Diagonal projections and rank histograms for measuring the distance between estimated copulas and observed data

Maël Forcier

May 2017

1 Introduction

To create scenarios for renewable energies production, we need to be able to measure the dependance through space and time or between different sources. To focus on the correlation, we use copulas. Giving a set of datas, different methods exist to choose the best copulas and parameters that will best fit and model our datas. Nevertheless, we need tools to verify and measure how each of the copulas fit our observed data. Moreover, we need a tool that focus on the tails where the dependance matters a lot for us. I will briefly present in section 1, the loglikelihood which is a classical way to measure the overall fit of a distribution. Then, I will describe more precisely a way to measure how a copula fit the datas in the tail using rank histogram on diagonal projections.

2 Loglikelihood

Present briefly the loglikelihood. Refer to a good paper explaining loglikelihood ?

3 Solving the problem of unreproducibility

Each day j, we look at a set of data including for example a description of the state of the system early in the morning and the observations of the 90 previous days. With this set of data, we are able to choose a distribution represented by its cumulative density function F_j with a set of parameters θ_j that we hope will predict the best what will happen on day j. Some techniques are presented in [2] or [3] but we will not focus on them in this article. We now want to verify if this distribution was well chosen.

Unfortunately, we only observe what happens on the day j once. Let us note O_j the day j observation. O_j is not sufficient to check if this random variable follow

the distribution F_j. Moreover, each day is different and for instance another day $i \neq j$ will give a different distribution F_i with different parameters θ_i. Thus, it is impossible to verify if each distribution is correct each day. Nevertheless, their are techniques to verify if our procedure is valid and if the estimation of distributions makes sense.

Let us define $U_j = F_j(O_j)$. It is easy to prove that U_j should have a uniform distribution. (This classical result is often used to generate a random variable X following cumulative density function F with uniform random variable U : $X = F^{-1}(X)$.) As all U_j are computed independantly, all the U_j must be independant and distributed uniformly.

We now have a set of observation $\mathbf{U} = (U_1, ..U_n)$ that should be independant and uniformly distributed on [0,1]. We can now compute the extent to which it follows a uniform distribution for example by computing the Earth Mover Distance between the empirical distribution of U and the uniform distribution or by using rank histograms.

4 Rank histograms

Imagine we have our problem in 1 dimension. Give an example in dimension 1. Present rank histograms refers to [1]

As one can see in [1], the biggest problem with rank histograms is that they are primarly useful only in one dimension. So we have to make projection. Projecting on the marginals is useless because what we care about is the dependance. Thus, we will project on the diagonals. This way we can measure the fit of the copulas in the corner that we are interested in.

5 Earth Mover Distance

Definition 1. *The **Earth Mover Distance** (EMD) between two histograms $P = ((x_i, p_i))_i$ and $Q = ((y_j, q_j))_j$ is :*

$$EMD(P,Q) = \frac{\min\limits_{(f_{i,j}) \in F} \sum_{i,j} f_{i,j} d_{i,j}}{\min(\sum_i p_i, \sum_i q_i)}$$

where $F = \{(f_{i,j}) | f_{i,j} \geq 0, \sum\limits_i f_{i,j} \leq P_i, \sum\limits_j f_{i,j} \leq Q_j, \sum\limits_{i,j} f_{i,j} = \min(\sum\limits_i p_i, \sum\limits_i q_i)\}$ *and $d_{i,j}$ is the distance between x_i and y_j.*

In our problem, we look at histograms of distribution with real density function. So, with probability 1 we will not have the same result twice. Thus, the weight of our histograms will just be 1 for each value : $\forall i, q_i = 1, \forall j, p_j = 1$.

Moreover, we will only consider vectors with same dimension n. We can now simplify the notation and define the EMD between two vectors :

Definition 2. *The **Earth Mover Distance** (EMD) between two vectors* $\boldsymbol{u} = (u_1, ..., u_n)$ *and* $\boldsymbol{v} = (v_1, ..., v_n)$ *of dimension n is :*

$$EMD(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{n} \min_{(f_{i,j}) \in F} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} d_{i,j}$$

where $F = \{(f_{i,j}) | f_{i,j} \geq 0, \sum_{i=1}^{n} f_{i,j} \leq 1, \sum_{j=1}^{n} f_{i,j} \leq 1, \sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} = n\}$
and $d_{i,j} = |u_i - v_j|$

Solving this linear program is possible but there is a faster way to compute the EMD thanks to the following property :

Property 1. *For any vectors* $\boldsymbol{u}$ *and* $\boldsymbol{v}$ *of dimension n :*

$$EMD(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{n} \sum_{i=1}^{n} |\tilde{u}_i - \tilde{v}_i| = \frac{1}{n} ||\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{v}}||_1$$

where $\tilde{\boldsymbol{x}}$ *is the sorted vector of* $\boldsymbol{x}$ *:*
$\{x_1, ..., x_n\} = \{\tilde{x}_1, ..., \tilde{x}_n\}$ *and* $\forall i \leq j, \tilde{x}_i \leq \tilde{x}_j$

Demonstration :

First we have :

$$\begin{aligned} F &= \{(f_{i,j}) | f_{i,j} \geq 0, \sum_{i=1}^{n} f_{i,j} \leq 1, \sum_{j=1}^{n} f_{i,j} \leq 1, \sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} = n\} \\ &= \{(f_{i,j}) | f_{i,j} \geq 0, \sum_{i=1}^{n} f_{i,j} = 1, \sum_{j=1}^{n} f_{i,j} = 1\} \end{aligned}$$

The way $\supset$ is trivial.

Let be $(f_{i,j})_{i,j} \in F$ and let us suppose that $\exists i_0, \sum_{i=1}^{n} f_{i_0,j} < 1 : \sum_{i=1}^{n} f_{i_0,j} = 1 - \epsilon$.
Thus, $\sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} \leq 1 - \epsilon + \sum_{i=1, i \neq i_0}^{n} 1 = n - \epsilon < n$
$(f_{i,j})_{i,j} \notin F$: Contradiction

We will now demonstrate that it exists $f_{i,j}$ integers that solve the minimum problem. This a classical demonstration using several theorems. For any precision, see the chapter 2 of [4].

Let us define $M = (M_{k,(i,j)}) \in \mathcal{M}_{2n,n^2}(\mathbb{R})$ with :

$$M_{k,(i,j)} = \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } k = j + n \\ 0 & \text{else} \end{cases}$$

Because $1 \leq i, j \leq n$ the cases are incompatible.

We now have :

$$F = \{(f_{i,j}) | f_{i,j} \geq 0, Mf = B\}$$

where $B = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}$

The coefficient of M are just -1,0 and 1 and M has just once 1 and once -1 on each of its column.
Thanks to the Poincaré lemma (see [4] chapter 2), we can say that M is totally unimodulary. Because B has integer coefficient, we can say that all extreme points of F have integer coefficients.
So, we finally have it exists $f_{i,j}$ integers that solve the minimization problem.

Let $(f_{i,j})_{i,j}$ solve the mnimization problem with integer coefficients.
$\forall i, \forall j, 0 \leq f_{i,j} \leq 1$ and $f_{i,j} \in \mathbb{N} \Rightarrow f_{i,j} = 0$ or $f_{i,j} = 1$
$\forall i, \sum_{j=1}^{n} f_{i,j} = 1 \Rightarrow \exists! j_0, f_{i,j_0} = 1$
$\forall j, \sum_{i=1}^{n} f_{i,j} = 1 \Rightarrow \exists! i_0, f_{i_0,j} = 1$

We now have :

$$\exists \sigma \in \mathfrak{S}_n, f_{i,j} = \begin{cases} 1 & \text{if } j = \sigma(i) \\ 0 & \text{else} \end{cases}$$

So,

$$EMD(\mathbf{u}, \mathbf{v}) = \min_{\sigma \in \mathfrak{S}_n} \sum_{i=1}^{n} d_{i,\sigma i} = \min_{\sigma \in \mathfrak{S}_n} \sum_{i=1}^{n} |u_i - v_{\sigma(i)}|$$

We now have to prove that this min is reached when $(u_i)_i$ is sorted in the same order than $(v_{\sigma(i)})_i$. Since the indexes have symetric roles and are just notations indicating coefficients, we can consider that u and v are sorted :
$\forall i \leq j, u_i \leq u_j, v_i \leq v_j$
With this notation, we need to prove that this minimum is reached for $\sigma = Id$.

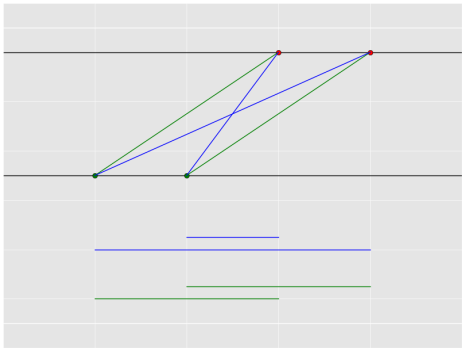

Figure 1: Graphic representation of case 1 : *The upper black line represents* $\boldsymbol{v}$ *and the lower one* $\boldsymbol{u}$*. The red points are from left to right* v_r *and* v_q *and The green points are from left to right* u_p *and* u_q*. The blue lines represent what the cost when using* σ *and the green lines the cost when using* $\tilde{\sigma}$*.*

Suppose that $\sigma \neq Id$ reaches the minimum, so $supp(\sigma) \neq \emptyset$ and $supp\sigma$ contains at least two elements. Let us define:

$$
\begin{aligned}
q &= \max supp(\sigma) \\
p &= \sigma^{-1}(q) \\
r &= \sigma(q)
\end{aligned}
$$

We have $q \leq p$ and $r \leq p$, so $u_q \leq u_p$ and $v_r \leq v_p$.

- Case 1 : $u_p \leq u_q \leq v_r \leq v_q$

$$
\begin{aligned}
|u_p - v_q| + |u_q - v_r| &= v_q - u_p + v_r - u_q \\
&= v_r - u_p + v_q - u_q \\
&= |u_p - v_r| + |u_q - v_q|
\end{aligned}
$$

- Case 2 : $u_p \leq v_r \leq u_q \leq v_q$

$$
\begin{aligned}
|u_p - v_q| + |u_q - v_r| &\leq |u_p - v_q| \\
&= v_q - u_p \\
&= v_q - u_q + u_q - u_p \\
&\leq v_q - u_q + v_r - u_p \\
&= |u_q - v_q| + |u_p - v_r|
\end{aligned}
$$

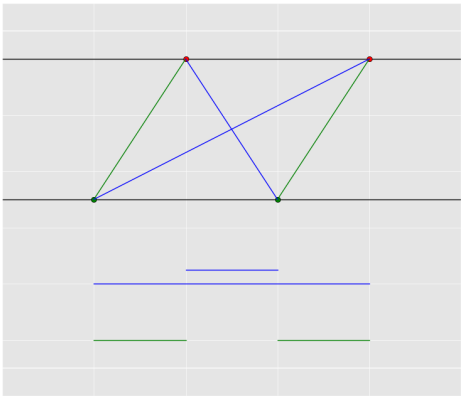

Figure 2: Graphic representation of case 2 : *The upper black line represents* $\boldsymbol{v}$ *and the lower one* $\boldsymbol{u}$*. The red points are from left to right* v_r *and* v_q *and The green points are from left to right* u_p *and* u_q*. The blue lines represent what the cost when using* σ *and the green lines the cost when using* $\tilde{\sigma}$*.*

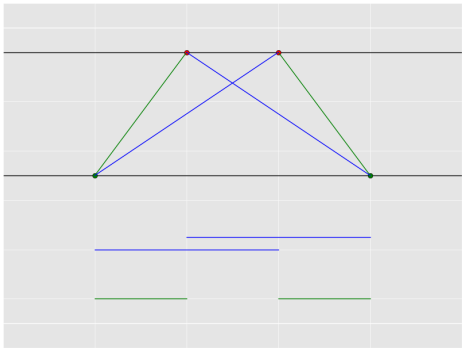

Figure 3: Graphic representation of case 3 : *The upper black line represents* $\boldsymbol{v}$ *and the lower one* $\boldsymbol{u}$*. The red points are from left to right* v_r *and* v_q *and The green points are from left to right* u_p *and* u_q*. The blue lines represent what the cost when using* σ *and the green lines the cost when using* $\tilde{\sigma}$*.*

- Case 3 : $u_p \leq v_r \leq v_q \leq u_q$

$$\begin{aligned}|u_p - v_q| + |u_q - v_r| &= v_q - u_p + u_q - v_r \\ &= u_q - u_p + v_q - v_r \\ &\leq u_q - u_p \\ &= u_q - v_q + v_q - u_p \\ &\leq u_q - v_q + v_r - u_p \\ &= |u_q - v_q| + |u_p - v_r|\end{aligned}$$

- Case 4 : $v_r \leq v_q \leq u_p \leq u_q$
  Same as case 1 by inversing the symetric roles of $\mathbf{u}$ and $\mathbf{v}$.
- Case 5 : $v_r \leq u_p \leq v_q \leq u_q$
  Same as case 2 by inversing the symetric roles of $\mathbf{u}$ and $\mathbf{v}$.
- Case 6 : $v_r \leq u_p \leq u_q \leq v_q$
  Same as case 3 by inversing the symetric roles of $\mathbf{u}$ and $\mathbf{v}$.

In all this cases, we have :

$$|u_p - v_q| + |u_q - v_r| \leq |u_q - v_q| + |u_p - v_r|$$

Let us define $\tilde{\sigma} = \sigma \circ (pq)$:

$$\begin{aligned}\forall i \notin \{p, q\}, \tilde{\sigma}(i) &= \sigma(i) \\ \tilde{\sigma}(p) &= r \\ \tilde{\sigma}(q) &= q\end{aligned}$$

Then we have

$$\begin{aligned}\sum_{i=1}^{n} |u_i - v_{\sigma(i)}| &= \sum_{i \notin \{p,q\}} |u_i - v_{\sigma(i)}| + |u_p - v_q| + |u_q - v_r| \\ &\leq \sum_{i \notin \{p,q\}} |u_i - v_{\sigma(i)}| + |u_q - v_q| + |u_p - v_r| \\ &= \sum_{i=1}^{n} |u_i - v_{\tilde{\sigma}(i)}|\end{aligned}$$

So, $\tilde{\sigma}$ reaches the minimum too and $supp(\tilde{\sigma}) = supp(\sigma) \setminus \{\max supp(\sigma)\}$. By doing this operation many times, we can remove all the elements of $supp(\sigma)$. So Id reaches the minimum. □

6 Projection on diagonal

Because we are very interested in extreme events, we will focus on tails of the multivariate distribution. To study their dependance, it is interesting to consider the corners of the space of copulas which is an hypercube.

6.1 Corner

Definition 3. *A **corner** of an hypercube* $[0,1]^d$ *is a point* $\boldsymbol{a} = (a_1, ..., a_d) \in \{0,1\}^d$ *:*

$$\forall i \in [\![1, d]\!], a_i = 0 \text{ or } a_i = 1$$

So there is 2^d corners in a hypercube of dimension d.

6.2 Diagonal

Definition 4. *A **diagonal** Δ is a segment which links to opposite corner $\boldsymbol{a}$ and $\boldsymbol{b}$:*

$$\Delta = [\boldsymbol{a}, \boldsymbol{b}] \text{ where } \forall i \in [\![1, d]\!], a_i = 0 \iff b_i = 1$$

Alternatively :

$$\Delta = \{(1-\lambda)\boldsymbol{a} + \lambda\boldsymbol{b}, \lambda \in [0,1]\}, \text{ where } \forall i \in [\![1, d]\!], a_i = b_i \text{ mod } 2$$

*Because one diagonal can be written [**a**,**b**] or [**b**,**a**], we will always consider $a_1 = 0$ so that each diagonal has a unique way notation.*

*We can also define the **direction** of a diagonal as the vector :*

$$U_\Delta = \frac{1}{\sqrt{d}}(\boldsymbol{b} - \boldsymbol{a})$$

6.3 Projection

Definition 5. *The **matrix of projection** on the linear space will be :*

$$M_\Delta = U_\Delta U_\Delta^\top$$

*Finally, the **projection on the diagonal** which is an affine space is the function P_Δ such that :*

$$P_\Delta(X) = M_\Delta(X - C) + C \text{ where } C = (\frac{1}{2}, ..., \frac{1}{2}) \tag{1}$$

So there are 2^{d-1} diagonals, directions and matrix of projection in an hypercube of dimension d.

The division by $\sqrt{d}$ in the definition of direction permits to have a unit vector.

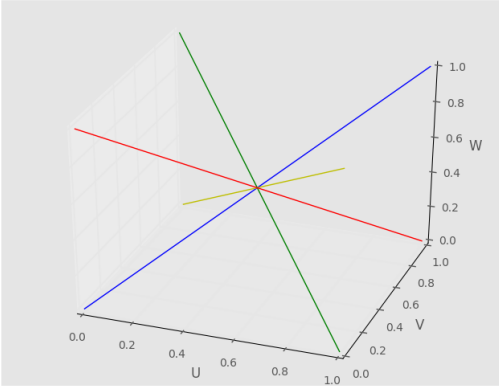

Figure 4: Examples of diagonals, *The blue one is [(0,0,0),(1,1,1)], the yellow one is [(0,1,0),(1,0,1)], the green one is [(0,1,1),(1,0,0)], and the red one is [(0,0,1),(1,1,0)]*

M is indeed a matrix thanks to the order of the factors (and not a scalar product as $U^\top U$).

One should not confuse the matrix of projection on the linear space and the traditional affine projection on the diagonal. That is why we need to translate everything with the center of the hypercube C.

6.4 Distribution on the diagonal

We now want to study the distribution of the points projected on the diagonal to compare it to a uniform distribution. Since the diagonal is a segment, each point x of the diagonal can be described by only one scalar number λ : $x = (1-\lambda)a+\lambda b$ (cf Definition of the diagonal).
λ can be understood as the normalised distance between a and x :

$$\|x - a\| = \|(1 - \lambda)a + \lambda b\| = \lambda\|a - b\| = \lambda\sqrt{d}$$

Where $\|.\|$ is a norm in our space.

But λ can be easily evaluated by taking the first coordonates of x :

$$x_1 = (1 - \lambda)a_1 + \lambda b_1 = (1 - \lambda) * 0 + \lambda * 1 = \lambda$$

This equality is possible thanks to our useful convention $(a_1, b_1) = (0, 1)$.

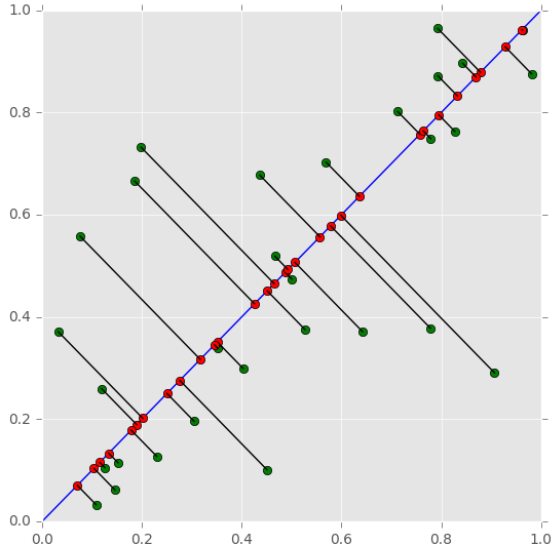

Figure 5: Projection of 30 points on the [(0,0),(1,1)] diagonal *The diagonal is in blue, the initial points are green and their projections are red.*

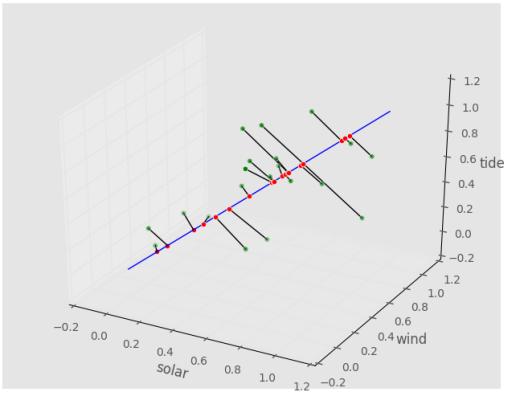

Figure 6: Projection of 20 points on the [(0,0,0),(1,1,1)] diagonal *The diagonal is in blue, the initial points are green and their projections are red.*

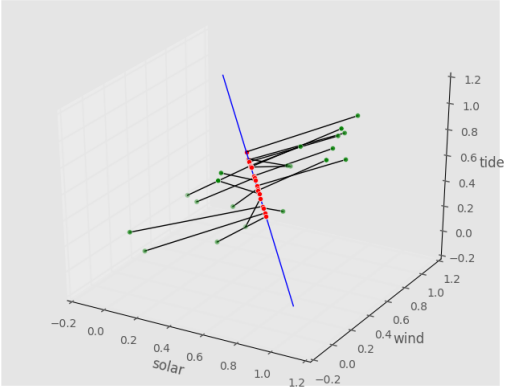

Figure 7: Projection of 20 points on the [(1,0,0),(0,1,1)] diagonal *The diagonal is in blue, the initial points are green and their projections are red.*

We now have a unique number that should be uniformly distributed on [0,1].

7 Our algorithm

For each day j do :

- Thanks to all day $i \leq j-1$, fit a parametric distribution model with copula C_j
- Generate n realizations U of the random variable with the copula dependance and uniform marginals :

  Generate $\mathbf{U} = (U_1, ..., U_n)$ with each $U_i = (U_{i,1}, ..., U_{i,d}) \in [0,1]^d$

  where $\forall i \in [\![1, n]\!]$,
  $\mathbb{P}(U_{i,1} \leq u_1, ..., U_{i,d} \leq u_d) = C(u_1, ..., u_d)$ and
  $(\forall j \in [\![1, d]\!], U_{i,j}$ is uniformly distributed on [0,1].)
  For example, n=10000

- For each diagonal Δ :
- Project all the U_i on the diagonal
  Define $\mathbf{V}_\Delta = (V_{\Delta,1}, ..., V_{\Delta,n}) = (P_\Delta(U_1), ..., P_\Delta(U_n))$
- Define the empirical distribution on this diagonal :

$F_\Delta(X) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{X \leq V_{\Delta,k}}$

where $a \leq b \iff \forall i \in [\![1, d]\!], a_i \leq b_i$

- Observe with the data what happened on day j
  Call this observation $O_j = (O_{j,1}, ..., O_{j,d})$
- Pass it in the copula space
  Define $Q_j = (F_1(O_{j,1}), ..., F_d(O_{j,d}))$
  where the F_i are the cumulative density functions of the marginals estimated with another method.
- Project Q_j on the diagonal
  Define $R_{\Delta,j} = P_\Delta(Q_j)$
- Either define $S_{\Delta,j} = F_\Delta(R_{\Delta,j})$
  and compute the distance between the empirical distribution of the $S_\Delta = (S_{\Delta,i})_{i \in days}$ and the uniform distribution on [0,1].
- Or make a rank histogram with the $R_\Delta = (R_{\Delta,i})_{i \in days}$ $F_\Delta^{-1}(P_j)$

8 Test code

In this section, I will explain how we computed this algorithm with different parameters. They are arguments of many test functions I wrote and are just strings defining options :

source : the type of power source we want it can be 'solar' or 'wind'

datatype : if the datas is power ('actuals'), errors ('errors), a normal distributed sample ('normal-sample') or a uniformly distributed sample ('uniform-sample'), the two last ones are options to make verifications.

segment$_m arginals$: $the way you segment the datas to fit the marginals, you can either take only the date at the h$
$this is not the way the datas are segmented to fit the copula.$

kind : which projection you want it can be on a diagonal ('diagonal'), a marginal ('marginal') or can even compose with the kendall function ('kendal').

index : the index of the diagonal or the marginal you want to project with. Diagonals are indexed in the order of diag. list of diags. Marginals are index as the coordonates. index does not matter for kendall function.

method : way you choose the data to fit the distributions, you can either fit copulas and marginals with the datas of the whole year and check the observation then ('wholeyear'), or you

References

[1] Thomas M. Hamill, *Interpretation of Rank Histograms for Verifying Ensemble Forecasts*, 2000.

[2] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, Henrik Bakken, *Pair-copula construction of multiple dependence*, 2007.

[3] Kjersti Aas, *Modeling the dependance structure of financial assets : A survey of four copulas*, 2004.

[4] Stéphane Gaubert, Frédéric Bonnans, *Recherche opérationnelle : aspects mathématiques et applications*, Editions de l'École polytechnique, 2016