SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Faculty of Civil Engineering

Reg. No.: SvF-5343-75468

Graphical probability models and machine learning

Master thesis

Study programme: Mathematical and Computational Modeling
Study field: 9.1.9. Applied Mathematics
Training workplace: Department of Mathematics and Constructive Geometry
Thesis supervisor: doc. Ing. Tomáš Bacigál, PhD.

Bratislava 2019 **Bc. Peter Fratrič**

Abstrakt

Účelom pravdepodobnostných grafických modelov je popísať štruktúru závislosti medzi náhodnými premennými. V posledných rokoch tzv. vine kopula model získal pozornosť aj pre viacrozmernú analýzu. V tejto práci používame vine kopula model na predpoveď odchýlky pozície atómu v proteíne, kde závislosť na fyzikálnych vlastnostiach proteínu je nelineárnou funkciou a teda použitie kopúl sú prirodzenou voľbou. Taktiež zavedieme pravdepodobnostný klasifikátor založený na vine kopule a použijeme ho na vysporiadanie sa s bimodálnym rozdelením výstupových dát. Tento prístup kombinuje techniky, ktoré sú nové a neboli ešte použité.

Abstract

Probabilistic graphical models are developed to describe dependence structure among random variables. In recent years vine copulas gained the attention for even high dimensional analysis. We use vine copula model to predict deviation of atomic positions in protein, where the dependence on physical properties of protein is non-linear and so the use of copulas is natural choice. We also develop probabilistic classifier based on vine copula and use it to tackle bimodal distribution of output data. This approach combines techniques that are rather novel or were not tested before.

Resumé

V tejto práci sa venujeme pravdepodobnostným grafickým modelom. V úvode približujeme pravdepodobnostné grafické modely všeobecne, uvádzame Bayesovu vetu ako najpodstatnejšiu časť teórie na ktorej budujeme náš matematický model a takisto si osvojujeme Bayesovský pohľad na strojové učenie. V úvode ďalej definujeme kopule ako matematický konštrukt, ktorý dáva spája marginálne distribučné funkcie, pristupujeme k rigoróznej definícií a uvádzame podstatné teoretické poznatky, ktoré sú neskôr použité na výstavbu modelu. Taktiež uvádzame niekoľko triviálnych aj netriviálnych príkladov kopúl s dôrazom na tie triedy, ktoré sa ešte neskôr vyskytnú pri praktickej aplikácií.

Dáta na ktorých prezentujeme naše teoretické poznatky tvoria hodnoty odmocniny stredných kvadratických odchýlook pozície v proteíne ako závislá premenná od fyzikálnych vlastností proteínu. Vzhľadom na nelineárny charakter závislostí medzi jednotlivými vlastnosťami proteínu sme sa jednoznačne rozhodli pre použitie kopúl. Keďže sa jedná o viacrozmerné dáta, rozhodli sme sa pre vine kopula grafický model, ktoré sa v poslednej dobe ukazujú ako dostatočne popisne flexibilný a zároveň bežným osobným počítačom odhadnuteľný. Pre porovnanie sme použili aj lineárnu regresiu a presnosť predpovede na testovacej vzorke sme porovnávali ako s lineárnou regresiou, tak s výsledkami publikácií, ktoré sa zaoberali týmito dátami s použitím iných modelov.

Naše dáta mali jednu špecifickú vlastnosť a to, že podmienené hustoty pravdepodobnosti založené na vine kopula modeli sme dostávali ako bimodálne rozdelenie. Tu nastal problém, ktoré lokálne maximum hustoty zvoliť za predpovedanú hodnotu. Podmienenú strednú hodnotu sme vylúčili, pretože dávala veľmi nepresné predpovede a ani z matematického hľadiska nedávala veľmi zmysel. Druhá možnosť bola vybrať ten podmienený modus, ktorý nadobúdal v danom bode vyššie lokálne maximum. Tento postup avšak nebol úplne spoľahlivý, pretože v niektorých prípadoch bolo vyššie lokálne maximum príliš špicaté a nižšie lokálne maximum výrazne menej špicaté. Teda, ak by sme na vopred zvolenom intervale zintegrovali podmienenú hustotu v okolí lokálneho maxima, mohlo by sa stať, že nižší lokálny extrém by mal vyššiu pravdepodobnosť na danom intervale. Pre vhodnú voľbu intervalu by táto možnosť mohla dať uspokojujúce výsledky, avšak pre vyššiu výpočtovú zložitosť rátať integrály pre každú hodnotu v testovacej vzorke sme od tejto

možnosti upustili. Namiesto toho sme odvodili rovnice pre pravdepodobnostnú klasifikáciu pre kopula model, ktorá mala rozhodnúť, ktorý modus je vhodnejší a pozorovali sme, že v rovniciach explicitne vystupujú marginálne hustoty na rozdiel od regresnej úlohy a teda sme usúdili, že pravdepodobnostná klasifikácia bude vhodná. Presnosť pravdepodobnostnej klasifikácie sme porovnali s klasickým naivným bayesovským klasifikátorom.

Zostavili sme algoritmus pozostávajúci z kombinácie regresnej úlohy odhadnutia rozdelenia pravdepodobnosti a klasifikačnej úlohy založenej na vine kopula grafickom modeli. Na záver práce sme porovnali výsledky.

Contents

1 Introduction

In this section we will introduce and explain motivation behind the study of probabilistic graphical models and their connection to machine learning. We will build on basic principles on which stands the probabilistic viewpoint of machine learning and graphical models called Bayesian inference. We will also make short introduction to copula theory and later we will use this theoretical knowledge in practice.

1.1 Motivation

Throughout recent years a massive boom of machine learning algorithms has occurred providing very good results for some problems that were too hard or too complex to be tackled directly. Machine learning is present many years but lack of large datasets and computational power did not let it grow to its full potential until a decade or two ago. If we restrict our attention only at supervised learning, that is having the input variables as well as output variables, then machine learning is in fact just fitting our model to data. The variety of models is rich. One can consider large models with many parameters like neural networks or models with smaller number of parameters but less general. Obviously the more general model the bigger is the need for large dataset although even smaller datasets can give good results if we know in advance which features are most dominant.

The class of models we will encounter in our research are probabilistic graphical models. These models are mostly used for estimating distribution function of random vector $\mathbf{X}$. Determining the dependence among random variables of random vector is the crucial task. We can consider only linear correlation but then we risk not describing the dependence structure precisely enough. For example if we consider random vector (X, Y) with dependence $Y = X^2$, where X is uniformly distributed on interval $[-1, 1]$ and so expected values equal to zero. We can calculate correlation coefficient $\rho_{X,Y}$.

$$\rho_{X,Y} = corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = cov(X, X^2) = E[X^3] - E[X]E[X^2] = 0$$

Hence two random variables are uncorrelated but not independent. However, it is not hard to show that if random variables are jointly normally distributed then zero correlation implies independence. Or more precisely stated their dependence structure is entirely described by correlation matrix, but this is

usually not the case in practical applications. In real world we can assume independence or just linear dependence. If we want to capture non-linear dependence the best option is using copulas. Copulas can be viewed as probabilistic graphical model - in particular vine copulas, but their mathematical definition is more profound and will be stated in next chapters. The aim of our research is to inspect predicting capabilities of copulas both in regression and classification problems for which we will use mathematical apparatus of probability theory and Bayesian inference. We will be mostly interested in vine copulas since there are only few papers making use of this copula class for prediction in supervised regression and none in classification problems. Our goal is not to be more precise in predicting than large general models but we hope to make good use of model that can be easy to interpret, describe dependence structure and also make good predictions.

1.2 Probabilistic graphical models

Consider a macroscopic physical system consisting of several variables. Interaction of these variables is specified by physical laws to certain degree of precision. If we describe this system by differential equations we may, or may not be able to solve these equations once we specify initial and boundary conditions. If the system we aim to make predictions about becomes too complex it is usually impossible to solve or even specify the mathematical model for the system. One approach to deal with such complexity is probabilistic graphical models.

We can now assign to each state of the system a random variable X. Formally speaking random variable is just a function $X : \Omega \to E$ that assigns a value from E to given state from set of states Ω. In probability theory we call this state an event and to be precise the assigned value need not to be a number, but an element of measurable space and the function should be itself measurable. We will not deal with such formalities and assume that all our random variables are real-valued measurable or discrete and measurable. Every measurement of the system refers to specific realisation of random variables in the system. Once we have our random variables specified forming a d-dimensional random vector $(X_1, ..., X_i, ..., X_d)$ and measurements recorded in dataset, we can define probability distribution function of random vector and (if exists) probability density function of random vector. We call these functions marginal distributions and marginal densities when we refer to a distribution or density of single variable. Note that single variable can

be affected by some other random variables. The aim of graphical models is determining joint probability function that would essentially describe the system.

We can visualize the system as a graph, where each node represents random variable and each edge between two nodes represents dependence between two random variables. Clearly for d random variables there can be $\frac{d(d-1)}{2}$ edges and thus in simple cases $\frac{d(d-1)}{2}$ parameters, but note that a dependence between two random variables can be specified by multiple parameters, so the number of parameters might grow very fast. As one can see the really hard part of this approach is determining the dependence structure from our dataset. Another issue is that the dependence in general does not need to be commutative meaning that if random variable X affects Y then Y might not affect X. From mathematical point of view this phenomenon is obvious since if the dependence is specified as $Y = f(X)$ then the function f might not be bijective. If the dependence does not commutes then we draw between two nodes directed edge, or possibly two directed edges of opposite direction if dependence from X to Y is different than from Y to X, however such complicated models are usually not considered. If the dependence commutes we naturally draw just one undirected as we can see on Figure 1 edge between two nodes, which is the same as directed edge both ways with the same weights.

Figure 1: Example of directed and undirected edge

Another interesting phenomenon we can encounter is conditional independence. This phenomenon occurs when we have atleast three random

variables (X, Y, Z) and mathematically can be expressed as $X = f(Y)$ and $Z = g(Y)$, where f, g are some functions and for simplicity assume that they are only univariate, but in real world applications they are typically dependent on more than just one random variable. In this case we can see that even though Y affects both X and Z, these two variables are essentially independent if we fix Y to have some specific value (Figure 2). We can write this in terms of probability as $P(X, Z|Y) = P(X|Y)P(Z|Y)$.

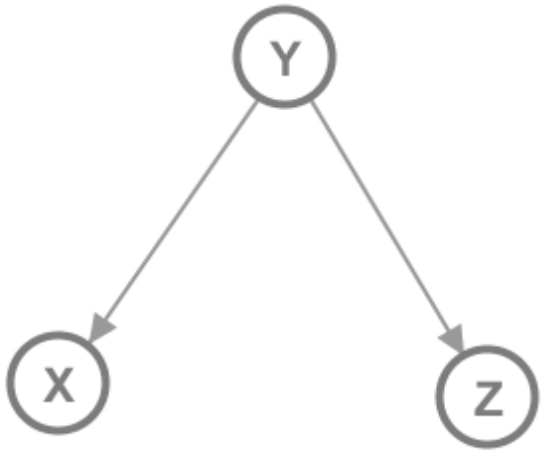

Figure 2: Conditional dependence

Most graphical models make several simplifying assumptions about the graph. For example if when we deal with Gaussian graphical model we assume that all variables have normal distribution and so all we need is correlation matrix of the random vector defining an undirected complete graph with weights on edges equal to correlation coefficient. Another well known model is Bayesian network, where we assume directed acyclic tree graph and probability distribution function is decomposed according to hierarchy of conditional dependencies defined by the graph. In this case edges have no weights and their primary purpose is specifying conditional dependence.

There is a lot of probabilistic graphical models used in real applications but we will not deal with all of them. We will be only concerned with vine copula graphical model, where the structure is typically defined by a tree graph and to each edge corresponds a function called copula that we will define later. Copula approach can be viewed as generalization of Gaussian graphical model, where we do not assume normal distribution of marginals. Dropping this assumption leads to the need of determining copula function that tells us how the marginal distributions are dependent on each other. Determining high dimensional copula function can be very hard. To deal with this obstacle we can use only bivariate copulas and model the dependence structure pairwise. Great promise in the future research show vine copulas and we have chosen this particular model for our thesis.

1.3 Bayesian inference

The backbone of probabilistic approach to machine learning and its further use in graphical models is Bayes' theorem. The most general form of Bayes' theorem can be written as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Where A, B are events and P is probability of event. We can very easily consider as event A validity of some hypothesis and event B to be occurrence of data corresponding to hypothesis. Then we can write Bayes' theorem in a convenient form:

$$P(hypothesis|data) = \frac{P(data|hypothesis)P(hypothesis)}{P(data)} \tag{2}$$

In this equation the *hypothesis* is for example a model that we believe describes the *data*. We can have many models available and it might be desirable to express how confident we are in our model and it is best represented by probability $P(hypothesis)$. This value is called *prior*. In order to make the model meaningful we want to make some predictions from the model. The probability $P(data|hypothesis)$ that the model produces data we have already observed is called *likelihood*. Multiplying prior times the likelihood and normalizing it by the constant $P(data)$ gives *posterior predictive distribution*, which tells us the probability how well the model fits the data.

To put this fundamental rule in mathematical notation we will denote θ vector of parameters for model m and by D we will denote the set of observed data $D = \{x_1, x_2, ..., x_n\}$. Rewriting equation (2) gives us:

$$P(\theta|D, m) = \frac{P(D|\theta, m)P(\theta|m)}{P(D|m)} \tag{3}$$

One can see that the confidence of our model is manifested through parameters of the model, but the question whether the model is the best one or not does not fall into this estimate. The choice of the model is in the end just an assumption we need to doubt and by some variant of "no free lunch" theorem follows that there does not exist a model good for every problem.

The Bayes rule itself follows from two basic rules of probability. Namely the product rule $P(x, y) = P(x)P(y|x)$ and the sum rule $P(x) = \sum_y P(x, y)$. The sum rule is sometimes called marginalization of probability and both rules will be extensively used throughout the whole thesis but formulated in terms of probability distributions and probability density functions. Given d-dimensional random vector $\mathbf{X} = (X_1, ..., X_d)$ with probability density function f we can calculate conditional density function $f(x_k|x_1 = X_1, ..., x_{k-1} = X_{k-1}, x_{k+1} = X_{k+1}, ...x_d = X_d)$ by formula:

$$f(x_k|x_1, ..., x_{k-1}, x_{k+1}, ..., x_d) = \frac{f(x_1, ..., x_d)}{\int_{-\infty}^{\infty} f(x_1, ..., x_d)dx_k} \tag{4}$$

where we have used shorter notation for conditional density.

1.4 Copulas

In recent years copula theory found its place in statistics. Here we note brief historical introduction mentioned in [1]. The history of copulas begins with Fréchet [2] who considered two random variables X_1, X_2 defined on the same probability space with distributions F_1, F_2 and studied the set $\Gamma(F_1, F_2)$ of bivariate distribution functions. One can see that joint distribution mainly depends on dependence of the two variables. If X_1 and X_2 are independent then clearly the distribution function is just product of its marginals $F(x_1, x_2) = F_1(x_1)F_2(x_2)$. Following some preliminary studies in 1956 the deepest result was obtained by Sklar introducing the notion, and the name, of *a copula* and proving fundamental theorem of copula theory that bears his name [3].

We can begin with stating definition viewing copula as a special case of distribution function.

Definition 1 *$C : [0,1]^d \to [0,1]$ is a d-dimensional copula if C is a joint cumulative distribution function of a d-dimensional random vector on the unit cube $[0,1]^d$ with uniform marginals. [4]*

We give examples of basic copulas associated with random vector $\mathbf{U} = (U_1, ..., U_d)$ to explore bounds of correlation.

- the *Independence copula* $\Pi_d(\mathbf{u}) = u_1 u_2 ... u_d$ whose components are independent and uniformly distributed on unit interval. One can easily observe that viewing copula as special case of joint distribution function the condition of independence being expressible as product of its marginals is satisfied.

- the *comonotonicity copula* $M_d(\mathbf{u}) = min(u_1, ..., u_d)$ represents extreme of strong dependency among random variables the vector, what can be expressed as $U_i = T_i(Y)$ where T_i is some monotone increasing transformation.

- for two dimensional case, we can define *countermonotonicity copula* $W_2(u_1, u_2) = max(u_1 + u_2 - 1, 0)$. This case refers to perfect negative dependence $U_2 = T(U_1)$ for strictly decreasing function T.

By natural generalization of countermonotonicity copula one can obtain $W_d(\mathbf{u}) = max(\sum_{i=1}^{d} u_i - d + 1, 0)$, which is not a copula in general, but provides lower bound for any given copula. These bounds are called Fréchet-Hoeffding bounds:

$$W_d(\mathbf{u}) \leq C(\mathbf{u}) \leq M_d(\mathbf{u}) \tag{5}$$

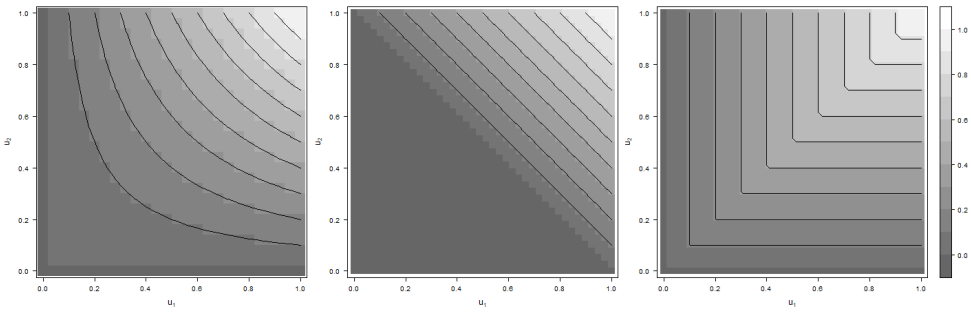

Figure 3: Contour plot of independence copula, comonotonicity copula and countermonotonicity copula

The backbone of copula theory considering its application is without doubt Sklar's theorem:

Theorem 1 *Let F be a distribution function with marginals $F_1, ..., F_d$. There exists a copula C such that for all $(x_1, ..., x_d) \in [-\infty, \infty]^d$,*

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d)) \tag{6}$$

C is uniquely determined on $Range(F_1) \times Range(F_2) \times ... \times Range(F_d)$ and hence it is unique when all marginals are continuous.

In subsequent subsections we will introduce some classes of copulas. We will consider only bivariate copulas.

1.4.1 Elliptical copulas

As written in [1] one can define elliptical copula as follows.

Definition 2 *Let $\mathbf{X}$ be d-dimensional random vector having elliptical distribution with mean vector μ, covariance matrix Σ and generator $g : [0, \inf) \to [0, \inf)$. Suppose that, for every $i \in \{1, 2, ..., d\}$, $(\frac{X_i}{\sigma_{ii}})$ has marginal distribution F_g. We call elliptical copula the distribution function of the random vector*

$$(F_g(\frac{X_1}{\sqrt{\sigma_{11}}}), F_g(\frac{X_2}{\sqrt{\sigma_{22}}}), ..., F_g(\frac{X_d}{\sqrt{\sigma_{dd}}}))$$

If $g(t) = (2\pi)^{-\frac{d}{2}} e^{-\frac{t}{2}}$ then we speak of Gaussian copula. Similarly, $g(t) = c(1 + \frac{t}{\nu})^{-\frac{d+\nu}{2}}$, for a suitable constant c, generates the multivariate t-Student distribution with ν degrees of freedom. [1].

Later on when we will use vine copula for classification we will use only density functions and so since distribution for elliptical copula is typically not in closed form the performance of our calculations will be better. We provide and example of bivariate Gaussian copula:

$$C_\theta(u_1, u_2) = \int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} \frac{1}{2\pi\sqrt{1-\theta^2}} (-\frac{s^2 - 2\theta st + t^2}{2(1-\theta^2)}) dsdt$$

where $\theta \in [-1, 1]$ and Φ^{-1} denotes the inverse of univariate Gaussian distribution [1].

1.4.2 Archimedean copulas

Definition 3 *A d-dimensional copula C is called Archimedean if it admits the representation:*

$$C(\mathbf{u}) = \Psi(\Psi^{-1}(u_1) + \Psi^{-1}(u_2) + ... + \Psi^{-1}(u_d))$$

for all $\mathbf{u} \in [0, 1]^d$ and for some Archimedean generator Ψ.

Archimedean generator is any decreasing and continuous function $\Psi : [0, \infty) \to [0, 1]$ that satisfies the conditions $\Psi(0) = 1$ and $lim_{t\to\infty}\Psi(t) = 0$ and which is strictly decreasing on $[0, inf\{t|\Psi(t) = 0\})$. By convention, $\Psi(+\infty) = 0$ and $\Psi^{-1}(0) = inf\{t \geq 0|\Psi(t) = 0\}$, where Ψ^{-1} denotes the pseudo-inverse of Ψ [1].

Len us give some examples of bivariate Archimedean copulas which will be occur once again in Appendix when estimating vine copula. In definition (3) we saw that Archimedean copula can be defined via generator function. For bivariate case by setting $\Psi(t) = -ln(1 - (1 - t)^\theta)$ for $\theta \geq 1$ we obtain Joe copula and by $\Psi(t) = -ln(\frac{e^{-\theta t}-1}{e^{-\theta}-1})$ for $\theta \neq 0$ we obtain Frank copula [4].

Example of Archimedean bivariate two-parametric case we note $BB1$ and $BB7$. Generator functions for these copulas are $\Psi(t) = (1 + t^{\frac{1}{\delta}})^{-\frac{1}{\theta}}$ for $\theta > 1$, $\delta > 0$ and $\Psi(t) = 1 - [1 - (1 + t)^{-\frac{1}{\delta}}]^{\frac{1}{\theta}}$ for $\theta \geq 1$, $\delta > 0$ [5].

1.4.3 Rotated, survival and extreme value copulas

It is intuitive that if we look at scatter plot of some copula one can rotate the plot and describe this new dependence via almost the same analytical

representation as has the initial plot. Rotations for copula density by 90, 180 and 270 degrees are given as fallows: [7]

$$c^{90}(u_1, u_2) = c(1 - u_2, u_1)$$
$$c^{180}(u_1, u_2) = c(1 - u_1, 1 - u_2)$$
$$c^{270}(u_1, u_2) = c(u_2, 1 - u_1)$$

Copula rotated by 180 degrees is called survival copula. This copula occupies special place among copulas obtained by rotation, because when we define survival function as for example in [5], then survival copula analogously relates joint survival function to its univariate margins and so provides a tool for survival analysis with good interpretation.

Another interesting class of copulas are extreme value copulas.

Definition 4 *A copula C is called extreme-value copula if there exists a copula C_F such that, for $n \to \infty$*

$$C_F(u_1^{\frac{1}{n}}, ..., u_d^{\frac{1}{n}}) \to C(u_1, ..., u_d)$$

$\forall (u_1, ..., u_d)^T \in [0, 1]^d$.

A way to construct extreme-value copulas is by defining a convex function $A : [0, 1] \to [\frac{1}{2}, 1]$ satisfying $max\{t, 1 - t\} \leq A(t) \leq 1$ for all $t \in [0, 1]$. The extreme value copula can be defined by:

$$C(u, v) = exp[ln(uv)A(\frac{ln(v)}{ln(uv)})]$$

and if we set $A(t) = 1 - \beta + (\beta - \alpha)t + [\alpha^r t^r + \beta^r (1 - t)^r]^{\frac{1}{r}}$, $0 \leq \alpha, \beta \leq 1$, $1 \leq r < \infty$, $t \in [0, 1]$. [8]

2 Regression Analysis

Let $X_1, ..., X_{d-1}$ be continuous random variables and Y random variable dependent on them. A regression model relates $Y \approx f(X_1, ..., X_{d-1}, \theta)$, where θ is vector of parameters. In this chapter we will consider simple linear regression model and more informative vine copula model. Both models will aim to determine root-mean-square deviation of atomic positions (RMSD) based on

properties of protein structure written in the table below, where $F1, ..., F9$ denotes physical properties and in our equations we later use standard notation $X_1, ..., X_9$ for random variables and $F_1, ..., F_9$ for marginal distribution functions.

$RMSD$	Root-mean-square deviation
$F1$	Total surface area
$F2$	Non polar exposed area
$F3$	Fractional area of exposed non polar residue
$F4$	Fractional area of exposed non polar part of residue
$F5$	Molecular mass weighted exposed area
$F6$	Average deviation from standard exposed area of residue
$F7$	Euclidian distance
$F8$	Secondary structure penalty
$F9$	Spacial Distribution constraints (N,K Value)

RMSD is an indicator in protein-structure-prediction-algorithms. We will not use full size of data set since we have only restricted computational resources and data are used primary to illustrate use of copulas in regression and classification problems, not to compete with large neural networks. Our data will consist of 10000 random samples from original dataset of size 45730.

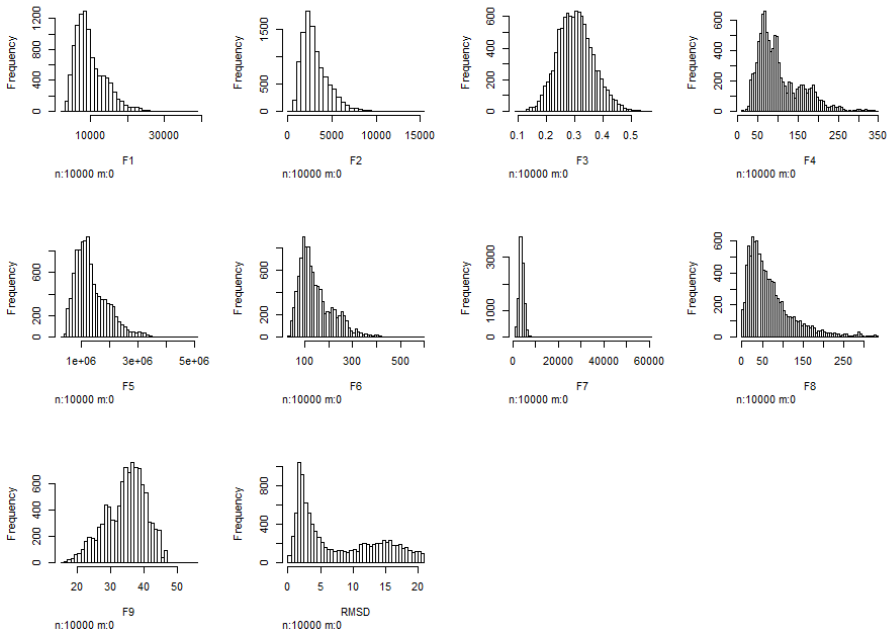

Figure 4: Histograms

Using R package *lmom* we have fitted these densities (Figure 4) with gamma distribution except for $F3$ where we have used normal distribution and $F4$, $F6$ where we have used three-parameter log-normal distribution. The $RMSD$ data we have used mixture normal distribution estimated with help of R package $mixR$.

2.1 Linear regression model

If function f is linear we speak of linear regression model of the form $Y = \theta_1 X_1 + ... + \theta_d X_{d-1} + \epsilon$, where ϵ is some random noise with mean value zero. Given input and output data we can estimate parameters of model hence we speak of supervised machine learning. If we assume that our predictor variables $X_1, ..., X_{d-1}$ have Normal distributions denoted by $N(\mu_i, \sigma_i^2)$ we can calculate expected value $E[Y]$. By linearity of expected value we can write $E[Y] = \theta_1 E[X_1] + ... + \theta_d E[X_{d-1}]$.

At this point let us introduce graphical model called Gaussian graphical model. This model is used to model probability density function for random

vector $\mathbf{X} = (X_1, ..., X_d)$ with assumption that all random variables have Normal distribution. Under this assumption we know that the dependence between random variables is entirely specified by correlation matrix Σ and so we can write:

$$f_{\mathbf{X}}(x_1, ..., x_d) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)} \tag{7}$$

Now we can calculate conditional probability density function for conveniently chosen random variable X_d. For this purpose we will use equation (4) so we can write:

$$f_{X_d|\mathbf{X}_{-d}}(x_d) = \frac{f_{\mathbf{X}}(x_1, ..., x_d)}{f_{\mathbf{X}_{-d}}(x_1, ..., x_{d-1})} \tag{8}$$

where $f_{\mathbf{X}_{-d}}(x_1, ..., x_{d-1})$ is joint normal distribution marginalized over X_d and by $\mathbf{X}_{-d}$ we denote random vector $\mathbf{X}$ without X_d. Omitting too much technical details we can observe that random variable X_d is determined by linear representation of random variables in random vector $(X_1, ..., X_{d-1})$ and so is equivalent to linear regression model in terms of expectation value and so by estimating parameters we ought to end up with the same result. Identifying components in vector $\mathbf{X}$ with nodes of the graph and edges defined by weights of correlation matrix we can visualize linear dependence structure of our model displayed in Figure 5. As we can see the linear dependence structure is between $F1-F5$ and $F2-F3$ corresponding to variables $X1-X5$ and $X2-X3$.

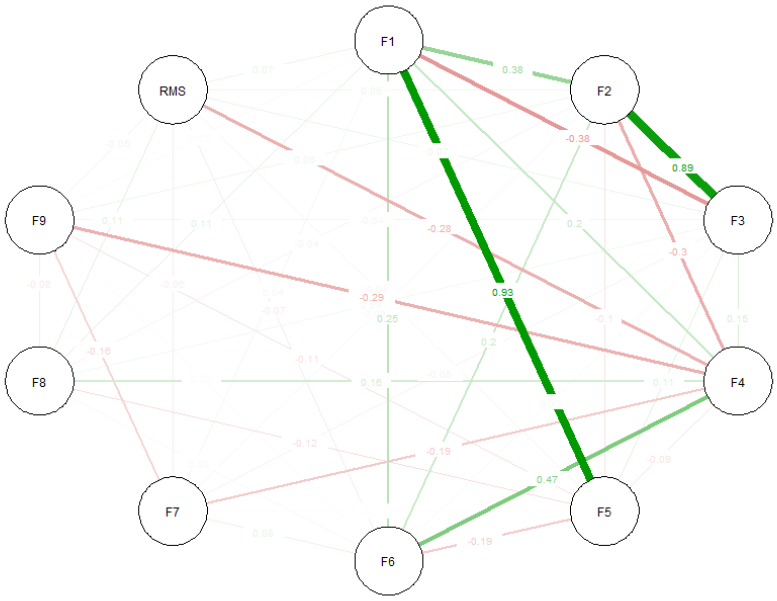

Figure 5: Correlation graph

We have fitted linear regression model on our RMSD-data and obtained linear equation with coefficients $\theta_0, ..., \theta_9$. The expectation value of y_{RMSD} is:

$$y_{RMSD} = 7.114 + 0.001755x_1 + 0.001364x_2 + 17.69x_3 - 0.1100x_4$$
$$-0.000005026x_5 - 0.02372x_6 - 0.0002776x_7 + 0.01524x_8 - 0.1274x_9$$

We can see that relatively largest weight is on coefficient standing next to x_3 and so we can say that RMDS is mainly linearly dependant on variable X_3. This however is not true because values of variable X_3 are normally distributed around mean 0.3 with small variance and other variables attain much larger values. Hence linear model is not giving us any deeper knowledge and will serve us only as reference.

On the Figure 6 we can see residuals of validation set from linear model.

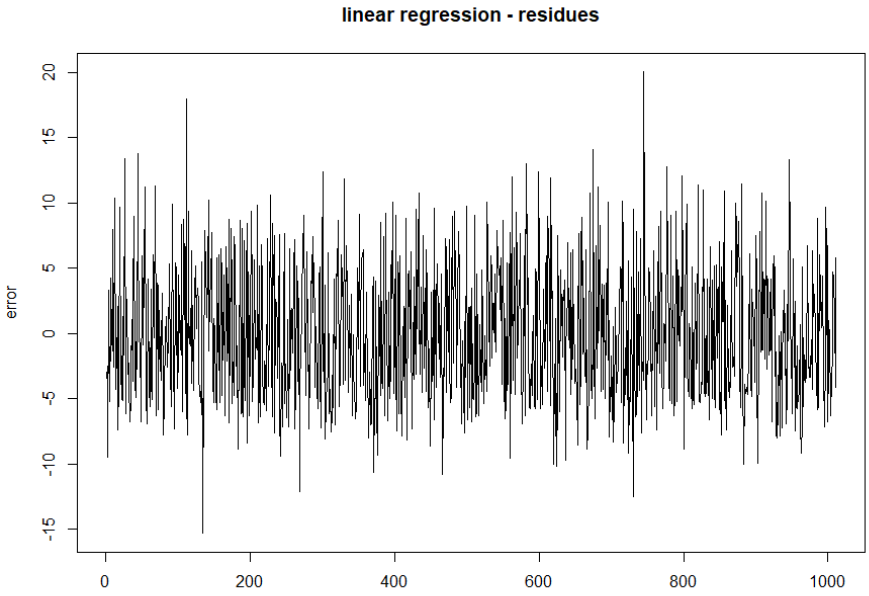

Figure 6: Residuals of linear regression model on test set.

2.2 Vine copula model

In the previous model we have not considered non-linear dependence among random variables. In equation (4) if we consider independence among random variables we get:

$$f(x_k|x_1, ..., x_{k-1}, x_{k+1}, ..., x_d) = \frac{\prod_{i=1}^{d} f_i(x_i)}{\int_{-\infty}^{\infty} f(x_1, ..., x_d)dx_k} \tag{9}$$

In theorem (1) we can differentiate distribution function to obtain:

$$\begin{aligned} f(x_1, ..., x_d) &= c(F_1(x_1), ..., F_d(x_d)) \prod_{i=1}^{d} \frac{\partial F_i(x_i)}{\partial x_i} \\ &= c(F_1(x_1), ..., F_d(x_d)) \prod_{i=1}^{d} f_i(x_i) \end{aligned}$$

where function c is density copula. We can see a product of marginal densities f_i with copula density c interpreted as a term to handle the dependencies.

Combining this equation with equation (4) yields:

$$f(x_k|x_1,...,x_{k-1},x_{k+1},...,x_d) = \frac{c(F_1(x_1),...,F_d(x_d))\prod_{i=1}^{d} f_i(x_i)}{\int_{-\infty}^{\infty} c(F_1(x_1),...,F_d(x_d))\prod_{i=1}^{d} f_i(x_i)dx_k}$$

Since all terms except x_k are constant they will cancel out to get:

$$f(x_k|x_1,...,x_{k-1},x_{k+1},...,x_d) = \frac{c(F_1(x_1),...,F_d(x_d))f_k(x_k)}{\int_{-\infty}^{\infty} c(F_1(x_1),...,F_d(x_d))f_k(x_k)dx_k} \quad (10)$$

Resulting in univariate conditional density function. These equations are valid for any copula and any random vector with continuous marginal densities. Further on we will use vine copulas as our copula functions.

The definition of vine copula is very complicated, but viewing vine copula as probabilistic graphical model we can use definition that can be found in [5]

Definition 5 *ν is regular vine on d elements, with $\varepsilon(\nu) = \varepsilon_1 \cup ... \cup \varepsilon_{d-1}$ denoting the set of edges of ν, if 1. $\nu = \{\tau_1,...,\tau_d\}$ [consists of d-1 trees]*
2. τ_1 is a connected tree with nodes $N_1 = \{1,...,d\}$, ad edges ε_1; for $l = 2,...,d-1$: T_l is a tree with nodes $N_l = \varepsilon_{l-1}$ [edges in a tree becomes nodes in the next tree]
3. for $l = 2,...,d-1$ and for $\{n_1,n_2\} \in \sharp(n_1 \ominus n_2) = 2$, where $\ominus$ denotes symmetric difference and $\sharp$ denotes cardinality [nodes joined in an edge differ by two elements]

Before we start using mathematical apparatus of regular vines we need to estimate marginal distribution functions and marginal densities. We will not dwell on this task for too long, since our main motivation is to transform every random variable to uniformly distributed random variable on interval $[0,1]$. Then we will need marginal density functions for further calculations. The result of transformation can be seen on scatter plots depicting pure dependence structure.

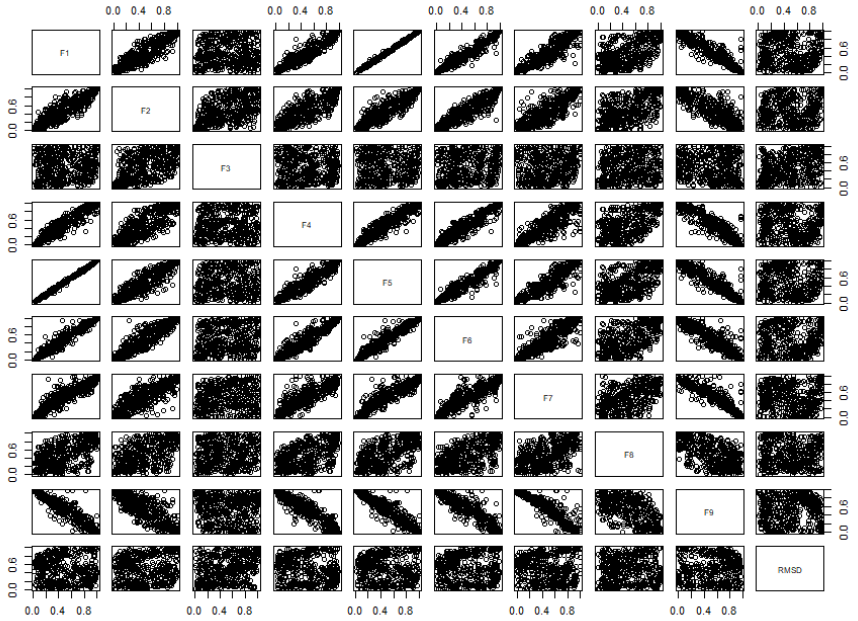

Figure 7: Scatter plot matrix

We can see in Figure 7 linear dependence between $F1 - F5$ and independence of $F3$ to all other variables. Other variables seems to be non-linearly dependent.

We have used well known R-package called VineCopula and fitted regular vine copula to our transformed data.

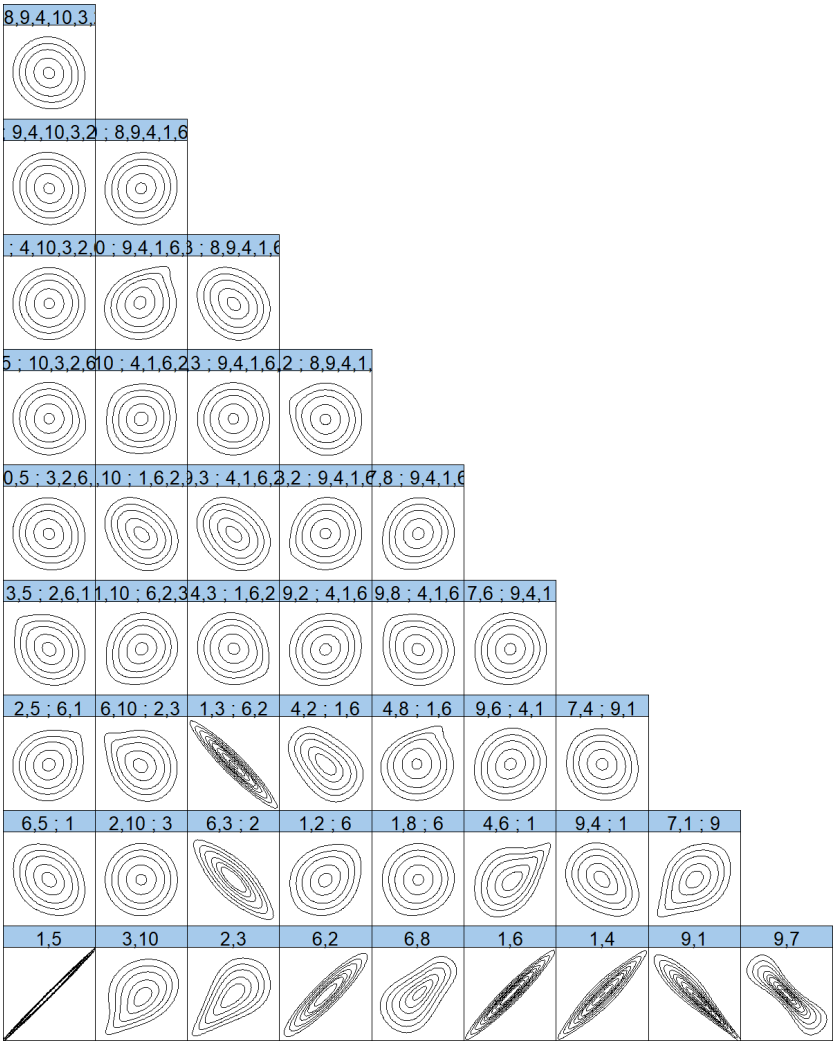

Figure 8: Contour plot of bivariate joint distribution with copula and normal marginals

The dependence structure is indeed very rich (Figure 9). We have displayed all trees of vine copula structure and we can see t-copula, Gaussian copula, Frank copula and many other copulas. For more information about these copulas we refer to [1], [4] or [5]. We list the whole structure of our vine copula in Appendix.

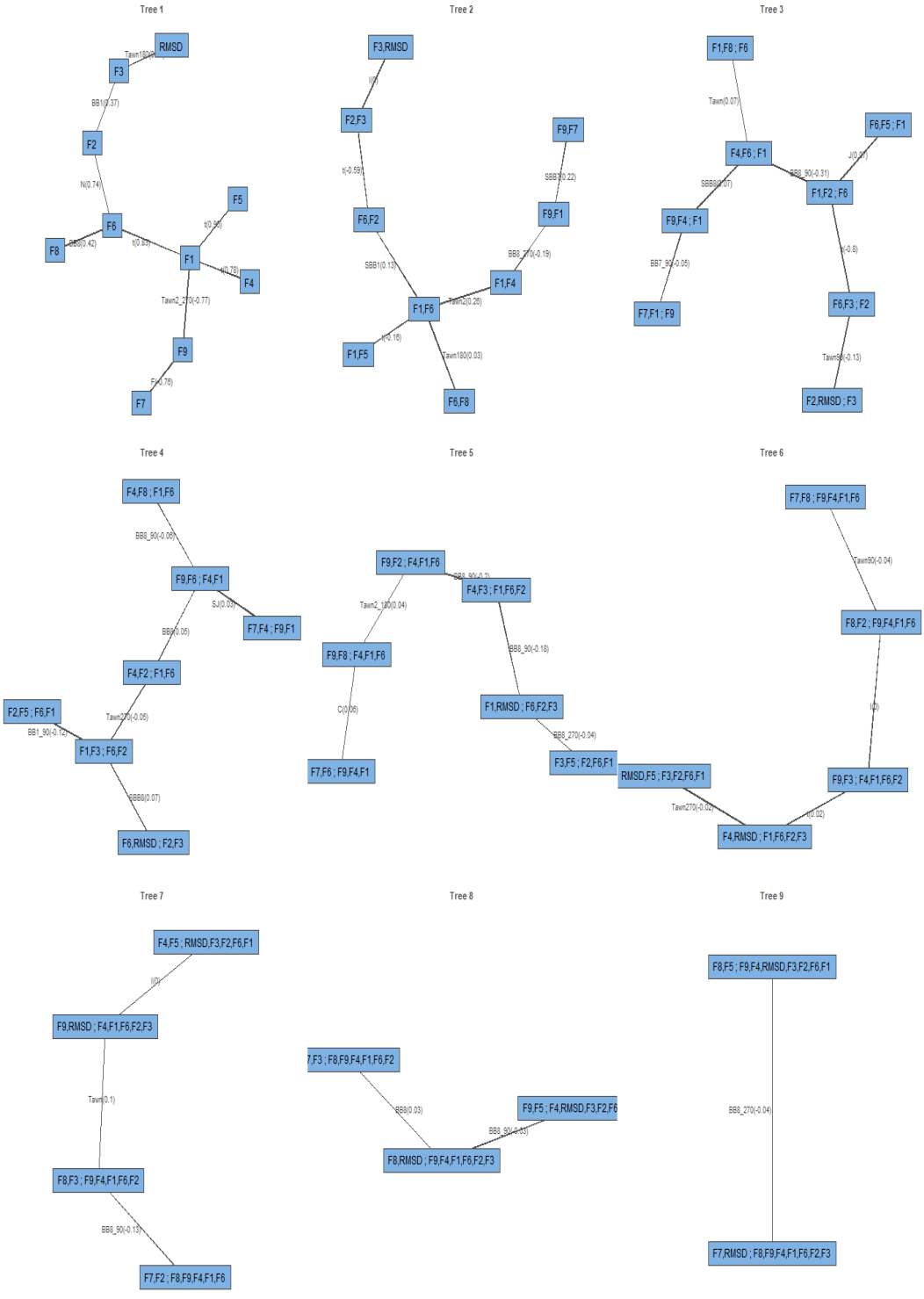

Figure 9: Regular vine copula

By formula (10) we have calculated RMSD conditional density function $f_{RMSD}(x_{RMSD}|x_1, ..., x_9)$ for several instances of random vector $(X_1, ..., X_9)$. On closer inspection of conditional density function (Figure 10) we can see that density function typically attains two local maxima in two points. In language of probability theory we call such points modes. One local maximum is somewhere between zero and five and second local maximum is above number five.

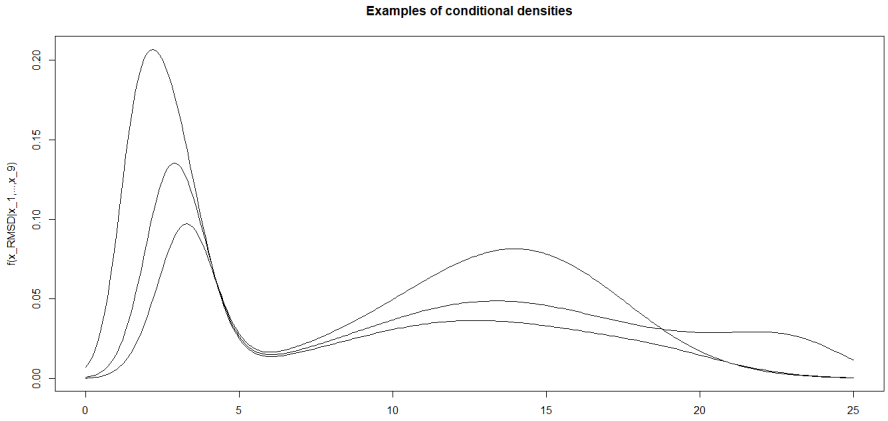

Figure 10: Examples of conditional distributions for three randomly chosen predictors

A mode is roughly defined as most probable value. For instance in case of normal distribution is mode equal to expectation value. If our conditional distribution function had only one mode the situation would be much simpler and we could state this value to be our prediction. However, situation is not in our favour and we have two modes so we need to create a procedure to tell us which mode to prefer. Creating such procedure is no easy task, since plenty of possibilities arise. One could consider only the global maximum, but by looking on our dataset on Figure 11 we see that most values are concentrated in the first peak and that we would need to give up correctly predicting values in second concentrated around second local maxima.

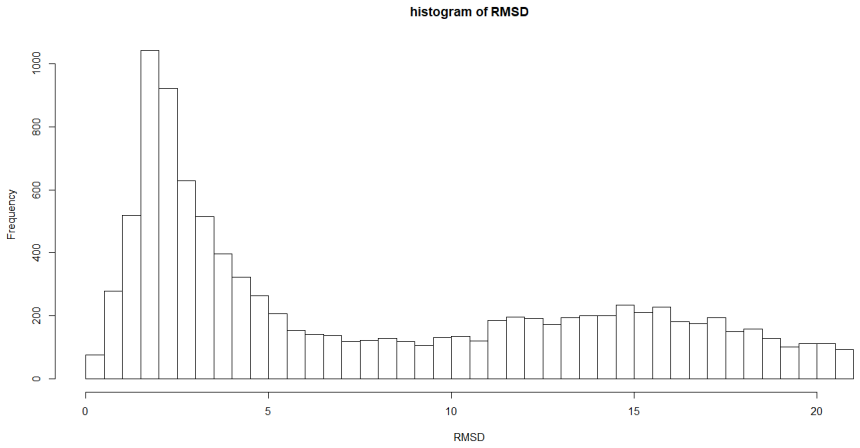

Figure 11: Histogram of $RMSD$

Another possibility would be taking expectation value as a prediction, but that would still favour lower values of RMSD as well and on average would give worse predictions. We have decided to transform this problem to supervised classification problem as will be seen in later chapter.

3 Probabilistic classification

Classification and regression are two very related tasks. Generally classifier is some rule, or a function f, that assigns to a sample x a class label y. We can write this simply as $y = f(x)$, where x comes from sample space X, while the class labels form finite set Y defined at the beginning. A typical example of such classification task is handwritten digit recognition where sample space is set of all images of given pixel resolution and class labels form a set $Y = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. This task can be solved by neural networks with high success rate where on the input goes all pixels of given image. Even though neural network calculates for every digit value in interval $[0, 1]$ and we can base our belief on this output value to given digit, we cannot consider it as a probability, because neural networks are in general not a distribution functions. This is where probabilistic classification gains the upper hand based on Bayesian inference. Once we obtain the probability

$P(class|data)$ we can predict class y with the highest probability by formula:

$$y = \underset{y_k}{\operatorname{argmax}}[P(class = y_k|data)] \tag{11}$$

The the main step when doing probabilistic classification is to calculate probability distribution. For this task we will later use regular vine copulas.

3.1 Naive Bayes

The simplest and most used classifier is naive Bayes classifier. We present it as analogical case to linear regression model. Naive Bayes classifier as name suggest is based on Bayes theorem and assumes independence of variables, thus being probabilistic classifier.

Consider random vector $\mathbf{X} = (X_1, ..., X_d)$ and K classes denoted by w_k for k-th class. From Bayes theorem (1) taking event A to be class and event B to be realization of random vector $\mathbf{x} = (x_1, ..., x_d)$ we can write:

$$P(w_k|\mathbf{x}) = \frac{P(\mathbf{x}|w_k)P(w_k)}{P(\mathbf{x})} \tag{12}$$

We ask the probability that given a realization of random vector $\mathbf{x}$ what is the probability that $\mathbf{x}$ is in class w_k. Probability $P(w_k)$ is prior probability of obtaining k-th class. We will deal with this term in classical fashion $P(w_k) = \frac{N_k}{N}$, where N is size of our dataset and N_k is observed size of data in class k. The term $P(\mathbf{x}|w_k)$ is interpreted as probability of obtaining realization $\mathbf{x}$ if we restrict our attention only on k-th class. Since we assume independence we can factor $P(\mathbf{x}|w_k)$ as product of marginal probabilities given k-th class. Term in the denominator is probability of realization $\mathbf{x}$ throughout all K classes. Since all classes are mutually exclusive we write $P(\mathbf{x}) = \sum_{j=1}^{K} P(\mathbf{x}|w_j)$. Putting it all together we can write:

$$P(w_k|\mathbf{x}) = \frac{\prod_{i=1}^{d} P(x_i|w_k)}{\sum_{i=1}^{K}\prod_{i=1}^{d} P(x_i)} \frac{N_k}{N} \tag{13}$$

We have fitted RMSD by bimodal normal distribution, which is just sum of two normal distributions with one mode and spit our RMSD according to local minimum value as seen on Figure 12 to two classes.

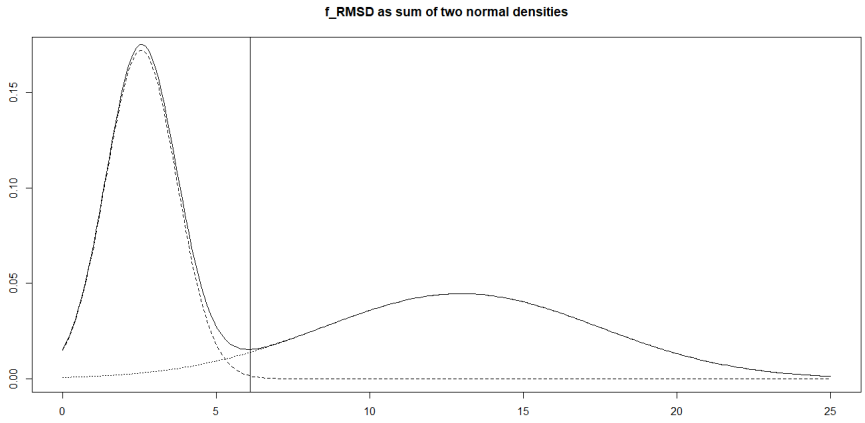

Figure 12: Splitting $RMSD$ mixture density

We have used R-package e1071 to fit naive Bayes model and make our predictions. In table bellow are shown results of predictions.

real \ predicted	class 1	class 2
class 1	270	98
class 2	280	363

Size of our test data is $n_{test} = 1011$. From table we can calculate that our predictions were 62.6% correct where 280 realizations in class 2 were incorrectly putted in class 1.

3.2 Regular vine probabilistic classification

Analogically as in regression model we will correct independence assumption we have made in naive Bayes model. We define probability $P(\mathbf{x}|w_k) = lim_{\epsilon \to \inf} F(x_1 - \epsilon \leq X_1 < x_1 + \epsilon, ..., x_d - \epsilon \leq X_d < x_d + \epsilon)|w_k)$, where F is joint distribution function of random vector $(X_1, ..., X_d)$. For small enough ϵ we can write (13) without independence assumption as:

$$P(w_k|\mathbf{x}) = \frac{f(x_1, ..., x_d|w_k)}{\sum_{j=1}^{K} f(x_1, ..., x_d|w_j)P(w_j)} \frac{N_k}{N} \tag{14}$$

We will deal with conditional density function $f(x_1, ..., x_d|w_k)$ as before, but this time we need to fit vine copula for each class. That is

$$f(x_1, ..., x_d|w_k) = c_k(F_1(x_1), ..., F_d(x_d)) \prod_{j=1}^{d} f_j(x_j) \tag{15}$$

where c_k is regular vine copula density for k-th class.

Setting $K = 2$ and $d = 9$ we can finally write our vine copula classifier for first class (predicted value of RMSD near first mode) as:

$$P(w_1|\mathbf{x}) = \frac{f(x_1, ..., x_9|w_1)}{f(x_1, ..., x_9|w_1)P(w_1) + f(x_1, ..., x_9|w_2)P(w_2)} \frac{N_1}{N} \tag{16}$$

Since both classes are mutually exclusive and we have only two we can calculate $P(w_2|\mathbf{x}) = 1 - P(w_1|\mathbf{x})$, which is computationally more effective.

Regular vine copula fitted on data in class 1 and another one on data in class 2 was structurally not very different than vine copula estimated in regression problem so we will not deal with closer inspection. Probabilistic classifier based on regular vine copula did better in classification task than naive Bayes model. We can see results on table bellow:

real \ predicted	class 1	class 2
class 1	430	162
class 2	120	299

From table we can calculate that our predictions were 72.1% correct where 120 realizations in class 2 were incorrectly putted in class 1 and 162 realizations in class 1 were incorrectly putted in class 2.

4 Combining regression and classification

To sum up what we did we will summarize steps of our approach using copulas:
1. Observe number of local maxima (more than one) in histogram of output data.
2. Fit output data with distribution and density function.
3. Split output data to classes. Number of classes is equal to number of local maxima.

4. Estimate copula for regression problem.
5. Estimate copula for classification problem. Estimating must be done for each class separately.
6. Use equation (10) and (14),(15) as such:

Algorithm 1 algorithms

procedure PREDICT(*test*) ▷ predicts RMSD based on test set predictors
  for each test case **do**
    $mL \leftarrow condDensityLocalMaximumOfClass1(test_case)$
    $mU \leftarrow condDensityLocalMaximumOfClass2(test_case)$
    $class \leftarrow whichClass(test_case)$
    if class=1 **then** $Predictions$.append(mL)
    if class=2 **then** $Predictions$.append(mU)
  return $Predictions$ ▷ returns vector of responses for given predictors

On author's github page https://github.com/fratric/diplomaThesis.git can be viewed implementation of this algorithm. By this procedure we have made predictions with combining vine copula regression model with Bayes classification and then vine copula regression model with vine copula classification. In this table we provide root-mean-square error and mean-average error for all models we have employed.

model	mean-average error	root-mean-square error
Linear model	4.510966	5.31519
Vine + Bayes	5.204006	6.873046
Vine + Vine	4.111231	5.999629

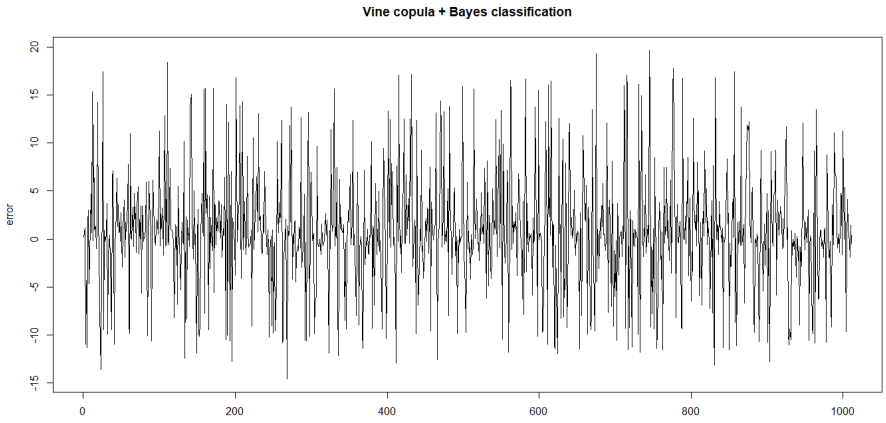

Figure 13: Residuals for vine copula regression model combined with naive Bayes classification

We can see that combination of Vine regression model with Bayes classifier performed very poorly. The main reason for this is poor classification. Both from the table and from Figure 13 and Figure 14 we can see that residuals for vine copula regression model have very large root-mean-square error compare to linear model. This can be explained by observation that if classifier, either Bayes or vine, classified incorrectly then vine regression model had no chance to predict correct value. On the other hand, we can from plot of residuals from Vine+Vine model that if we would restrict our attention only to correctly classified variables then vine regression model would be very precise. This means that main issue with our model is classification and so a progress in this direction is required.

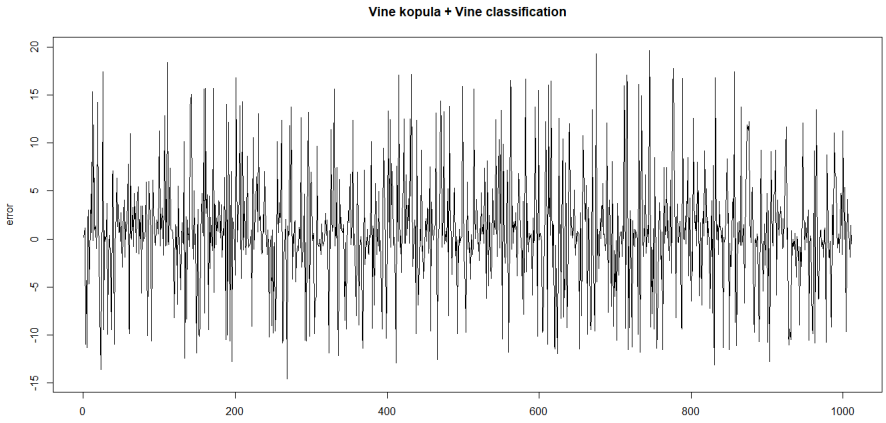

Figure 14: Residuals for vine copula regression model combined with vine copula classification

5 Conclusions

In this thesis we have demonstrated use of copulas on regression problems and on probabilistic classification problems. The use of Gaussian copulas on probabilistic classification was already used in few papers in recent years and applications of Vine or factor copulas are particularly rare.

The dataset we have used was already tackled in paper [6] with mean-average error equal to 3.845204 and surpassing neural network [6] with mean-square error equal to 4.202547 and linear model with mean-square error equal to 4.510966 is satisfactory especially when vine copula also describes the dependence structure providing us with better intuition about physics hidden behind the curtain of data.

6 Appendix

Regular vine copula with following pair-copulas estimated by method RVineStructureSelect by Akaike information criterion in R-package VineCopula. For more information https://cran.r-project.org/web/packages/VineCopula/

Tree 1:	copula type
1,5	t (par = 1, par2 = 10.53, tau = 0.96)
3,10	Rotated Tawn type 1 180 degrees (par = 1.63, par2 = 0.46, tau = 0.23)
2,3	BB1 (par = 1.06, par2 = 1.04, tau = 0.37)
6,2	Gaussian (par = 0.92, tau = 0.74)
6,8	BB8 (par = 3.27, par2 = 0.85, tau = 0.42)
1,6	t (par = 0.97, par2 = 18, tau = 0.83)
1,4	t (par = 0.94, par2 = 6.51, tau = 0.78)
9,1	Rotated Tawn type 2 270 degrees (par = -4.73, par2 = 0.97, tau = -0.77)
9,7	Frank (par = -14.66, tau = -0.76)

Tree 2:	copula type
6,5;1	t (par = -0.25, par2 = 19.78, tau = -0.16)
2,10;3	Independence
6,3;2	t (par = -0.8, par2 = 9.1, tau = -0.59)
1,2;6	Survival BB1 (par = 0.21, par2 = 1.04, tau = 0.13)
1,8;6	Rotated Tawn type 1 180 degrees (par = 1.13, par2 = 0.11, tau = 0.03)
4,6;1	Tawn type 2 (par = 1.7, par2 = 0.49, tau = 0.26)
9,4;1	Rotated BB8 270 degrees (par = -1.94, par2 = -0.81, tau = -0.19)
7,1;9	Survival BB7 (par = 1.35, par2 = 0.17, tau = 0.22)

Tree 3:	copula type
2,5;6,1	Joe (par = 1.13, tau = 0.07)
6,10;2,3	Rotated Tawn type 1 90 degrees (par = -1.36, par2 = 0.35, tau = -0.13)
1,3;6,2	t (par = -0.95, par2 = 17.73, tau = -0.8)
4,2;1,6	Rotated BB8 90 degrees (par = -3.74, par2 = -0.63, tau = -0.31)
4,8;1,6	Tawn type 1 (par = 1.66, par2 = 0.09, tau = 0.07)
9,6;4,1	Survival BB8 (par = 1.37, par2 = 0.75, tau = 0.07)
7,4;9,1	Rotated BB7 90 degrees (par = -1.02, par2 = -0.08, tau = -0.05)

Tree 4:	copula type
3,5;2,6,1	Rotated BB1 90 degrees (par = -0.04, par2 = -1.11, tau = -0.12)
1,10;6,2,3	Survival BB8 (par = 1.22, par2 = 0.92, tau = 0.07)
4,3;1,6,2	Rotated Tawn type 1 270 degrees (par = -1.16, par2 = 0.21, tau = -0.05)
9,2;4,1,6	BB8 (par = 1.16, par2 = 0.89, tau = 0.05)
9,8;4,1,6	Rotated BB8 90 degrees (par = -1.15, par2 = -0.97, tau = -0.06)
7,6;9,4,1	Survival Joe (par = 1.05, tau = 0.03)

Tree 5:	copula type
10,5;3,2,6,1	Rotated BB8 270 degrees (par = -1.18, par2 = -0.81, tau = -0.04)
4,10;1,6,2,3	Rotated BB8 90 degrees (par = -3.07, par2 = -0.52, tau = -0.18)
9,3;4,1,6,2	Rotated BB8 90 degrees (par = -5.03, par2 = -0.35, tau = -0.2)
8,2;9,4,1,6	Rotated Tawn type 2 180 degrees (par = 1.22, par2 = 0.1, tau = 0.04)
7,8;9,4,1,6	Clayton (par = 0.13, tau = 0.06)

Tree 6:	copula type
4,5;10,3,2,6,1	Rotated Tawn type 1 270 degrees (par = -1.38, par2 = 0.02, tau = -0.02)
9,10;4,1,6,2,3	t (par = 0.03, par2 = 9.8, tau = 0.02)
8,3;9,4,1,6,2	Independence
7,2;8,9,4,1,6	Rotated Tawn type 1 90 degrees (par = -1.31, par2 = 0.08, tau = -0.04)

Tree 7:	copula type
9,5;4,10,3,2,6,1	Independence
8,10;9,4,1,6,2,3	Tawn type 1 (par = 1.41, par2 = 0.2, tau = 0.1)
7,3;8,9,4,1,6,2	Rotated BB8 90 degrees (par = -2.56, par2 = -0.51, tau = -0.13)

Tree 8:	copula type
8,5;9,4,10,3,2,6,1	Rotated BB8 90 degrees (par = -1.24, par2 = -0.62, tau = -0.03)
7,10;8,9,4,1,6,2,3	BB8 (par = 1.11, par2 = 0.84, tau = 0.03)

Tree 9:	copula type
7,5;8,9,4,10,3,2,6,1	Rotated BB8 270 degrees (par = -1.14, par2 = -0.85, tau = -0.04)

References

[1] Jaworski P., Durante F., Hördle W., Rychlik T., Copula Theory and its Application, Proceedings of the Workshop Held in Warsaw, 25 − 26 September 2009, Springer

[2] Fréchet, M.: Sur les tableaux de corrélation dont les marges sont données. Ann. Univ. Lyon. Sect. A.14(3), 53-77 (1951)

[3] Sklar, A.: Fonctions de répartition ándimensions et leurs marges. Publ. Inst. Stat. Univ. Paris 8, 229-231 (1959)

[4] Nelsen R. B., An Introduction to Copulas, New York: Springer, (1999)

[5] Joe H., Dependence Modeling with Copulas, CRC Press, Taylor & Francis Group, (2014)

[6] M. S. Irajia, H. Amerib, RMSD Protein Tertiary Structure Prediction with Soft Computing, Modern Education and Computer Science Press, (2016)

[7] M. Killiches, D. Kraus, C. Czado, Examination and visualisation of the simplifying assumption for vine copulas in three dimensions, Australian & New Zealand Journal of Statistics 59(1), (October 2016)

[8] J. A. Tawn, Bivariate extreme value theory: models and estimation, Biometrika, 75(3), 397-415. (1988)