\{(1+\theta s)^{-1/}, 0\}$	$[-1,\infty)\backslash\{0\}$
Frank	$-\log(\frac{e^{-\theta t}-1}{e^{-\theta}-1})$	$-\frac{1}{\theta}\log[1 + e^{-s}(e^{-\theta} - 1)]$	$\mathbb{R}\backslash\{0\}$

Table 1.2: Archimedean Copulas formula.

Copula	$C_\theta(u,v)$	$c_\theta(u,v)$
Gumbel	$\exp -((log(u))^\theta + ((log(v))^\theta))^{1/\theta}$	$(uv)^{-1}(log(u)log(v))^{\theta-1}(w^{2/\theta-2} + (\theta-1)w^{2/\theta-2})C_\theta(u,v)$
Clayton	$\max\{(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}, 0\}$	$\frac{u^{(\theta+1)}(uv)^\theta}{(u^\theta+v^\theta-(uv)^\theta)^{1/\theta+2}}$
Frank	$-\frac{1}{\theta}\log[\frac{1+e^{-\theta}-(1-e^{-\theta u}(1-e^{-\theta v}))}{1-e^{-\theta}}]$	$\frac{\theta(1-e^{-\theta})e^{-\theta(u+v)}}{(1-e^{-\theta}-(1-e^{-\theta u})(1-e^{\theta v}))^2}$

The relationships between the Copula's generator and the measures of dependence permitted us to compute the value assuming a Copula model (Gumbel, Clayton, ...). Genest and MacKay in [34] demonstrated that the population version of Kendall's tau is given by:

$$\tau = 1 + 4\int_0^1 \frac{\varphi(t)}{\varphi'(t)}dt. \tag{1.46}$$

While the notion of tail dependence in term of Archimedean Copula's generator is expressed in the following theorems.

Theorem 1.4.2. *Let C be be an Archimedean Copula with generator φ defined as follow:*

$$C(u,v) = \varphi^{-1}(\varphi(u) + \varphi(v)). \tag{1.47}$$

If C has upper tail dependence, then $\frac{\partial \varphi^{-1}}{\partial t}(0) = -\infty$ and its coefficient is given by:

$$\lambda_U = 2 - 2\lim_{s\to 0} \frac{\frac{\partial \varphi^{-1}}{\partial t}(2s)}{\frac{\partial \varphi^{-1}}{\partial t}(s)}. \tag{1.48}$$

However, if $\frac{\partial \varphi^{-1}}{\partial t}(0)$ is finite, then C does not have an upper tail dependence.

Theorem 1.4.3. *Let C be an Archimedean Copula with generator φ defined as in theorem 1.4.2, then then the coefficient of lower tail dependence of the Copula C is given by*

$$\lambda_L = 2\lim_{s\to \infty} \frac{\frac{\partial \varphi^{-1}}{\partial t}(2s)}{\frac{\partial \varphi^{-1}}{\partial t}(s)}. \tag{1.49}$$

For the proof of these theorems, see [21].
Using (1.46), the measures of concordance Kendall's tau τ and the coefficients of tail dependence can be derived for the Archimedean Copulas families as shown in Table 1.3. We can see that the Clayton Copula shows lower tail dependence and the Gumbel-Hougaard upper tail dependence. However, and similarly to the Gaussian Copula, the Frank Copula does not show any tail dependence.

Table 1.3: Measures of dependence for the Archimedean Copulas.

Copula	τ	λ_U	λ_L	Limiting and Special Cases
Gumbel	$\frac{\theta-1}{\theta}$	$2-2^{-1/\theta}$	0	$C_1 = \Pi, C_\infty = M$
Clayton	$\frac{\theta}{\theta+2}$	0	$2^{-1/\theta}$	$C_{-1} = W, C_0 = \pi, C_\infty = M$
Frank	$1 + \frac{4(D_1{}^1(\theta)-1)}{\theta}$	0	0	$C_{-\infty} = W, C_0 = \pi, C_\infty = M$

[1] $D_1(\theta) = \frac{1}{\theta}\int_0^\theta \frac{t}{e^t-1}dt$

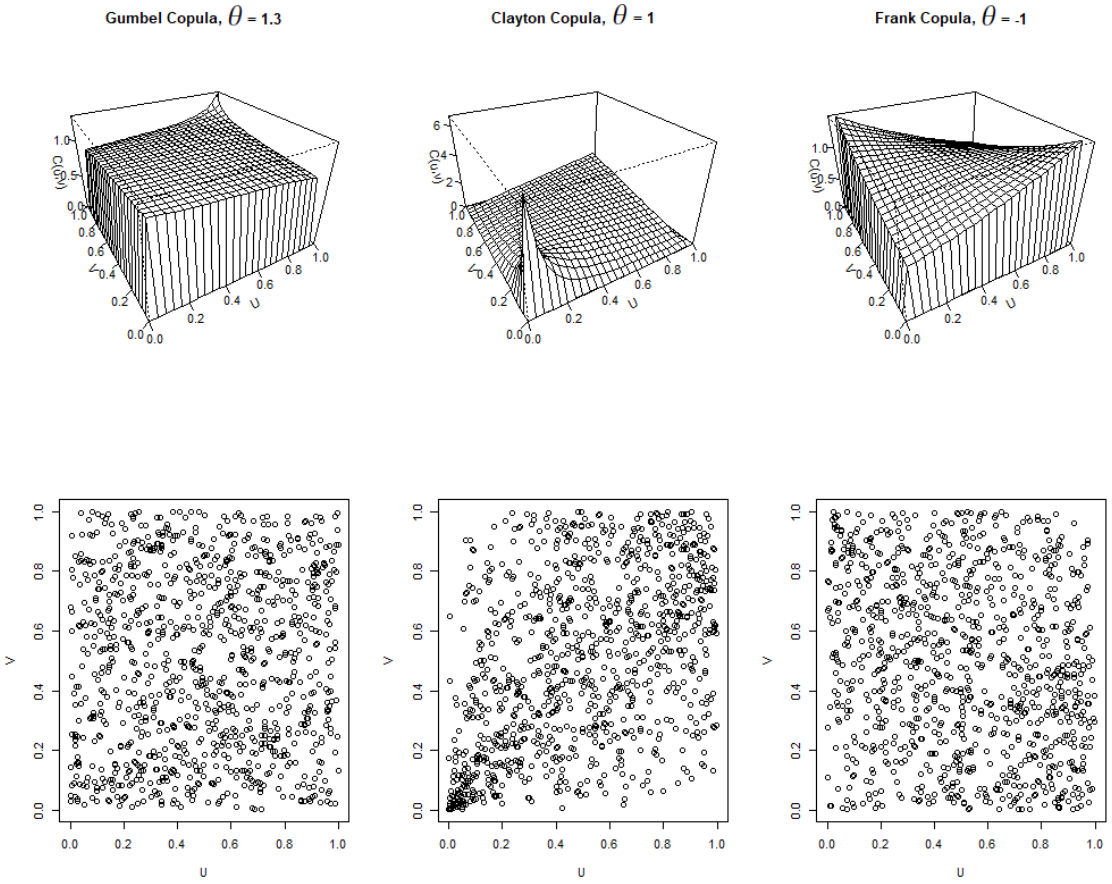


Figure 1.3: Archimedean Copulas, n=1000.

1.5 Empirical Copula

The empirical Copula is a non-parametric method for estimating the Copula function from data. It is based on the idea of transforming the marginal distributions of a bivariate random variable into uniform distributions and then computing the empirical distribution function of the transformed data [22].

To estimate the empirical Copula, we start by considering a sample of n observations of a bivariate random vector (X,Y). We assume that X and Y are continuous and have continuous marginal distribution functions $F_X(x)$ and $F_Y(y)$, respectively. We then transform X and Y into uniform distributions $U[0,1]$ using the marginal distribution functions, i.e.,

$$U_i = F_X(X_i), i = 1,2,...,n, V_i = F_Y(Y_i), i = 1,2,...,n. \tag{1.50}$$

We now form a new data set (U_i,V_i), and calculate the empirical distribution function of the joint distribution of the transformed variables. This empirical distribution function is known as

the empirical bivariate Copula $\hat{C}$, and is defined as:

$$\hat{C}(u,v) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{U_i \leq u, V_i \leq v}, \tag{1.51}$$

where $\mathbb{1}_i$ denotes the number of observations with $U_i \leq u$ and $V_i \leq v$.

The empirical bivariate Copula $\hat{C}$ is a non-parametric estimator of the true Copula C, and is consistent in the sense that it converges to the true Copula as the sample size n increases to infinity. It is also known to be uniformly consistent, meaning that the maximum distance between the empirical Copula and the true Copula converges to zero with probability one.

1.6 Conclusion

This chapter provided a brief discussion of the theory of bivariate Copulas and introduced the mathematical formula needed for our research, which are presented in Chapters 3 and 4. Additionally, Appendix 4.4 includes some R code to further explain how to simulate bivariate Copulas. For a more detailed understanding of Copulas, we recommend consulting sources such as [65] and [22].

CHAPTER 2

DIMENSIONALITY REDUCTION

2.1 Introduction

Throughout time, mathematicians found out that datasets don't contain only relevant information, it tend to be larger and more complex [31], and thus can be reduced. Therefore, the need for powerful reduction tools and methods was rising, which gave birth to many efficient methods for the Dimensionality Reduction process. These techniques aim to reduce the number of attributes of datasets without losing relevant information about the data. They play an essential role in the pre-processing steps, as they eliminate the redundancy and the noise of the data, improve the classification accuracy and decrease the computational time, especially when working with high-dimensional datasets, as it tend to be much more complicated than with low-dimensional ones. While the choice of an appropriate Dimensionality Reduction technique for the data became a must to attain the best model for the data, and the optimal accuracy [6]. Dimensionality Reduction finds an essential purpose in statistics and machine learning, and as a result, it has been developed for these fields and through them for decades. This made Dimensionality Reduction methods a necessary tool for large data analysis, as they generate a smaller and less noisy image of the large data that conserve its most important features. It was applied in different fields as text [83], digital images [84], speech signals [11] and videos [67].
Authors divided the Dimensionality Reduction techniques into two different types, named feature extraction and feature selection. These two patterns are described in details in the following subsections.

2.2 Feature extraction

Feature extraction is a technique that transforms the data by projecting a high-dimensional data into a lower dimensional subspace called the principal subspace. This process generates new features based on the raw data. The projection is done by algebraic transformation under an optimization criteria [20] [93]. These methods make it possible to preserve the original relation between features hence not loosing any important amount of information [1]. Feature extraction searches for significant diminished structures of the data, this process guarantees capturing more important information, with a reduced number of attributes compared to features selection. Overall, feature extraction is a technique that may offers less overfitting and better accuracy for classification, in contrast to feature selection technique. Despite that, feature extraction techniques have their share of bad sides too. First of all, some of the extraction algorithms that supposedly use non-linear methods, are unable to complete the mapping back from the high-dimensional space to the low-dimensional one, making the training of some correct and useful classification models impossible. Moreover, features extraction methods are often unsupervised, so they lose their label information after the Dimensionality Reduction, which is essential for certain type of applications in order to make a concluded prediction model. Also, since it produces new and completely different features, we lose the data description linked to the original features, and this can be a heavy drawback for many datasets [7] [18]. We distinguish two types of feature extraction techniques: Linear and Non-linear techniques, to each its own characteristics.

2.2.1 Linear Dimensionaltiy Reduction techniques

The fact that linear Dimensionality Reduction methods work under simple geometric interpretations, made them the key of analyzing high-dimensional data. They also preserve important characteristics of the data, like the correlation, margin between the classes, and the covariance. Thanks to that, linear Dimensionality Reduction has found many uses, such as : data compression, cleaning noise, visualizing the data structure. Dealing with most data types with a bunch of different methods available, gives linear Dimensionality Reduction a great complexity and a lot of possibilities.

2.2.1.1 SVD (Singular Value Decomposition)

(**SVD**) [72] is an unsupervised linear feature extraction technique. It is one of the most known tool in numerical algebra for data pre-processing and Dimensionality Reduction, because it's very adaptive and based on simple and interpretable linear algebra. This decomposition is done by making a transformation that maps the data into a new more simple coordinate data. We also find it in Fourier transformation (**FFT**) "data driven-transformation", machine learning like ranking search results in the most relevant way which is used in search engines. It is the basis for many facial recognition algorithms (identify faces in pictures).

SVD works as follow, let X be a data matrix which is defined as a collection of column vectors $\{X_1, X_2, ..., X_m\}$. **SVD** (Singular Value Decomposition) takes the matrix X and decompose it into the product of three new matrices $X = U\Sigma V^T$, where U and V are the left and right matrices of singular vector. These two matrices are orthogonal with dimensions $n \times m$, and are hierarchically ordered by importance, i.e. (U_1 has more information than U_2, U_2 has more information than U_3, etc...), U contains information about the columns. While V holds the information about the rows. Σ on the other hand is a diagonal non negative $m \times m$ matrix. Which is also hierarchically ordered by importance and value ($\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0$). The **SVD** gives a unique result, and is guaranteed to exist.

Choosing the number of vectors to retain, also known as the truncation value r, can be done with different common methods. Like picking the r directly from the σ_j plot as the elbow value, where we retain only the singular values that are above that r. Or like capturing 90% or 99% of the variance explained plot [9]. However, these methods are heuristic and most of the time they don't work well (a higher rank of r will result in more accuracy but a more complex model. While a lower rank of r would give us less accuracy but with a less complex model). To deal with that and get a balance between modeling complexity and accuracy, In [32], authors proposed an optimal way to truncate, while having some assumptions on the data. They could look for the best spot to retain the most information of X without over-fitting the data. They stated that the data X can be written as:

$$X = X_{True} + \gamma X_{Noise}, \tag{2.1}$$

where X_{Noise} is the noise present in the data matrix X, that X_{Noise} is normally distributed with a mean of 0 and a variance of 1, and γ is the noise coefficient. The r value this time is obtained with the help of the median σ_{med} of the values of σ in the matrix Σ, and the aspect ratio β of the matrix X (the shape of the matrix), they infer that the max singular value of the noise distribution named the threshold δ. While assuming that the value σ_{med} is in the noise floor, will give us the optimal rank of r, where:

$$\begin{aligned} \delta &= w(\beta)\sigma_{med}, \\ w(\beta) &= 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43, \end{aligned} \tag{2.2}$$

where w represents the correction value of the aspect ratio β. The latter (based on the shape of the data matrix) is defined as :

$$\beta = \begin{cases} \frac{n}{m} & \text{if } n < m, \\ \frac{m}{n} & \text{if } n > m. \end{cases} \tag{2.3}$$

The Singular Value Decomposition **SVD** was applied in different fields, such as in gene expression data [62], signal processing [92], [5] and in NLP [64].

Constrained **SVD** was introduced (**CSVD**) [36] to deal with orthogonal and sparsity issue by merging some constraints for more efficiency of the baseline method **SVD**. While in [45], authors introduce a multi-level **SVD** based imputation method in order to improve the efficiency of the pre-processing phase in various areas.

2.2.1.2 PCA (principal Components Analysis)

PCA is an unsupervised linear technique that uses eigenvectors to identify a set of uncorrelated attributes called Principal Components (**PCs**). This method is a data-driven hierarchical coordinate system based on data that represents the statistical variation. It captures the maximum amount of variance in the data matrix X. It is known for being a powerful technique for Dimensionality Reduction [47] and it is commonly used in data science and machine learning applications. It first appeared in 1901 [70] and was developed by Hotelling [42], and it is still frequently used today. In order to apply **PCA** on our dataset, we will be following these steps. Let X be a matrix of $n \times m$ dimensions:

In the first step we calculate the mean of columns, which is defined as:

$$\overline{x_i} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}, \tag{2.4}$$

where x_{ij} represents the observation of the i^{th} row and the j^{th} column of matrix X. The next step consists on making mean-centered data, as defined in here:

$$B = X - \overline{X}, \tag{2.5}$$

where $\overline{X} = \{\overline{x_1}, \overline{x_2}, ..., \overline{x_n}\}^T$, and then we compute the covariance matrix of the mean-centered data B

$$C = B^T B. \tag{2.6}$$

This last step will lead us to the calculation of the eigen vectors and the eigenvalues, and also to the construction of the Principal Components Matrix T, defined as :

$$\begin{aligned} T &= BV, \\ CV &= VD, \end{aligned} \tag{2.7}$$

where V and D are the matrices of eigen vectors and eigen values respectively, V is called the loading matrix which describe the amount of variance of the data matrix X that the principle components matrix V captures. The statistical representation T can also be achieved by calculating the SVD of the mean subtracted data, where

$$\text{if } B = U\Sigma V^T, \text{ then } T = U\Sigma. \tag{2.8}$$

Many truncation methods exist, and they are the same ones mentioned in the **SVD** section. Another common rule called the kaiser rule [91]. It says that the eigenvalues shows how important each component is and how much information it holds. It also says that an eigenvalue under 1 holds no more information than a single variable, and based on this rule, the **PCA** components under 1 are ignored.

On the other hand, **PCA** suffers from the fact that each component is a linear combination of all the original variables. Which made the interpretation of the results very difficult. To help with that, a new method called Sparse Principal Component Analysis (**SPCA**) made its apparition. It uses the **LASSO** (see section 2.3) to produce modified Principal Components with Sparse loadings [96].

Rotation techniques were also introduced to support practitioners with interpreting the Principal Components [47]. Neural Networks with Hebbian learning have been proposed for an adaptive **PCA** [16]. But, the performance of these techniques is heavily dependent on the learning parameters, which require a lot of time to compute and to be determined. In [69], a new approach was developed with the purpose of reducing the computation time, making it faster and easier, titled "simple **PCA**". It gives approximate solutions without the need of calculating a variance-covariance matrix, and is independent on the learning parameters.

PCA still has other problems, like the use of an unsupervised algorithm, and it doesn't take advantage of the label information when classifying. Therefore, another technique was proposed in [27], it is a discriminant analysis based on Fisher's criterion, named "**FDA**", with the aim to maximize between-class scatter while minimizing it within-class. However, it is only suitable for two-class classifications, and will find hard time facing multi-class problems. A newer method called "**LDA**" was later published in [75], as an extension of **FDA**. It solves the multi-class issue, but draws back when dealing with distributions more complex than Gaussian. And to overcome that, [68] established in 2007 a pairwise formulation for **LDA** titled "Neighborhood Min-Max Projection" (**NMMP**). Another problem with the **LDA** was that it needed a high enough amount of train data to manage the Small Sample Size Problem (**SSSP**) [30], a necessity that causes problems with small-scale data. But, it was dealt with later in [60], where authors proposed the Angle Linear Discriminant Embedding (**ALDE**). One other problem with **LDA** was that it was weak to outliers when it uses L2-norm in the objective function. [95] proposed a new formulation to make the method more robust against them.

An extension of **PCA** using local linear approach named Local **PCA** (**LPCA**) [48] was proposed, where results indicate higher overall performance of **LPCA** compared with **PCA** using speech and image dataset. Generalized **PCA** (**GPCA**) was introduced to treat the large data space with uncorrelated number of subspace [87]. Moreover, in [2], Constrative **PCA** was introduced aiming for a lower dimensional system from the original data and to get better perception. It showed good results in diffrent application fields. While **SPCA** was developed in [23], the technique uses a hybrid process based on robust and scalable algorithm.

2.2.1.3 Other linear Dimensionality Reduction techniques

An unsupervised linear Dimensionality Reduction technique was proposed [15], named Independent Component Analysis (**ICA**). It models the data as a linear mixture of non-Gaussian independent source and extracts the independent components from linear transformation of the original data. Another unsupervised linear feature extraction technique based on the linear approximation of non-linear Laplacian Eigen map was introduced under the name of Local Preserving Projections (**LPP**) [39], with the goal to fix variation problems and achieve optimal preservation of neighborhood structure of data.
In 2018, authors introduced a Locality-Regularized Linear Regression Discriminant Analysis (**LL-RDA**) [44], that was made possible by maximizing and minimizing the inter-class and intra-class reconstruction of local scatters respectively at the same time. While in [74], a new **ICA** method using the Copula-based Hoeffeding's measure of dependence was developed as a contrast function and it was applied as pre-processing in time series clustering.

2.2.2 Non-linear Dimensionality Reduction techniques

But still after all, these linear methods couldn't handle complex non-linear data. Which is why non-linear Dimensionality Reduction techniques were needed and introduced. Lately, non-linear Dimensionality Reduction techniques have been getting a great deal of attention, because they can process complex non-linear data, unlike their linear counterparts. This gives them an upper hand for real world applications as we often have to deal with complex and non-linear observations [94] [55].

2.2.2.1 KPCA

The most known non-linear method is called **KPCA** [77]. It works when some datasets that aren't linearly separable, can be made so if their attributes are projected into a higher dimensional space. Once there, we can apply **PCA** on these new linear attributes. By using arithmetic operations on the original space, we can make the attributes become linearly separable in the new space. This method works under the condition of minimizing the reconstruction error in the new attribute space, for a centered dataset $\Phi(x_i)$:

$$\min \sum_{i}^{t} \| \Phi(x_i) - U_q U_q^T \Phi(x_i) \| . \tag{2.9}$$

After projecting the data, we apply the **SVD** method on the centered features dataset $\Phi(X)$ in the high-dimensional space, where Φ is the mapping of the original dataset and the data projected in the higher dimensional space, therefore:

$$\Phi(X) = U \Sigma V^T, \tag{2.10}$$

where the eigen vectors of the covariance matrix are calculated using the following matrix product:

$$K = \Phi(X)\Phi(X)^T. \tag{2.11}$$

And the eigen vectors using the dot product matrix as in **PCA** [78]. The only requirement is that we calculate the dot product $\phi(x_i).\phi(x_j)$ efficiently. A common method for doing so, is to use of the Gaussian kernel scale of radial basis function as a way to describe locality between data points [71]. this last is defined as:

$$k(x_i, x_j) = \exp(\frac{- \parallel x_i - x_j \parallel^2}{2\sigma^2}), \tag{2.12}$$

where σ is the width parameter of the function, that can be calculated using cross-validation. For that, a precise rule indicates to start at the right side of the Scree Plot (the lowest eigen values) and look at the points that fit (approximately) on a straight line, going to the left we should look for the last point of that line, which will be our indicator of the number of components to retain. This rule and the scree plot were proposed by Cattell (1966) [9] and revised by Cattell and Jaspers (1967) [10].

However, **KPCA** suffers from extremely high computation time. Aiming to minimize that Subset **KPCA** (**SKPCA**) was developed [88]. While others have found away to dynamically replace old blocks of data under the name Block Adaptive **KPCA** (**BAKPCA**) [90].

2.2.2.2 Other non-linear Dimensionality Reduction techniques

Based on the specral theory, a non-linear dimentionality reduction technique was developed named **IsoMap** (Isometric Mapping) [85]. This method aims to preserve the geodesic distances in the lower-dimension. Another common unsupervised method is Locally Linear Embedding (**LLE**). It focuses on preserving only local properties of the data. These techniques in addition to another non-linear techniques were all studied against **PCA** [73]. They concluded that the non-linear methods of Dimensionality Reduction are still unable to perform better than the original linear technique **PCA**.

2.3 Feature selection

The other class of Dimensionality Reduction techniques is called feature selection. They are filtration techniques that rank the datasets attributes in sort of which to keep and which to eliminate. The most common selection techniques are the Sequential Backward **SB** and Forward Selection **FS**, the Sequential Backward technique **SB** [61] takes all the datasets attributes and start eliminating the irrelevant features one at a time based on the criterion function. While

the sequential forward selection start with an empty set and sequentially select features to add one by one. However, these techniques suffer from drawbacks, such as the large amount of time complexity $O(n^2)$ as a consequence of the greedy sequential search method. for that, authors proposed heuristic search methods. In [81], a genetic algorithm-based wrapped feature selection for medical classification was proposed. They concluded that their approach outperformed other methods in term of accuracy and algorithm running time. In [86], authors introduced a correlation-based attribute selection using Genetic algorithm as an optimal search tool for selecting subset of attributes. They proved that their approach is even more accurate when dealing with large datasets. There have been also work involving **SVD** for feature selection, as in [57], where authors proposed sparsified **SVD** to reduce the dimensionality more with a lower computing time named **S2R** method. In [37], authors introduced the StepWise selection (**SW**). The main approaches of this method are the Forward Selection **FS**, Sequential Backward **SB** and a combination of the two, it makes a base model and then allows features to enter and leave it one at a time following some criteria [53], until the program settles on a model that it finds best. It will then return the results as a set of features to keep. Another powerful technique named **LASSO** technique was proposed, it sums the absolute values of the model parameters, then regularizes them down until some of the variables coefficients reach zero. Once the process is done, the variables that still have a non-zero coefficient are chosen to be part of the final model [28]. The **LASSO** method can be formulated as the following optimization problem:

$$minimize||Y - X\beta||^2 + \lambda||\beta||_1, \tag{2.13}$$

where Y is the response variable, X is the design matrix of predictors, β is the vector of coefficients, and λ is the penalty parameter. The l_1 norm penalty in the second term promotes sparsity by encouraging many coefficients to be exactly zero, leading to a simpler and more interpretable model. However, the lasso method also has some limitations and challenges, such as: the optimal value of λ depends on the data and the goals of the analysis, and may require cross-validation or other tuning methods to be determined.

Similar method had been proposed in [19] named Least Angle Regression (**LARS**), and had been applied as a Dimensionality Reduction technique since then [59], where standard Least-Squares (**LS**) forward selection has the potential to be too greedy, and **LARS** tackles this issue by making it simple to specify the **LASSO** sequence of variables to enter the model.

2.4 Copulas based Dimensionality Reduction

Over the last few years, researchers showed interested in feature selection using Copulas functions. A greedy supervised feature selection method using multivariate Copula based mutual information was proposed [54], they compared their results against well-known mutual information-based feature selection methods. It displayed better performance in term of classification accuracy

and its noise tolerance property. In [63], authors developed a Copula-based Random Forest **RF** method to select the relevant features. After that, the selected features were classified to a label-valued outcome using a Random Forest **RF** algorithm. Similar work had been proposed to improve the the principal component analysis (**PCA**) method for Dimensionality Reduction [8] **MC-PCA**, where they fitted the data to the multivariate Copulas and simulate new data from it, the simulated data is then be reduced using **PCA** method. They concluded that their method enhance the performance of the principal component analysis method. However the efficiency is not tested and a classification accuracy of the obtained PCs is required.

The major problem in dealing with high-dimensional data is to find a way to measure dependence between variables without imposing constraints to estimate the marginal distributions. To deal with that, authors introduced An unsupervised linear Dimensionality Reduction method based on Copulas and **LU**- decomposition (**LU-C**) in [43]. This technique was introduced to measure the dependence between variables without the need to estimate marginal distribution in order to detect and eliminate redundant data, increase learning accuracy and improve decision making process with maintaining of the integrity of the original data. They used statistical and classification techniques to improve the effectiveness of their method, statically by measuring the standard deviation of reduced datasets, and classification by fitting the reduced data to the models: **ANN**, k-Nearest Neighbors (**k-NN**) and Naive Bayesian (**NB**). The goal of this paper is to develop sampling based Dimensionality Reduction technique that can deal with very high-dimensional datasets by taking account the heterogeneous aspects of the data and the integrity of the original information. This approach provides a way to use binary linear programming formulation to find a lower linear space of dimensions (column X_j) of the original matrix for maximization of redundant column. To identify and remove redundant variables from datasets, the authors introduced the following technique: Let X be a dataset defined in $n \times m$ matrix, X^i and X_j are the i^{th} row and j^{th} column of matrix X and let $Y = Y_1, Y_2, ..., Y_m$ be the decision variables that:

$$Y_i = \begin{cases} 1 & \text{if the redundancy of dimension } X_j \text{ is detected,} \\ 0 & \text{else.} \end{cases} \tag{2.14}$$

Using the Gaussian Copula defined in chapter 1, they measured the dependence (the correlation matrix Σ with $m \times m$ dimensions) between the columns and eliminate dependent columns (if Y_j=1, Y will be $m \times m$ data matrix containing 0/1 indicators) of the main matrix X and a threshold value of Σ is used to compare the dimensions $\{X_1, X_2, ..., X_m\}$. The optimal solution to reduce the dimensionality is to maximize the number of columns that will be eliminated, i.e the objective function is presented as follow:

$$Max(\sum_{j=1}^{m}(Y_j)), Y_j \in \{0,1\}, j = 1, ..., m. \tag{2.15}$$

Under the following constraints:

$$\begin{cases} \sum_{k \in B_c} \alpha_k X_k = 0 \iff \alpha_k \neq 0, \forall k \in B_c, \\ B_c = \{j \in \{1,...,m\} / y_j = 0\}, \\ Y_j \in \{0,1\}, \forall j \in \{1,...,m\}, \end{cases} \tag{2.16}$$

where $(m-k)$ presents the deleted dimensions, k the dimension of the subspace of B_c and α is a vector representing the coefficients of the linear combination of dimensions. The first constraint verifies the linear independence of the column $X_i, i = 1,...,k$ belonging to the subset Bc, the second one shows that for $i = 1,...,k$, X_i is not redundant and the third constraint shows that Y_j is the m-dimensional vector of binary decision variables. The method that the authors proposed is divided into two steps, step one consist on constricting the dependent sample subsets $S_{i,(i=1,...,k')}$, as follow:

1. Calculate the empirical Copula to visualize the dependence.
2. Determine the theoretical Copula using the data based on scatter plot of the empirical Copula and the marginal distributions of the datasets.
3. Analyze the dependence between the variables using the Copula's parameter and regroup dimensions having the strong correlation relationship (the correlation parameter ($|\theta| > 0.7$) in each sample subset $S_{i,(i=1,...,k')}$.

To define the coefficients of linear sample combinations and come up with a low linear space ($X_{i,i=1,...,k)}$ of the original matrix, in Step 2, the authors proposed the LU-decomposition (forward substitution) to solve the linear system equations $\alpha \times S' = C$, where C is a column vector in the dependent sample subsets $S_{i,(i=1,...,k')}$, α_j is an output vector representing the coefficients of the linear combination of dimensions and $S'_{i,(i=1,...,k'-1)}$ is a lower triangular matrix without column C, this system is defined as:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \\ \end{pmatrix} \begin{pmatrix} x_{11} & & & \\ x_{21} & x_{22} & & \\ \vdots & \vdots & \ddots & \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}.$$

Therefore,

$$\begin{cases} \alpha_1 = \frac{c_1}{x_{11}}, \\ \alpha_i = \frac{1}{x_{ii}}[c_i - \sum_{j=1}^{i-1} \alpha_j x_{ij}], i = 2,...,n. \end{cases} \tag{2.17}$$

The authors used 4 real-world datasets available in [17], to apply their approach and compare it with knowing Dimensionality Reduction techniques. Also, to improve the efficiency of the

proposed technique, they used statistical and classification methods. The first dataset (Pima Diabetes Database) contains 2 variables present the result of diabetes test where 268 (34.9%) tested positive and 500 (65.1%) tested negative. The second dataset (waveform Database) is presented in a matrix with 33367×21 where the variables are between 0 and 6. The third dataset (Human activity recognition using smart phone) presents a result of an experiment and it contains two different matrix with 7352×128 and 7352×384 dimensions, each matrix will be reduced separately. The fourth dataset (Thyroid disease diagnosis problem) contains measurements of the thyroid gland hormones presented in a matrix with 7200×561. To better visualize the dependence, the authors calculated the empirical Copula and plotted the result for the 5 different datasets and using the goodness-of-fit test, they concluded that all the datasets have the same distribution as the Gaussian Copula and it will be used as the theoretical Copula in their approach. The authors used several techniques to reduce the dimensionality and their approach in order to compare between the results of reduction obtained from the different technique, these results are presented in the table below:

Table 2.1: Dimensionality Reduction results (number of columns reduced).

	SVD	PCA	SPCA	LU-C
Pima Diabetes	0	4	2	5
Waveform	5	15	15	17
Human Activity 1	23	102	93	107
Human Activity 2	112	337	338	339
Thyroid Disease	180	500	519	520

The truncation of the three features extraction is made as follow: the r value for **SVD** is selected by capturing the only positive non-zero eigen-values on the diagonal matrix, while the Kaiser rule is used for **PCA**, and for **SPCA**, the reduction process is performed by retaining the number of non-zero loadings and the percent of explained variance. From Table 2.1, **SVD** gives the lowest number of Dimensionality Reduction, while **SPCA** and **PCA** work very well and have good result. However the proposed approach **LU-C** provides the best results of reduction with the 5 different datasets and it overcome the weakness of the other techniques. To describe the performance of the Dimensionality Reduction for all the methods, the authors used standard deviation of a set of results as a statistical precision tool, they concluded that the proposed approach performed better than the other 3 methods because it has the smallest bias and standard deviations are more stable, also it overcome the weakness of **SVD** and **PCA** when we are dealing with large database. To improve the effectiveness of dimentionality reduction and compare the proposed approach with the other methods, the authors used classification models **ANN**, **K-NN** and **NB** and they concluded that their approach yields the highest precision and lowest recall.

LU-C shows good results and reduces a high amount of redundancy compared to other powerful techniques. However, in the second stage of this technique, authors used LU-decomposition as

a tool to find the coefficients of linear sample combinations, this method gives a good solution but includes a lot of operation (n^2 operation), which takes time and past by a lot of operations, especially in the data setting. Also, the method doesn't covers non-linear dependency, which can lead to redundancy and noise in data. Additionally, **LU-C** is not reproducible since the data used are modified (the used datasets doesn't have the same dimensions as in [17]). Beside that, statistical precision is not suitable for feature selection techniques, since the input and output variables are the same. On the other hand, **LU-C** is an unsupervised feature selection technique compared against the feature extraction methods **SVD**, **PCA** and **SPCA**. The choice of truncation for **SVD** is not optimal, authors may consider using the technique explained in (2.2). Beside that, the comparison would be more appropriate to be performed against feature selection techniques since **LU-C** is a feature selection technique, such as: **LASSO**, **SW** and **LARS**.

2.5 Conclusion

This chapter gave a general overview about most common feature extraction and selection techniques to reduce dimensions of large data, and also, how authors used Copulas in the pre-processing field. We've seen that each method has strengths and weakness. Our goal is to overcome their weakness. Therefore, 2 feature selection techniques are proposed in the next chapter (chapter 3), and another feature extraction technique is proposed in chapter 4.

CHAPTER 3

FEATURE SELECTION BASED ON BIVARIATE COPULAS

3.1 Introduction

After we've learnt in the previous chapter how different Dimensionality Reduction techniques work, and the theory behind them. We'll have a much more practical approach in this chapter. Aiming to demonstrate the benefits of our 2 proposed approach: Bivariate Copulas-based Feature Selection (**BCFS**) and Grouped Bivariate Copulas-based Feature Selection (**GBCFS**). To do so, these methods are compared against similar well known methods. We take a more in-depth look at the algorithms behind the approaches, and apply all the former methods on different real life datasets. We then sort the results in order to observe their perks and strong points, how they face each individual dataset, and how they perform in real life situation. A discussion to analyse the results and to compare the methods will conclude the chapter while achieving its goal.

3.2 BCFS

Starting by the first proposed approach **BCFS**. As we explained in section 2.4, the Dimensionality Reduction technique based on Copulas and LU-decomposition (**LU-C**) proposed in [43] gives good results against well-known methods. However it includes a complex optimization problem and passes by a lot of operations, this leads to a long processing time ($O(m^2n^2)$). To improve that we propose a new filtering method with less complex model and time complexity, it is built using an algorithm programmed in R, and uses bivariate Copula as a tool to detect redundancy between each two attributes in order to eliminate one of them. **BCFS** will have to outperform other methods of Dimensionality Reduction by having more accuracy, and better reduction of the data.

3.2.1 The method

This section focuses on introducing the **BCFS** technique using the bivariate Copulas and the algorithm behind it.
Let X be the input matrix of $n \times m$ dimensions containing redundant variables. In order to transform the matrix X's attributes into random variables between $[0,1]$. We use the pseudo observation transformation defined in (3.1) by forcing the variates to fall inside the open unit hypercube.

$$u_{ij} = \frac{r_{ij}}{n+1}, \tag{3.1}$$

where $i \in \{0,...,n\}$, $j \in \{0,...,m\}$ and r_{ij} denotes the rank of X_{ij} among all X_{kj}, where $k \in \{0,...,n\}$. Next, in order to visualize the dependency between the pairs attributes, we use (1.51) to calculate and plot the bivariate empirical Copulas for each pair of attributes After that we follow these steps:

1. Determine the bivariate theoretical Copulas for each pair using the data based on the scatter plot of the bivariate empirical Copula and the marginal distributions of the datasets.
2. Pick the first pair of attributes.
3. Calculate Kendall's tau τ.
4. Deduce the bivariate theoretical Copula's parameter using (1.36).
5. Eliminate one of the correlated attributes if $|\theta| >= 0.5$, where θ is the Copula's parameter. Otherwise skip to the next pair of attributes and go back to step 3.
6. After all the attributes are tested, we get a new reduced data as output with uncorrelated attributes holding the same information as the input matrix.

Algorithm 1: Dimensionality Reduction using **BCFS**.

Input: Dataset matrix X.
Output: Matrix of reduced dataset X.

```
1 begin
    θ=NULL.
    for i := 1 to m do
2     for j := 1 to m do
3       θ_ij = sin(π/2 × τ_ij).
4       if |θ_ij| >= 0.5 then
5         Delete one of the attributes.
6 end
```

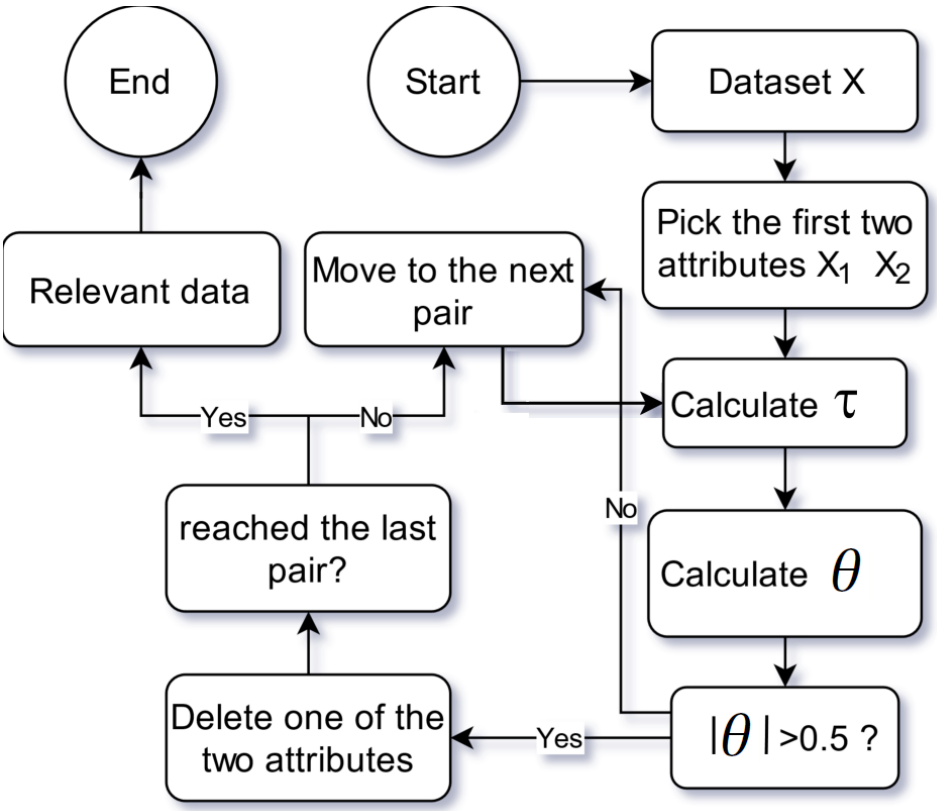


Figure 3.1: Flowchart of **BCFS**.

Figure 3.1 and Algorithm 1 represent the **BCFS** method. Taking as an input the data matrix. This algorithm checks for inter-correlation between two attributes and eliminate one attribute each time correlation is detected.

The choice of which attribute to eliminate between the two is random, as the first one detected will be directly flagged for elimination. We then apply the same procedure to all the possible pairs, leaving us with new relevant and uncorrelated datasets representing the same information as the input matrix. We used the relationship given in (1.36), because the Gaussian Copula correspond to our dataset (see section 3.4 for more details).

To improve our algorithm, and to reduce its time complexity, we use a method proposed in [50] and has been described with more details in [3] and [14] named Fast Kendall's tau instead of the commonly used method for calculating the Kendall's tau τ (time complexity of $O(n^2)$). It uses a process called sorting by exchanging that decreases the time complexity to $O(n \log n)$. The equation of Fast Kendall's tau is presented in (3.2).

$$\tau = \frac{4A}{n(n-1)} - 1, \tag{3.2}$$

where A defines the concordant pairs. This leads to an initial time complexity of $O(m^2 n \log n)$ for the entire Algorithm 1. But, due to the nature of this algorithm, the time complexity is variable and decreases each time an attribute is eliminated. The memory complexity on the other hand is $O(m \times n)$. In Figure 3.2, we can see an illustration of how our approach **BCFS** treats the data, using the matrix X as an input for Algorithm 1, we eliminate k redundant attributes where $1 \leq k \leq m-1$. As an output we get a reduced and relevant data where $1 \leq l \leq m-k$.

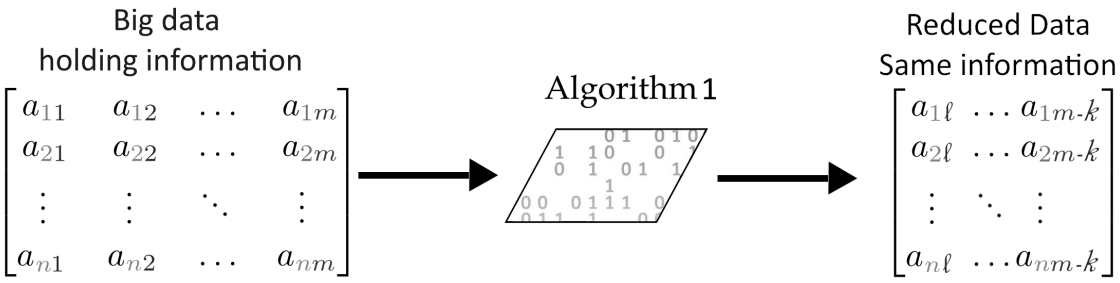


Figure 3.2: Illustration of **BCFS**.

3.3 GBCFS

In this section, we discuss the second proposed approach **GBCFS** in more details published in [24], and see how the mathematical assets are combined with our program to achieve the desired outcome.

3.3.1 The method

Let X be a matrix of $n \times m$ dimensions. **GBCFS** is an optimization work of the previous method **BCFS**. This time, instead of directly deleting one of the detected correlated attributes (as we've seen in the previous section), we take one given attribute X_k, where $k \in \{1,...,m\}$, and group all attributes that are correlated with it. We then do the same for the rest of $k \in \{1,...,m\}$, and class all these groups in one vector. Afterwards, we choose the largest group (the one containing the most correlated elements) and eliminate all attributes correlated with X_k from the data matrix X, we then redetermine the next largest group and keep deleting until no correlation is left in the data. The whole process follows these steps:

1. Determine the bivariate theoretical Copulas for each pair of attributes.
2. Measure the bivariate theoretical Copula's parameter θ between the attribute X_k where $k \in \{1,...,m\}$ and the rest of the attributes $\{X_1,...,X_{k-1},X_{k+1},...,X_m\}$.
3. If $|\theta| >= 0.5$ (correlation detected), store the correlated attributes with X_k in G_k, where G_k is the group of the attributes that are correlated with X_k, and G represents all the groups of the correlated attributes.

4. Pick the largest group in G, named G_l.

5. Eliminate all the attributes in the data matrix X and the groups G, which were selected in the group G_l.

6. If there are groups left in G, go back to step 4, otherwise go to the next step.

7. Obtain a new uncorrelated output matrix X of $n \times (m-p)$ dimensions, where p is the number of deleted attributes after all the process is done, this new matrix holds the same information as the input matrix X.

In step 1, in order to fit the data to the theoretical Copula, the attributes must be random variables between $[0,1]$. To do so, we use (3.1) (as in **BCFS** technique). While in step 2, the Copula's parameter is determined using the relationship between Kendall's tau τ and the Copula's parameter. For a less time complexity, (3.2) is used (as in **BCFS** technique).
Algorithm 2 and Fig 3.3 clarify the proposed method **GBCFS**, taking as an input a large data

Algorithm 2: Dimensionality Reduction using **GBCFS**.

Input: Dataset matrix X.
Output: Matrix of reduced dataset X.

```
1 begin
    G <- −m × m matrix.
    θ=NULL.
    l=NULL.
    for i := 1 to m do
2     for j := 1 to m do
3       Calculate the Copula's parameter θ_ij.
        if |θ_ij| >= 0.5 then
4         G_ij = j
5 for i := 1 to m do
6     Pick G_l the largest group in G.
      for j := 1 to m and j ∈ G_l do
7     eliminate X_j from X.
        eliminate G_j from G.
8 end
```

matrix X_{nm} containing redundant attributes, and an output of only relevant data matrix $X_{n(m-p)}$ clear of p redundant features, and holding the same information as the input large data matrix X. The time complexity of this algorithm is equal to $O(m^2 nlogn)$, with memory complexity of $O(m \times n)$.

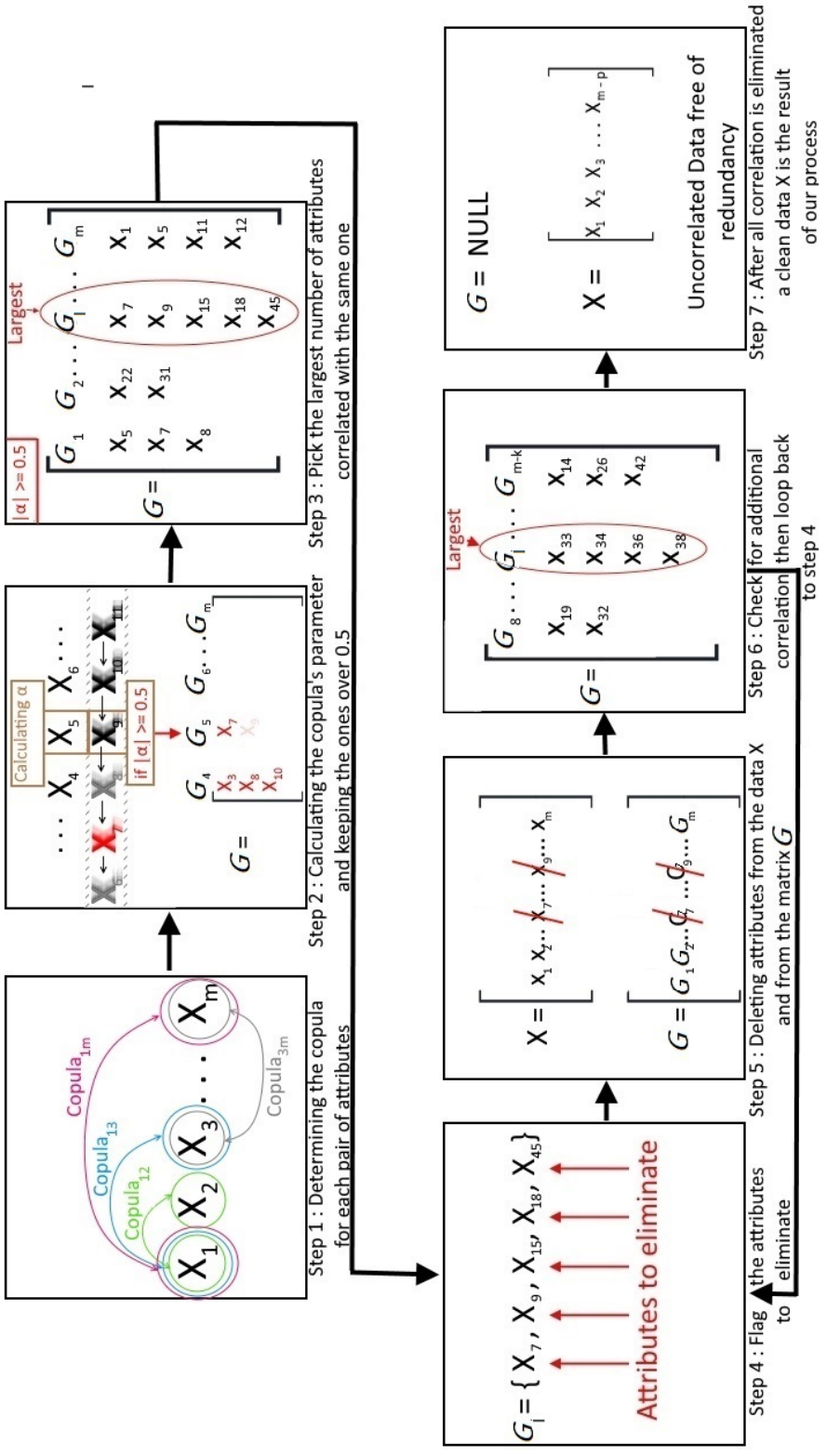


Figure 3.3: Illustration of **GBCFS**.

3.4 Experimental results

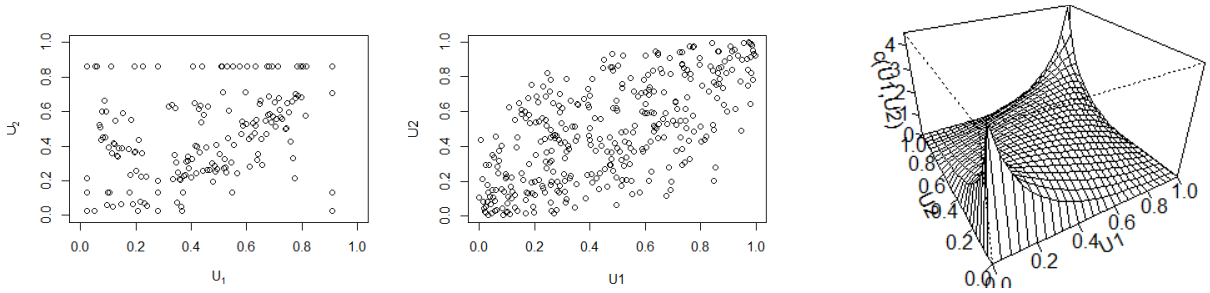
The aim of this section is to show the performance of the two proposed method **BCFS** and **GBCFS** by implementing it to real world datasets, we firstly apply the methods, and then compare it against each other and other methods introduced in chapter 2, which are, the unsupervised technique **LU-C**, and the supervised techniques: **LASSO**, **SW** and **LARS** in term of reduction and classification accuracy. All results shown below were taken out from simulations run on RStudio using R version 4.1.3 [9] (64bit), and a PC with the following specs: CPU: Intel Core i5-9300H (4 Cores, 8 Threads, up to 4.10 GHz), RAM: 8GB DDR4 (2666 MHz), GPU: GTX 1050, Disk: SSD and OS: Win 10 (64bit). The datasets Ionosphere, Sonar, Wpbc and Waveform were selected from UCI machine repository [17]. While Scene datasets was obtained from LIBSVM repository. A short description about the data is given in Table 3.1.

Table 3.1: Summary of the used datasets.

Data	No. rows	No. of attributes	No. of class
Ionosphere	351	34	2
Sonar	208	60	2
Wpbc	198	32	2
Waveform	5000	50	3
Scene	2407	294	14

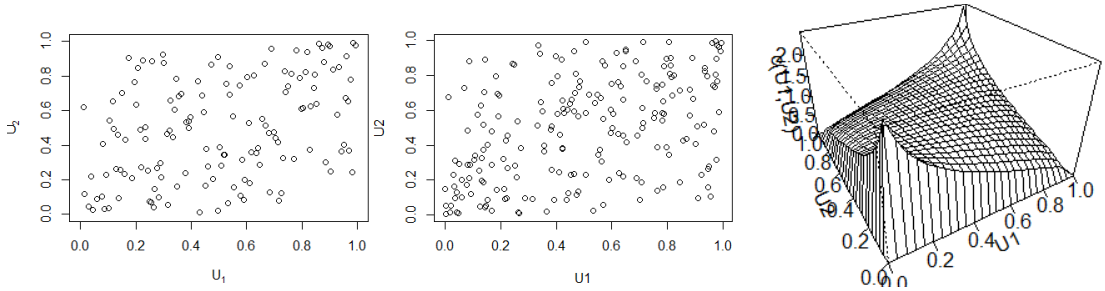
3.4.1 Fitting to Copulas

Starting from step 1 of our method as shown in Figure 3.1 and 3.3, we fit our data to several Copulas using the package "Copula" [41]. Figures 3.4, 3.5,3.6, 3.7 and 3.8 represent the scatter-plots of the empirical Copula generated using (1.51) (3.4a,3.5a, 3.6a, 3.7a and 3.8a), of the theoretical Gaussian Copulas generated using (1.32) (3.4b,3.5b, 3.6b, 3.7b and 3.8b), and of the Gaussian density Copulas using (1.34) (3.4c,3.5c, 3.6c, 3.7c, 3.8c) for the Ionosphere, Sonar, Wpbc, Waveform and Scene datasets respectively. From these scatter-plots, we can assume that the Gaussian Copula is the most suitable for these five datasets. To confirm our theory, we run the goodness of fit test [35] on the bivariate empirical Copulas and the bivariate theoretical Gaussian Copulas. This process verifies that our choice of Copulas is the most appropriate for all the datasets.



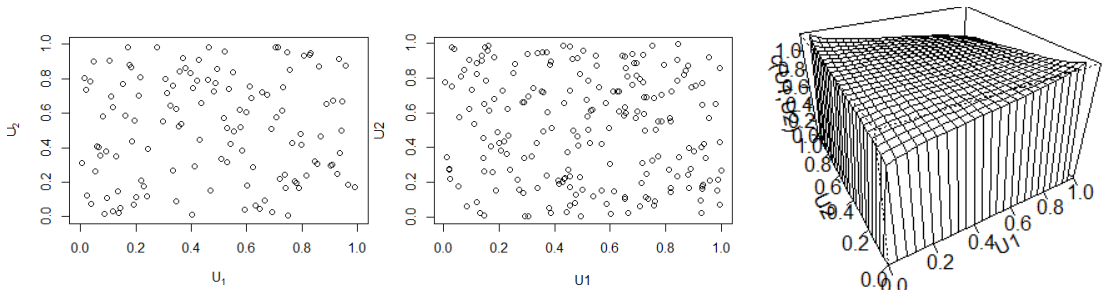
(a) Empirical Copula. (b) Normal theoretical Copula. (c) Normal Copula density.

Figure 3.4: The attributes pair (X_4, X_{22}), $\theta = 0.655$ of Ionosphere dataset.



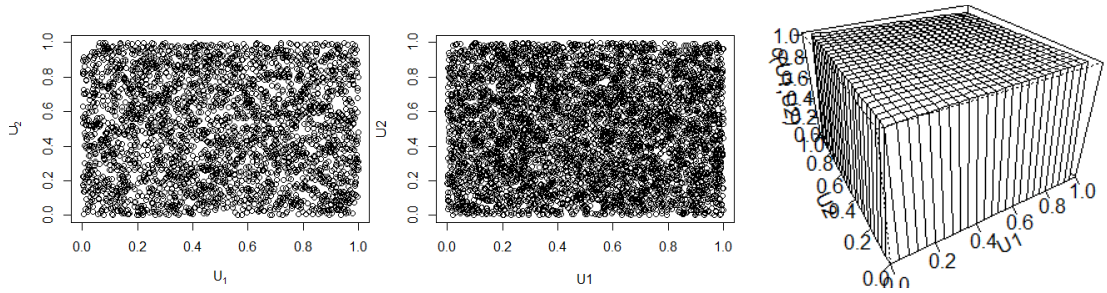
(a) Empirical Copula. (b) Normal theoretical Copula. (c) Normal Copula density.

Figure 3.5: The attributes pair (X_7, X_{15}), $\theta = 0.377$ of Sonar dataset.



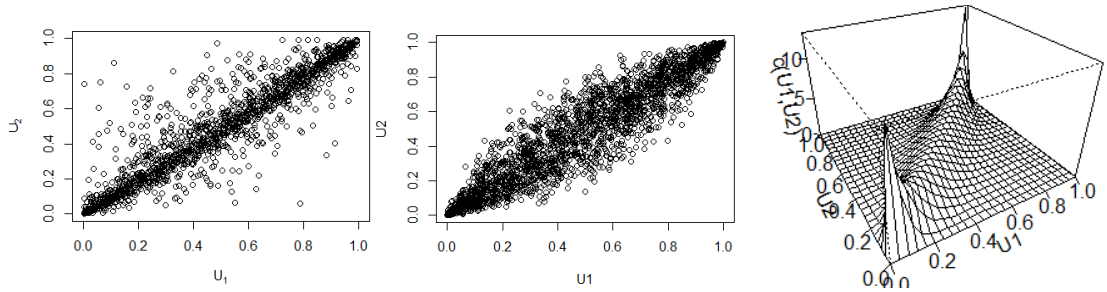
(a) Empirical Copula. (b) Normal theoretical Copula. (c) Normal Copula density.

Figure 3.6: The attributes pair (X_6, X_{25}), $\theta = -0.055$ of Wpbc dataset.



(a) Empirical Copula. (b) Normal theoretical Copula. (c) Normal Copula density.

Figure 3.7: The attributes pair (X_1, X_2), $\theta = -0.007$ of Waveform dataset.



(a) Empirical Copula. (b) Normal theoretical Copula. (c) Normal Copula density.

Figure 3.8: The attributes pair (X_1, X_2), $\theta = 0.940$ of Scene dataset.

3.4.2 Dimensionality Reduction

After selecting the best Copula for our data, we apply the two Algorithms 1 and 2, we use the package "pcaPP" [26] to perform the fast Kendall's tau τ in order to increase the algorithm's speed. We run the other unsupervised reduction method: **LU-C** technique, and the supervised methods: **LASSO**, **SW** and **LARS** using the package "lars" [38], where the hyper-parameter is tuned by selecting the smallest rmse for each model using 10 cross-validation method, and the StepWise selection **SW** with the best model criteria [53]. In order to investigate the efficiency of the proposed method, we compute the running time for each methods. The obtained results are given in Tables 3.2 and 3.3 for the unsupervised and supervised methods respectively.

Table 3.2: Dimensionality Reduction using unsupervised methods.

Datasets	Original data	GBCFS		BCFS		LU-C	
	D	D	T	D	T	D	T
Ionosphere	34	7	0.134s	11	0.132s	22	0.259s
Sonar	60	20	0.279s	25	0.200s	37	1.423s
Wpbc	32	7	0.092s	9	0.060s	14	0.250s
Waveform	50	27	1.579s	28	1.461s	38	7m 9s
Scene	294	30	36.44s	40	10.18s	104	25m 47s

Table 3.3: Dimensionality Reduction using supervised methods.

Datasets	Original data	LASSO		SW		LARS	
	D	D	T	D	T	D	T
Ionosphere	34	14	0.0508s	21	0.558s	14	0.149s
Sonar	60	27	0.131s	24	0.476s	28	0.217s
Wpbc	32	10	0.056s	15	0.255s	10	0.125s
Waveform	50	33	0.056s	29	3.162s	32	0.142s
Scene	294	128	20.28s	68	35.10s	134	24.17s

3.4.3 Classification accuracy

After reducing the dimensions, three classification models were used to demonstrate the performance of the proposed approaches **BCFS** and **GBCFS**, and also the other methods. These models are: "Artificial Neural Network **ANN**" [4], "Random Forest **RF**" [56] and "AdaBoost **AB**". This simulation was achieved using the following R packages: "neuralnet" [29], "caret" [52], "dplyr" [89] and "fastAdaboost" [12]. The reduced data are normalized and shuffled to reduce the risk of over-fitting, and the models are performed using 10 folds cross-validation in order to obtain the most verifiable results. Table 3.4 summaries the obtained mean accuracy values of the 10-folds for each model.

Table 3.4: The Values of model accuracy.

Models	Data	Original	GBCFS	BCFS	LU-C	LASSO	SW	LARS
ANN	Ionosphere	0.834	**0.918**	0.891	0.858	0.850	0.854	0.849
	Sonar	0.719	0.806	**0.816**	0.726	0.790	0.775	0.769
	Wpbc	0.742	**0.871**	0.824	0.773	0.751	0.695	0.751
	Waveform	0.832	**0.835**	0.833	0.824	0.827	0.814	0.820
	Scene	0.628	**0.665**	0.656	0.604	0.625	0.635	0.617
RF	Ionosphere	0.926	0.923	**0.934**	0.918	0.932	0.923	0.931
	Sonar	0.827	0.808	**0.827**	0.802	0.800	0.798	0.803
	Wpbc	0.793	**0.808**	0.803	0.798	0.793	0.763	0.793
	Waveform	0.816	**0.827**	0.826	0.818	0.820	0.824	0.816
	Scene	0.708	**0.754**	0.735	0.668	0.708	0.702	0.717
AB	Ionosphere	0.911	0.920	**0.931**	0.914	0.918	0.909	0.910
	Sonar	0.779	0.812	**0.831**	0.800	0.802	0.765	0.798
	Wpbc	0.693	**0.799**	0.792	0.785	0.726	0.718	0.726
	Waveform	0.826	**0.852**	0.840	0.812	0.826	0.824	0.816
	Scene	0.631	**0.644**	0.636	0.612	0.622	0.589	0.599

3.4.4 Discussion

After running our two algorithms (**BCFS** and **GBCFS**), we get the results given in Table 3.3. They indicate that **BCFS** eliminated a lot of redundant attributes in a low amount of time. It deleted 23 attributes from the Ionosphere datasets, 35 attributes from the Sonar datasets, 23

attributes from the Wpbc datasets, 22 attributes from the Waveform datasets and 254 attributes from the Scene datasets. Leading us to a new reduced matrix of 11, 25, 9, 28 and 40 dimensions for those datasets respectively. On the other hand, GBCFS reduced a large amount of redundancy in also a low amount of time. As it eliminated 27 attributes from the Ionosphere datasets, 40 attributes from the Sonar datasets, 25 attributes from the Wpbc datasets, 23 attributes from the Waveform datasets and 264 from the Scene datasets. Leaving us only with 7, 20, 7, 27 and 30 relevant attributes for those datasets respectively. Table 3.4 shows the accuracy of the results obtained above. It demonstrates the efficiency of both **BCFS** and **GBCFS**, as we notice an improvement in the accuracy compared to the original data for most of the simulation values in all three different classification models for 2 and multi-class datasets.
Alongside our methods, we ran four other feature selection techniques in the simulations in order to compare the reduction results and the efficiency. As shown in Table 3.2 and 3.3, these methods reduced an important amount of redundancy. However, **GBCFS** reduced the most, thanks to its advantages over the other methods. Which consists on the fact that **BCFS** and **LU-C** methods eliminate the attributes randomly and **GBCFS** doesn't. While **LASSO**, **LARS** technique and **SW** selection are supervised methods and the proposed techniques aren't. We also notice that the supervised techniques are fast for small and large data, where their time complexity depends on tuning the hyper-parameter. Contrarily to **LU-C** method, as **LU-C** technique has the highest computational time of O(m^2n^2). In Table 3.4, we can observe a higher accuracy on most models with the **GBCFS** method, indicating an improvement after the reduction of attributes. We do notice some falling behind in few accuracy values indeed, but the difference is really small, and the majority of values are higher with the **GBCFS**. From this point, we can affirm with confidence that the proposed methods **BCFS** and **GBCFS** are efficient techniques for reducing dimensions, and improving the accuracy of the classification models. It also reduce the computational time of the models since it provide us with a smaller and more relevant version of the data.

3.5 Conclusion

In this chapter, we proposed two new unsupervised non-linear filtering feature selection techniques under the name of **BCFS** and **GBCFS**. This techniques eliminate redundant attributes of large data based on inter-correlation, which is detected using bivariate Copulas. The proposed method **GBCFS** is an improvement of **BCFS** technique, as it offers a non-random elimination of attributes, which is determined in a way to optimize and achieve better results. **GBCFS** was compared against **BCFS**, **LU-C**, **LASSO**, **LARS** and **SW** selection in term of Dimensionality Reduction and computational time by applying them on five real-world datasets, and in term of accuracy using different classification models. The results indicate that **GBCFS** performs better in most situations. We see an improvement in the quality of reduction since we introduced the grouping algorithm, providing us with more reduction as well as a better accuracy than **BCFS**

method (which means increased efficiency), and making this method the best in most of the obtained results. Despite not being the highest in very few classification results, it stays really close to the best, and it is a negligible drawback compared to the important gain in Dimensionality Reduction. All of this made **GBCFS** the right method to optimize the pre-processing step in data mining, by capturing only relevant information and cleaning the data from redundancy, which in turn improves data analysis, machine learning algorithms, and the performance of models with an important computational time.

CHAPTER 4

FEATURE EXTRACTION BASED ON BIVARIATE COPULAS

4.1 Introduction

Beside the two feature selection techniques **BCFS** and **GBCFS**, we developed an unsupervised feature extraction technique named **BCFS** based **PCA** (**BCFS-PCA**) published in [25]. The method aims to improve the perfomance of **PCA** in term of reduction and information extraction. The proposed method is compared against the baseline method **PCA** and another method that combines multivariate Copulas and **PCA** to see how it improves **PCA**, and also against the feature extraction technique **SVD** and **KPCA**, where these four last methods are introduced in chapter 2. The comparison is made using real world data according to the Dimensionality Reduction, and the classification accuracy using Random Forest **RF** model of the new reduced data.

4.2 Methodology

This section aims to explain the steps we followed for the proposed technique. The method consists on following a two-stages process by combining **BCFS** and **PCA** for Dimensionality Reduction **BCFS-PCA**. We start by the **BCFS** technique to select only relevant attributes. For that, a theoretical bivariate Copula estimation for each two attributes of the data matrix X is needed. We choose the bivariate Gaussian Copula expressed in (1.34), which corresponds to our experimental results. The correlation coefficients between the variables are estimated using (1.36).

This relationship facilitates the definition of the inter-correlation between the variables as explained in chapter 1, which leads to eliminating correlated attributes and reduces redundancy in the data X. After that, the second stage consist on performing the **PCA** method to the reduced

data using **BCFS** technique, as explained in subsection 2.2.1.2. An illustration of the proposed method is given in Figure 4.1.

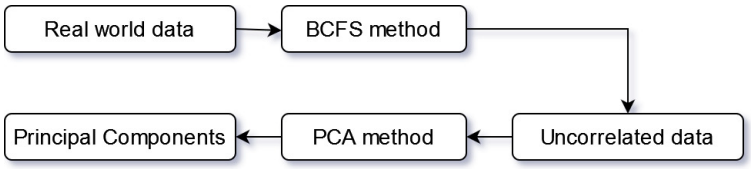


Figure 4.1: An illustration of **BCFS-PCA**.

4.3 Experimental results

In order to investigate the performance of the the proposed technique **BCFS-PCA** against **MC-PCA**, **PCA**, **SVD** and **KPCA**, we implemented it to two real-world datasets (small and large data). Then, the obtained Principal Components are compared with the help of the diagram scree plot, but also with their accuracy value through the Random Forest **RF** classifier.

4.3.1 Small data

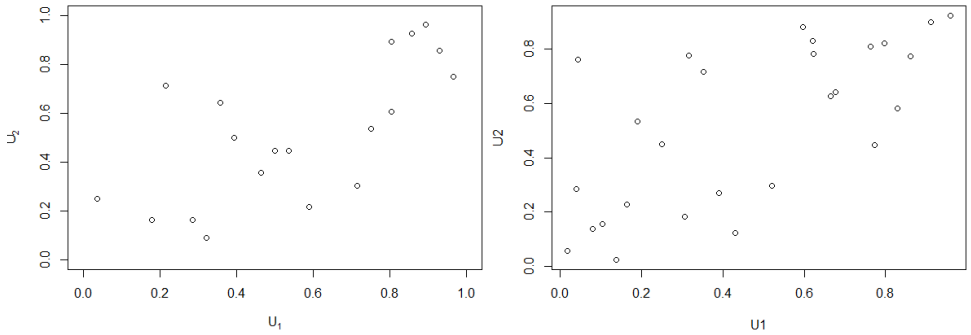
In this part, we used a small data for a more in-depth look into the correlation matrix and the process of the selection of variables for both methods **BCFS-PCA** and **MC-PCA**, which is much clearer in a data with less attributes.

Decathlon2 dataset: The Decathlon2 dataset is a matrix of 27 observations and 13 attributes available in the package "factoextra" [49], where the last attribute corresponds to the class column. This attributes present athletes performance during two sporting meetings, where the columns from 1 to 10 define the scores of the athletes for the 10 decathon events. The next 2 attributes correspond to the rank and the points earned respectively, and the final column contains a category variable for the athletic event: 2004 Olympic Game and 2004 Decastar.

4.3.1.1 Fitting to Copulas

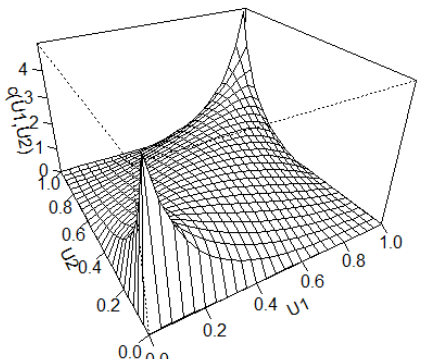
Using the package "Copula" [41], we fit the data to different Copulas. Figures 4.2a, 4.2b and 4.2c show the plots of the empirical Copula density, the Gaussian theoretical Copula and the Gaussian empirical Copula of the decathlon2 dataset, which illustrate a good fit to the Gaussian Copula for this data. In order to confirm that, a goodness of fit test [35] is performed on the multivariate estimated Gaussian Copula, and each estimated bivariate Gaussian Copula.
Table 4.1 presents the correlation matrix of the Gaussian Copula corresponding to our dataset. These coefficients were obtained using (1.36). **BCFS** reduces dimensions by eliminating the correlated attributes from the original dataset. Therefore, using the correlation matrix, the method eliminates these 7 redundant attributes: $\{X_2, X_4, X_5, X_6, X_7, X_{11}, X_{12}\}$, leading to a new

reduced matrix with 5 columns: $\{X_1, X_3, X_8, X_9, X_{10}\}$. On the other hand, the MC method eliminates the non correlated attributes from the generated data of the Gaussian Copula, based on the estimated correlation matrix. It deletes the variables: $\{X_9, X_{10}\}$ and selects the correlated variables: $\{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{11}, X_{12}\}$.



(a) Empirical Copula.

(b) Normal theoretical Copula.



(c) Normal Copula density.

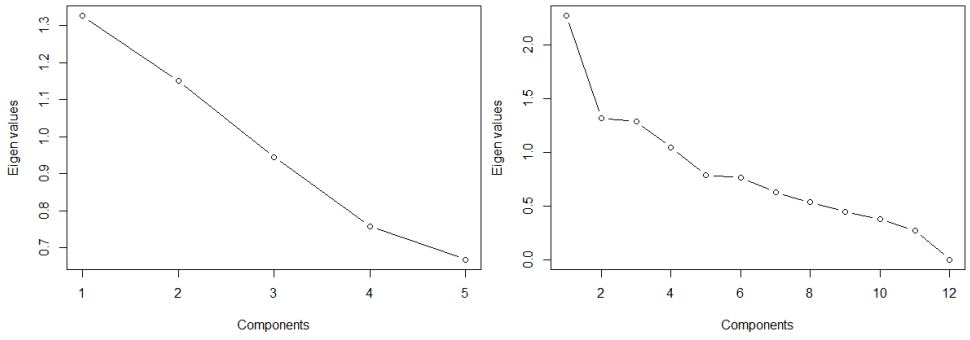
Figure 4.2: The attributes pair (X_1, X_6), $\theta = 0.697$ from "decathlon2" dataset.

Table 4.1: Estimated Gaussian Copula's parameter for Decathlon2 datasets.

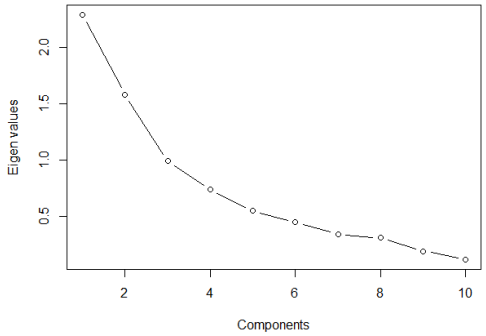
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
X_1	1	**-0.71**	-0.36	-0.32	**0.54**	**0.70**	-0.28	0.10	-0.16	-0.16	0.41	**-0.78**
X_2		1	0.36	0.22	**-0.50**	**-0.58**	0.18	0.07	0.09	0.31	**-0.53**	**0.66**
X_3			1	**0.64**	-0.12	-0.21	**0.73**	-0.23	0.48	-0.03	**-0.58**	**0.74**
X_4				1	-0.25	-0.17	0.46	**-0.57**	0.20	-0.19	**-0.50**	**0.51**
X_5					1	**0.70**	-0.14	0.13	0.12	0.23	0.45	**-0.54**
X_6						1	-0.42	0.06	0.28	-0.12	0.37	**-0.55**
X_7							1	-0.39	0.27	0.07	**-0.61**	**0.56**
X_8								1	0.02	0.39	-0.04	-0.01
X_9									1	-0.04	-0.40	0.44
X_{10}										1	-0.30	0.01
X_{11}											1	**-0.76**
X_{12}												1

4.3.1.2 Performing PCA.

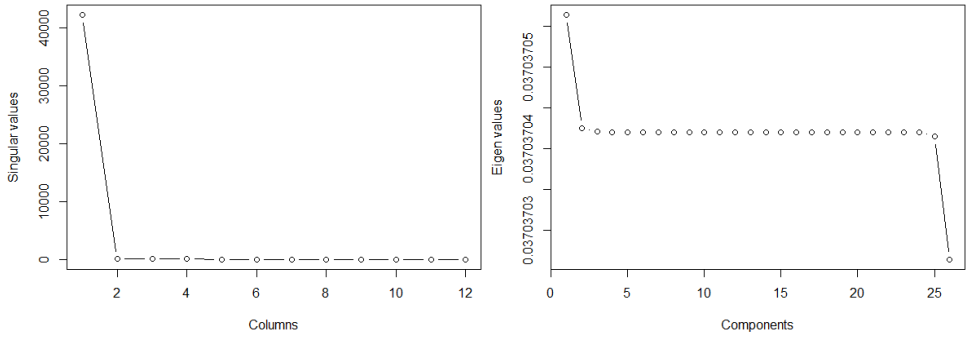
After fitting the data to the corresponding Copula, and reducing the dimensions, we proceed to the next step and perform **PCA** to the reduced data using both methods and the original dataset, and also **SVD** and **KPCA**. Table 4.2 and Figure 4.3 represent the ratios of population variance related to principal components and the scree plot respectively.



(a) **BCFS-PCA.** (b) **PCA**.



(c) **MC-PCA**.



(d) **SVD**. (e) **KPCA**.

Figure 4.3: Scree plot of decathlon2 datasets.

Table 4.2: **PCs** of Decathlon2 datasets.

	PC1	PC2	PC3	PC4	PC5
BCFS-PCA	1.3270483	1.1504737	0.9452370	0.7582466	0.6685374
PCA	2.27262402	1.32180748	1.29068176	1.04960508	0.78881606
MC-PCA	2.2878610	1.5811571	0.9888946	0.7331171	0.5447678
SVD	4225.095	82.89466	24.13306	22.39354	11.95112
KPCA	0.03703705	0.03703704	0.03703704	0.03703704	0.03703704

	PC6	PC7	PC8	PC9	PC10
BCFS-PCA	/	/	/	/	/
PCA	0.76774029	0.63021515	0.53611073	0.45157436	0.38070509
MC-PCA	0.4462264	0.3371441	0.3046095	0.1864597	0.1146765
SVD	6.430171	2.286536	1.604018	1.133431	0.8659114
KPCA	0.03703704	0.03703704	0.03703704	0.03703704	0.03703704

	PC11	PC12	
BCFS-PCA	/	/	/
PCA	0.27450132	0.00512123	/
MC-PCA /	/	/	
SVD	0 0.7247563	0.1520537	/
KPCA	0.03703704	0.03703704	...

4.3.1.3 Classification accuracy

In order to measure the performance of our approach against the other methods, we perform a classification method for accuracy. We do that by using the model Random Forest **RF** [56] through the package "caret" [52]. We firstly shuffle and normalize the reduced datasets to minimize the risk of over-fitting, **RF** is then performed under 10 folds cross-validation to obtain the most truthful results. The accuracy values of the models are given in Table 4.3, and plotted to visualize the performance in Figure 4.4.

Table 4.3: **RF** accuracy of decathlon2 dataset.

No. attributes	BCFS-PCA	PCA	MC-PCA	SVD	KPCA
1	0.6333	0.5185	0.4814	0.5701	0.5073
2	0.7037	0.4667	0.556	0.6583	0.6416
3	0.7167	0.5667	0.4583	0.6667	0.6583
4	0.7583	0.6917	0.4417	0.6833	0.7083
5	0.8148	0.6833	0.5083	0.7333	0.6917
6	/	0.7250	0.5750	0.7167	0.6667
7	/	0.7917	0.6417	0.7500	0.7250
8	/	0.7750	0.5083	0.8083	0.7333
9	/	0.7407	0.5167	0.7750	0.7000
10	/	0.7778	0.5583	0.8000	0.7250
11	/	0.7917	/	0.7333	0.7167
12	/	0.8083	/	0.7917	0.7667
...	/	/	/	/	...

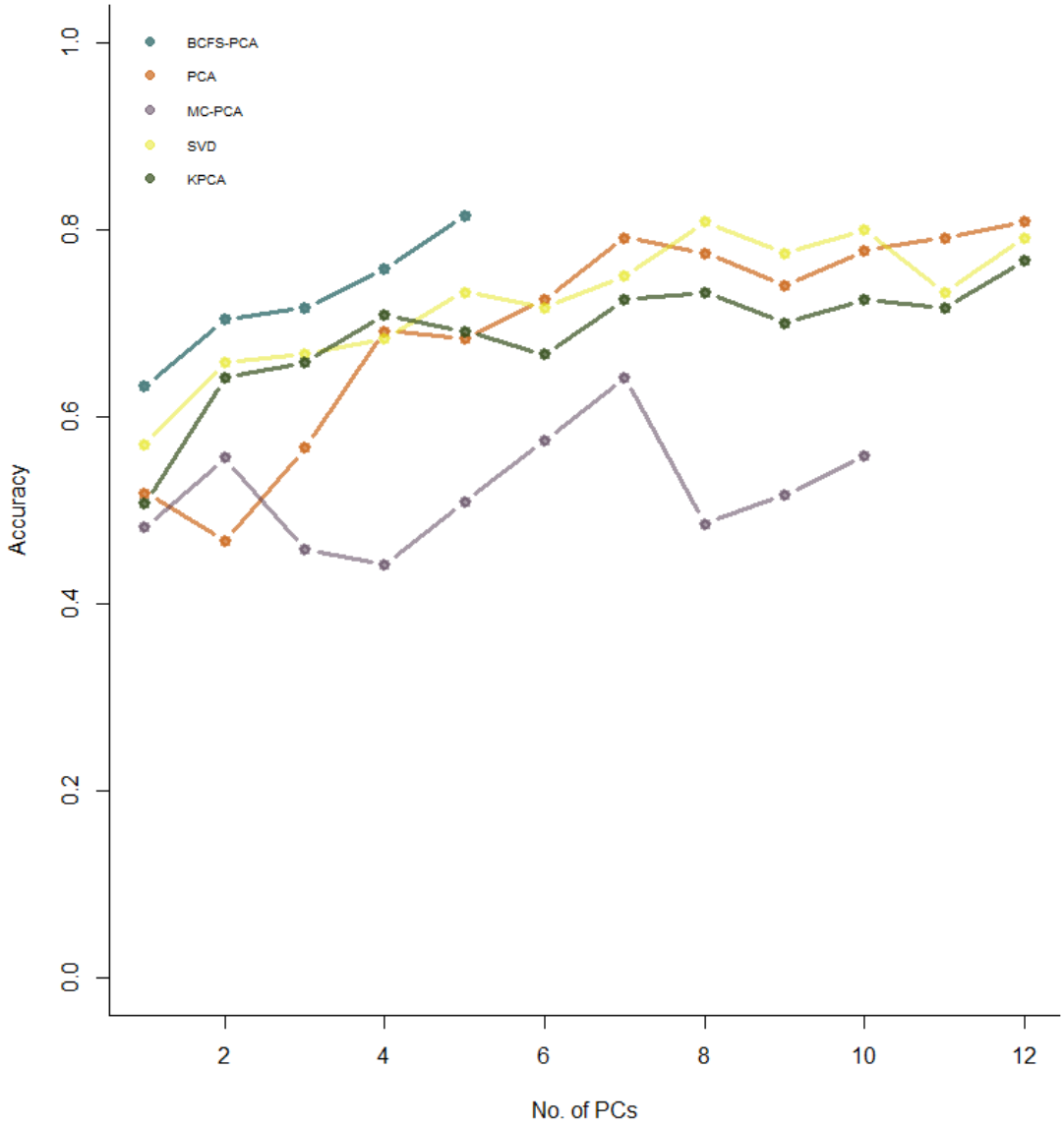


Figure 4.4: Accuracy of decathlon2 dataset.

4.3.2 Large data

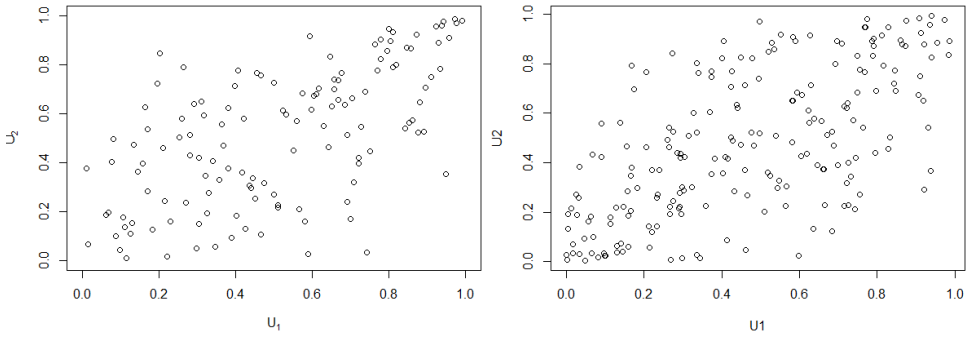
To see how the method performs on larger data, we introduce the next dataset:

Sonar datasets: The "Sonar" data is composed of 111 patterns available at [17], obtained by bouncing Sonar signals off a metal cylinder from different angles. The "Sonar.rocks" data contains 97 other patterns obtained from rocks instead. Each pattern is described by 60 numbers ranging from 0.0 to 1.0. The label associated with each observation contains the letter "R" for rock and

"M" for the metal cylinder. The labels numbers on the other hand, are in an increasing order depending on the aspect angle.

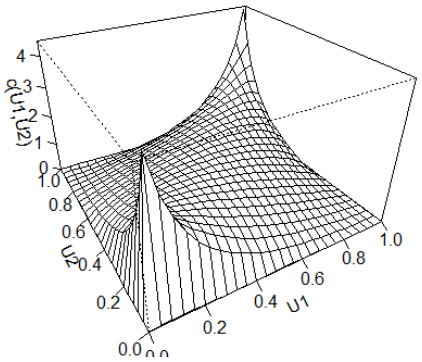
4.3.2.1 Fitting to Copulas

As we did for decathlon2 datasets, we fit the data to the corresponding Copula which in this case is the Gaussian Copula. Figures 4.5c, 4.5b and 4.5a represent the plots of the Gaussian Copula density, theoretical Copula and the empirical Copula of the Sonar datasets for the couple (X_2, X_3). While Table 4.4 shows the reduction results after applying the **BCFS** and **MC** method.



(a) Empirical Copula.

(b) Normal theoretical Copula.



(c) Normal density Copula.

Figure 4.5: The attributes pair (X_2, X_3), $\theta = 0.659$ from "Sonar" dataset.

Table 4.4: Selected variables for Sonar dataset.

Datasets	Original data	BCFS	MC
Sonar	60	25	54

4.3.2.2 Performing PCA

In this part, we apply **PCA** to the output of both **BCFS** and **MC**. Table 4.5 and Figure 4.6 show the ratios of the population of the Principal Components, and the scree plot, using the five methods **BCFS-PCA**, **PCA**, **MC-PCA**, **SVD** and **KPCA** respectively.

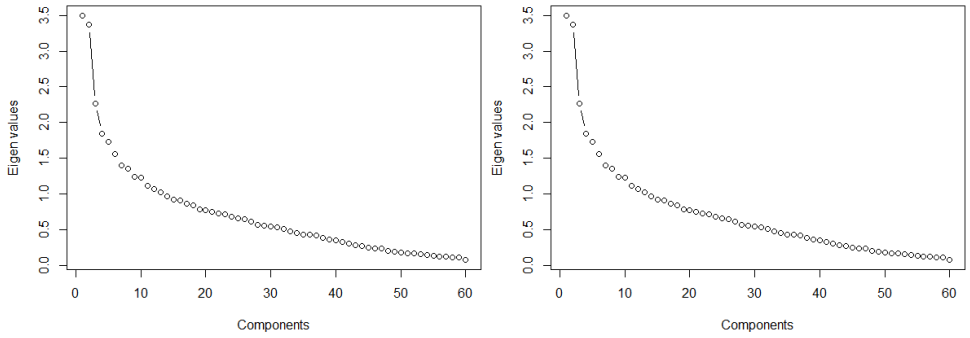
Table 4.5: **PCs** of Sonar datasets.

	PC1	PC2	PC3	PC4	PC5
BCFS-PCA	2.4446911	1.8391889	1.4040616	1.2486493	1.0824996
PCA	3.49398540	3.36724365	2.26494881	1.84594534	1.73327666
MC-PCA	3.36788430	3.26025401	2.40882551	1.74063719	1.50395664
SVD	40.62628292	10.70937595	8.55873747	5.22294450	4.38161550
KPCA	0.072314673	0.049871618	0.021547068	0.016311559	0.013741493

	PC6	PC7	PC8	PC9	PC10
BCFS-PCA	1.0429182	1.0007814	0.9504712	0.9073542	0.8734395
PCA	1.56172889	1.40263862	1.35199094	1.24079676	1.22256345
MC-PCA	1.45839122	1.31614855	1.25237476	1.14865434	1.12353354
SVD	4.01841431	3.95531866	3.08788940	2.84903749	2.63780853
KPCA	0.011535316	0.010854956	0.007850860	0.006113269	0.005425382

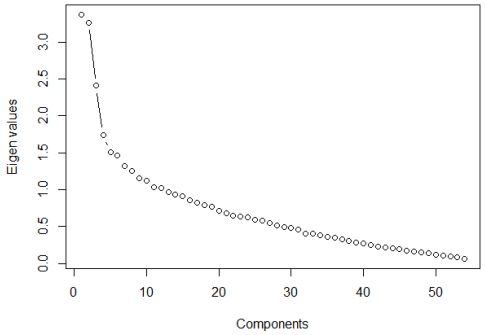
	PC11	PC12	PC13	PC14	PC15
BCFS-PCA	0.8234273	0.8084142	0.7373793	0.7316819	0.6882025
PCA	1.11587102	1.06826681	1.02381040	0.96077600	0.92556913
MC-PCA	1.03029094	1.01924852	0.96477626	0.93419626	0.91377904
SVD	2.44678089	2.27954487	2.07433464	1.86076429	1.78354939
KPCA	0.005085615	0.004663958	0.004291771	0.003362361	0.002964842

	PC16	PC17	...
BCFS-PCA	0.6779643	0.6597789	...
PCA	0.90364549	0.86067703	...
MC-PCA	0.85950587	0.82533601	...
SVD	1.67637642	1.61361118	...
KPCA	0.002652953	0.002523338	...

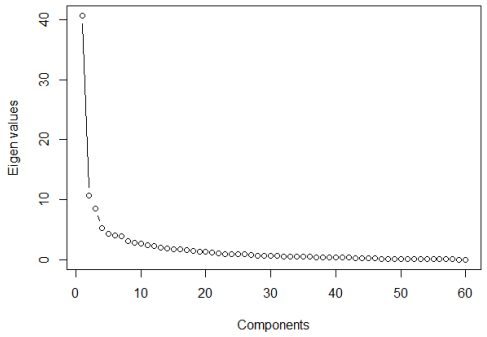


(a) **BCFS-PCA.**

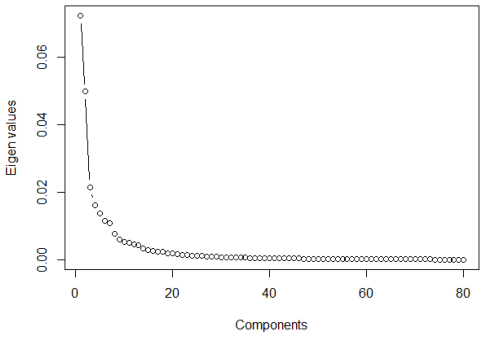
(b) **PCA**.



(c) **MC-PCA**.



(d) **SVD**.



(e) **KPCA**.

Figure 4.6: Scree plot of Sonar datasets.

4.3.2.3 Classification accuracy

In order to compare the five methods, we fit the **PCs** obtained from the Sonar large data to the classification model "Random Forest" **RF**, and we measure the accuracy for each selected **PCs** (the same method and packages we used for decathlon2 dataset are applied in this case). The results are given in Table 4.6 and Figure 4.7.

Table 4.6: **RF** accuracy of Sonar dataset.

No. of PCs	BCFS-PCA	PCA	MC-PCA	SVD	KPCA
1	0.6273	0.5556	0.4340	0.6005	0.6093
2	0.6490	0.6058	0.4904	0.6305	0.6345
3	0.7692	0.7307	0.4952	0.6936	0.6303
4	0.8365	0.7307	0.5527	0.7639	0.7501
5	0.8413	0.8077	0.5048	0.7937	0.8123
6	0.8125	0.7885	0.4615	0.7787	0.7837
7	0.8221	0.8125	0.5000	0.7749	0.8067
8	0.8317	0.7981	0.4856	0.7640	0.8175
9	0.8125	0.7981	0.4760	0.7637	0.8035
10	0.8269	0.8173	0.5144	0.8044	0.8079
11	0.8223	0.8368	0.5590	0.7542	0.8169
12	0.8361	0.8127	0.5234	0.7782	0.8175
13	0.8175	0.8166	0.5088	0.7635	0.8164
14	0.8177	0.8142	0.5483	0.7743	0.8066
15	0.8328	0.8130	0.5563	0.8081	0.8027
16	0.8421	0.8181	0.5531	0.7790	0.8129
17	0.8232	0.8166	0.5160	0.7648	0.8216
18	0.8229	0.8174	0.5879	0.7803	0.8139
19	0.8170	0.8035	0.5726	0.8179	0.8025
20	0.8261	0.8186	0.5667	0.8175	0.8213
...	...	...	...	...	...

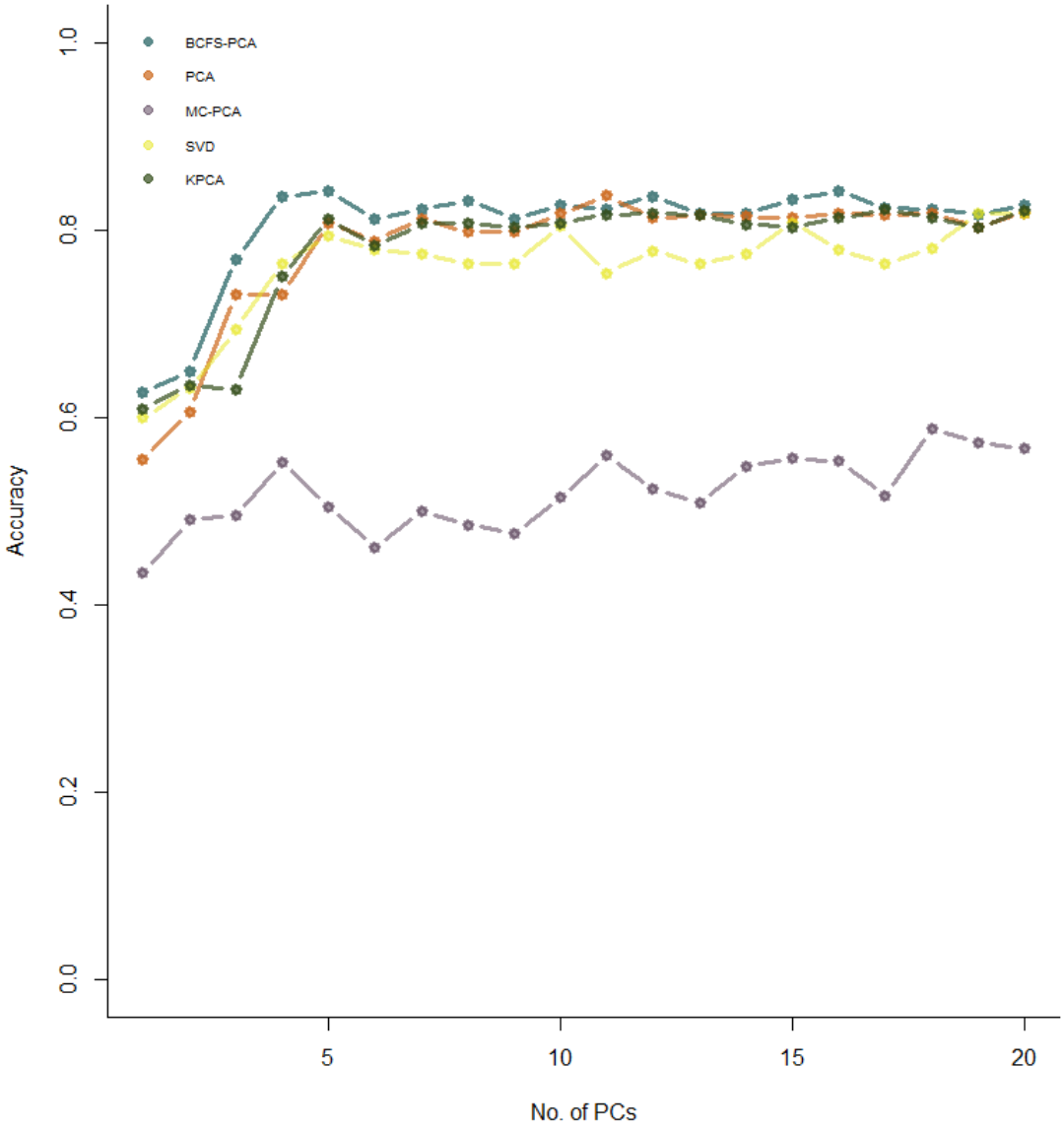
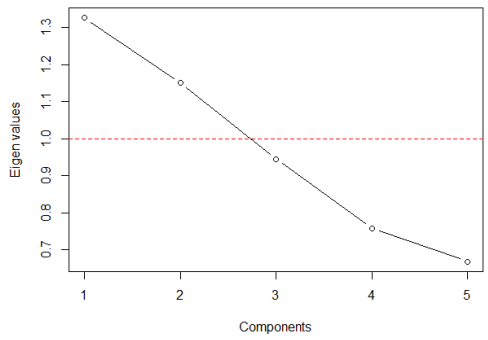


Figure 4.7: Accuracy of Sonar dataset

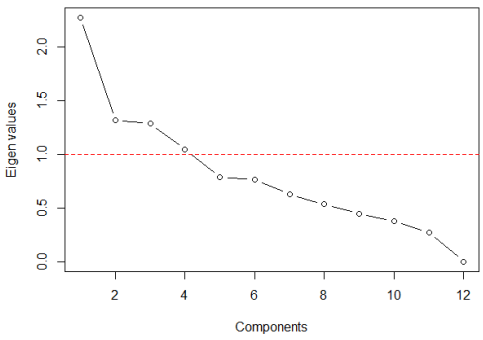
4.3.3 Discussion

1. **Decathlon2 datasets:** After applying the Dimensionality Reduction method **BCFS**, we applied **PCA** on the 5 features selected of the **BCFS** method. The proposed method is compared against **MC-PCA**, and the baseline methods **PCA**, **SVD** and **KPCA**. The acquired **PCs** are given in Table 4.2. Based on the Kaiser rule [91], the eigenvalues show how important each component is and the amount of information it holds. It also states that

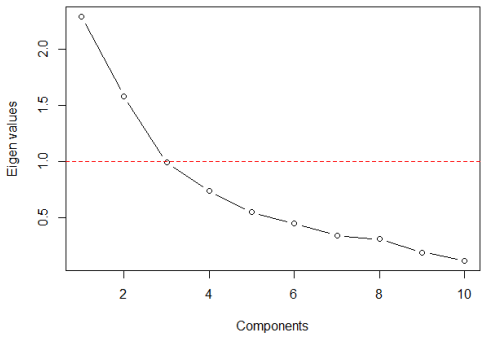
an eigenvalue with a value less than 1 holds no more information than a single variable. Following that statement, only 2 **PCs** are selected in **BCFS-PCA**, 3 **PCs** in **MC-PCA**, and 4 **PCs** in **PCA** method. For **SVD** technique, the truncation value r is obtained using (2.2) explained in chapter 2. The captured **PCs** correspond to the values of eigenvalues larger or equal to the threshold $\delta = 9.1542$ in Table 4.2. In other words, only 5 **PCs** are selected. For **KPCA**, only 3 **PCs** are selected using the elbow rule explained in chapter 2. Figure 4.8 illustrates the truncation, where the selected **PCs** correspond to the eigenvalues above the the truncated red line.



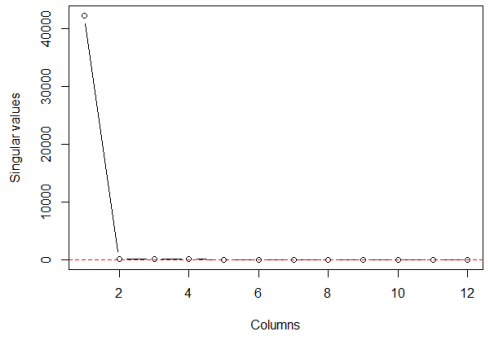
(a) **BCFS-PCA**.



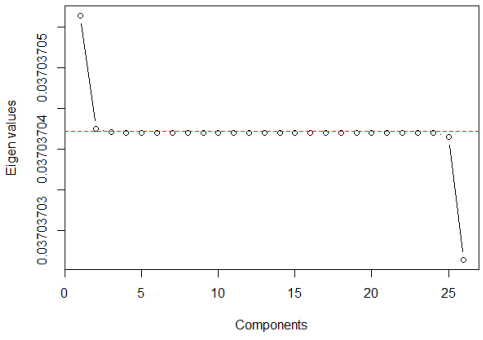
(b) **PCA**.



(c) **MC-PCA**.



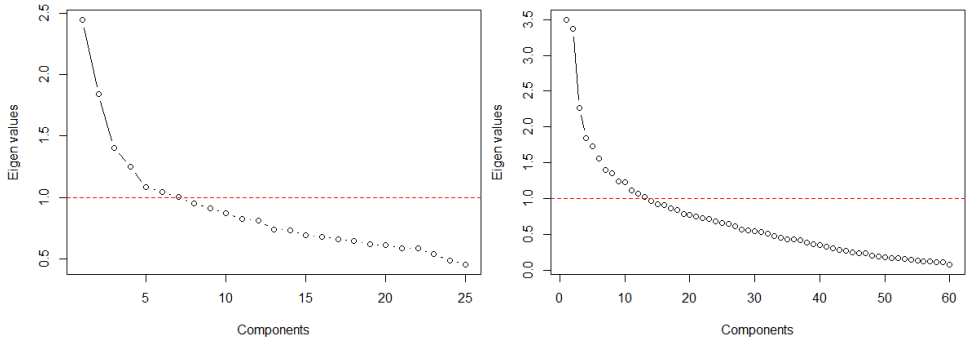
(d) **SVD**, $\delta = 9.1542$.



(e) **KPCA**.

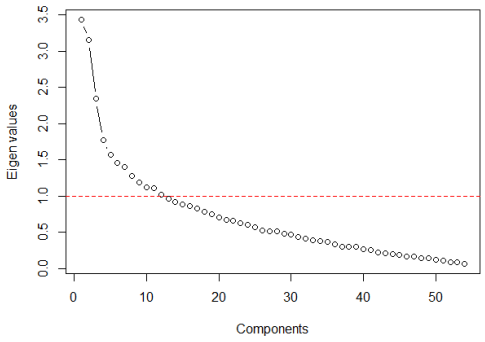
Figure 4.8: Selected **PCs** for decathlon2 datasets.

After extracting the **PCs**, a classification accuracy analysis is performed to compare the different methods, where the results are given in 4.3 and Figure 4.4. We notice that **MC-PCA** gave a low accuracy for several selected **PCs**, and did not increase the performance of **PCA** method. Howerver on the other hand, **BCFS-PCA** showed a higher accuracy compare to the four methods under different number of selected **PCs** for the model **RF**.

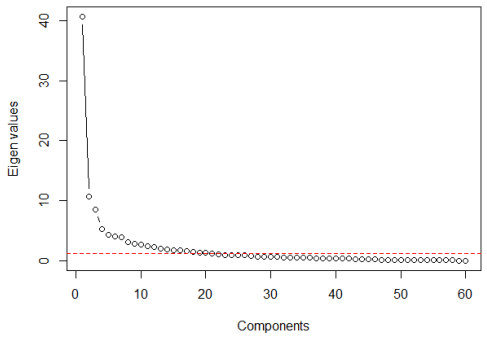


(a) **BCFS-PCA.**

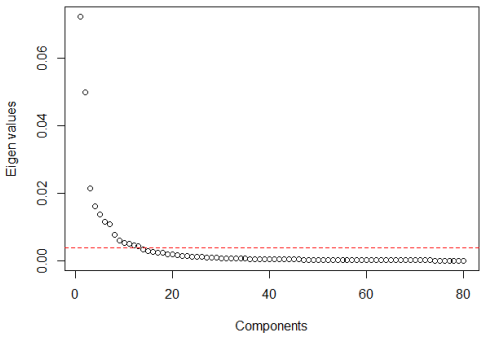
(b) **PCA.**



(c) **MC-PCA**.



(d) **SVD**, $\delta = 1.1893$.



(e) **KPCA**.

Figure 4.9: Selected **PCs** of Sonar datasets.

2. **Sonar datasets:** The first stage of the proposed method **BCFS** selected 25 attributes out of

60, as it eliminated 35 redundant attributes. The **MC** method on the other hand eliminated only 6 attributes and selected 54 variables. In the second stage, and after applying **PCA** on the results acquired from stage one, we get the **PCs** given in Table 4.5. By applying the same truncation methods used for decathlon2 datasets, we conclude that we can select only 7 **PCs** using **BCFS-PCA**, 12 **PCs** using **PCA** and **MC-PCA** methods, 20 **PCs** using **SVD** with a threshold equal to $\delta = 1.1893$, and 13 **PCs** using **KPCA**. The truncation is shown in Figure 4.9

Table 4.6 shows the accuracy values obtained after fitting the **PCs** to **RF**. We can observe that the **BCFS-PCA** outperformed the other four methods for various numbers of selected **PCs**. However, it should be noted that **MC-PCA** indicated low values of accuracy, hence it did not improve the performance of **PCA** at all.

The achieved results show that **BCFS-PCA** not only selects less attributes and eliminates more redundancy, but also enhances the extracted **PCs** in term of classification accuracy and modeling, for both small and large data.

4.4 Conclusion

The main goal of this chapter is to improve the performance of the feature extraction method **PCA**, by reducing as much irrelevant features as possible. The method is an enhancement of the baseline technique **PCA**, that combines it with the Dimensionality Reduction method **BCFS**. It demonstrated better results in clearing redundancy compared to **MC-PCA**, **PCA**, **SVD** and **KPCA**, but also in improving the classification model **RF**. Using real-world datasets as a sample for the experimental results, **BCFS-PCA** selected the minimum number of attributes, and had a higher efficient accuracy on the model **RF** for different amounts of selected **PCs**. For that reason, **BCFS-PCA** is an ideal method to optimize the data pre-processing, as it enhances data analysis by extracting the strictly needed information only. This implies more optimized machine learning algorithms, and better performance of models with a significant computational time.

Code

Simulation of the Gassian bivariate Copula.

The following R code correspond to the simulation of the bivariate Gaussian Copula giving in chapter 1 on page 15 with normal margins.

```
#Set the seed for reproducibility.
>set.seed(180)
 #Choose a value for the correlation coefficient, theta, where -1 <= theta <= 1.
>theta <- 0.5

# Compute the Cholesky decomposition of the correlation matrix Sigma.
 >Sigma <- matrix(c(1, theta, theta, 1), ncol = 2)
>A <- chol(Sigma)

# Generate two independent standard normal random variables Z1 and Z2.
    >Z <- matrix(rnorm(n = 1000*2), nrow = 2)

# Transform the random variables Z1 and Z2 using the Cholesky decomposition matrix A.
> X <- A %*% Z

# Compute the cumulative distribution function (CDF) of the standard normal
    distribution for each transformed variable X and Y.
>U <- pnorm(X[1,])
>V <- pnorm(X[2,])

>Gauss.cop1<- cbind(U,V)

# Print the first five rows of the bivariate Gaussian Copula.
>head(Gauss.cop1)
          U         V
[1,] 0.3745051 0.6559069
```

```
[2,] 0.8727994 0.4487043
[3,] 0.2657744 0.2967683
[4,] 0.5969068 0.2609988
[5,] 0.6884883 0.4078758
[6,] 0.2457123 0.4702010
```

Copula visualization

The plot of the Gaussian Copula with $\theta = 0.5$ presented in Figure 1.1a is generated using the following code:

```
# Plot the Gaussian Copula.
>plot(U, V, xlab = "U", ylab = "V")
```

While the corresponding density Copula presented in Figure 1.1b is obtained using the following code:

```
# Create a Gaussian Copula object with the specified correlation.
>norm.cop <- normalCopula(param = theta, dim = 2)

# Plot the Gaussian Copula density.
>persp(norm.cop,dCopula, xlab ="U", ylab = "V" , zlab="c(U,V)")
```

Using the estimated Kendall's tau.

The relationship between kendall's tau τ and the Copula's parameter θ giving in (1.36), allows us to obtain a more suitable Copula. To do so, we follow this code:

```
# Set the seed for reproducibility.
>set.seed(180)

# Generate a 1000 x 2 matrix of standard normal random variables.
Z <- matrix(rnorm(n = 1000*2), ncol = 2)

# Compute the Kendall's tau correlation coefficient between the two variables.
>tau <- cor(Z, method = "kendall")[1,2]

# Compute the Copula parameter theta from the Kendall's tau correlation coefficient.
>theta <- sin((pi/2) * tau)

# Create a normal Copula object with the computed parameter theta.
>Gauss.cop2 <- normalCopula(dim = 2, param = theta)

# Generate 1000 samples from the normal Copula.
>u <- rCopula(1000, Gauss.cop2)
```

```
# Print the first five rows of the bivariat e Gaussian Copula.
>head(u)
       [,1]       [,2]
[1,] 0.940995038 0.07102679
[2,] 0.513410765 0.58586152
[3,] 0.866764730 0.79797598
[4,] 0.009021473 0.46479192
[5,] 0.394972091 0.60580576
[6,] 0.309862720 0.03904465
```

For Spearman's rho ρ we use the same code, however we change these two lines in the previous code using (1.37):

```
>rho.s <- cor(Z, method = "spearman")[1,2]
>theta <- sin((pi/6) * rho.s)
```

Comparison

Now, let's compare between these two gaussian Copulas and see which one suits the data better.

```
#Calculating Kendall's tau of the 2 normal samples.
>tau<-cor(Z,method="kendall")
> tau
         [,1]         [,2]
[1,]  1.000000000 -0.006094094
[2,] -0.006094094 1.000000000


#Calculating Kendall's tau of the 2 first generated Gaussian Copula "Gauss.cop1"
    samples.
>cor(Gauss.cop1,method = "kendall")
       U         V
U 1.0000000 0.3064104
V 0.3064104 1.0000000


#Calculating Kendall's tau of the 2 second generated Gaussian "Gauss.cop2" Copula
    samples (using Kendall's tau).
>cor(u,method = "kendall")
         U         V
U 1.0000000 0.3170531
V 0.3170531 1.0000000
```

We can see that the estimated value of correlation obtained using Kendall's tau "Gauss.cop2" is closer to the correlation between the normal distributed couples (Z_1, Z_2). This means that Kendall's tau is key find when it comes to estimating the Copula's parameter.

Simulation of the bivariate t-Copula.

The bellow R code correspond to the simulation of the bivariate Student Copula giving in chapter 1 on page 17 with normal margins.

```
#For same sample of Z.
>set.seed(180)
# Set correlation coefficient and degrees of freedom.
>theta<- 0.5
>nu <- 3

# Correlation matrix Sigma.
>Sigma <- matrix(c(1, theta, theta, 1), ncol=2)

# Cholesky decomposition L of correlation matrix Sigma.
>L <- chol(Sigma)

# Simulate independent standard normal random variables.
>Z <- matrix(rnorm(n = 1000*2), ncol = 2)

# Simulate chi-squared random variable.
>S <- rchisq(n=1000, df=nu)

# Compute the correlated standard normal random variables Y1 and Y2.
>Y <- Z %*% t(L)

# Compute the correlated t-distributed random variables X1 and X2.
>X <- sqrt(nu/S) * Y

# Transform X1 and X2 to t-distributed random variables U and V.
>U <- qt(pt(X[,1], df=nu), df=nu)
>V <- qt(pt(X[,2], df=nu), df=nu)

# Combine U and V into a matrix.
>t.cop1 <- cbind(U,V)

# Print the six first values of U and V.
>head(t.cop1)
          U          V
```

```
[1,] -0.3312392 0.08440337
[2,]  5.2889578 5.51481815
[3,]  0.6981046 -0.35067600
[4,] -0.2097863 -0.22959658
[5,] -0.9390116 -1.07764164
[6,] -0.7998438 -0.41368443
```

Copula visualization

The plot of the Student Copula with $\theta = 0.5$ presented in Figure 1.2a is obtained using the same code of Gaussian Copula, i.e:

```
# Plot the sample.
>plot(U, xlab="U", ylab="V")
```

Similarly, the density plot of the Student Copula shown in Figure 1.2b is generated using the same code of plotting the Gaussian Copula. However, the "tCopula" function is used instead of "normalCopula", i.e:

```
# Create a Student Copula object with the specified correlation theta and the degree
    of freedom nu.
>st.cop <- tCopula(param = theta, dim = 2,df=nu)

# Plot the Student Copula density.
>persp(st.cop,dCopula, xlab ="U", ylab = "V" , zlab="c(U,V)")
```

Using the estimated Kendall's tau.

In order to estimate the t-Copula's parameter, and as we did for the Gaussian Copula. We use (1.36). The code goes as follow:

```
# Set the seed for reproducibility.
set.seed(180)

# Generate a 1000 * 2 matrix of standard normal random variables.
>Z <- matrix(rnorm(n = 1000*2), ncol = 2)

# Compute the Kendall's tau correlation coefficient between the two variables.
>tau <- cor(Z, method = "kendall")[1,2]

# Compute the Copula parameter theta from the Kendall's tau correlation coefficient.
>theta <- sin((pi/2) * tau)

# Create a normal Copula object with the computed parameter theta.
>t.cop2 <- tCopula(dim = 2, param = theta)
```

```
# Generate 1000 samples from the normal Copula.
>u <- rCopula(1000, t.cop2)

# Print the first five rows of the bivariate t-Copula.
>head(u)
       [,1]       [,2]
[1,] 0.90917508 0.101969239
[2,] 0.52273411 0.642276741
[3,] 0.91158510 0.856908103
[4,] 0.08188587 0.476154696
[5,] 0.44346909 0.556955628
[6,] 0.08123545 0.001859567
```

Comparison

In order to compare between the two Copulas, and as we did for the Gaussian Copula. We calculate the correlation using kendall's tau of the obtained samples from "t.cop1" and "t.cop2", and see which value is closer to the correlation value between Z_1 and Z_2 "tau".

```
#Calculating Kendall's tau of the 2 normal samples.
tau<-cor(Z,method="kendall")
> tau
         [,1]         [,2]
[1,]  1.000000000 -0.006094094
[2,] -0.006094094 1.000000000


#Calculating Kendall's tau of the 2 first generated Student Copula "t.cop1" samples.
>cor(t.cop1,method = "kendall")
       U         V
U 1.0000000 0.2693213
V 0.2693213 1.0000000


#Calculating Kendall's tau of the 2 second generated Student "t.cop2" Copula samples
    (using Kendall's tau).
>cor(u,method = "kendall")
        [,1]        [,2]
[1,]  1.00000000 -0.02005205
[2,] -0.02005205 1.00000000
```

It is clear that "t.cop2" is more suitable for the random variables Z_1 and Z_2 compared to "t.cop1".

General Conclusion

The most powerful trait of machine learning is that the machine can be built to make very complex decisions just by learning them from real life data, which due to its source, is filled with inconsistency, redundancy, and noise. However, if we seek high-quality decisions, we must provide high-quality data. Luckily, data pre-processing was made to ensure this essential condition, it's an operation that cleans the input data through many steps before we use it for the learning. One of the most crucial steps in pre-processing is the Dimensionality Reduction, which aims to eliminate noise and undesired redundancy by removing irrelevant or redundant information from the data. Therefore, the major goal of this thesis is to deal with this issue and derive new Dimensionality Reduction techniques by eliminating the redundant attributes using sampling methods. The introduced techniques in chapter 3 and 4 are based on the theory of Copula, since this latter provide us with the possibility to detect redundancy and eliminate it using the correlation between the variables without the need to impose constraints to specify the types of marginal distributions

The first proposed technique named Bivariate Copulas based Feature Selection **BCFS** introduced in section 3.2 is a Dimensionality Reduction method that eliminates redundancy by modeling correlation using Copulas. **BCFS** Indicated good results against the unsupervised feature selection technique **LU-C** and the supervised feature selection techniques **LASSO**, **SW** and **LARS**, due to its low time complexity ($Om^2 n \log n$) and being able to capture more redundancy and eliminate it. Also, it improved the models accuracy using several data and outperformed the baseline methods. However, this technique eliminates redundancy randomly as we explained in section 3.2. To deal with that, we optimized the technique by proposing another Dimensionality Reduction technique named Grouped Bivariate Copulas based Feature Selection **GBCFS**. The technique is introduced in details in section 3.3. **GBCFS** showed better results compared to **BCFS** and the other techniques in term of reduction and elimination of noise and redundancy. As a result, it improved the models accuracy for almost all the datasets.

Another Dimensionality Reduction technique was proposed in chapter 4 under the name **BCFS-PCA**. It is a feature extraction technique that combines the **BCFS** method proposed in chapter 3 and the feature extraction method **PCA** introduced in chapter 2. The established approach is an improvement of **PCA**. The achieved results indicate that not only **BCFS** improved the reduction of **PCA**, but, **BCFS-PCA** outperformed the linear feature extraction techniques **MC-PCA**, **SVD** and the non-linear feature extraction technique **KPCA** for small and huge data. Also, in term of

accuracy using Random Forest **RF** model.
The obtained results in chapter 3 and 4 imply that copula is a powerful tool when it comes to data pre-processing and machine learning, due to its capacity of capturing non-linear dependence easily, without using marginal distribution. However, handling all attributes at once can lead us to better results using multivariate analysis instead of bivariate analysis, therefore, future work will focus on developing a new unsupervised filtering technique using multivatiate Copulas instead, it will allow as to treat all the attributes at once using the multivariate Copulas correlation matrix. We will also focus on using the other type of Copulas such as Student Copula and Archimedean copulas. One can also use the theory of Copula to improve the other feature extraction techniques as we did for **PCA**.

BIBLIOGRAPHY

[1] Abd-Alsabour, N. (2018).
On the role of dimensionality reduction.
J. Comput., 13(5):571–579.

[2] Abid, A., Zhang, M. J., Bagaria, V. K., and Zou, J. (2018).
Exploring patterns enriched in a dataset with contrastive principal component analysis.
Nature communications, 9(1):1–7.

[3] Abrevaya, J. (1999).
Computation of the maximum rank correlation estimator.
Economics letters, 62(3):279–285.

[4] Agatonovic-Kustrin, S. and Beresford, R. (2000).
Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research.
Journal of pharmaceutical and biomedical analysis, 22(5):717–727.

[5] Ahmadkhani, S. and Adibi, P. (2016).
Face recognition using supervised probabilistic principal component analysis mixture model in dimensionality reduction without loss framework.
IET Computer Vision, 10(3):193–201.

[6] Ayesha, S., Hanif, M. K., and Talib, R. (2020).
Overview and comparative study of dimensionality reduction techniques for high dimensional data.
Information Fusion, 59:44–58.

[7] Aziz, R., Verma, C., and Srivastava, N. (2017).
Dimension reduction methods for microarray data: a review.
AIMS Bioengineering, 4(2):179–197.

[8] Badakhshan Farahabadi, F., Fathi Vajargah, K., and Farnoosh, R. (2021).
Dimension Reduction Big Data Using Recognition of Data Features Based on Copula Function and Principal Component Analysis.

Advances in Mathematical Physics, 2021.

[9] Cattell, R. B. (1966).
The scree test for the number of factors.
Multivariate behavioral research, 1(2):245–276.

[10] Cattell, R. B. and Jaspers, J. (1967).
A general plasmode (no. 30-10-5-2) for factor analytic exercises and research.
Multivariate Behavioral Research Monographs.

[11] Chang, C.-D., Wang, C.-C., and Jiang, B. C. (2012).
Singular value decomposition based feature extraction technique for physiological signal analysis.
Journal of medical systems, 36(3):1769–1777.

[12] Chatterjee, S. (2016).
fastAdaboost: A fast implementation of adaboost.
R package version, 1(0).

[13] Cherubini, U., Luciano, E., and Vecchiato, W. (2004).
Copula methods in finance.
John Wiley & Sons.

[14] Christensen, D. (2005).
Fast algorithms for the calculation of kendall's τ.
Computational Statistics, 20(1):51–62.

[15] Comon, P. (1994).
Independent component analysis, a new concept?
Signal processing, 36(3):287–314.

[16] Diamantaras, K. I. and Kung, S. Y. (1996).
Principal component neural networks: theory and applications.
John Wiley & Sons, Inc.

[17] Dua, D. and Graff, C. (2017).
UCI machine learning repository.

[18] Eesa, A. S., Abdulazeez, A. M., and Orman, Z. (2017).
A dids based on the combination of cuttlefish algorithm and decision tree.
Science Journal of University of Zakho, 5(4):313–318.

[19] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004).
Least angle regression.

The Annals of statistics, 32(2):407–499.

[20] Elhadad, M. K., Badran, K. M., and Salama, G. I. (2017).
A novel approach for ontology-based dimensionality reduction for web text document classification.
International Journal of Software Innovation (IJSI), 5(4):44–58.

[21] Embrechts, P., Lindskog, F., and McNeil, A. (2001).
Modelling dependence with copulas.
Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich, 14:1–50.

[22] Embrechts, P., Mikosch, T., and Klüppelberg, C. (1997).
Modelling Extremal Events for Insurance and Finance.
Springer, Berlin.

[23] Erichson, N. B., Zheng, P., Manohar, K., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2020).
Sparse principal component analysis via variable projection.
SIAM Journal on Applied Mathematics, 80(2):977–1002.

[24] Femmam, K., Brahimi, B., and Femmam, S. (2023).
An optimized feature selection technique based on bivariate copulas “gbcfs”.
Journal of Combinatorial Optimization, 45(2):74.

[25] Femmam, K. and Femmam, S. (2023).
Improving the dimensionality reduction of pca using bivariate copulas.
Advances and Applications in Statistics, 86(1):47–64.

[26] Filzmoser, P., Fritz, H., and Kalcher, K. (2021).
pcaPP: Robust PCA by Projection Pursuit.
R package version 1.9-74.

[27] Fisher, R. A. (1936).
The use of multiple measurements in taxonomic problems.
Annals of eugenics, 7(2):179–188.

[28] Fonti, V. and Belitser, E. (2017).
Feature selection using lasso.
VU Amsterdam Research Paper in Business Analytics, 30:1–25.

[29] Fritsch, S., Guenther, F., and Guenther, M. F. (2019).
Package ‘neuralnet’.
Training of Neural Networks.

[30] Fukunaga, K. (1990).
Introduction to statistical pattern recognition.
Computer science and scientific computing. Academic Press, Boston, 2nd ed edition.

[31] Gao, L., Song, J., Liu, X., Shao, J., Liu, J., and Shao, J. (2017).
Learning in high-dimensional multimedia data: the state of the art.
Multimedia Systems, 23(3):303–313.

[32] Gavish, M. and Donoho, D. L. (2014).
The optimal hard threshold for singular values is $4/\sqrt{3}$.
IEEE Transactions on Information Theory, 60(8):5040–5053.

[33] Genest, C. and Favre, A.-C. (2007).
Everything you always wanted to know about copula modeling but were afraid to ask.
Journal of hydrologic engineering, 12(4):347–368.

[34] Genest, C. and MacKay, J. (1986).
The joy of copulas: Bivariate distributions with uniform marginals.
The American Statistician, 40(4):280–283.

[35] Genest, C., Rémillard, B., and Beaudoin, D. (2009).
Goodness-of-fit tests for copulas: A review and a power study.
Insurance: Mathematics and economics, 44(2):199–213.

[36] Guillemot, V., Beaton, D., Gloaguen, A., Löfstedt, T., Levine, B., Raymond, N., Tenenhaus, A., and Abdi, H. (2019).
A constrained singular value decomposition method that integrates sparsity and orthogonality.
PloS one, 14(3):e0211463.

[37] Harrell Jr, F. E. (2015).
Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis.
Springer.

[38] Hastie, T. and Efron, B. (2022).
lars: Least Angle Regression, Lasso and Forward Stagewise.
R package version 1.3.

[39] He, X., Yan, S., Hu, Y., and Zhang, H.-J. (2003).
Learning a locality preserving subspace for visual recognition.
In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 385–392. IEEE.

[40] Hoeffding, W. (1941).
Masstabinvariante korrelationsmasse für diskontinuierliche verteilungen.
Archiv für mathematische Wirtschafts-und Sozialforschung, 7:49–70.

[41] Hofert, M., Kojadinovic, I., Maechler, M., and Yan, J. (2020).
copula: Multivariate Dependence with Copulas.
R package version 1.0-1.

[42] Hotelling, H. (1933).
Analysis of a complex of statistical variables into principal components.
Journal of educational psychology, 24(6):417.

[43] Houari, R., Bounceur, A., Kechadi, M.-T., Tari, A.-K., and Euler, R. (2016).
Dimensionality reduction in data mining: A copula approach.
Expert Systems with Applications, 64:247–260.

[44] Huang, P., Li, T., Shu, Z., Gao, G., Yang, G., and Qian, C. (2018).
Locality-regularized linear regression discriminant analysis for feature extraction.
Information sciences, 429:164–176.

[45] Husson, F., Josse, J., Narasimhan, B., and Robin, G. (2019).
Imputation of mixed data with multilevel singular value decomposition.
Journal of Computational and Graphical Statistics, 28(3):552–566.

[46] Joe, H. (1997).
Multivariate models and multivariate dependence concepts.
CRC press.

[47] Jolliffe, I. T. (1995).
Rotation of principal components: choice of normalization constraints.
Journal of Applied Statistics, 22(1):29–35.

[48] Kambhatla, N. and Leen, T. K. (1997).
Dimension reduction by local principal component analysis.
Neural computation, 9(7):1493–1516.

[49] Kassambara, A. and Mundt, F. (2020).
factoextra: Extract and Visualize the Results of Multivariate Data Analyses.
R package version 1.0.7.

[50] Knight, W. R. (1966).
A computer method for calculating kendall's tau with ungrouped data.
Journal of the American Statistical Association, 61(314):436–439.

[51] Kotz, S., Balakrishnan, N., and Johnson, N. L. (2004).
Continuous multivariate distributions, Volume 1: Models and applications, volume 1.
John Wiley & Sons.

[52] Kuhn, M. (2022).
caret: Classification and Regression Training.
R package version 6.0-92.

[53] Kuhn, M. and Johnson, K. (2020).
Feature engineering and selection: a practical approach for predictive models.
CRC Press.

[54] Lall, S., Sinha, D., Ghosh, A., Sengupta, D., and Bandyopadhyay, S. (2021).
Stable feature selection using copula based mutual information.
Pattern Recognition, 112:107697.

[55] Lee, J. A. and Verleysen, M. (2007).
Nonlinear dimensionality reduction, volume 1.
Springer.

[56] Liaw, A., Wiener, M., et al. (2002).
Classification and regression by randomForest.
R news, 2(3):18–22.

[57] Lin, P., Zhang, J., and An, R. (2014).
Data dimensionality reduction approach to improve feature selection performance using sparsified svd.
In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 1393–1400. IEEE.

[58] Ling, C.-H. (1965).
Representation of associative functions.
Publ. Math. Debrecen, 12:189–212.

[59] Liu, C., Yang, S. X., and Deng, L. (2015a).
A comparative study for least angle regression on NIR spectra analysis to determine internal qualities of navel oranges.
Expert Systems with Applications, 42(22):8497–8503.

[60] Liu, S., Feng, L., and Qiao, H. (2015b).
Scatter balance: An angle-based supervised dimensionality reduction.
IEEE transactions on neural networks and learning systems, 26:277–89.

[61] Marill, T. and Green, D. (1963).

On the effectiveness of receptors in recognition systems.
IEEE transactions on Information Theory, 9(1):11–17.

[62] Meng, C., Zeleznik, O. A., Thallinger, G. G., Kuster, B., Gholami, A. M., and Culhane, A. C. (2016).
Dimension reduction techniques for the integrative analysis of multi-omics data.
Briefings in Bioinformatics, 17(4):628–641.

[63] Mesiar, R. and Sheikhi, A. (2021).
Nonlinear random forest classification, a copula-based approach.
Applied Sciences, 11(15):7140.

[64] Mills, P. (2017).
Singular value decomposition (svd) tutorial: Applications, examples, exercises.

[65] Nelsen, R. B. (2007).
An introduction to copulas.
Springer Science & Business Media.

[66] Nelsen, R. B. and Nelsen, R. B. (1999).
Archimedean copulas.
An Introduction to Copulas, pages 89–124.

[67] Ng, S. (2017).
Principal component analysis to reduce dimension on digital image.
Procedia computer science, 111:113–119.

[68] Nie, F., Xiang, S., and Zhang, C. (2007).
Neighborhood minmax projections.
In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 993–998.

[69] Partridge, M. and Calvo, R. A. (1998).
Fast dimensionality reduction and simple pca.
Intelligent data analysis, 2(3):203–214.

[70] Pearson, K. (1901).
Principal components analysis.
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 6(2):559.

[71] Pochet, N., De Smet, F., Suykens, J. A., and De Moor, B. L. (2004).
Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction.
Bioinformatics, 20(17):3185–3195.

[72] Poole, D. (2014).
Linear algebra: A modern introduction.
Nelson Education.

[73] Postma, E., van den Herik, H., and van der Maaten, L. (2009).
Dimensionality reduction: a comparative review.
Journal of Machine Learning Research, 10(1–41):66–71.

[74] Rahmanishamsi, J., Dolati, A., and Aghabozorgi, M. R. (2018).
A copula based ica algorithm and its application to time series clustering.
Journal of Classification, 35(2):230–249.

[75] Rao, C. R. (1948).
The utilization of multiple measurements in problems of biological classification.
Journal of the Royal Statistical Society: Series B (Methodological), 10(2):159–193.

[76] Scarsini, M. (1984).
On measures of concordance.
Stochastica, 8(3):201–218.

[77] Schölkopf, B., Smola, A., and Müller, K.-R. (1997).
Kernel principal component analysis.
In *International conference on artificial neural networks*, pages 583–588. Springer.

[78] Schölkopf, B., Smola, A., and Müller, K.-R. (1998).
Nonlinear component analysis as a kernel eigenvalue problem.
Neural computation, 10(5):1299–1319.

[79] Schweiser, B. (1961).
Associative functions and statistical triangle inequalities.
Publicationes Mathematicae, Debrecen, 8:169–186.

[80] Schweizer, B. and Wolff, E. F. (1981).
On nonparametric measures of dependence for random variables.
The annals of statistics, 9(4):879–885.

[81] Singh, D. A. A. G., Leavline, E. J., Priyanka, R., and Priya, P. P. (2016).
Dimensionality reduction using genetic algorithm for improving accuracy in medical diagnosis.
International Journal of Intelligent Systems and Applications, 8(1):67.

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

[83] Tang, B., Shepherd, M., Milios, E., and Heywood, M. I. (2005).
Comparing and combining dimension reduction techniques for efficient text clustering.
In *Proceeding of SIAM international workshop on feature selection for data mining*, pages 17–26.

[84] Tang, Y. and Rose, R. (2008).
A study of using locality preserving projections for feature extraction in speech recognition.
In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1569–1572.

[85] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000).
A global geometric framework for nonlinear dimensionality reduction.
science, 290(5500):2319–2323.

[86] Tiwari, R. and Singh, M. P. (2010).
Correlation-based attribute selection using genetic algorithm.
International Journal of Computer Applications, 4(8):28–34.

[87] Vidal, R., Ma, Y., and Sastry, S. (2005).
Generalized principal component analysis (gpca).
IEEE transactions on pattern analysis and machine intelligence, 27(12):1945–1959.

[88] Washizawa, Y. (2009).
Subset kernel principal component analysis.
In *2009 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE.

[89] Wickham, H., François, R., Henry, L., and Müller, K. (2022).
dplyr: A Grammar of Data Manipulation.
R package version 1.0.9.

[90] Xie, L., Li, Z., Zeng, J., and Kruger, U. (2016).
Block adaptive kernel principal component analysis for nonlinear process monitoring.
AIChE Journal, 62(12):4334–4345.

[91] Yeomans, K. A. and Golder, P. A. (1982).
The guttman-kaiser criterion as a predictor of the number of common factors.
Journal of the Royal Statistical Society. Series D (The Statistician), 31(3):221–229.

[92] YongchangWang and Zhu, L. (2017).
Research and implementation of svd in machine learning.
In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 471–475.

[93] Zebari, D. A., Haron, H., Zeebaree, S. R., and Zeebaree, D. Q. (2019).
Enhance the mammogram images for both segmentation and feature extraction using wavelet transform.
In *2019 International Conference on Advanced Science and Engineering (ICOASE)*, pages 100–105. IEEE.

[94] Zhang, X.-L. (2015).
Nonlinear dimensionality reduction of data by deep distributed random samplings.
In *Asian Conference on Machine Learning*, pages 221–233. PMLR.

[95] Zhao, H., Wang, Z., and Nie, F. (2019).
A new formulation of linear discriminant analysis for robust dimensionality reduction.
IEEE Transactions on Knowledge and Data Engineering, 31(4):629–640.

[96] Zou, H., Hastie, T., and Tibshirani, R. (2006).
Sparse principal component analysis.
Journal of Computational and Graphical Statistics, 15(2):265–286.

Abstract

The task of modeling high dimensional datasets has become increasingly difficult and challenging due to the large amount of redundancy present in the data. This redundancy often leads to the presence of noise and inaccurate data modeling and analysis results. While numerous statistical methods have been proposed to address this problem, many of them involve multiple operations and have high time complexity, often resulting in poor classification performance. To deal with that, in this thesis, three Dimensionality Reduction based on the inter-correlation between the huge data attributes are proposed, where this correlation is modeled using the theory of Copulas. The first two Dimensionality Reduction techniques aim to reduce redundancy by selecting only relevant attributes. While the third proposed technique is a feature extraction process that combines Principal Component Analysis **PCA** and the bivariate Copulas. All these techniques are performed using real-world datasets and compared against powerful Dimensionality Reduction methods in term of reduction, information capturing and models accuracy of the obtained reduced data to evaluate the effectiveness of each technique.

Keywords

Copulas, Feature Selection, Feature Extraction, Dimensionality Reduction, Inter-correlation, **PCA**.

RÉSUMÉ

La tâche de modélisation des données de haute dimension est devenue de plus en plus difficile en raison de la grande quantité de redondance présente dans les données. Cette redondance conduit souvent à la présence de bruit et à des résultats de modélisation et d'analyse de données inexactes. Bien que de nombreuses méthodes statistiques aient été proposées pour résoudre ce problème, beaucoup d'entre elles impliquent plusieurs opérations et ont une complexité temporelle élevée, ce qui se traduit souvent par de mauvaises performances de classification. Pour résoudre ce problème, dans cette thèse, trois techniques de réduction des dimensions basées sur l'intercorrélation entre les attributs des grosses données sont proposées, où cette corrélation est modélisée à l'aide de la théorie des copules. Les deux premières techniques de réduction des dimensions visent à réduire la redondance en ne sélectionnant que les attributs pertinents. Tandis que la troisième technique proposée est un processus d'extraction des caractéristiques qui combine l'analyse en composantes principales (ACP) et les copules bivariées. Toutes ces techniques sont réalisées à l'aide données réels et comparées à des méthodes puissantes de réduction des dimensions en termes de réduction, de capture d'information et d'exactitude des modèles des données réduites obtenues pour évaluer l'efficacité de chaque technique.

Keywords

Copules, Selection des Caractéritiques, Extraction des Caractéristiques, Réduction des Dimensions, Intercorrélation, ACP.

الملخص:

مهمة تصميم نماذج لمجموعات بيانات عالية الأبعاد أصبحت مهمة صعبة وتحدي متزايد بسبب الكم الهائل من التكرار الموجود في البيانات، الذي يؤدي في كثير من الأحيان إلى وجود ضوضاء ونتائج تحليلية غير دقيقة للبيانات. وعلى الرغم من اقتراح العديد من الطرق الإحصائية لحل هذه المشكلة، فإن العديد منها يتطلب عمليات متعددة ولها تعقيد عالي، مما يؤدي في كثير من الأحيان إلى أداء ضعيف للتصنيف. وللتعامل مع هذه المشكلة، يتم في هذه الرسالة اقتراح ثلاث تقنيات لتخفيض الأبعاد بناءً على الترابط بين سمات البيانات الكبيرة، حيث يتم تمثيل هذا الترابط باستخدام نظرية الكوبولاس. تهدف التقنيات الأولى والثانية إلى تقليل التكرار عن طريق اختيار السمات ذات الصلة فقط. فيما يتعلق بالتقنية الثالثة المقترحة، فإنها عملية استخراج للميزات تجمع بين تحليل العناصر الرئيسية والكوبيولا ثنائية المتغيرات. يتم تنفيذ كل هذه التقنيات باستخدام مجموعات بيانات من العالم الحقيقي ومقارنتها مع طرق قوية لتقليل الأبعاد من حيث الحد من الأبعاد والتقاط المعلومات ودقة النماذج التي تم الحصول عليها من البيانات المخفضة لتقييم فعالية كل تقنية.

الكلمات المفتاحية

الكوبولاس، استخراج الميزات، تقليل الأبعاد، ترابط،تحليل العناصر الرئيسية